



ON-LINE DISTANCE COURSE ON BIOINFORMATICS

Course Objectives:

The course in Bioinformatics includes studying the basics of applied science, combining computer science with modern biology, based on studying the methods and models by which informatics contributes to the development of molecular biology. Some aspects of bioinformatics, such as Web-based programming, computer modeling, as well as the study of operating systems and programming languages are also covered in the training. The training of students in the "Bioinformatics" specialty will be focused on the opportunity to use the wealth of new information and to develop new techniques and methodology for intelligent information analysis, processing and knowledge discovery.

Learning Outcomes:

At the conclusion of the course, the student **be able** to:

- in-depth knowledge in the field of bioinformatics.
- skills to develop and apply theoretical methods, mathematical modeling and computational techniques for the simulation of biological systems and processes.
- interdisciplinary training and research opportunity in various areas of bioinformatics, including topics such as DNA and protein databases, protein structure and function, computational neuroscience, biomechanics, genetics, and management of agricultural and natural systems.
- serious theoretical training in the field of informatics and mathematics, and solid practical skills meeting modern European standards and requirements.
- formation of affinity and abilities for independent research and design activity.basis for continuing education
- way of thinking and affinity (openness) to the rapidly changing requirements of the information society.
- to use mathematical models and software packages in solving real business, engineering and management problems in continuous and discrete macrosystems.
- to participate in the development of basic software products and packages.to adapt and implement ready-made software products and systems.
- to solve optimization tasks of a different nature..

STRUCTURE OF THE COURSE OF BIOINFORMATICS

COURSE - 146 hours

MODULES – 4 modules

TOPICS – 12 topics + 10 elective

LESSONS – 75 lessons

| | | | | | |
|--------|--------|------------------------------------|---|-----------|--|
| | | | | Duration | |
| COURSE | | BIOINFORMATICS | | 146 hours | |
| | | | | | |
| | | Title of the module/ topic/ lesson | | | |
| MODULE | 1. | Molecular basis of life | | 50 | |
| | Topics | 1. | <i>Structure of Matter</i> | 10 | |
| | | | 1. Quantum theory of the atom. Ionic bond | 2 | |
| | | | 2. Chemical bonding in diatomic and polyatomic molecules. Geometric configuration of molecules | 2 | |
| | | | 3. Intermolecular forces. Hydrogen bonding, electrostatic and van der Waals interactions. | 2 | |
| | | | 4. Brownian motion and kinetic theory | 2 | |
| | | | 5. From Solid to Liquid | 2 | |
| | | 2. | <i>Fundamentals of Molecular Biology</i> | 10 | |
| | | | 1. Chemical bases of the cell | 2 | |
| | | | 2. The cell. Basic cell types. Cell construction. Organelles of eukaryotic cells. Cytoskeleton - components and structural functions | 2 | |
| | | | 3. Lipids. Construction and functions. Biomembranes and cell architecture. Lipid structure and structural organization of biomembranes. Protein | 2 | |

| | | | | | | |
|--|--------|-----------------|-----------------------------|--|-----------|--|
| | | | | components and main functions in biomembranes. Transport of ions and small molecules across cell membranes | | |
| | | | 4. | Cellular energy. Oxidation of glucose and fatty acids to carbon dioxide | 2 | |
| | | | 5. | Signaling the cell surface. Signal molecules and receptors. G-protein bound receptors. Signal pathways controlling gene activity | 2 | |
| | | 3. | Proteins and enzymes | | 20 | |
| | | | 1. | Structure and properties of amino acids | 2 | |
| | | | 2. | Structure of proteins | 2 | |
| | | | 3. | Chemical bonds in the protein molecule | 2 | |
| | | | 4. | Protein functions | 2 | |
| | | | 5. | Mechanism of enzyme catalysis | 3 | |
| | | | 6. | Types of enzyme reactions | 2 | |
| | | | 7. | Kinetics of enzyme reactions | 3 | |
| | | | 8. | Regulation of enzyme reactions | 2 | |
| | | | 9. | Enzyme classification | 2 | |
| | | | | | | |
| | | 4. | Biophysical Problems | | 10 | |
| | | | 1. | Thermodynamics | 2 | |
| | | | 2. | Kinetics | 2 | |
| | | | 3. | Quantum mechanics | 2 | |
| | | | 4. | Spectroscopy | 2 | |
| | | | 5. | Understanding biological systems using physical chemistry | 2 | |
| | 2. | Genetics | | | 32 | |
| | Topics | 1. | Molecular Genetics | | 12 | |
| | | | 1. | DNA: The Genetic Material, Experimental Proof of the Genetic, Function of DNA, Genetic Role of DNA in Bacteriophage | 2 | |
| | | | 2. | DNA Structure: The Double Helix | 2 | |

| | | | | | |
|--|--------|---|---|-----------|--|
| | | 3. | An Overview of DNA Replication | 2 | |
| | | 4. | Genes and Proteins, Inborn Errors of Metabolism as a Cause of Hereditary Disease, Mutant Genes and Defective Proteins | 2 | |
| | | 5. | Gene Expression: The Central Dogma, Transcription, Translation, The Genetic Code | 2 | |
| | | 6. | Mutation, Protein Folding and Stability | 2 | |
| | 2. | Theoretical Modeling in Genomics | | 12 | |
| | | 1. | Identification of Muscle-Specific Transcriptional Regulatory Regions | 2 | |
| | | 2. | A Systematic Analysis of Gene Functions by the Metabolic Pathway Database | 2 | |
| | | 3. | Polymer Dynamics of DNA, Chromatin, and Chromosomes | 2 | |
| | | 4. | Sequence Patterns Diagnostic of Structure and Function | 2 | |
| | | 5. | Recognizing Functional Domains in Biological Sequences | 2 | |
| | | 6. | Computer Simulations of Protein-DNA Interactions | 2 | |
| | 3. | Computer Genomics | | 8 | |
| | | 1. | DNA Decoding | 2 | |
| | | 2. | Sequence Alignment | 2 | |
| | | 3. | Genome Assembly and Annotation | 2 | |
| | | 4. | Biological Interaction Network | 2 | |
| | 3. | Bioinformatic tools | | 54 | |
| | Topics | 1. | Introduction into Bioinformatics | 6 | |
| | | 1. | History of Bioinformatics | 2 | |
| | | 2. | Objectives of Bioinformatics | 2 | |
| | | 3. | Components of Bioinformatics | 2 | |
| | | 1. | Docking Problem | 12 | |
| | | 1. | Where to Find Protein and Ligand Structures | 2 | |
| | | 2. | Where Can Ligands Dock | 2 | |

| | | | | | | |
|--|--------|--------------------------|-------------------------------------|---|-----------|--|
| | | | 3. | How Does a Docking Program Work | 2 | |
| | | | 4. | The Scoring Functions | 2 | |
| | | | 5. | The Importance of Considering Receptor Flexibility During Docking | 2 | |
| | | | 6. | Main Characteristics of Selected Docking Software | 2 | |
| | | 2. | Protein Folding Problem | | 12 | |
| | | | 1. | Protein folding | | |
| | | | 2. | Sequence Alignment | | |
| | | | 3. | Multiple sequence alignment | | |
| | | 3. | Artificial Neural Networks | | 12 | |
| | | | 1. | General Concepts of ML and ANN | | |
| | | | 2. | Perceptron | | |
| | | | 3. | Multi-Layer Perceptron - Backpropagation | | |
| | | | 4. | Deep Learning with Python | | |
| | | | 5. | Convolutional Neural Networks | | |
| | | 4. | Algorithms in Bioinformatics | | 12 | |
| | | | 1. | Biological Networks | | |
| | | | 2. | Algorithms | | |
| | | | 3. | Network Properties | | |
| | | | 4. | Random Networks | | |
| | | | 5. | Small World Networks | | |
| | | | 6. | Scale Free Networks | | |
| | | | 7. | Clustering | | |
| | | | 8. | Network Alignment | | |
| | 4. | Elective Subjects | | | 10 | |
| | Topics | 1. | Introduction into Bio Java | | 1 hour | |
| | | 2. | Introduction into BioPython | | 1 hour | |
| | | 3. | Operations Research | | 1 hour | |
| | | 4. | Modern Methods in Computer Biology | | 1 hour | |
| | | 5. | Statistical Analysis | | 1 hour | |
| | | 6. | Computer Molecular Modeling | | 1 hour | |
| | | 7. | Bioinformatics Computer Laboratory | | 1 hour | |

| | | | | | |
|--|--|-----|--------------------------------------|--------|--|
| | | 8. | Quantitative Pharmacology | 1 hour | |
| | | 9. | Scripting Languages | 1 hour | |
| | | 10. | Models of Artificial Neural Networks | 1 hour | |



Module 1. Molecular basis of life

Topic 1. Structure of Matter

Lesson 1. Quantum theory of the atom. Ionic bond



Contents

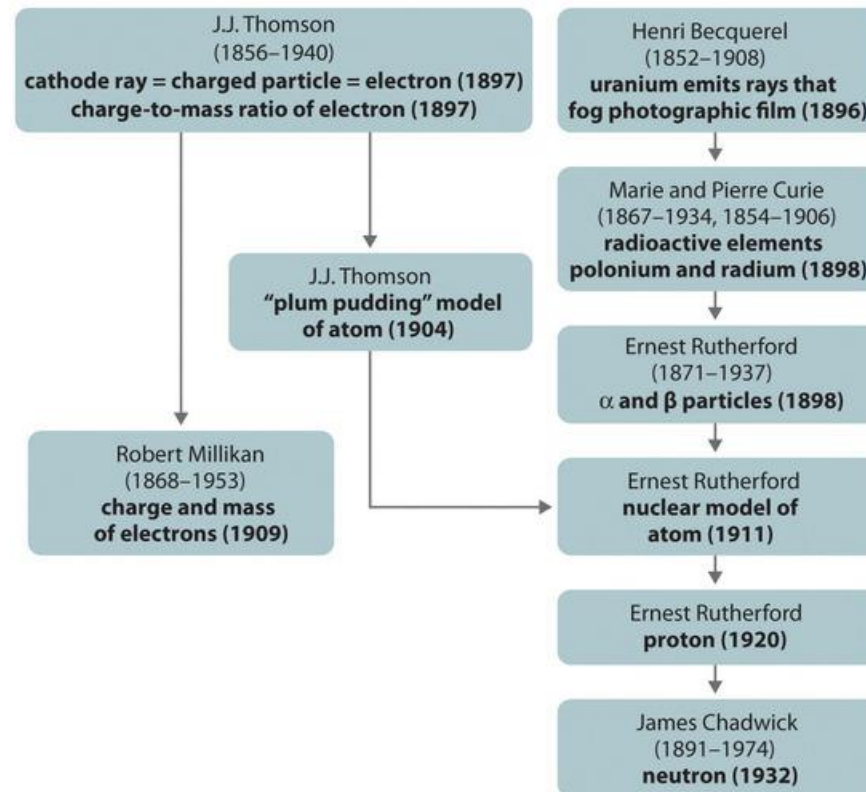
- Bohr's Model
- The Quantum Mechanical Atom
- Wavefunctions
- Quantum Numbers
- Atomic Orbitals
- Ionic Bonding



Introduction

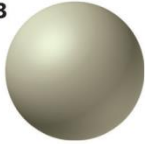
Atoms, the smallest particles of an element that exhibit the properties of that element, consist of negatively charged electrons around a central nucleus composed of more massive positively charged protons and electrically neutral neutrons. Radioactivity is the emission of energetic particles and rays (radiation) by some substances. Three important kinds of radiation are α particles (helium nuclei), β particles (electrons traveling at high speed), and γ rays.

A Summary of the Historical Development of Models of the Components and Structure of the Atom



The Evolution of Atomic Theory, as Illustrated by Models of the Oxygen Atom

1803



Dalton proposes the indivisible unit of an element is the atom.

1904



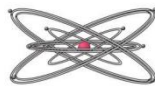
Thomson discovers electrons, believed to reside within a sphere of uniform positive charge (the plum pudding model).

1911



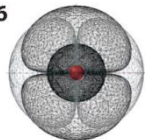
Rutherford demonstrates the existence of a positively charged nucleus that contains nearly all the mass of an atom.

1913



Bohr proposes fixed circular orbits around the nucleus for electrons.

1926



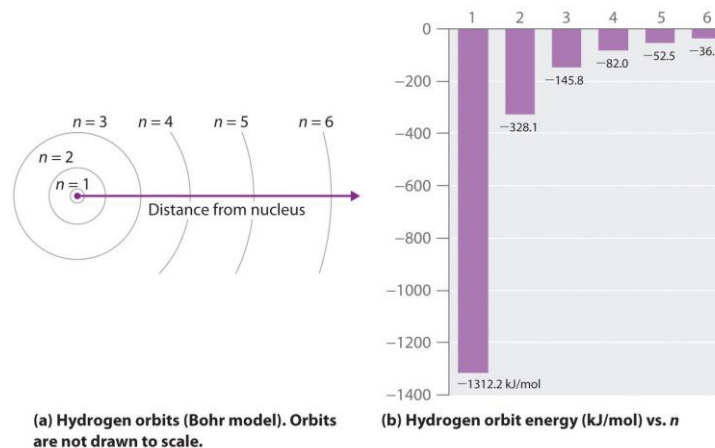
In the current model of the atom, electrons occupy regions of space (orbitals) around the nucleus determined by their energies.

Bohr's Model

In 1913, a Danish physicist, Niels Bohr (1885–1962; Nobel Prize in Physics, 1922), proposed a theoretical model for the hydrogen atom that explained its emission spectrum. Bohr's model required only one assumption: The electron moves around the nucleus in circular orbits that can have only certain allowed radii. Rutherford's earlier model of the atom had also assumed that electrons moved in circular orbits around the nucleus and that the atom was held together by the electrostatic attraction between the positively charged nucleus and the negatively charged electron. Although we now know that the assumption of circular orbits was incorrect, Bohr's insight was to propose that the electron could occupy only certain regions of space.

Using classical physics, Niels Bohr showed that the energy of an electron in a particular orbit is given by

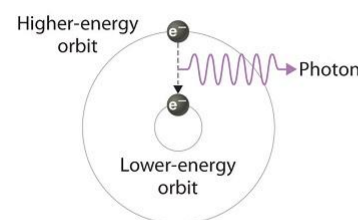
$$E_n = \frac{-Rhc}{n^2}$$



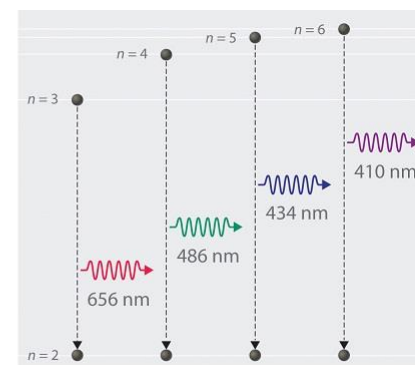
where R is the Rydberg constant, h is Planck's constant, c is the speed of light, and n is a positive integer corresponding to the number assigned to the orbit, with $n = 1$ corresponding to the orbit closest to the nucleus. In this model $n = \infty$ corresponds to the level where the energy holding the electron and the nucleus together is zero. In that level, the electron is unbound from the nucleus and the atom has been separated into a negatively charged (the electron) and a positively charged (the nucleus) ion. In this state the radius of the orbit is also infinite.

Bohr's Model

As n decreases, the energy holding the electron and the nucleus together becomes increasingly negative, the radius of the orbit shrinks and more energy is needed to ionize the atom. The orbit with $n = 1$ is the lowest lying and most tightly bound. The negative sign in equation indicates that the electron-nucleus pair is more tightly bound (i.e. at a lower potential energy) when they are near each other than when they are far apart. Because a hydrogen atom with its one electron in this orbit has the lowest possible energy, this is the ground state (the most stable arrangement of electrons for an element or a compound) for a hydrogen atom. As n increases, the radius of the orbit increases; the electron is farther from the proton, which results in a less stable arrangement with higher potential energy. A hydrogen atom with an electron in an orbit with $n > 1$ is therefore in an excited state, defined as any arrangement of electrons that is higher in energy than the ground state. When an atom in an excited state undergoes a transition to the ground state in a process called decay, it loses energy by emitting a photon whose energy corresponds to the difference in energy between the two states.



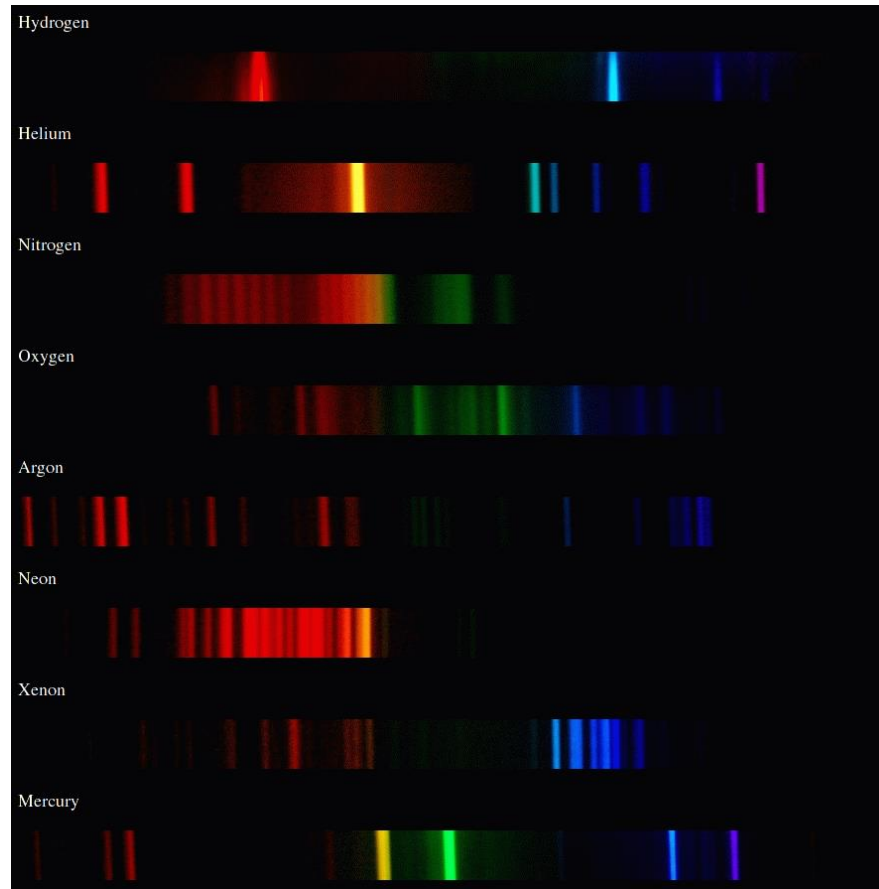
(a) Electronic emission transition



(b) Balmer series transitions

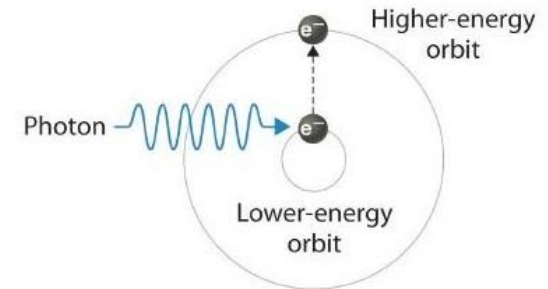
Unfortunately, Bohr could not explain why the electron should be restricted to particular orbits. Also, despite a great deal of tinkering, such as assuming that orbits could be ellipses rather than circles, his model could not quantitatively explain the emission spectra of any element other than hydrogen. In fact, Bohr's model worked only for species that contained just one electron: H, He^+ , Li^{2+} , and so forth.

The atomic emission spectra for various elements. Each thin band in each spectrum corresponds to a single, unique transition between energy levels in an atom.

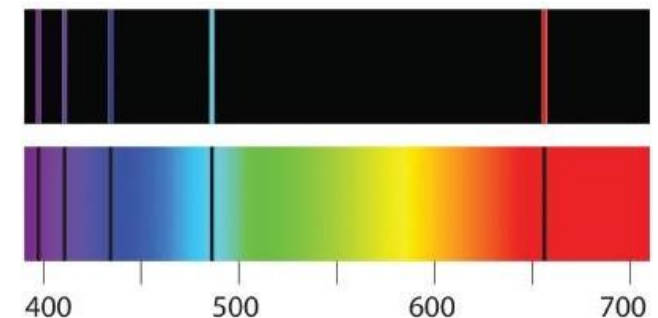


The Energy States of the Hydrogen Atom

If white light is passed through a sample of hydrogen, hydrogen atoms absorb energy as an electron is excited to higher energy levels (orbits with $n \geq 2$). If the light that emerges is passed through a prism, it forms a continuous spectrum with black lines (corresponding to no light passing through the sample) at 656, 468, 434, and 410 nm. These wavelengths correspond to the $n = 2$ to $n = 3$, $n = 2$ to $n = 4$, $n = 2$ to $n = 5$, and $n = 2$ to $n = 6$ transitions. Any given element therefore has both a characteristic emission spectrum and a characteristic absorption spectrum, which are essentially complementary images.



(a) Electronic absorption transition



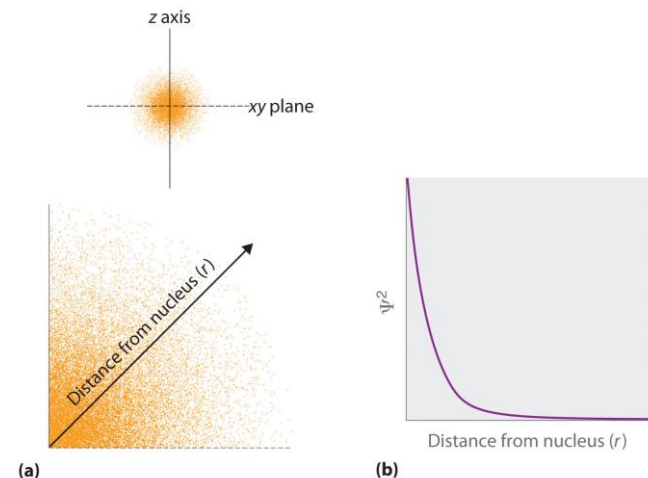
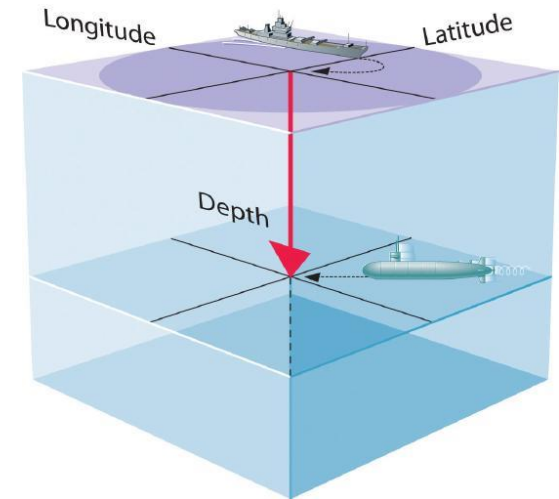
(b) H₂ emission spectrum (top), H₂ absorption spectrum (bottom)

The Quantum Mechanical Atom

The paradox described by Heisenberg's uncertainty principle and the wavelike nature of subatomic particles such as the electron made it impossible to use the equations of classical physics to describe the motion of electrons in atoms. Scientists needed a new approach that took the wave behavior of the electron into account. In 1926, an Austrian physicist, Erwin Schrödinger (1887–1961; Nobel Prize in Physics, 1933), developed wave mechanics, a mathematical technique that describes the relationship between the motion of a particle that exhibits wavelike properties (such as an electron) and its allowed energies.

Wavefunctions

- A wavefunction uses three variables to describe the position of an electron.
- The magnitude of the wavefunction at a particular point in space is proportional to the amplitude of the wave at that point
- The square of the wavefunction at a given point is proportional to the probability of finding an electron at that point, which leads to a distribution of probabilities in space.
- Describing the electron distribution as a standing wave leads to sets of quantum numbers that are characteristic of each wavefunction.
- Each wavefunction is associated with a particular energy.



Quantum Numbers

The **principal quantum number (n)** tells the average relative distance of an electron from the nucleus. As n increases for a given atom, so does the average distance of an electron from the nucleus.

The second quantum number is often called the **azimuthal quantum number (l)**. The value of l describes the shape of the region of space occupied by the electron. The allowed values of l depend on the value of n and can range from 0 to $n - 1$.

The third quantum number is the magnetic quantum number (m_l). The value of m_l describes the orientation of the region in space occupied by an electron with respect to an applied magnetic field. The allowed values of m_l depend on the value of l : m_l can range from $-l$ to l in integral steps.

Each wavefunction with an allowed combination of n , l , and m_l values describes an **atomic orbital**, a particular spatial distribution for an electron. For a given set of quantum numbers, each principal shell has a fixed number of subshells, and each subshell has a fixed number of orbitals.



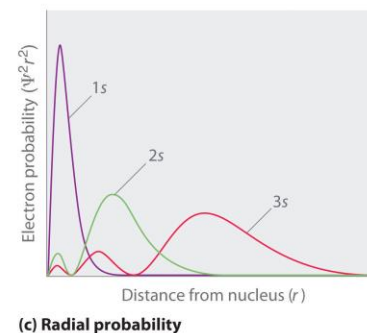
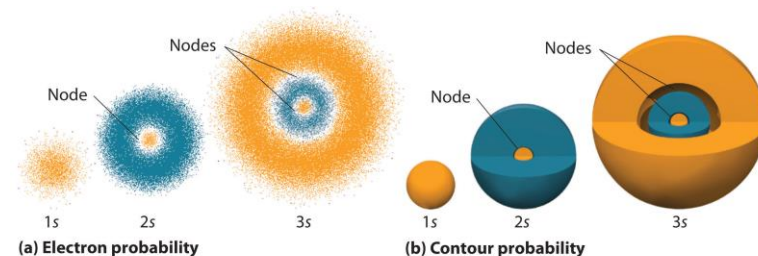
Atomic Orbitals

An orbital is the quantum mechanical refinement of Bohr's orbit. In contrast to his concept of a simple circular orbit with a fixed radius, orbitals are mathematically derived regions of space with different **probabilities** of containing an electron.

s Orbitals ($l=0$)

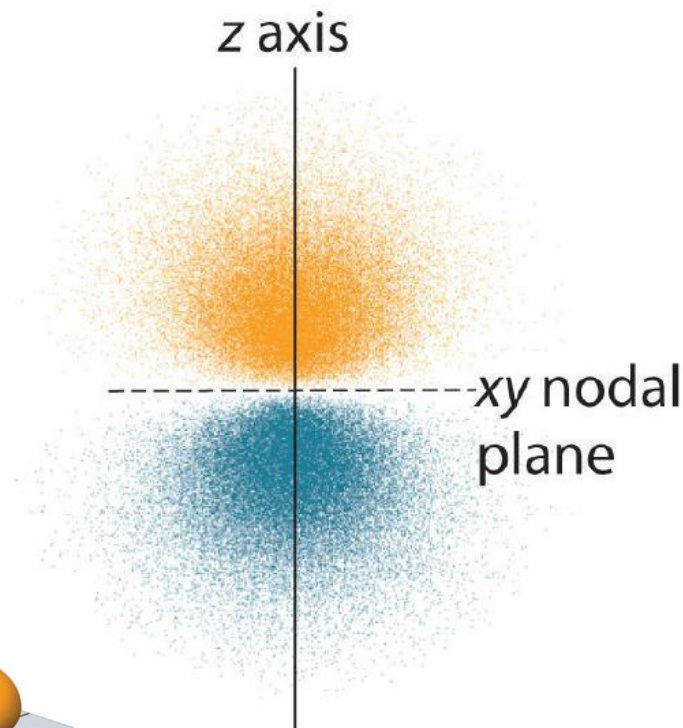
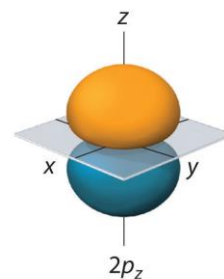
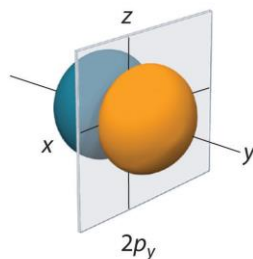
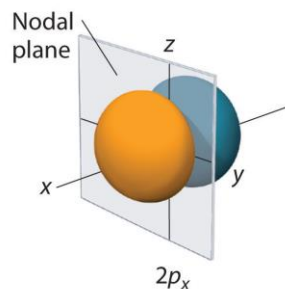
Three things happen to s orbitals as n increases:

- They become larger, extending farther from the nucleus.
- They contain more nodes. This is similar to a standing wave that has regions of significant amplitude separated by nodes, points with zero amplitude.
- For a given atom, the s orbitals also become higher in energy as n increases because of their increased distance from the nucleus.



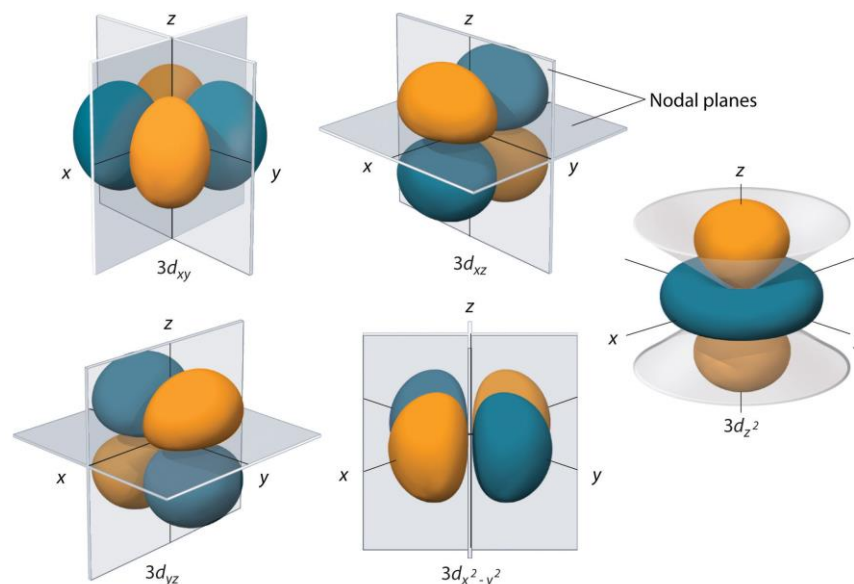
p Orbitals ($l=1$)

Only s orbitals are spherically symmetrical. As the value of l increases, the number of orbitals in a given subshell increases, and the shapes of the orbitals become more complex. Because the 2p subshell has $l = 1$, with three values of m_l (-1 , 0 , and $+1$), there are three 2p orbitals.



d Orbitals ($l=2$)

Subshells with $l = 2$ have five d orbitals; the first principal shell to have a d subshell corresponds to $n = 3$. The five d orbitals have m_j values of -2 , -1 , 0 , $+1$, and $+2$.



ERASMUS+

Orbital Energies



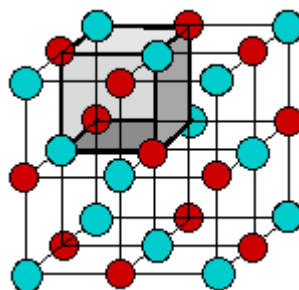
Orbital Energy Level Diagram for the Hydrogen Atom with a single electron. Each box corresponds to one orbital. Note that the difference in energy between orbitals decreases rapidly with increasing values of n .

Ionic Bonding

The ionic bond is formed when the "sharing" is so unequal that an electron from atom A is completely lost to atom B, resulting in a pair of ions:

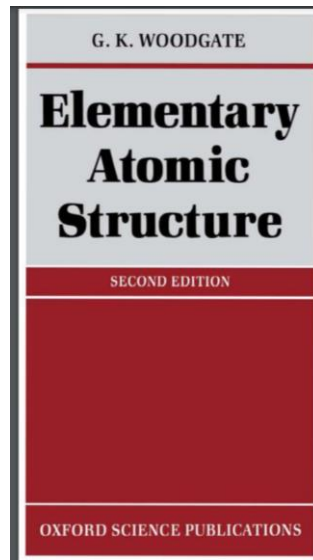


According to the ionic electrostatic model, solids such as NaCl consist of positive and negative ions arranged in a crystal lattice. Each ion is attracted to neighboring ions of opposite charge, and is repelled by ions of like charge; this combination of attractions and repulsions, acting in all directions, causes the ion to be tightly fixed in its own location in the crystal lattice.

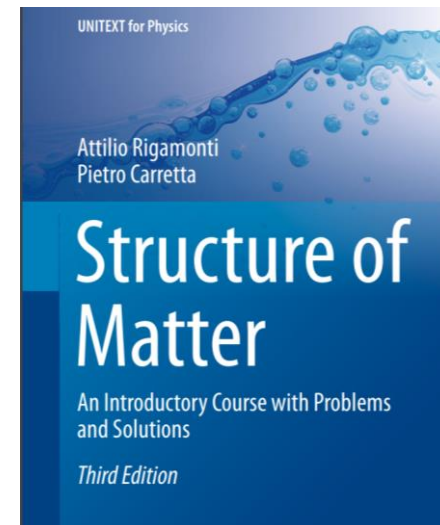


Review Questions

- What is Historical Development of Models of the Components and Structure of the Atom?
- Explain The Quantum Mechanical Atom
- What are Wavefunctions?
- Explain the Term Atomic Orbitals and give examples.



G. K. Woodgate(2002)
Elementary Atomic
Structure, Oxford
University Press Inc.,
New York



Rigamonti, A., & Carretta, P.
(2015). Structure of Matter.
UNITEXT for Physics, Springer
International Publishing
Switzerland



Module 1. Molecular basis of life

Topic 1. Structure of Matter

Lesson 2. Chemical bonding in diatomic and polyatomic molecules. Geometric configuration of molecules



Contents

- The Born-Oppenheimer Approximation
- Potential Energy Curves and Surfaces
- Bonding in Polyatomic Molecules
- Hybridization
- Molecular geometry



Introduction

The basis for understanding chemical bonding and the structures of molecules is the electron orbital description of the structure and valence of atoms, as provided by quantum mechanics. Using quantum mechanics to predict the chemical bonding patterns, optimal geometries, and physical and chemical properties of molecules is a large and active field of research known as molecular quantum mechanics or more commonly as quantum chemistry. Quantum chemistry calculations allow the geometries of molecules to be computed as well as a wide range of properties. Quantum chemistry can also be used in a novel way, in which the electrons are treated using quantum mechanics but the nuclei are treated as classical particles. We use quantum mechanics to calculate the internuclear forces but then use these forces in Newton's Second Law to study the motion of the nuclei during chemical reactions. This gives us a microscopic window into the specific motions, the complex dance, executed by the nuclei during a simple or complex chemical process.

The Born-Oppenheimer Approximation

The Born-Oppenheimer approximation is one of the basic concepts underlying the description of the quantum states of molecules. This approximation makes it possible to separate the motion of the nuclei and the motion of the electrons. This is not a new idea for us. We already made use of this approximation in the particle-in-a-box model when we explained the electronic absorption spectra of cyanine dyes without considering the motion of the nuclei. Then we discussed the translational, rotational and vibrational motion of the nuclei without including the motion of the electrons. In this chapter we will examine more closely the significance and consequences of this important approximation. Note, in this discussion nuclear refers to the atomic nuclei as parts of molecules not to the internal structure of the nucleus.

The Born-Oppenheimer Approximation

For a diatomic molecule as an example, the Hamiltonian operator is grouped into three terms.

$$\hat{H}(r, R) = \hat{T}_{nuc}(R) + \frac{e^2}{4\pi\epsilon_0} \frac{Z_A Z_B}{R} + \hat{H}_{elec}(r, R)$$

$$\hat{T}_{nuc}(R) = -\frac{\hbar^2}{2m_A} \nabla_A^2 - \frac{\hbar^2}{2m_B} \nabla_B^2$$

$$\hat{H}_{elec}(\vec{r}, \vec{R}) = -\frac{\hbar^2}{2m} \sum_i \nabla_i^2 + \frac{e^2}{4\pi\epsilon_0} \left(-\sum_i \frac{Z_A}{r_{Ai}} - \sum_i \frac{Z_B}{r_{Bi}} + \frac{1}{2} \sum_i \sum_{j \neq i} \frac{1}{r_{ij}} \right)$$

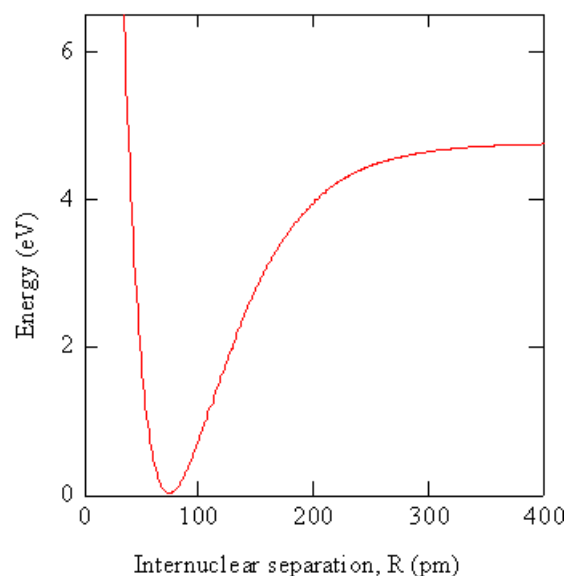
The Born-Oppenheimer approximation says that the nuclear kinetic energy terms in the complete Hamiltonian can be neglected in solving for the electronic wavefunctions and energies. Consequently, the electronic wavefunction $\phi_e(r, R)$ is found as a solution to the electronic Schrödinger equation

$$\hat{H}_{elec}(r, R) \phi_e(r, R) = E_e(R) \phi_e(r, R)$$

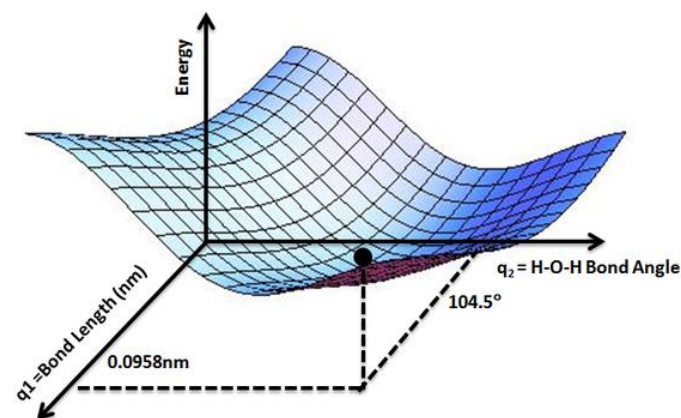
Even though the nuclear kinetic energy terms are neglected, the Born-Oppenheimer approximation still takes into account the variation in the positions of the nuclei in determining the electronic energy and the resulting electronic wavefunction depends upon the nuclear positions, R . As a result of the Born-Oppenheimer approximation, the molecular wavefunction can be written as a product

$$\psi_{ne}(r, R) = X_{ne}(R) \phi_e(r, R)$$

Potential Energy Curves and Surfaces



The potential energy function for a diatomic molecule. In practice the electronic Schrödinger equation is solved using approximations at particular values of R to obtain the wavefunctions $\phi_e(r, R)$ and potential energies $V_e(R)$.



The potential energy surface for a water molecule: Shows the energy minimum corresponding to optimized molecular structure for water- O-H bond length of 0.0958 nm and H-O-H bond angle of 104.5°.

Bonding in Polyatomic Molecules

Basically two ways to approach polyatomics.

First is to use delocalized M.O.'s where e^- are not confined to a single bond (region between 2 atoms) but can wander over 3 or more atoms. We will use this approach later for C bonding.

Second is to use hybridization of atomic orbitals and then use these to form localized (usually) bonds.

Hybridization combines orbitals on the SAME nucleus to form new orbitals called hybrids. Hybrids have characteristics of both the atomic orbitals from which they are formed.

Example: sp hybrid is formed by combining (adding) a $2s$ and a $2p$ wave function or orbital on a single atom.

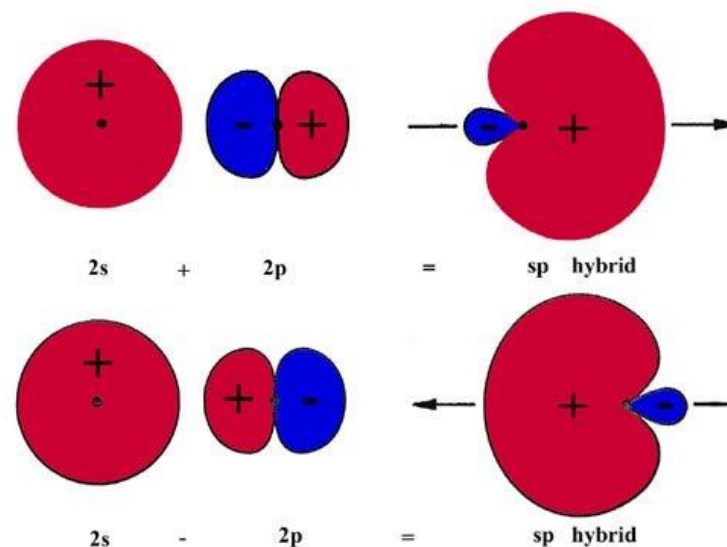
Hybridization

The BeH_2 molecule is linear and the two Be-H bonds are equivalent. The valence bond description of BeH_2 accounted for the two-fold valency of Be (which has the ground state configuration $1s^2 2s^2$) by assuming the bonding to occur with a promoted configuration of Be:



At first sight this suggests that the two Be-H bonds should be dissimilar and not necessarily 180° apart because one bond results from the overlap with a 2s orbital and the other with a 2p orbital on Be.

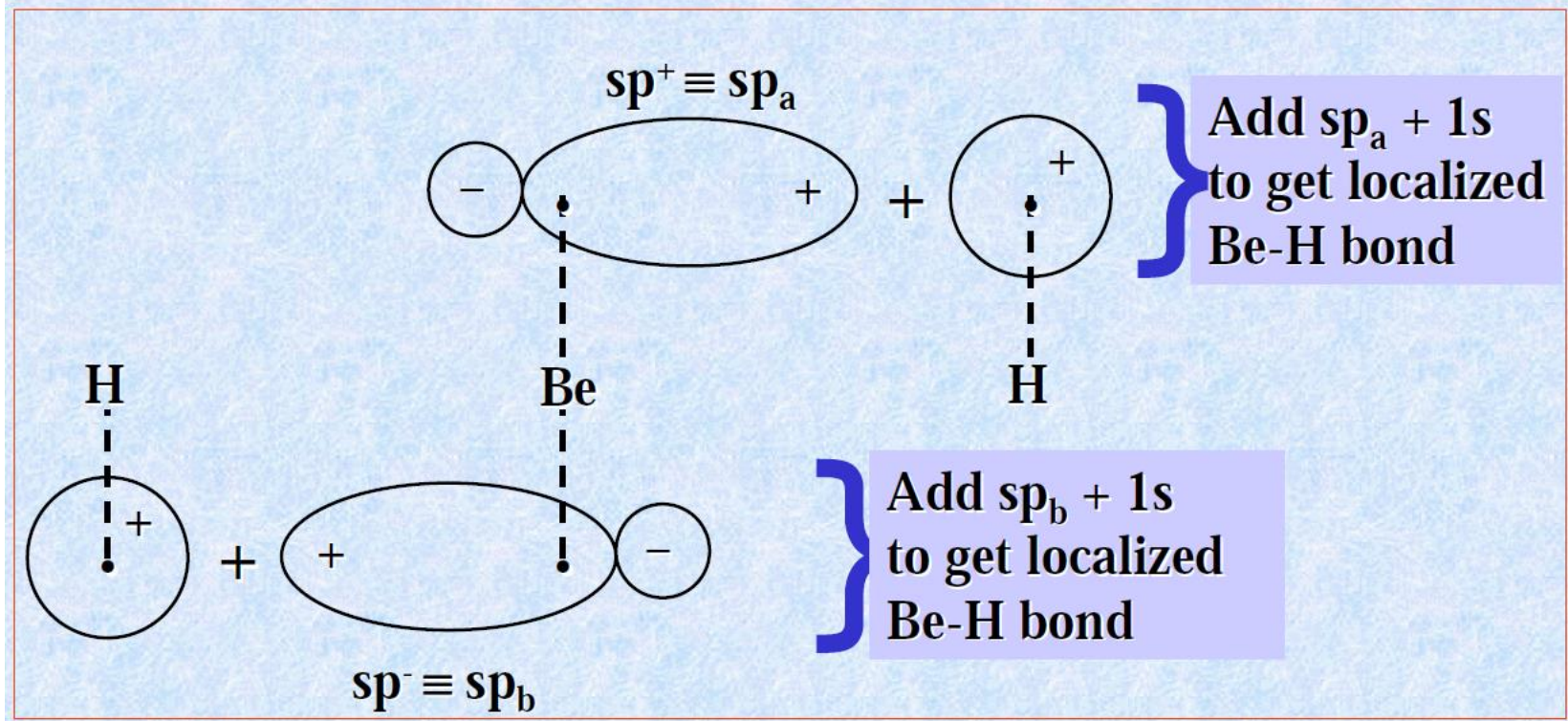
The construction of the hybrid orbitals is accomplished by taking the sum and the difference of the 2s orbital and one of the 2p orbitals, say the $2p_x$ orbital, both orbitals being centred on the Be nucleus.



Localized BeH_2 orbitals:

Begin by hybridizing Be 2s, 2p orbitals to give two sp hybrid orbitals -- each pointing in opposite direction:

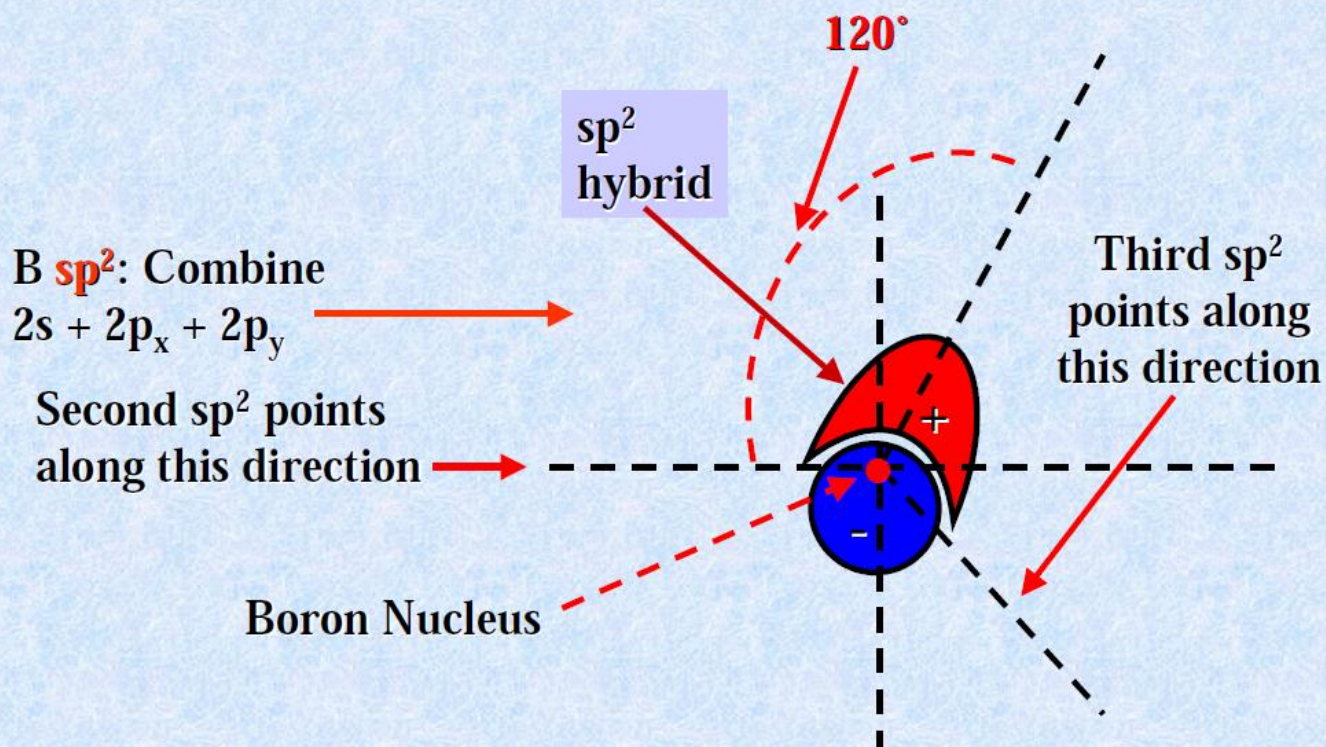
$$sp^+ = 2s + 2p \equiv sp_a \quad \text{and} \quad sp^- = 2s - 2p \equiv sp_b \longrightarrow$$



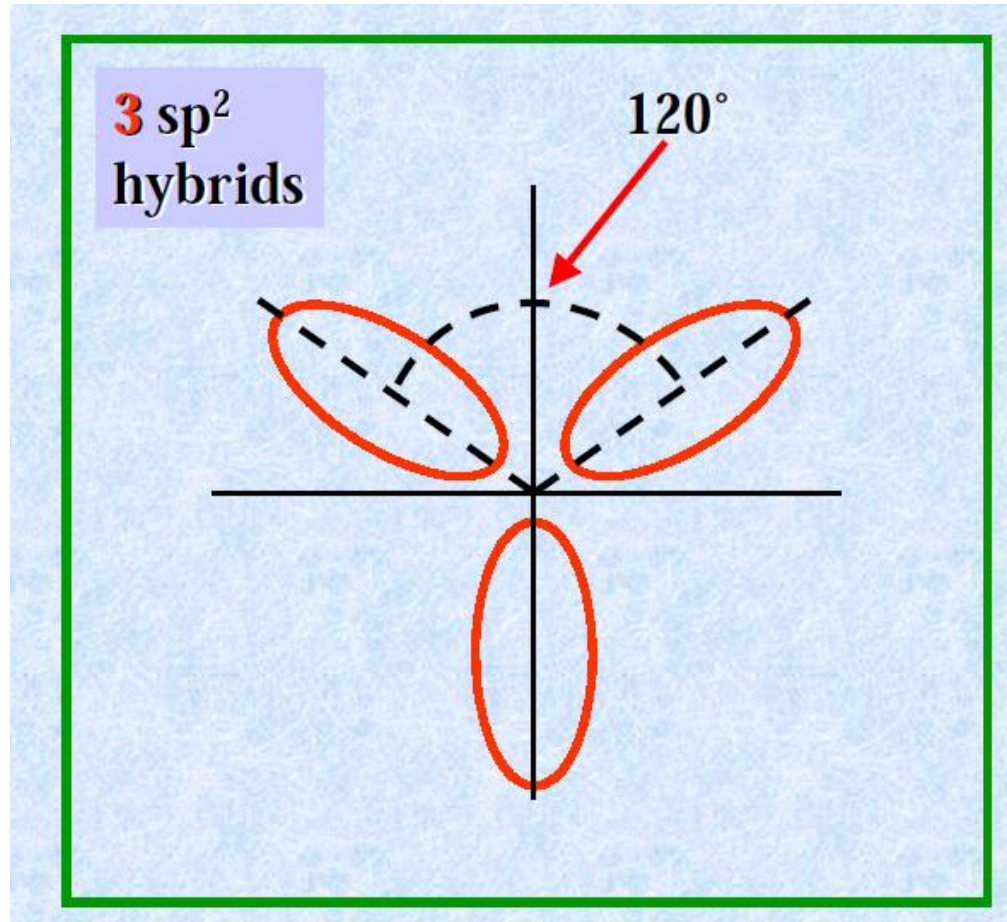
BH₃ Fragment:

Can explain by **sp² hybrid** which provides 3 localized bonds.

Note: (B 1s²2s²2p - 3 valence e⁻)



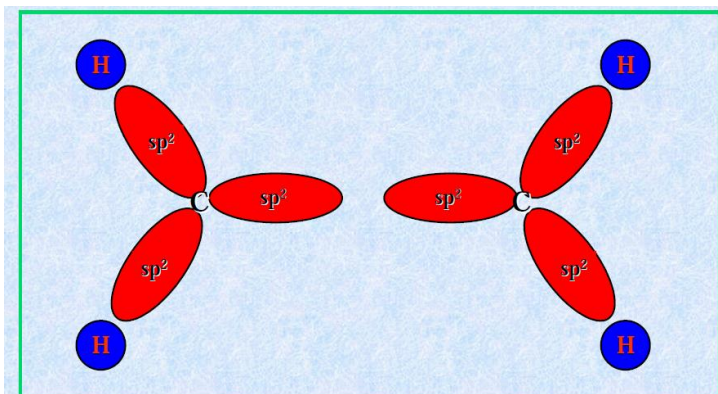
Three of these in a plane pointing at 120° with respect to each other



Summary of Hybridization Results

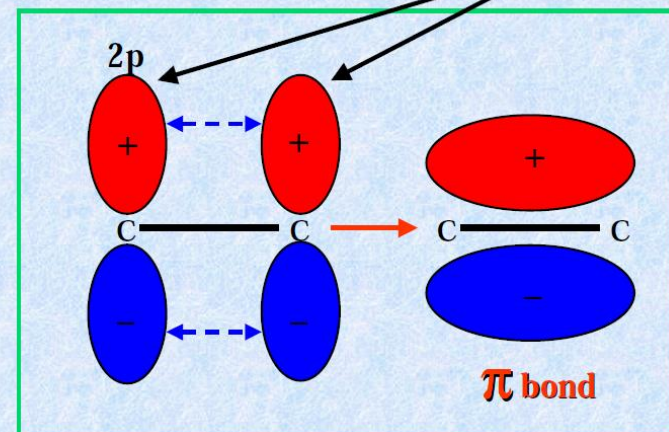
| Example | Groups Attached to Center Atom | Hybrid | Geometry |
|----------------|--------------------------------|--------|--|
| BeH_2 | 2 | sp | linear H-Be-H |
| BH_3 | 3 | sp^2 | trig. plane (120°) |
| CH_4 | 4 | sp^3 | tetrahedral ($109^\circ 28'$ H-C-H angles) |

Carbon can also exhibit sp^2 hybridization: C_2H_4 (ethylene)



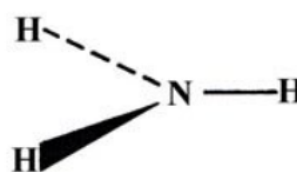
This leaves us still with 1 e^- on each C and one unused 2p orbital.

Now **overlap** 2p
orbitals to give
 π bond.



Molecular geometry

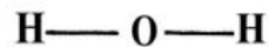
The theory of valency is known as valence bond theory. One further feature of this theory is that it may be used to predict (or in some cases, rationalize) the observed geometries of molecules. The geometry of a molecule means the relative arrangement of the nuclei in three-dimensional space. For example, assuming the two O-H bonds in the water molecule to be similar and hence of the same length, the angle formed by the two O-H bonds (the HOH angle) could conceivably possess any angle from 180° to some relatively small value. All we demand of our simple theory is that it correctly predict whether the water molecule is linear (bond angle = 180°) or bent (bond angle less than 180°). Or as another example, it should predict whether the ammonia molecule is planar or pyramidal.



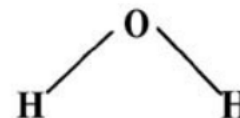
ERASMUS+

Molecular geometry

The observed geometry of a molecule is that which makes the energy of the system a minimum. Thus those geometries will be favoured which (i) concentrate the largest amount of charge density in the binding region and thus give the strongest individual bonds, and (ii) keep the nuclei as far apart as possible (consistent with (i)), and hence reduce the nuclear repulsions. Consider again the two possibilities for the water molecule. It is clear that the linear form (a) will have a smaller energy of nuclear repulsion from the hydrogens than will the bent form (b).



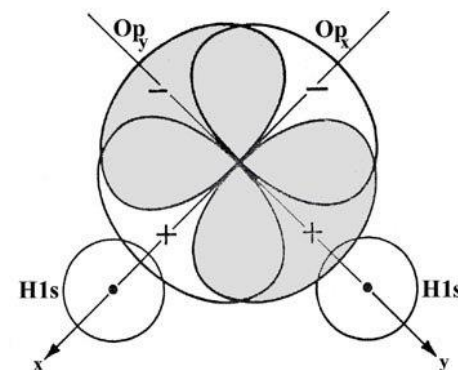
(a)



(b)

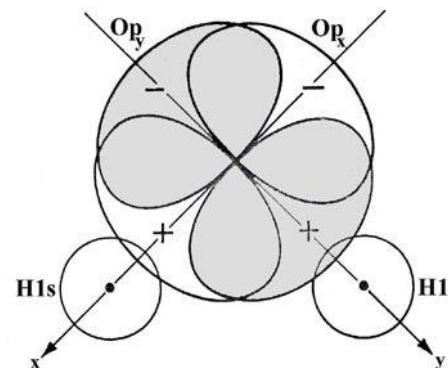
Molecular geometry

If the amount of electron density which could be concentrated in the regions between the nuclei in each O-H bond (i.e., the strength of each O-H bond) was independent of the bond angle, then clearly the linear form of the water molecule would be the most stable. This would be the situation if all the atomic orbitals which describe the motions of the electrons were rigidly spherical and centred on the nuclei. But this is not the case. As was stressed earlier in our discussion of atomic orbitals, the motion of electrons possessing angular momentum because they occupy orbitals with $l \neq 0$ is concentrated along certain axes or planes in space. In particular the three p orbitals are a maximum along the three perpendicular axes in space. The valence bond theory of the water molecule describes the two O-H bonds as resulting from the overlap of the H 1s orbitals with the two half-filled 2p orbitals of the oxygen atom. Since the two 2p orbitals are at right angles to one another, valence bond theory predicts a bent geometry for the water molecule with a bond angle of 90° .



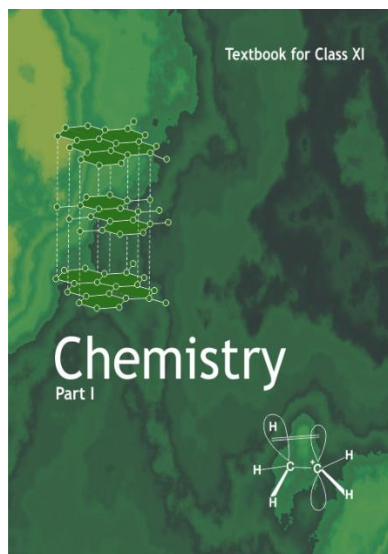
Molecular geometry

The actual bond angle in the water molecule is 104.5° . The opening of the angle to a value greater than the predicted one of 90° can be accounted for in terms of a lessening of the repulsion between the hydrogen nuclei. The assumption we have made is that the maximum amount of electron density will be transferred to the binding region and hence yield the strongest possible bond when the hydrogen and oxygen nuclei lie on the axis which is defined by the direction of the 2p orbital. For a given internuclear separation, this will result in the maximum overlap of the orbitals. Because an orbital with $l = 1$ restricts the motion of the electron to certain preferred directions in space, bond angles and molecular geometry will be determined, to a first rough approximation, by the inter-orbital angles.

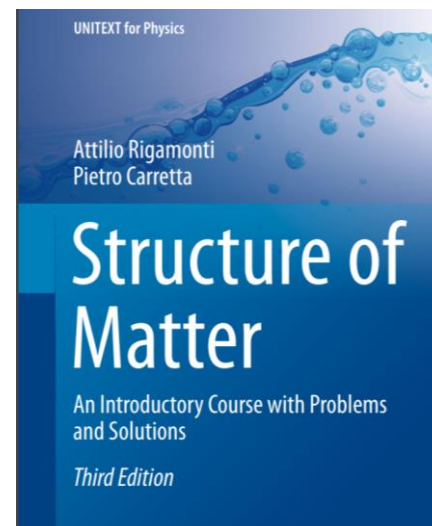


Review Questions

- How the chemical bonds are formed?
- Explain the term molecular orbitals
- Explain the hybridization
- What the geometry of the molecules depends on?



<https://www.neetprep.com/ncert/1591-Chemical-Bonding-Molecular-Structure-Chemical-Bonding-Molecular-Structure--NCERT-Chapter-PDF>



Rigamonti, A., & Carretta, P. (2015). Structure of Matter. UNITEXT for Physics, Springer International Publishing Switzerland



Module 1. Molecular basis of life

Topic 1. Structure of Matter

Lesson 3. Intermolecular forces. Hydrogen bonding, electrostatic and van der Waals interactions



Contents

- Introduction
- Hydrogen Bonding
- Hydrophobic Interactions
- Van der Waals Forces
- Specific Interactions



Introduction

Intermolecular forces are the attractive or repulsive forces between molecules. They are separated into two groups; short range and long range forces. Short range forces happen when the centers of the molecules are separated by three angstroms (10^{-8} cm) or less. Short range forces tend to be repulsive, where the long range forces that act outside the three angstroms range are attractive. Long range forces are also known as Van der Waals forces. They are responsible for surface tension, friction, viscosity and differences between actual behavior of gases and that predicted by the ideal gas law. Intermolecular forces are responsible for most properties of all the phases. The viscosity, diffusion, and surface tension are examples of physical properties of liquids that depend on intermolecular forces. Vapor pressure, critical point, and boiling point are examples of properties of gases. Melting and sublimation are examples of properties of solids that depend on intermolecular forces.

Hydrogen Bonding

A hydrogen bond is a special type of dipole-dipole attraction which occurs when a hydrogen atom bonded to a strongly electronegative atom exists in the vicinity of another electronegative atom with a lone pair of electrons. These bonds are generally stronger than ordinary dipole-dipole and dispersion forces, but weaker than true covalent and ionic bonds.

For a hydrogen bond to occur there must be both a hydrogen donor and an acceptor present. The donor in a hydrogen bond is the atom to which the hydrogen atom participating in the hydrogen bond is covalently bonded, and is usually a strongly electronegative atom such as N, O, or F. The hydrogen acceptor is the neighboring electronegative ion or molecule, and must possess a lone electron pair in order to form a hydrogen bond.

Since the hydrogen donor is strongly electronegative, it pulls the covalently bonded electron pair closer to its nucleus, and away from the hydrogen atom. The hydrogen atom is then left with a partial positive charge, creating a dipole-dipole attraction between the hydrogen atom bonded to the donor, and the lone electron pair on the acceptor.

Types of hydrogen bonds

Hydrogen bonds can occur within one single molecule, between two like molecules, or between two unlike molecules.

Intramolecular hydrogen bonds: Intramolecular hydrogen bonds are those which occur within one single molecule. This occurs when two functional groups of a molecule can form hydrogen bonds with each other. In order for this to happen, both a hydrogen donor and an acceptor must be present within one molecule, and they must be within close proximity of each other in the molecule. For example, intramolecular hydrogen bonding occurs in ethylene glycol ($\text{C}_2\text{H}_4(\text{OH})_2$) between its two hydroxyl groups due to the molecular geometry.

Intermolecular hydrogen bonds: Intermolecular hydrogen bonds occur between separate molecules in a substance. They can occur between any number of like or unlike molecules as long as hydrogen donors and acceptors are present and in positions in which they can interact. For example, intermolecular hydrogen bonds can occur between NH_3 molecules alone, between H_2O molecules alone, or between NH_3 and H_2O molecules.

Properties and effects of hydrogen bonds

- higher boiling points
- higher viscosity

Factors preventing Hydrogen bonding

- Electronegativity: Hydrogen bonding cannot occur without significant electronegativity differences between hydrogen and the atom it is bonded to.
- Atom Size: The size of donors and acceptors can also effect the ability to hydrogen bond.

Hydrogen Bonding in Nature

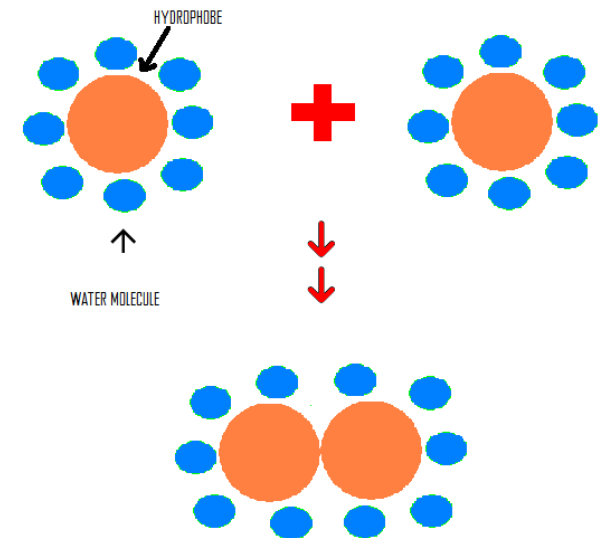
Hydrogen bonding plays a crucial role in many biological processes and can account for many natural phenomena such as the unusual properties of water. In addition to being present in water, hydrogen bonding is also important in the water transport system of plants, secondary and tertiary protein structure, and DNA base pairing.

- Plants
- Proteins
- DNA

Hydrophobic Interactions

Hydrophobic interactions describe the relations between water and hydrophobes (low water-soluble molecules). Hydrophobes are nonpolar molecules and usually have a long chain of carbons that do not interact with water molecules. The mixing of fat and water is a good example of this particular interaction. The common misconception is that water and fat doesn't mix because the Van der Waals forces that are acting upon both water and fat molecules are too weak. However, this is not the case. The behavior of a fat droplet in water has more to do with the enthalpy and entropy of the reaction than its intermolecular forces.

American chemist Walter Kauzmann discovered that nonpolar substances like fat molecules tend to clump up together rather than distributing itself in a water medium, because this allow the fat molecules to have minimal contact with water.



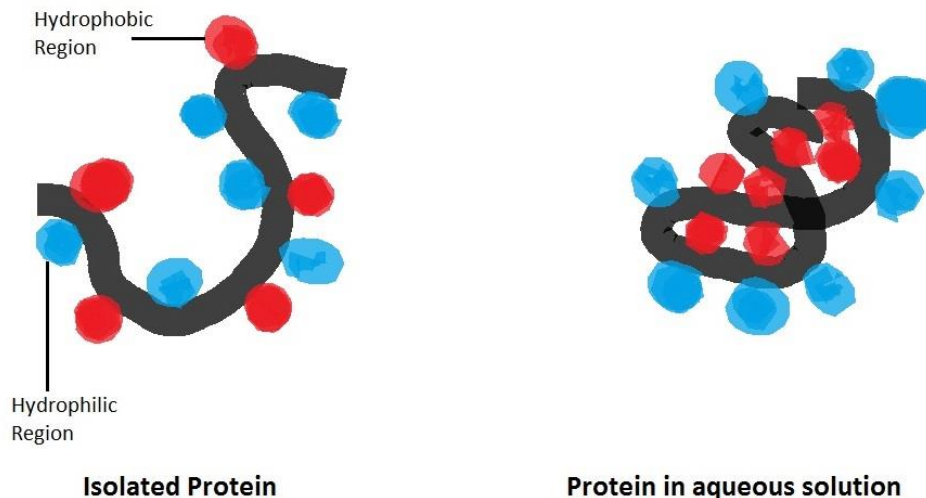
Strength of Hydrophobic Interactions

Hydrophobic interactions are relatively stronger than other weak intermolecular forces (i.e., Van der Waals interactions or Hydrogen bonds). The strength of Hydrophobic Interactions depend on several factors including (in order of strength of influence):

1. Temperature: As temperature increases, the strength of hydrophobic interactions increases also. However, at an extreme temperature, hydrophobic interactions will denature.
2. Number of carbons on the hydrophobes: Molecules with the greatest number of carbons will have the strongest hydrophobic interactions.
3. The shape of the hydrophobes: Aliphatic organic molecules have stronger interactions than aromatic compounds. Branches on a carbon chain will reduce the hydrophobic effect of that molecule and linear carbon chain can produce the largest hydrophobic interaction. This is so because carbon branches produce steric hindrance, so it is harder for two hydrophobes to have very close interactions with each other to minimize their contact to water.

Biological Importance of Hydrophobic Interactions

Hydrophobic Interactions are important for the folding of proteins. This is important in keeping a protein stable and biologically active, because it allows the protein to decrease its surface area and reduce the undesirable interactions with water. Besides from proteins, there are many other biological substances that rely on hydrophobic interactions for their survival and functions, like the phospholipid bilayer membranes in every cell of your body!



Van Der Waals Interactions

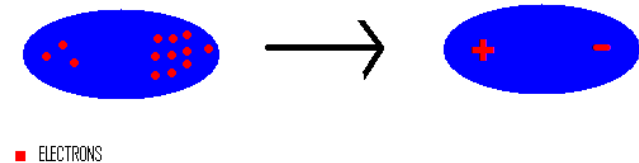
Van der Waals forces are driven by induced electrical interactions between two or more atoms or molecules that are very close to each other. Van der Waals interaction is the weakest of all intermolecular attractions between molecules. However, with a lot of Van der Waals forces interacting between two objects, the interaction can be very strong.

Causes of Van der Waals Forces

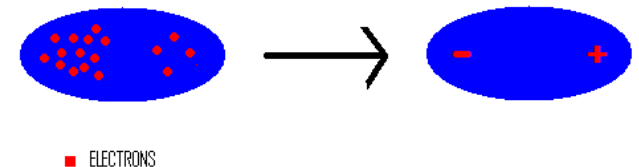
Quantum Mechanics strongly emphasizes the constant movement of electrons in an atom through the Schrödinger Equation and the Heisenberg's Uncertainty Principle. The Heisenberg's Uncertainty Principle proposes that the energy of the electron is never zero; therefore, it is constantly moving around its orbital. The square of the Schrödinger Equation for a particle in a box suggests that it is probable of finding the electron (particle) anywhere in the orbital of the atom (box).

These two important aspects of Quantum Mechanics strongly suggest that the electrons are constantly are moving in an atom, so dipoles are probable of occurring. A dipole is defined as molecules or atoms with equal and opposite electrical charges separated by a small distance.

It is probable to find the electrons in this state:



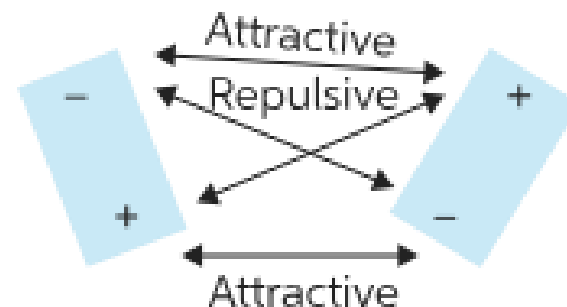
This is how spontaneous (or instantaneous) dipoles occur. When groups of electrons move to one end of the atom, it creates a dipole. These groups of electrons are constantly moving so they move from one end of the atom to the other and back again continuously. Therefore, the opposite state is as probable of occurring.



Opposite state due to fluctuation of dipoles:

Electrostatic interactions

The intermolecular forces of attraction in which complete or partial ionic species are attracted to each other are termed as electrostatic interactions. These attraction forces do not include any sharing of electrons between atoms. So, they are also named as non-covalent bonds. The term electrostatic interactions comprise both attractive and repulsive forces ionic species, which means ions with opposite charges are attracted to each other, whereas, the similar charges repel from each other. In liquids, molecules that are held together by intermolecular interactions are comparably weaker than the intramolecular interactions within polyatomic ions and atoms of molecules. These are very important in describing the formation of different molecules. Due to changes in intermolecular interactions, the transitions occur between the solid and liquid or the liquid and gas phase.

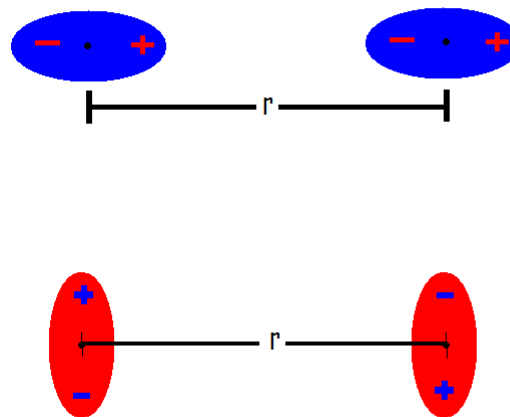


Types of Electrostatic interactions

- The interactions between two dipolar molecules through space is called **dipole-dipole interaction**.
- In 1930, Fritz London, a German physicist stated that there are non-permanent fluctuations distribution of electrons within atoms and non-polar molecules, which lead to short-lived rapid dipole moments formation and produces attractive forces known as **London dispersion forces**.
- A strong dipole-dipole type of interaction is called **hydrogen bonding**.
- **Ion-Ion Interaction** These types of interactions are responsible for ionic bonding.
- **Ion-dipole Interaction** When ionic substances dissolve in a polar substance, such as water and ion-dipole interactions will take place.

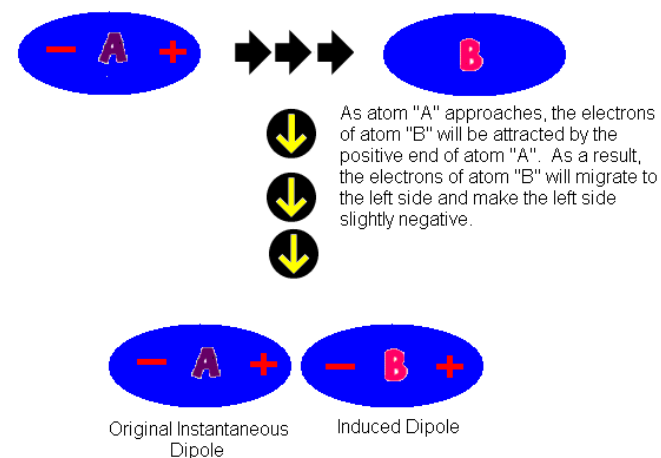
Dipole-Dipole Interaction

Dipole-Dipole interactions occur between molecules that have permanent dipoles; these molecules are also referred to as polar molecules. The figure below shows the electrostatic interaction between two dipoles.



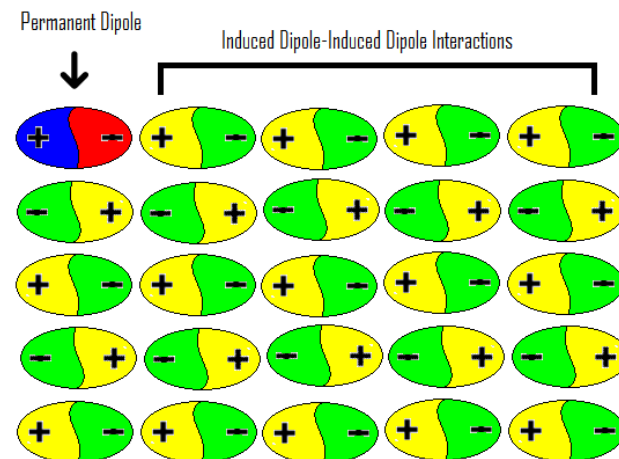
Induced Dipoles

An induced dipole moment is a temporary condition during which a neutral nonpolar atom (i.e. Helium) undergo a separation of charges due to the environment. When an instantaneous dipole atom approaches a neighboring atom, it can cause that atom to also produce dipoles. The neighboring atom is then considered to have an induced dipole moment.



Spontaneous Dipole-Induced Dipole Interaction

Spontaneous dipole-induced dipole interactions are also known as dispersion or London forces (name after the German physicist Fritz London). They are large networks of intermolecular forces between nonpolar and non-charged molecules and atoms (i.e. alkanes, noble gases, and halogens). Molecules that have induced dipoles may also induce neighboring molecules to have dipole moments, so a large network of induced dipole-induced dipole interactions may exist. The image below illustrates a network of induced dipole-induced dipole interactions.



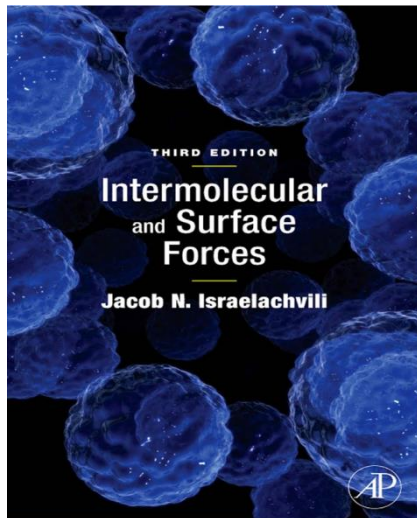


Applications of Electrostatic interactions

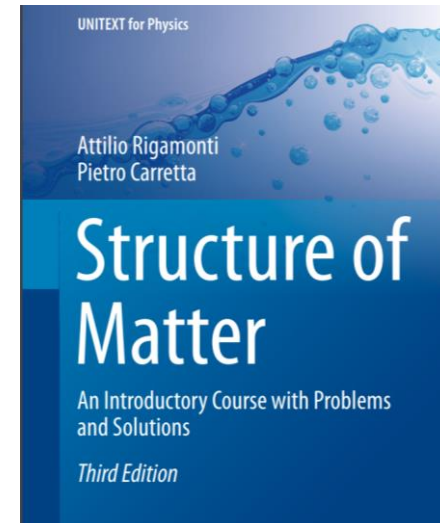
Electrostatic interactions become significant for gases at high pressures as these are accountable for their observed deviations from the ideal gas law at high pressures. Such interactions are important in maintaining the 3-D structure of larger molecules. For example, nucleic acids and proteins. In several biological processes, where larger molecules bind specifically but transiently to one, such interactions are involved. Additionally, these interactions also deeply influence crystallinity and design of materials, in general, for many organic molecules synthesis and particularly for self-assembly.

Review Questions

- What types of electrostatic interactions exist?
- Explain the characteristics of van der Waals interactions
- Explain the importance of hydrogen bonding in life science



Jacob N. Israelachvili (2011)
Intermolecular and surface forces
Elsevier Inc.



Rigamonti, A., & Carretta, P.
(2015). Structure of Matter.
UNITEXT for Physics, Springer
International Publishing
Switzerland



Module 1. Molecular basis of life

Topic 1. Structure of Matter

Lesson 4. Brownian motion and kinetic theory



Contents

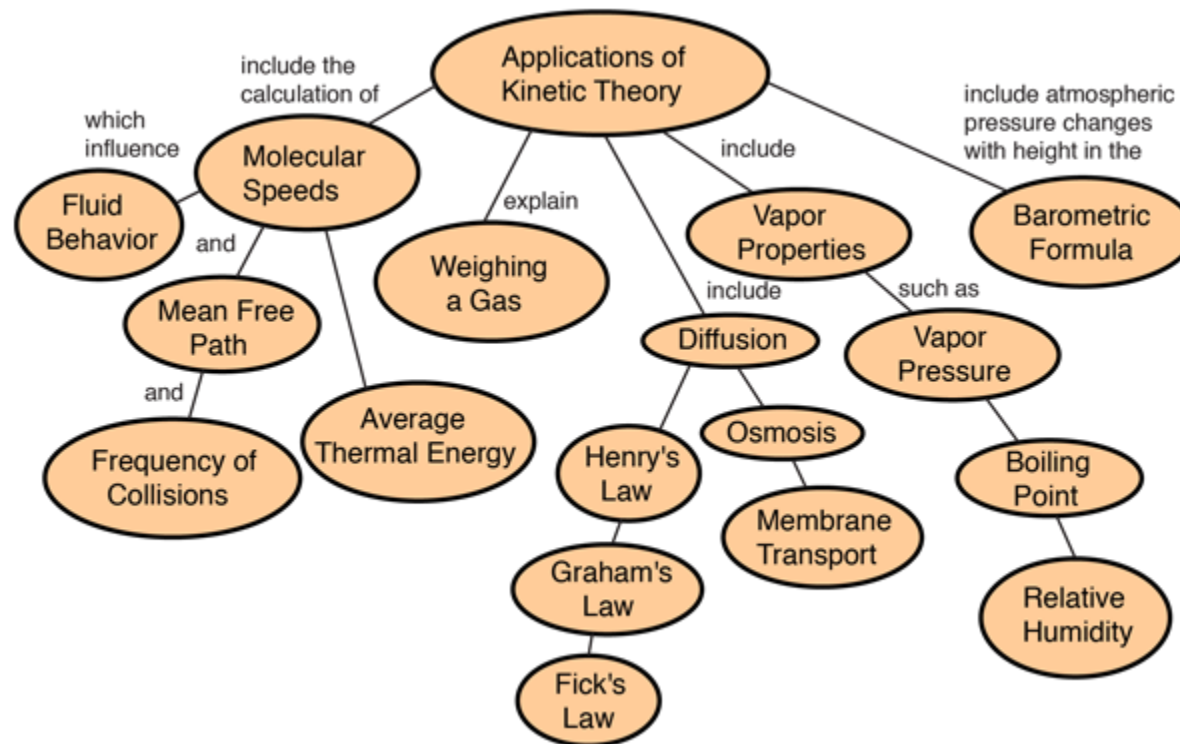
- Introduction
- Kinetic Theory
- The Ideal Gas
- Brownian Motion
- The Structure of Gases and Liquids



Introduction

Intermolecular forces are the attractive or repulsive forces between molecules. They are separated into two groups; short range and long range forces. Short range forces happen when the centers of the molecules are separated by three angstroms (10^{-8} cm) or less. Short range forces tend to be repulsive, where the long range forces that act outside the three angstroms range are attractive. Long range forces are also known as Van der Waals forces. They are responsible for surface tension, friction, viscosity and differences between actual behavior of gases and that predicted by the ideal gas law. Intermolecular forces are responsible for most properties of all the phases. The viscosity, diffusion, and surface tension are examples of physical properties of liquids that depend on intermolecular forces. Vapor pressure, critical point, and boiling point are examples of properties of gases. Melting and sublimation are examples of properties of solids that depend on intermolecular forces.

Application Of Assumptions Of The Kinetic Theory



Assumptions of the Kinetic Molecular Theory

The kinetic theory involves a number of assumptions that focus on being able to talk about an ideal gas.

- Molecules are treated as point particles. Specifically, one implication of this is that their size is extremely small in comparison to the average distance between particles.
- The number of molecules (N) is very large, to the extent that tracking individual particle behaviors is not possible. Instead, statistical methods are applied to analyze the behavior of the system as a whole.
- Each molecule is treated as identical to any other molecule. They are interchangeable in terms of their various properties. This again helps support the idea that individual particles don't need to be kept track of, and that the statistical methods of the theory are sufficient to arrive at conclusions and predictions.
- Molecules are in constant, random motion. They obey Newton's laws of motion.
- Collisions between the particles, and between the particles and walls of a container for the gas, are perfectly elastic collisions.
- Walls of containers of gases are treated as perfectly rigid, do not move, and are infinitely massive (in comparison to the particles).

The Ideal Gas Law

The kinetic theory of gases is significant, in that the set of assumptions above lead us to derive the ideal gas law, or ideal gas equation, that relates the pressure (p), volume (V), and temperature (T), in terms of the Molar Gas constant, 8.31 J/molK (R) and the number of mols (n). The resulting ideal gas equation is:

$$pV = nRT$$

Average Molecular Kinetic Energy

The average kinetic energy of each molecule of a gas in a container can also be calculated using the equation below:

$$E_k = \frac{1}{2}m(c_{\text{rms}})^2 = \frac{3}{2}kT = \frac{3}{2} \frac{RT}{N_A}$$

- E_k = the **average kinetic energy** in joules (J)
- m = the **mass of each gas molecule** in kilograms (kg)
- k = the **Boltzmann constant** ($= 1.38 \times 10^{-23} \text{ kgm}^2\text{s}^{-2}\text{K}^{-1}$)
- T = the **temperature** in kelvin (K)
- R = the **molar gas constant** ($= 8.31 \text{ kgm}^2\text{s}^{-2}\text{K}^{-1}\text{mol}^{-1}$)
- N_A = **Avogadro's number** ($= 6.02 \times 10^{23}$)

Rates of diffusion and kinetic theory

Diffusion is the process by which the molecules of one substance mix with those of another as a result of their motion. The substances move freely from a region of high concentration to a region of low concentration at their own pace. The rate of diffusion depends on the temperature and the density of the substances involved.

Diffusion

Diffusion is the process whereby gas molecules mix up intimately with one another as a result of the kinetic energy of the molecules. According to Graham's law of diffusion, the time rate at which gases diffuse is inversely proportional to the square root of its density provided that temperature is kept constant. This is written mathematically as:

$$t \propto \frac{1}{\sqrt{d}}$$

t = rate of diffusion

d = molecular density of the gas

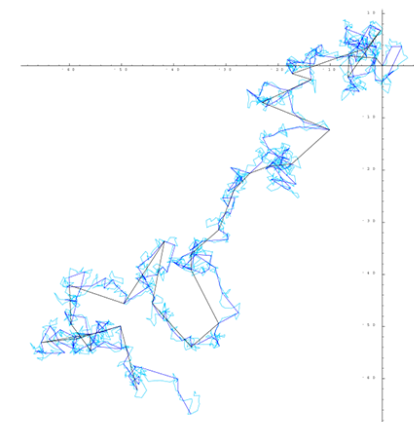
This means that, the heavier the gas, the slower the movement of the gas.

Kinetic Theory of Gases Postulates

- Gases have been made up of a large number of tiny particles (atoms and molecules). Such particles are extremely small in comparison to the distance between them. The individual particle size is defined as negligible, and the majority of the volume occupied by the gas is empty space.
- These molecules seem to be in constant random motion, resulting in collisions with one another and with the container's walls. Whenever the molecules of gas collide with the walls of a container, they impart momentum to the walls and this results in the generation of a measurable force. So, if we divide this force by the area, we get the pressure.
- Collisions among molecules and walls are completely elastic. It thus means that when the molecules collide, they retain their kinetic energy. Molecules don't ever slow down and will always move at the same rate.
- The average kinetic energy of the gas particles has been affected by temperature. That is, the higher the temperature, the greater the average kinetic energy of the gas.
- Apart from when they collide, the molecules have no attraction or repulsion for one another.

Brownian Motion

Brownian motion is the seemingly random motion of particles, atoms, or molecules that emerges out of the random collisions of those particles. Brownian motion can be observed as light shines through a window. Particles of dust or pollen can be seen in the light floating in the air, following what seems to be random jittery patterns. The dust particles aren't moving on their own, but are colliding with molecules of the air keeping the dust in motion.



Causes of Brownian Motion

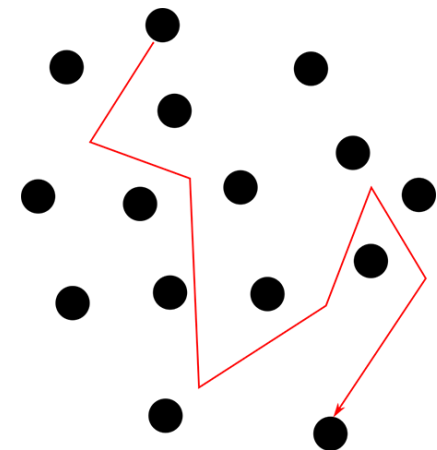
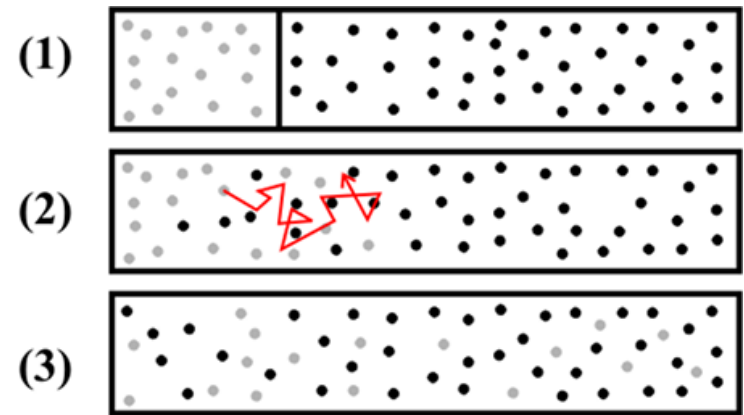
In order to understand Brownian motion and how the motion of the dust particles in the air actually isn't all that random, a few things need to be understood about particles such as atoms and molecules. Atoms and molecules behave in different ways depending on their state of matter. In order to understand Brownian motion, the structures of gases and liquids and how their molecules or atoms move must first be understood.

Brownian Motion Examples

- The motion of pollen grains on still water.
- Movement of dust motes in a room (although largely affected by air currents)
- Diffusion of pollutants in the air.
- Diffusion of calcium through bones.

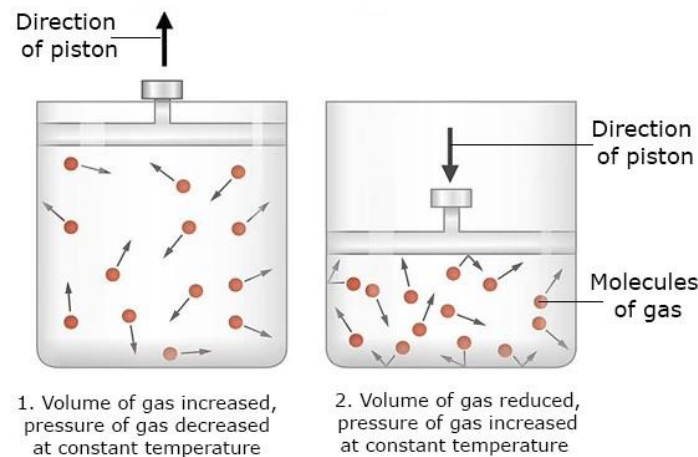
The Structure of Gases and Liquids

All matter is in motion according to kinetic theory. Particles of a solid are compact and do not freely move past each other. However, the molecules or atoms of a solid are still vibrating which emerges and is detected as heat. This kinetic energy or heat in particles is more apparent and consequential in liquids or gases. Matter in liquid and gases are freer to move and tend toward equilibrium. That is, atoms and molecules in liquids and gases move toward areas of lower concentrations and away from areas of higher concentrations. This is called diffusion; this phenomenon is observable in liquids when one liquid dissolves in another, and in gases as smoke spreads to fill a room.



Pressure of a Gas

If the volume of a fixed mass of a gas is reduced at constant temperature, the molecules will be compressed and become more closely packed together as shown in the figure below. Under this condition, the time rate of collision of the molecules with themselves and the walls of the container increases thereby increasing the pressure of the gas. If on the other hand, the volume is increased by raising the piston up at constant temperature, the time rate of collision of the molecules will decrease. They will take longer time to collide with the walls of the container. This will lead to a decrease in the pressure. This agrees with Boyle's law which states that, 'the volume of a fixed mass of a gas at constant temperature is inversely proportional to its pressure.'



Osmosis

Osmosis is the tendency of a solvent to pass from a dilute solution, through a semi permeable membrane into a concentrated solution. Semi-permeable membrane is a substance such as cellophane, parchment or vegetable material which would allow some molecules of liquid to diffuse through them but not others. Such a membrane may allow the molecules of a solvent to pass through it but not those of a solute.

Viscosity

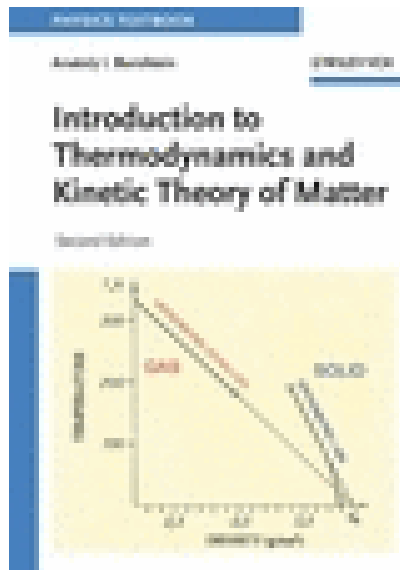
Viscosity by definition is internal friction between layers of fluids in motion. Liquids that are dense pour more slowly than those that are less dense. E.g honey will pour more slowly than water because it is denser than kerosene. This means that honey is a more viscous liquid than water. Viscosity can be demonstrated if we consider a ball bearing falling through some liquids.

Applications of Surface Tension and Viscosity

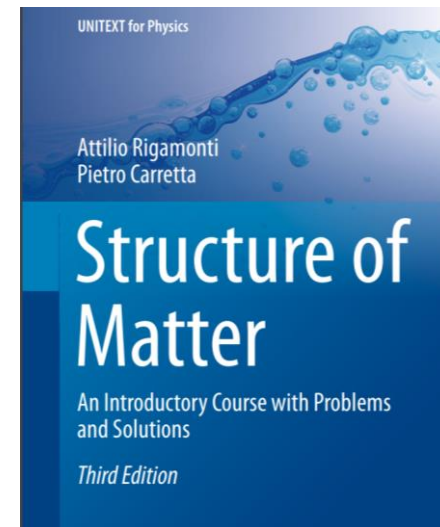
1. The knowledge of surface tension is applied in industries in making some materials such as, umbrellas, canvass, rain coats and waterproof tents
2. It is difficult to wash dirty clothes or oily clothes with water only. That is why we use soap and detergents to wash. Soap and detergents weakens the surface tension of water and enables it to float away dirt or oil from the material
3. Viscous liquids are used as lubricants. Examples are grease and engine oil.
4. Viscosity is applied in the design of boats, ships and aircraft.

Review Questions

- What are Assumptions of the Kinetic Molecular Theory
- Explain the Kinetic Theory of Gases
- Explain Brownian Motion Examples



A. I. Burshstein(2005)
Introduction to Thermodynamics
and Kinetic Theory of Matter
WILEY-VCH Verlag GmbH & Co.
KGaA



Rigamonti, A., & Carretta, P.
(2015). Structure of Matter.
UNITEXT for Physics, Springer
International Publishing
Switzerland



Module 1. Molecular basis of life

Topic 1. Structure of Matter

Lesson 5. From Solid to Liquid

Contents

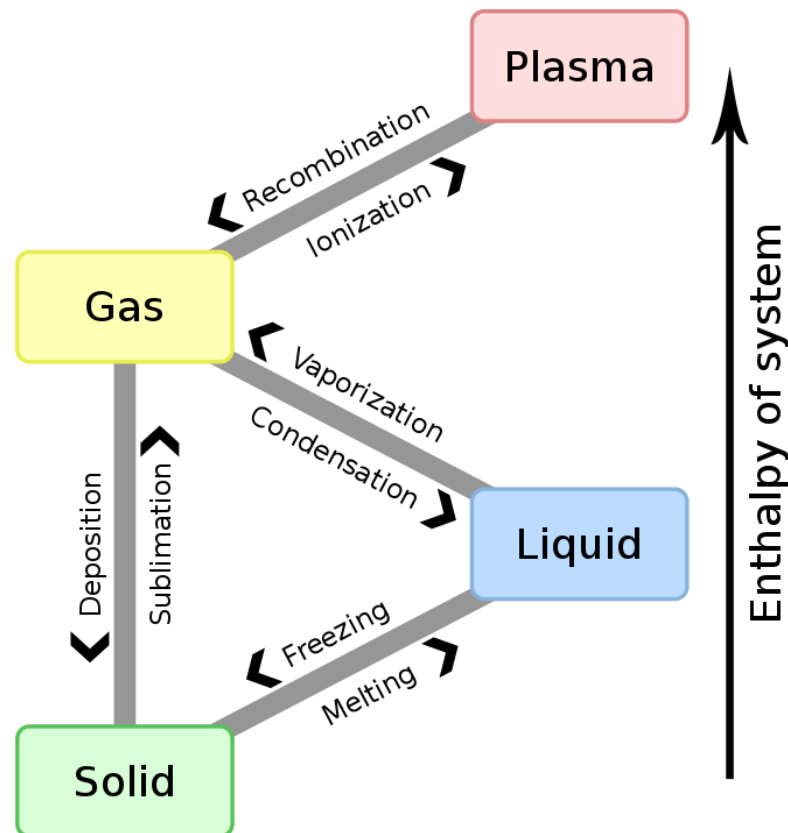
- Introduction
- States of matter
- Characteristics of the Three States of Matter
- Enthalpy of fusion
- Liquid-gas phase change
- Enthalpy of vaporization
- Heating Curves



Introduction

Substances can change phase — often because of a temperature change. At low temperatures, most substances are solid; as the temperature increases, they become liquid; at higher temperatures still, they become gaseous.

States of matter





Process of change of state from solid to liquid is called:

- (a). Melting
- (b). Freezing
- (c). Boiling
- (d). Condensation

Characteristics of the Three States of Matter

| Characteristic | Solid | Liquid | Gas |
|--|-------------------------------|--------------------------|----------------------------------|
| shape | definite | indefinite | indefinite |
| volume | definite | definite | indefinite |
| relative intermolecular interaction strength | strong | moderate | weak |
| relative particle positions | in contact and fixed in place | in contact but not fixed | not in contact, random positions |

Enthalpy of fusion

The process of a solid becoming a liquid is called melting (an older term that you may see sometimes is fusion). The opposite process, a liquid becoming a solid, is called solidification. For any pure substance, the temperature at which melting occurs — known as the melting point — is a characteristic of that substance. It requires energy for a solid to melt into a liquid. Every pure substance has a certain amount of energy it needs to change from a solid to a liquid. This amount is called the enthalpy of fusion (or heat of fusion) of the substance, represented as ΔH_{fus} . The unit of ΔH_{fus} is kilojoules per mole, so we need to know the quantity of material to know how much energy is involved. The ΔH_{fus} is always tabulated as a positive number. However, it can be used for both the melting and the solidification processes as long as you keep in mind that melting is always endothermic (so ΔH will be positive), while solidification is always exothermic (so ΔH will be negative).

Enthalpies of Fusion for Various Substances

| Substance (Melting Point) | ΔH_{fus} (kJ/mol) |
|---------------------------|----------------------------------|
| Water (0°C) | 6.01 |
| Aluminum (660°C) | 10.7 |
| Benzene (5.5°C) | 9.95 |
| Ethanol (−114.3°C) | 5.02 |
| Mercury (−38.8°C) | 2.29 |

What happens when a solid becomes a liquid?

During melting, energy goes exclusively to changing the phase of a substance; it does not go into changing the temperature of a substance. Hence melting is an isothermal process because a substance stays at the same temperature. Only when all of a substance is melted does any additional energy go to changing its temperature.

In a solid, individual particles are stuck in place because the intermolecular forces cannot be overcome by the energy of the particles. When more energy is supplied (e.g., by raising the temperature), there comes a point at which the particles have enough energy to move around but not enough energy to separate. This is the liquid phase: particles are still in contact but are able to move around each other. This explains why liquids can assume the shape of their containers: the particles move around and, under the influence of gravity, fill the lowest volume possible (unless the liquid is in a zero-gravity environment).

“Liquids and Gravity.” (a) A liquid fills the bottom of its container as it is drawn downward by gravity and the particles slide over each other. (b) A liquid floats in a zero-gravity environment. The particles still slide over each other because they are in the liquid phase, but now there is no gravity to pull them down.



(a)



(b)

Liquid-gas phase change

The phase change between a liquid and a gas has some similarities to the phase change between a solid and a liquid. At a certain temperature, the particles in a liquid have enough energy to become a gas. The process of a liquid becoming a gas is called boiling (or vapourization), while the process of a gas becoming a liquid is called condensation. However, unlike the solid/liquid conversion process, the liquid/gas conversion process is noticeably affected by the surrounding pressure on the liquid because gases are strongly affected by pressure. This means that the temperature at which a liquid becomes a gas, the boiling point, can change with surrounding pressure. Therefore, we define the normal boiling point as the temperature at which a liquid changes to a gas when the surrounding pressure is exactly 1 atm, or 760 torr. Unless otherwise specified, it is assumed that a boiling point is for 1 atm of pressure.

Enthalpy of vaporization

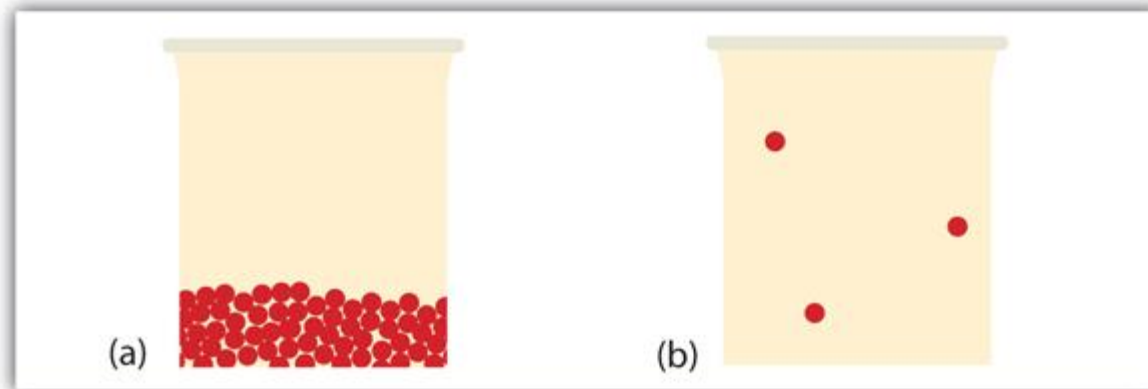
Like the solid/liquid phase change, the liquid/gas phase change involves energy. The amount of energy required to convert a liquid to a gas is called the enthalpy of vaporization (or heat of vaporization), represented as ΔH_{vap} . The unit for ΔH_{vap} is also kilojoules per mole, so we need to know the quantity of material to know how much energy is involved. The ΔH_{vap} is also always tabulated as a positive number. It can be used for both the boiling and the condensation processes as long as you keep in mind that boiling is always endothermic (so ΔH will be positive), while condensation is always exothermic (so ΔH will be negative).

Enthalpies of Vaporization for Various Substances

| Substance (Normal Boiling Point) | ΔH_{vap} (kJ/mol) |
|----------------------------------|----------------------------------|
| Water (100°C) | 40.68 |
| Bromine (59.5°C) | 15.4 |
| Benzene (80.1°C) | 30.8 |
| Ethanol (78.3°C) | 38.6 |
| Mercury (357°C) | 59.23 |

What happens when a liquid becomes a gas?

When a liquid becomes a gas, the particles separate from each other, with each particle going its own way in space. This is how gases tend to fill their containers. Indeed, in the gas phase most of the volume is empty space; only about one one-thousandth of the volume is actually taken up by matter.



Sublimation

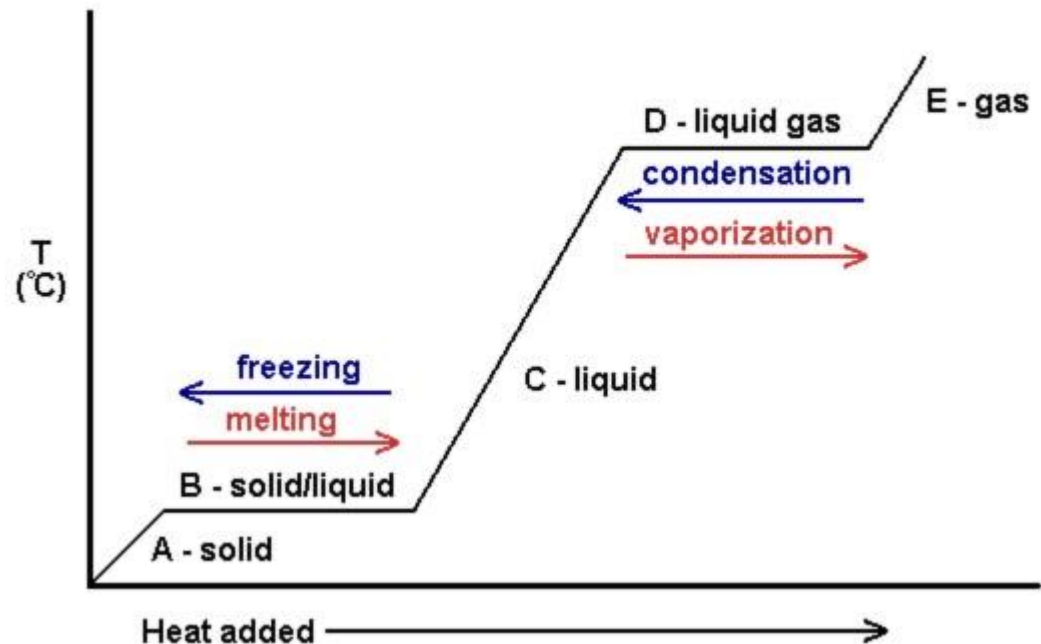
Under some circumstances, the solid phase can transition directly to the gas phase without going through a liquid phase, and a gas can directly become a solid. The solid-to-gas change is called sublimation, while the reverse process is called deposition. Sublimation is isothermal, like the other phase changes. There is a measurable energy change during sublimation; this energy change is called the enthalpy of sublimation, represented as ΔH_{sub} . The relationship between the ΔH_{sub} and the other enthalpy changes is as follows:

$$\Delta H_{\text{sub}} = \Delta H_{\text{fus}} + \Delta H_{\text{vap}}$$

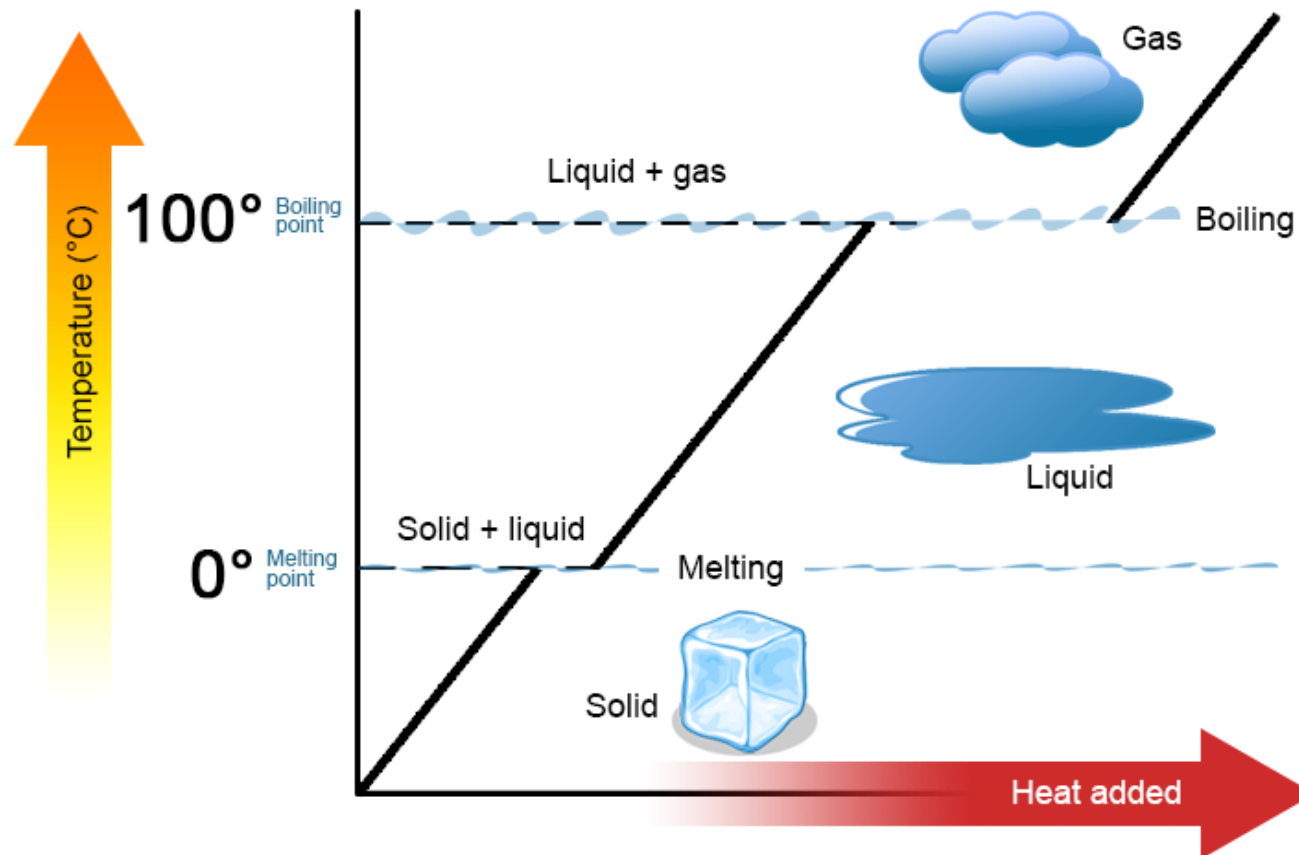
As such, ΔH_{sub} is not always tabulated because it can be simply calculated from ΔH_{fus} and ΔH_{vap}

Heating Curves

A plot of the temperature versus the amount of heat added is known as a heating curve. These are commonly used to visually show the relationship between phase changes and enthalpy for a given substance.

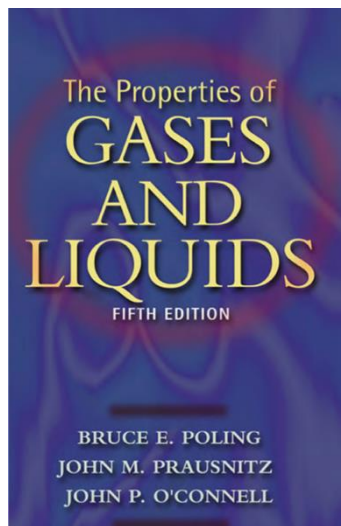


Heating curve for water. As heat is added to solid water,

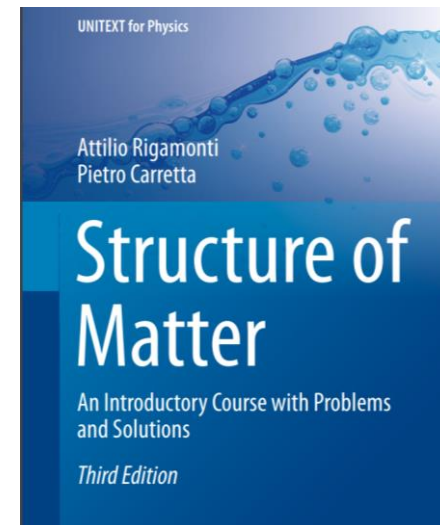


Review Questions

- What is the difference between melting and solidification?
- What is the difference between boiling and condensation?
- Describe the molecular changes when a solid becomes a liquid.
- Describe the molecular changes when a liquid becomes a gas.



Bruce E. Poling, John M. Prausnitz,
John P. O'Connell (2001) The
properties of gases and liquids
McGraw-Hill



Rigamonti, A., & Carretta, P.
(2015). Structure of Matter.
UNITEXT for Physics, Springer
International Publishing
Switzerland



Module 1. Molecular basis of life

Topic 2. Fundamentals of Molecular Biology

Lesson 1. Chemical bases of the cell



Contents

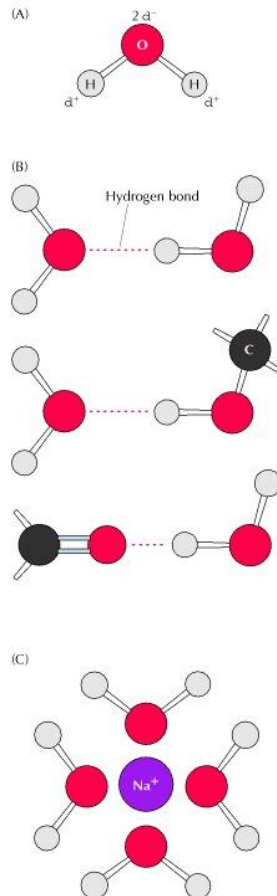
- Introduction
- Carbohydrates
- Lipids
- Nucleic Acids
- Proteins



Introduction

Cells are composed of water, inorganic ions, and carbon-containing (organic) molecules. Water is the most abundant molecule in cells, accounting for 70% or more of total cell mass. Consequently, the interactions between water and the other constituents of cells are of central importance in biological chemistry. The critical property of water in this respect is that it is a polar molecule, in which the hydrogen atoms have a slight positive charge and the oxygen has a slight negative charge

Characteristics of water



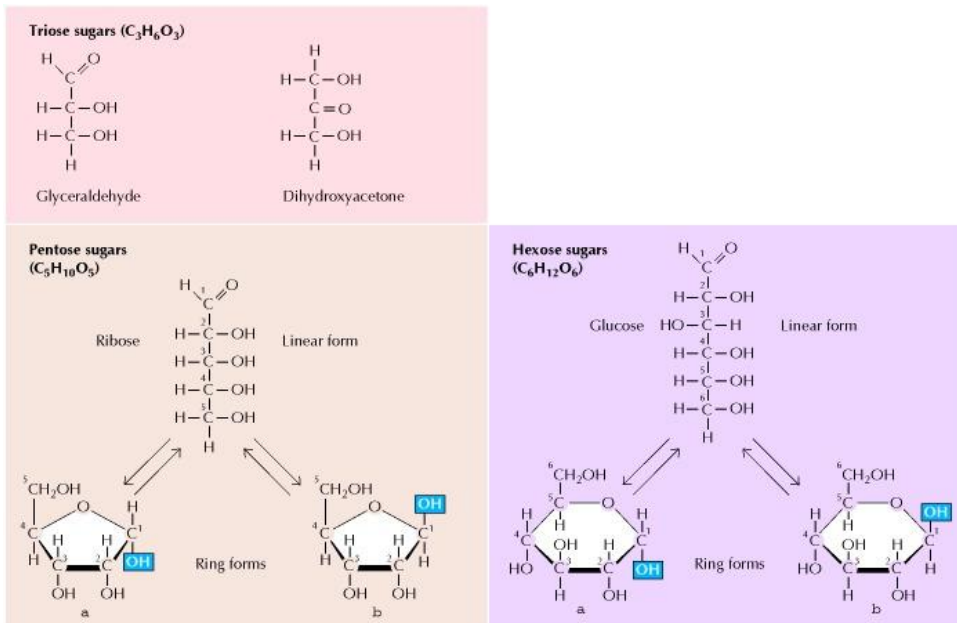
(A) Water is a polar molecule, with a slight negative charge (δ^-) on the oxygen atom and a slight positive charge (δ^+) on the hydrogen atoms. Because of this polarity, water molecules can form hydrogen bonds (dashed lines) either with each other or with other polar molecules (B), in addition to interacting with charged ions (C).

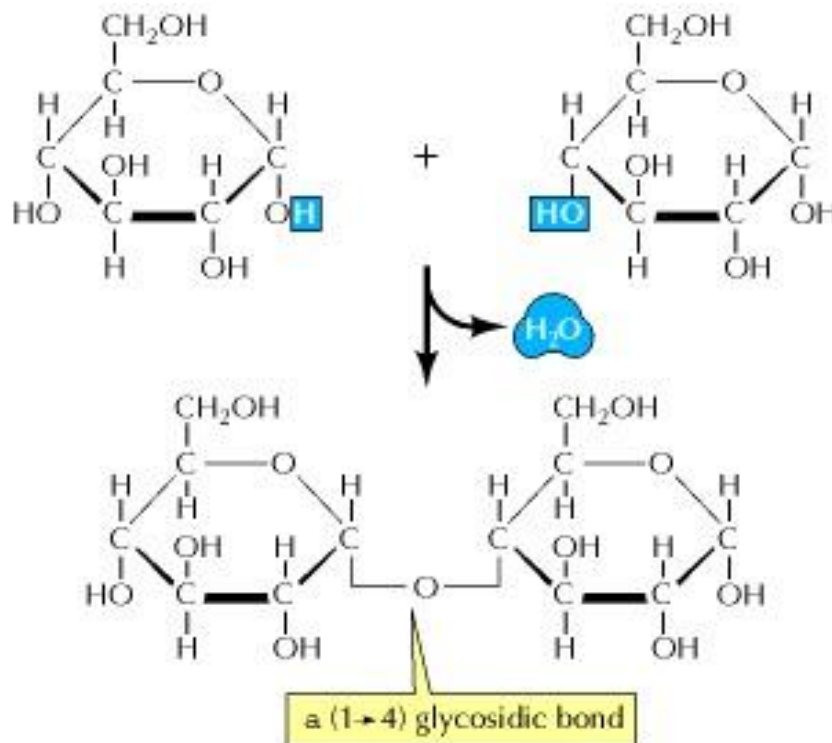
The inorganic ions of the cell, including sodium (Na⁺), potassium (K⁺), magnesium (Mg²⁺), calcium (Ca²⁺), phosphate (HPO₄²⁻), chloride (Cl⁻), and bicarbonate (HCO₃⁻), constitute 1% or less of the cell mass. These ions are involved in a number of aspects of cell metabolism, and thus play critical roles in cell function.

Carbohydrates

Structure of simple sugars

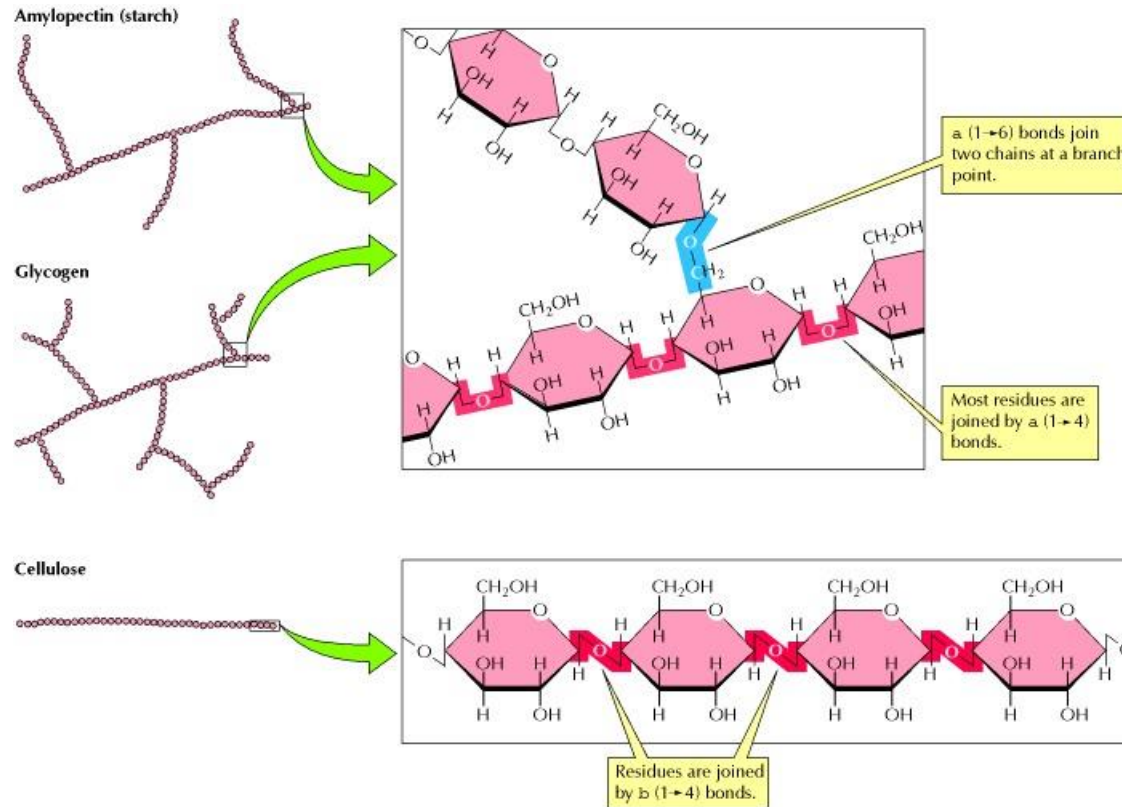
Representative sugars containing three, five, and six carbons (triose, pentose, and hexose sugars, respectively) are illustrated. Sugars with five or more carbons can cyclize to form rings, which exist in two alternative forms (α and β) depending on the configuration of carbon 1.





Formation of a glycosidic bond

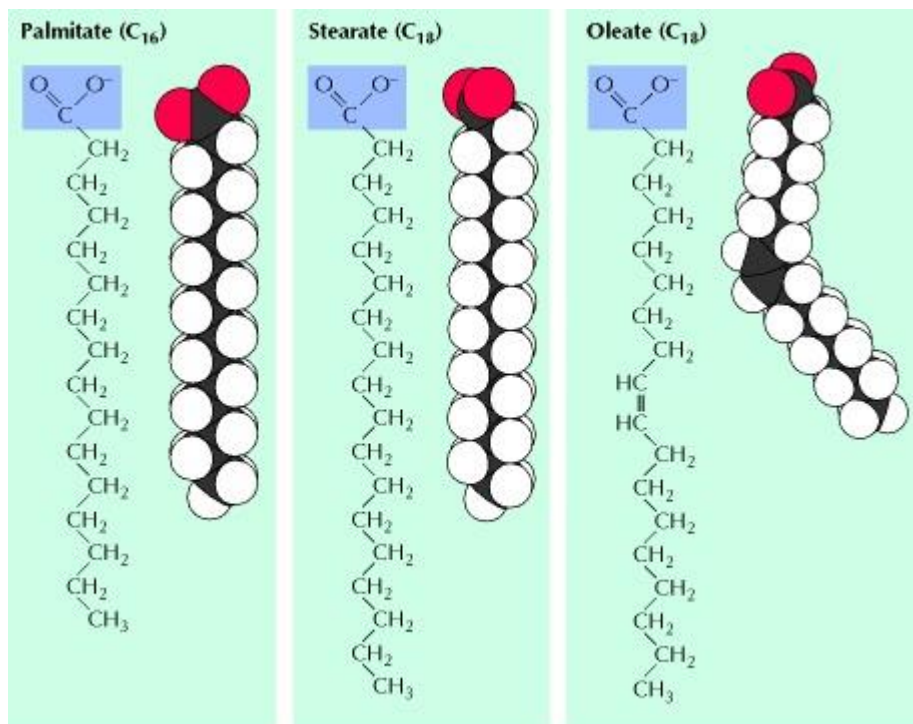
Two simple sugars are joined by a dehydration reaction (a reaction in which water is removed). In the example shown, two glucose molecules in the α configuration are joined by a bond between carbons 1 and 4, which is therefore called an α (1 \rightarrow 4) glycosidic bond.



Structure of polysaccharides

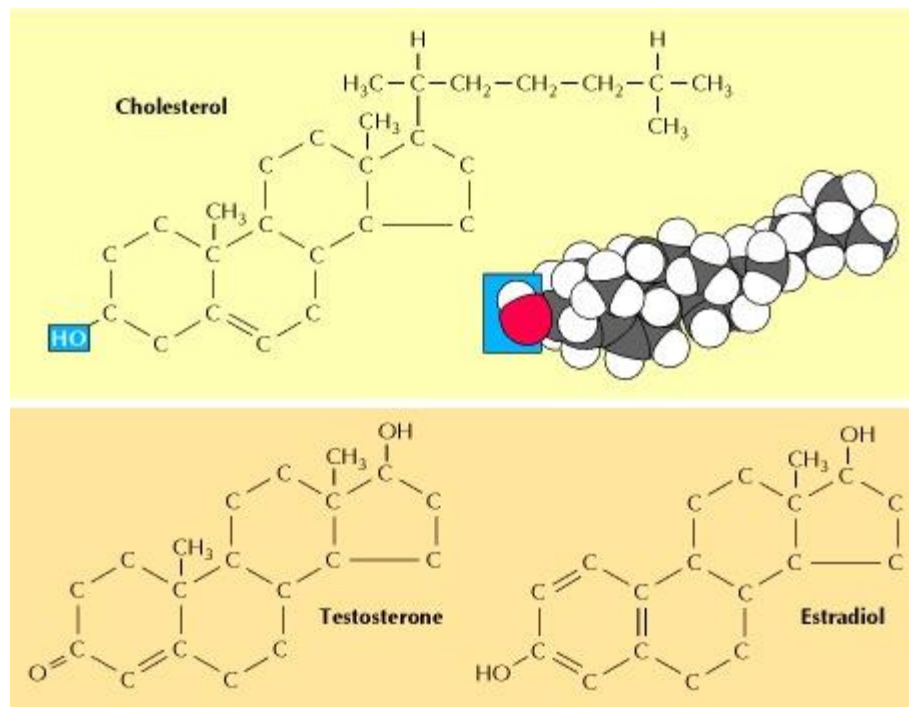
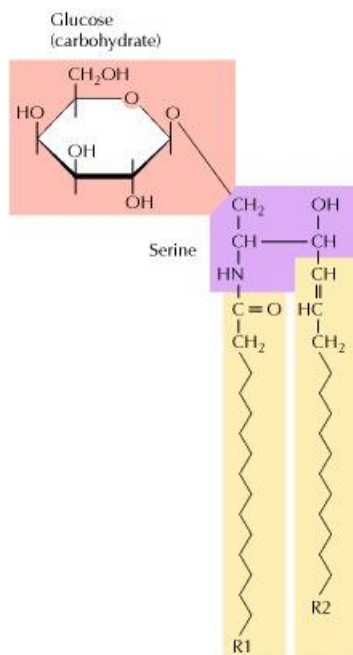
Polysaccharides are macromolecules consisting of hundreds or thousands of simple sugars. Glycogen, starch, and cellulose are all composed entirely of glucose residues, which are joined by α (1→4) glycosidic bonds in glycogen and starch but by β (1→4) bonds in cellulose. Glycogen and one form of starch (amylopectin) also contain occasional α (1→6) bonds, which serve as branch points by joining two separate α (1→4) chains.

Lipids



Structure of fatty acids

Fatty acids consist of long hydrocarbon chains terminating in a carboxyl group (COO-). Palmitate and stearate are saturated fatty acids consisting of 16 and 18 carbons, respectively. Oleate is an unsaturated 18-carbon fatty acid containing a double bond between carbons 9 and 10. Note that the double bond introduces a kink in the hydrocarbon chain.



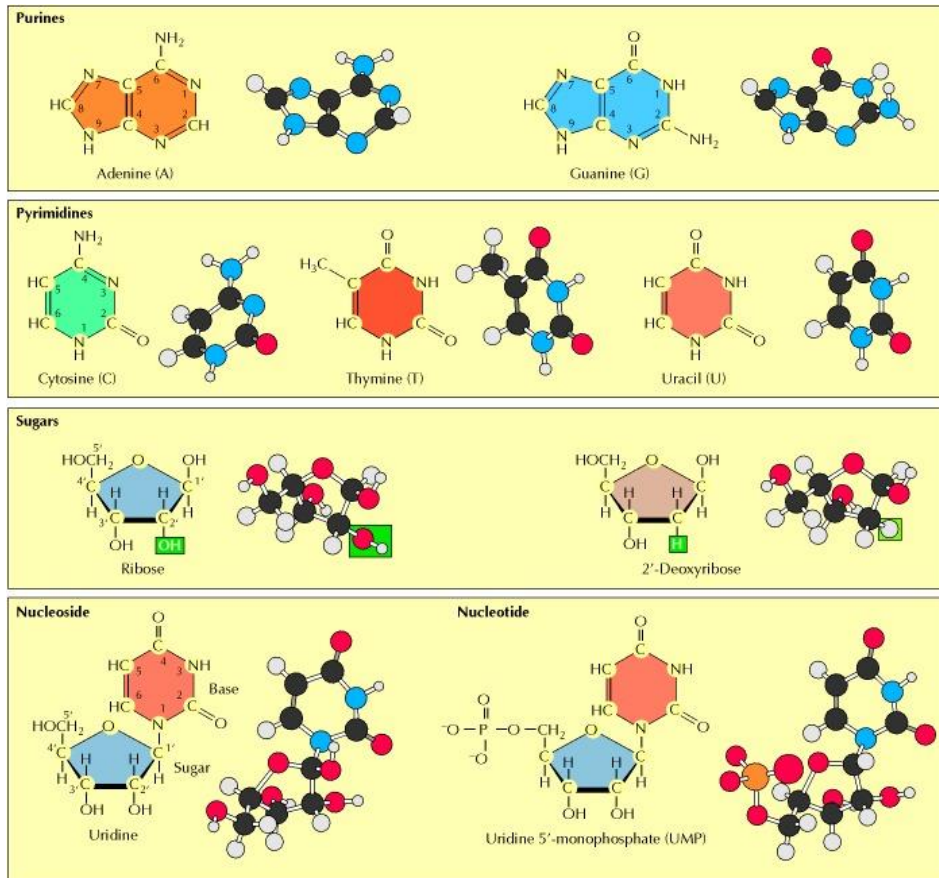
Structure of glycolipids

Two hydrocarbon chains are joined to a polar head group formed from serine and containing carbohydrates (e.g., glucose).

Cholesterol and steroid hormones

Cholesterol, an important component of cell membranes, is an amphipathic molecule because of its polar hydroxyl group. Cholesterol is also a precursor to the steroid hormones, such as testosterone and estradiol (a form of estrogen). The hydrogen atoms bonded to the ring carbons are not shown in this figure.

Nucleic Acids

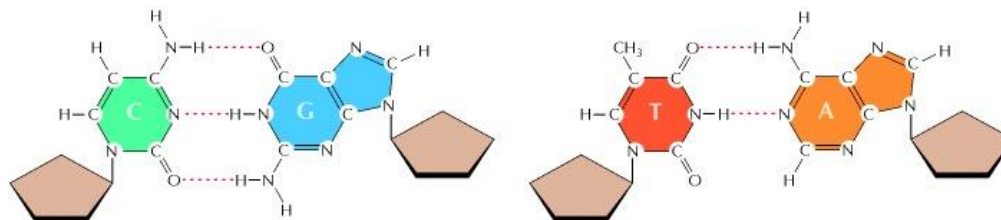
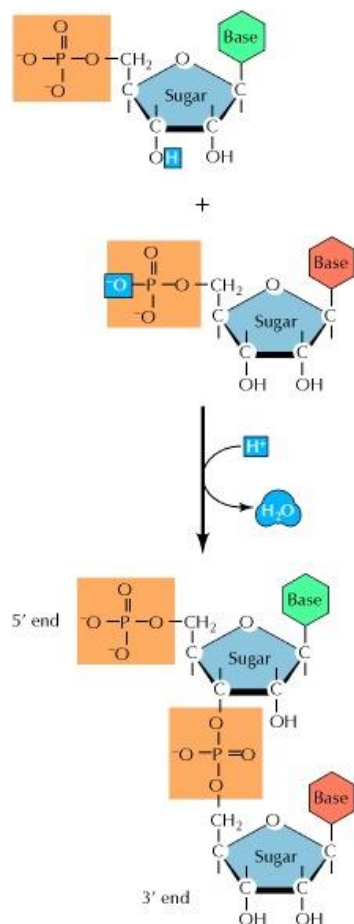


Components of nucleic acids

Nucleic acids contain purine and pyrimidine bases linked to phosphorylated sugars. A nucleic acid base linked to a sugar alone is a nucleoside. Nucleotides additionally contain one or more phosphate groups.

Polymerization of nucleotides

A phosphodiester bond is formed between the 3' hydroxyl group of one nucleotide and the 5' phosphate group of another. A polynucleotide chain has a sense of direction, one end terminating in a 5' phosphate group (the 5' end) and the other in a 3' hydroxyl group (the 3' end).



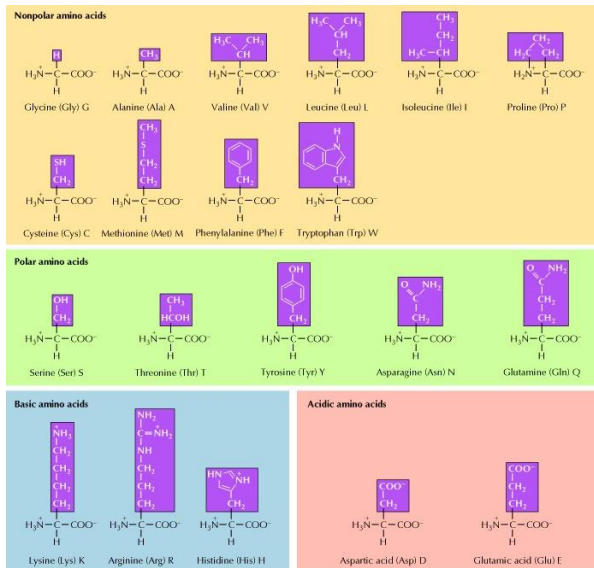
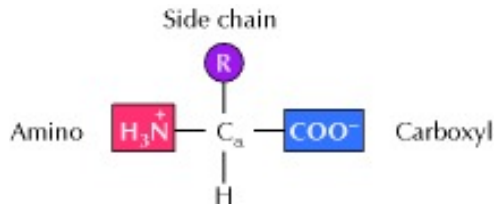
Complementary pairing between nucleic acid bases

The formation of hydrogen bonds between bases on opposite strands of DNA leads to the specific pairing of guanine (G) with cytosine (C) and adenine (A) with thymine (T).

Proteins

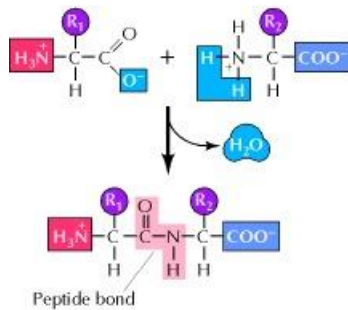
Structure of amino acids

Each amino acid consists of a central carbon atom (the α carbon) bonded to a hydrogen atom, a carboxyl group, an amino group, and a specific side chain (designated R). At physiological pH, both the carboxyl and amino groups are ionized, as shown.



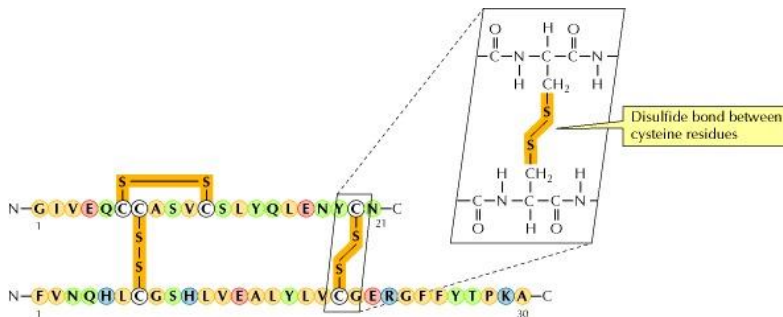
The amino acids

The three-letter and one-letter abbreviations for each amino acid are indicated. The amino acids are grouped into four categories according to the properties of their side chains: nonpolar, polar, basic, and acidic.



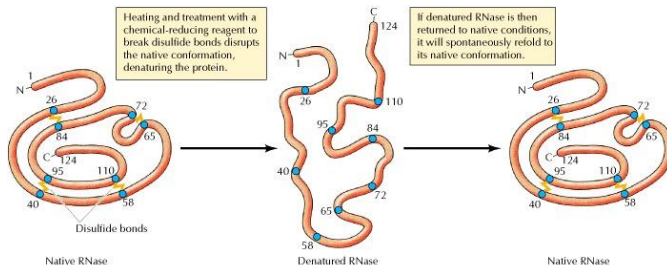
Formation of a peptide bond

The carboxyl group of one amino acid is linked to the amino group of a second.

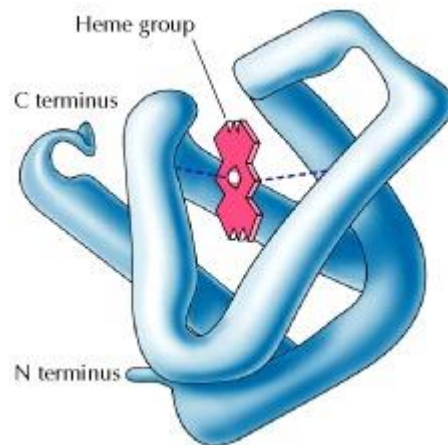


Amino acid sequence of insulin

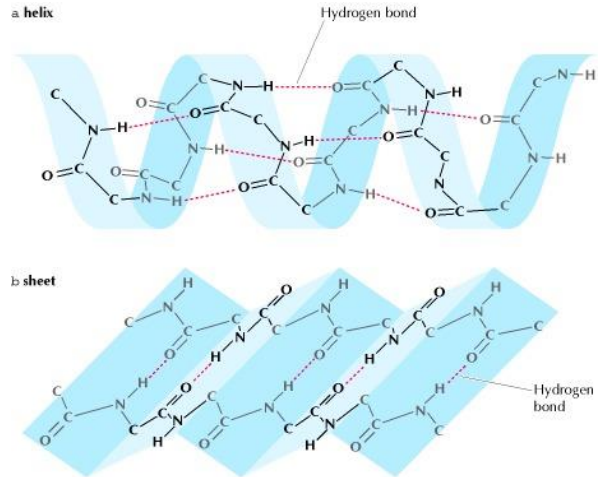
Insulin consists of two polypeptide chains, one of 21 and the other of 30 amino acids (indicated here by their one-letter codes). The side chains of three pairs of cysteine residues are joined by disulfide bonds, two of which connect the polypeptide chains.



Protein denaturation and refolding
Ribonuclease (RNase) is a protein of 124 amino acids (indicated by numbers). The protein is normally folded into its native conformation, which contains four disulfide bonds (indicated as paired circles representing the cysteine residues).



Three-dimensional structure of myoglobin
Myoglobin is a protein of 153 amino acids that is involved in oxygen transport. The polypeptide chain is folded around a heme group that serves as the oxygen-binding site.

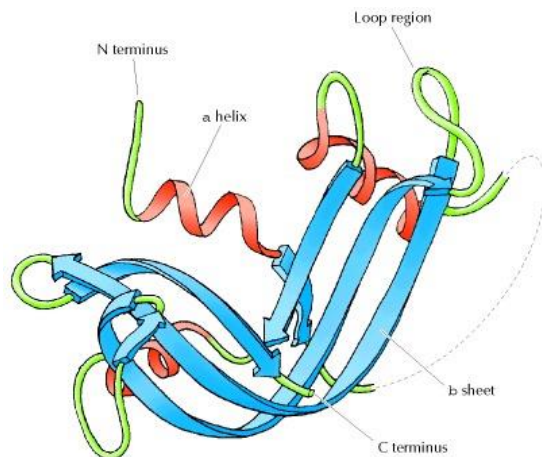


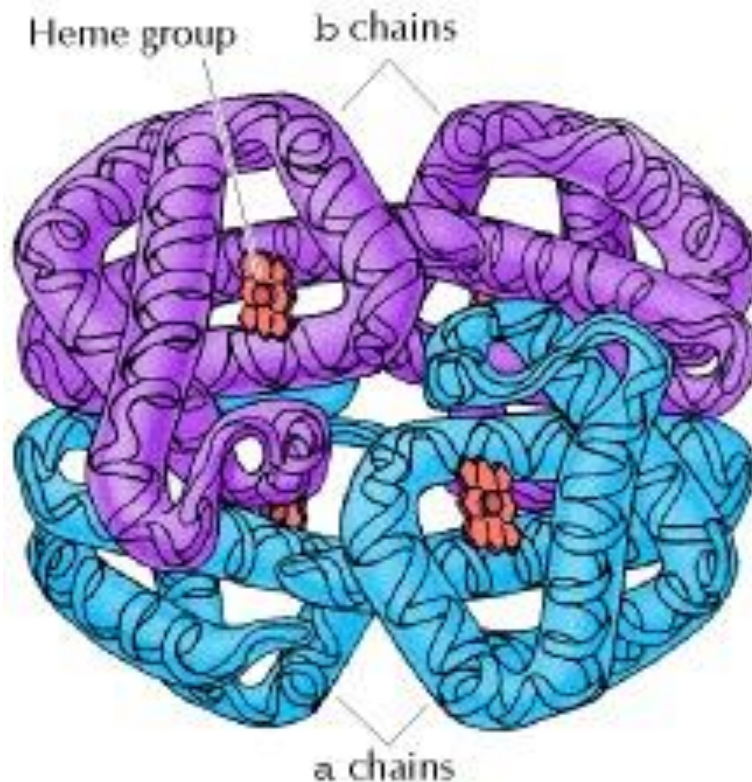
Secondary structure of proteins

The most common types of secondary structure are the α helix and the β sheet. In an α helix, hydrogen bonds form between CO and NH groups of peptide bonds separated by four amino acid residues. In a β sheet, hydrogen bonds connect two parts of a polypeptide chain lying side by side. The amino acid side chains are not shown.

Tertiary structure of ribonuclease

Regions of α -helix and β -sheet secondary structures, connected by loop regions, are folded into the native conformation of the protein. In this schematic representation of the polypeptide chain as a ribbon model, α helices are represented as spirals and β sheets as wide arrows.





Quaternary structure of hemoglobin
Hemoglobin is composed of four polypeptide chains, each of which is bound to a heme group. The two α chains and the two β chains are identical.



Review Questions

- What are the basic molecular of life?
- What is their structure?
- What are their functions?

- Cooper GM. The Cell: A Molecular Approach. 2nd edition. Sunderland (MA): Sinauer Associates; 2000. The Molecular Composition of Cells. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK9879/>



Module 1. Molecular basis of life

Topic 2. Fundamentals of Molecular Biology

Lesson 2. The cell. Basic cell types. Cell
construction. Organelles of eukaryotic cells.
Cytoskeleton - components and structural
functions



Contents

- Introduction
- The Cell Theory
- Prokaryotic and eukaryotic cell structures
- Cytoplasmic Membrane
- Organelles



Introduction

Cells are the basic units of all life. They make up every organ of every animal, plant, fungus, and bacteria. Cells in a body are like the building blocks of a house. They also have a specific basic structure that is shared by most cells. Cells usually consist of:

The cell membrane - this is a lipid bilayer that marks the limits of the cell. Within it, we can find the other two basic components of the cell: the DNA and the cytoplasm. All cells have a cell or plasma membrane.

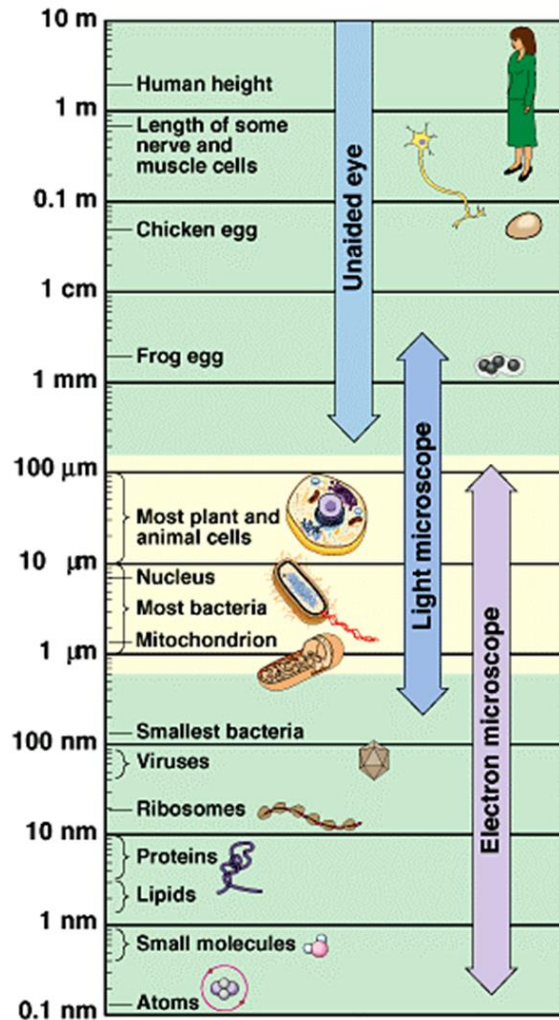
DNA - the DNA contains the instructions so that the cell can function. The genetic material can be protected within the nucleus (eukaryotic cells) or floating in the cytoplasm (prokaryotic cells). Most cells have DNA, but red blood cells, for example, don't.

Cytoplasm - the cytoplasm is the viscous substance within the plasma membrane in which the other components of a cell (the DNA/nucleus and other organelles) are floating.

The Cell Theory

Proposed by Matthais Schleiden and Theodor Schwann in 1839:

- All living things are made up of cells.
- Cells are the smallest working unit of all living things.
- All cells come from preexisting cells through cells division.



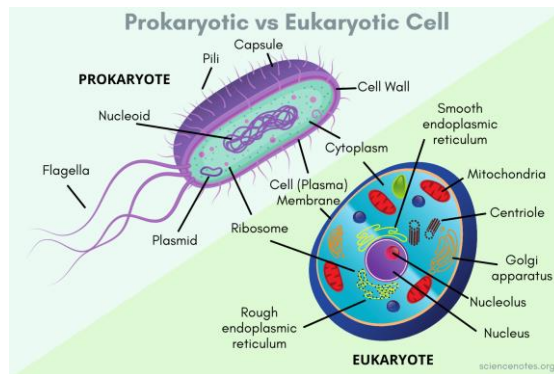
How We Study Cells?

- microscope
- cell fractionation

Most cells are between 1-100 µm in diameter which can be visualized by light microscope.

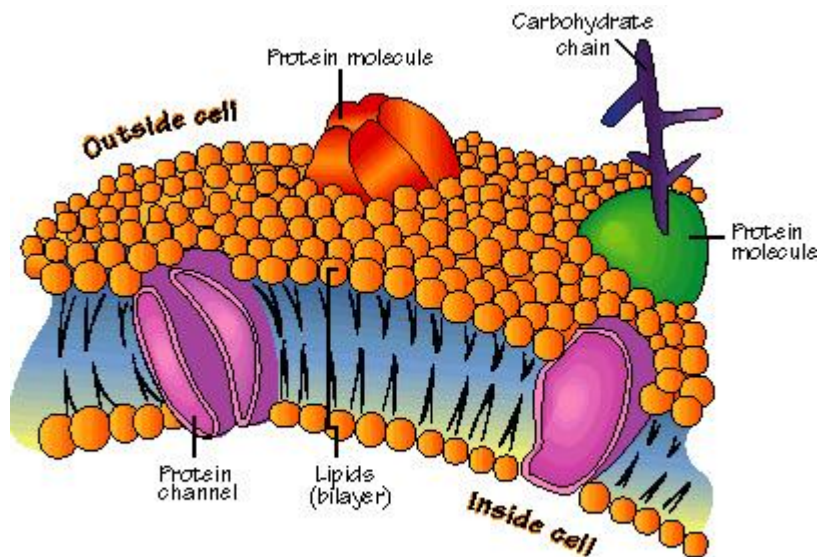
Prokaryotic and eukaryotic cell structures

The definition of prokaryote roughly translates from Greek as: 'without kernel' meaning 'without nucleus'. Hence, prokaryotes never have a nucleus. Prokaryotes are usually unicellular, which means that bacteria, for example, are only made up of one single cell. There are, however, exceptions to that rule where the organism is unicellular but has a nucleus, so it's a eukaryote. Yeast is one example.



On the other hand, eukaryote in Greek translates to “true nucleus”. This means that all eukaryotes have a nucleus. Except for yeast, eukaryotes are multicellular as they can be made up of millions of cells. Humans, for example, are eukaryotes, and so are plants and animals. In terms of cell structure, eukaryotes and prokaryotes share some traits but are different in others. The following table shows the similarities and differences while also giving us a general overview of the cell structures we will be discussing in this article.

Cytoplasmic Membrane



It is also called plasma membrane or cell membrane. The plasma membrane is a semi-permeable membrane that separates the inside of a cell from the outside.

In eukaryotic cells, the plasma membrane consists of proteins, carbohydrates and two layers of phospholipids (i.e. lipid with a phosphate group). These phospholipids are arranged as follows:

The polar, hydrophilic (water-loving) heads face the outside and inside of the cell. These heads interact with the aqueous environment outside and within a cell.

The non-polar, hydrophobic (water-repelling) tails are sandwiched between the heads and are protected from the aqueous environments.

Scientists Singer and Nicolson described the structure of the phospholipid bilayer as the 'Fluid Mosaic Model'. The reason is that the bilayer looks like a mosaic and has a semi-fluid nature that allows lateral movement of proteins within the bilayer.

Functions

- The plasma membrane is selectively permeable i.e. it allows only selected substances to pass through.
- It protects the cells from shock and injuries.
- The fluid nature of the membrane allows the interaction of molecules within the membrane. It is also important for secretion, cell growth, and division etc.
- It allows transport of molecules across the membrane. This transport can be of two types:
- Active transport – This transport occurs against the concentration gradient and therefore, requires energy. It also needs carrier proteins and is a highly selective process.
- Passive transport – This transport occurs along the concentration gradient and therefore, does not require energy. Thus, it does not need carrier proteins and is not selective.

Functions

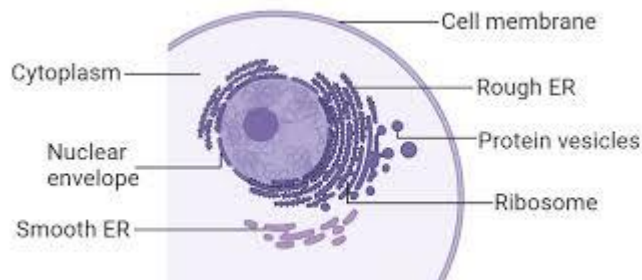
- The plasma membrane is selectively permeable i.e. it allows only selected substances to pass through.
- It protects the cells from shock and injuries.
- The fluid nature of the membrane allows the interaction of molecules within the membrane. It is also important for secretion, cell growth, and division etc.
- It allows transport of molecules across the membrane. This transport can be of two types:
- Active transport – This transport occurs against the concentration gradient and therefore, requires energy. It also needs carrier proteins and is a highly selective process.
- Passive transport – This transport occurs along the concentration gradient and therefore, does not require energy. Thus, it does not need carrier proteins and is not selective.

Organelles

Organelles are structures within a cell that are surrounded by a membrane and carry out different functions for the cell. For example, mitochondria are in charge of generating energy for the cell, while the Golgi apparatus is involved in sorting proteins, among other functions.

Endoplasmic reticulum (ER)

It is a network of small, tubular structures. It divides the space inside of Eukaryotic cells into two parts – luminal (inside ER) and extra-luminal (cytoplasm).

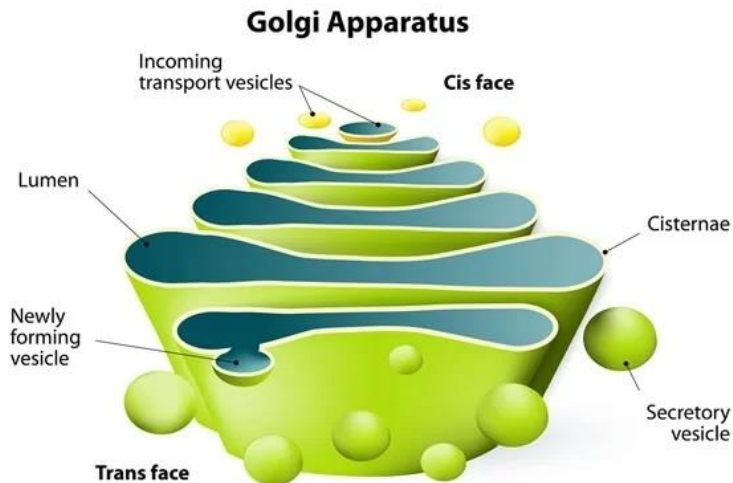


Functions

SER is involved in lipid synthesis and RER is involved in protein synthesis.

RER helps in folding proteins and transports it to the Golgi apparatus in vesicles.

Golgi Apparatus



It is named after the scientist who discovered it, Camillo Golgi. Golgi is made of many flat, disc-shaped structures called cisternae. It is present in all eukaryotic cells except human red blood cells and sieve cells of plants.

Structure: The cisternae are arranged in parallel and concentrically near the nucleus as follows:

- Cis face (forming face) – It faces the plasma membrane and receives secreted material in vesicles.
- Trans face (maturing face) – It faces the nucleus and releases the received material into the cell.

Functions

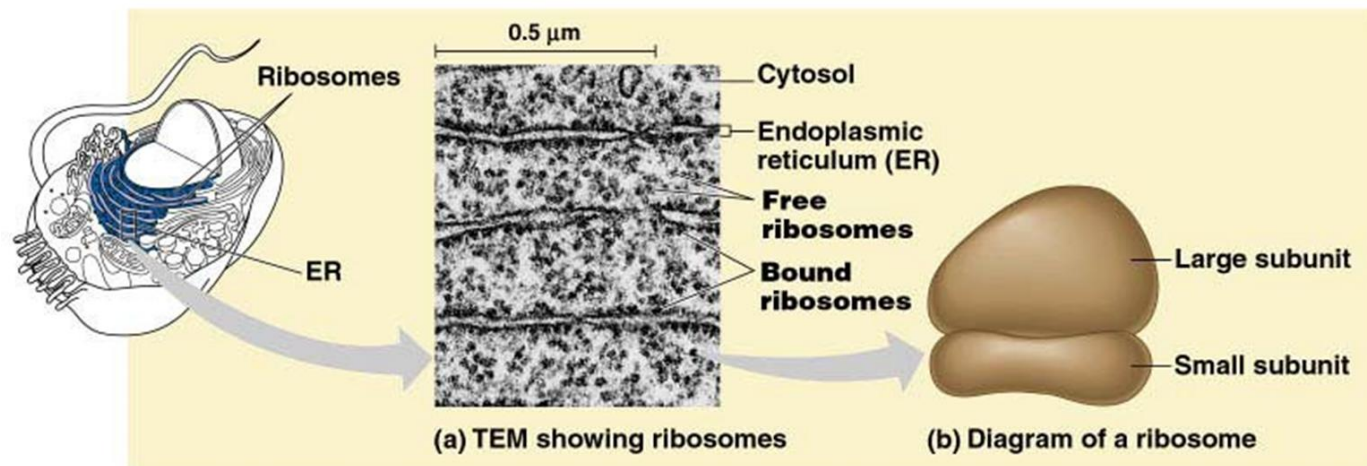
- An important site for packaging material within the cell.
- Proteins are modified in the Golgi.
- An important site for the formation of glycolipids (i.e. lipids with carbohydrate) and glycoproteins (i.e. proteins with carbohydrates).

Ribosomes

These structures are not bound by a membrane. Ribosomes are also called ‘Protein factories’ since they are the main site of protein synthesis.

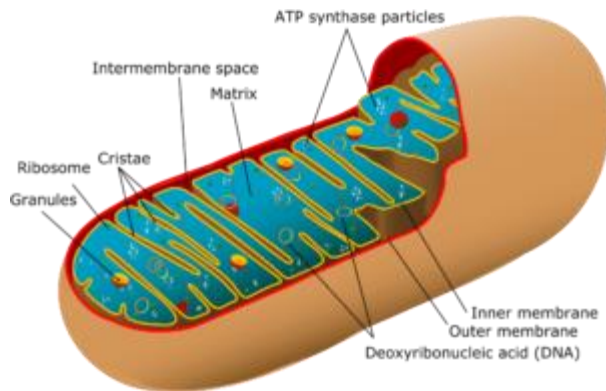
Structure: They are made of ribonucleic acids and proteins. Eukaryotic ribosomes are of the 80S type, with 60S (large subunit) and 40S (small subunit).

Functions: A Major site for synthesis of proteins and polypeptides (chain of amino acids).



Mitochondria

Description: They are membrane-bound organelles, also known as ‘powerhouses of the cell’.



Structure: It has two membranes – outer and inner. The outer membrane forms a continuous boundary around the mitochondria. The inner membrane is semi-permeable and divided into folds called ‘cristae’. The membranes divide the lumen of the mitochondria into an inner and outer compartment. The inner compartment is called matrix and outer compartment forms the intermembrane space.

Functions

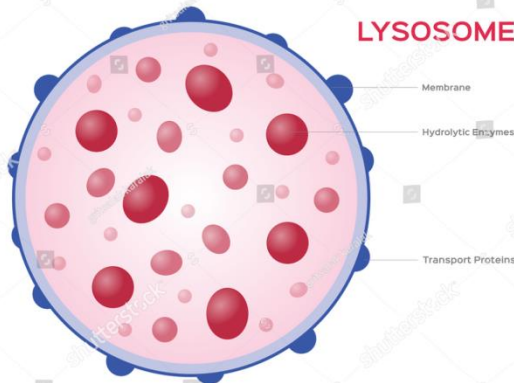
They produce energy (ATP) and therefore are called the ‘powerhouse of the cell’.

Helps in regulating cell metabolism.

Mitochondria possess their own DNA, RNA and components required for protein synthesis.

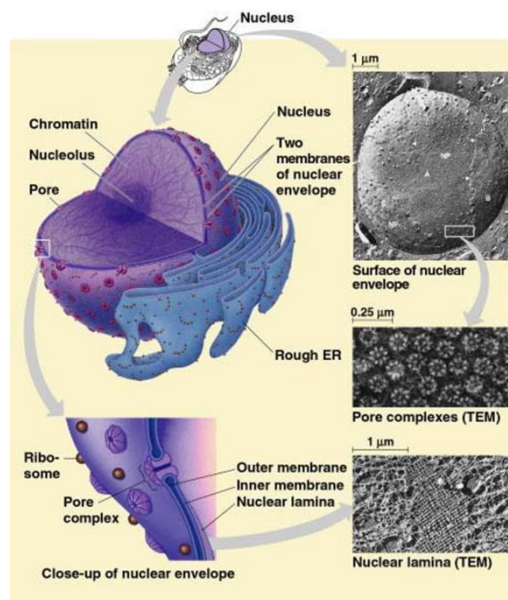
Lysosomes

They are membrane-bound vesicles formed in the Golgi apparatus. Lysosomes are also called 'suicidal bags' since they are rich in hydrolytic enzymes such as lipases, proteases, carbohydrates etc. These enzymes are optimally active at acidic pH (less than 7).



Function: The main function of lysosomes is to digest lipids, proteins, carbohydrates and nucleic acids.

Nucleus



Functions

It stores genetic information (in the form of DNA) necessary for development and reproduction.

It contains all information necessary for protein synthesis and cellular functions.

Nucleus is the main organelle of a cell. It is a double membrane structure with all the genetic information. Therefore, it is also called the 'brain' of a cell. The nucleus is found in all eukaryotic cells except human RBCs and sieve cells of plants.

A nucleus has the following parts:

Nuclear envelope – It is a double membrane structure that surrounds the nucleus. The outer membrane is continuous with the endoplasmic reticulum. The inner membrane has small pores called 'nuclear pores'.

Nucleoplasm – It is the fluid material in the nucleus that contains the nucleolus and chromatin.

Nucleolus – Nucleoli are not membrane-bound and are active sites for ribosomal RNA synthesis.

Chromatin – It consists of DNA and proteins called 'histones'. The DNA is organised into chromosomes. Chromosomes have certain constriction sites called 'centromeres'. Based on the position of the centromere, they can be divided as follows:

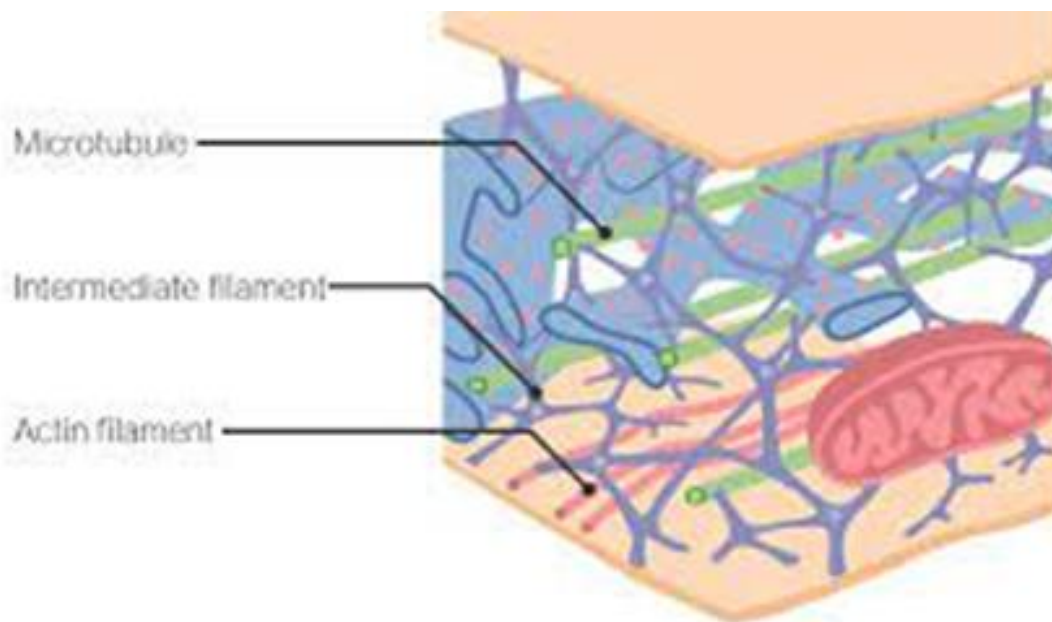
Metacentric – With centromere in the centre and having equal chromosome arms.

Sub-metacentric – Centromere is slightly off-centre creating one short and one long arm.

Acrocentric – Centromere is extremely off-centre with one very long and one very short chromosome arm.

Telocentric – Centromere is placed at one end of the chromosome. Humans do not possess telocentric chromosomes.

Cytoskeleton



Cytoskeleton

- a network of fibers extending throughout the cytoplasm

- function:

- provide mechanical strength to the cell establish cell shape

- locomotion (several types of cell motility) intracellular transport of organelles

- 3 main types of fiber:

1. microtubules: determine the positions of membrane-enclosed organelles and intracellular transport

2. microfilament: determine the shape of the cell and necessary for the whole cell locomotion

3. intermediate filament: provide mechanical strength and resistance to shear stress

Review Questions

- What is the difference between prokaryotic and eukaryotic cells?
- Describe the organelles and what are their functions?



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.

Module 1. Molecular basis of life

Topic 2. Fundamentals of Molecular Biology

Lesson 3. Lipids. Construction and functions.
Biomembranes and cell architecture. Lipid structure and structural organization of biomembranes. Protein components and main functions in biomembranes. Transport of ions and small molecules across cell membranes

Contents

- Introduction
- Lipids
- Construction and functions. Biomembranes and cell architecture.
- Protein components and main functions in biomembranes.
- Transport of ions and small molecules across cell membranes



Introduction

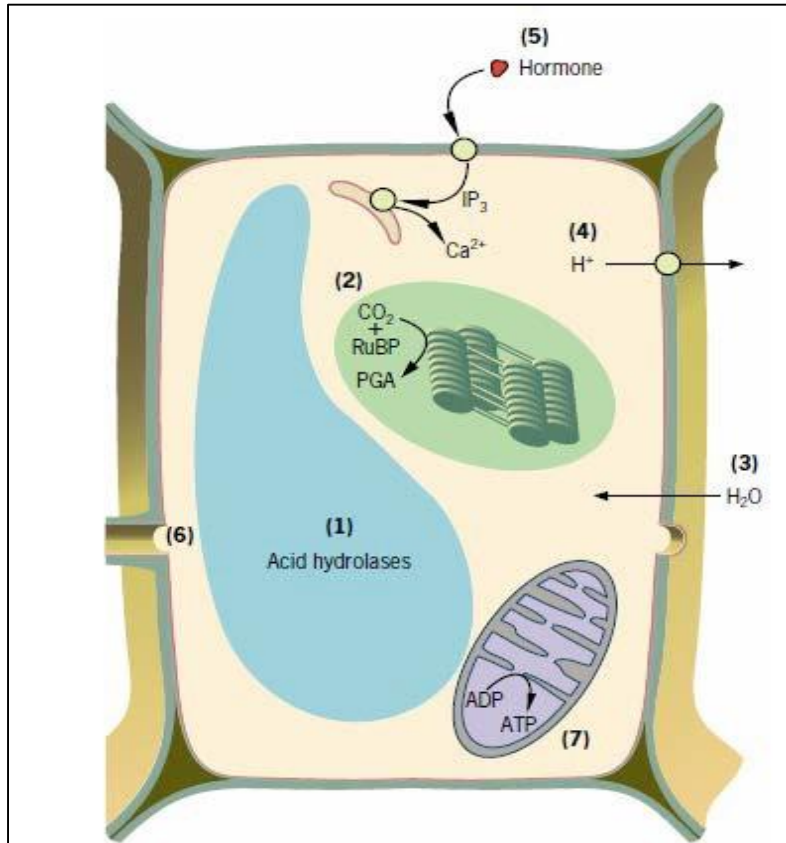
Cells are the basic units of all life. They make up every organ of every animal, plant, fungus, and bacteria. Cells in a body are like the building blocks of a house. They also have a specific basic structure that is shared by most cells. Cells usually consist of:

The cell membrane - this is a lipid bilayer that marks the limits of the cell. Within it, we can find the other two basic components of the cell: the DNA and the cytoplasm. All cells have a cell or plasma membrane.

DNA - the DNA contains the instructions so that the cell can function. The genetic material can be protected within the nucleus (eukaryotic cells) or floating in the cytoplasm (prokaryotic cells). Most cells have DNA, but red blood cells, for example, don't.

Cytoplasm - the cytoplasm is the viscous substance within the plasma membrane in which the other components of a cell (the DNA/nucleus and other organelles) are floating.

✓ All cells are surrounded by membranes



➤ Selective barrier

➤ But also important for:

1. Compartmentalization
2. Biochemical activities
3. Transport of dissolved substances
4. Transport of ions
5. Signal transduction
6. Cell-cell interaction
7. Energy conversion

➤ Dynamic structures:

1. Constant movements
2. Continuous building and degradation of their components

❖ All biological membranes have common basic structure:

✓ very thin layer of lipid and protein molecules connected with non-covalent interactions

✓ dynamic and fluid structures

✓ biochemical composition: **lipids, proteins, sugars**

- membranes with **similar functions** (e.g. from the same organelles) are **similar** in different cells

- membranes with different functions (e.g. different organelles) are very different within the same cell

✓ Lipids

- double bilayer (thickness 5-10 nm)

- basic fluid structure

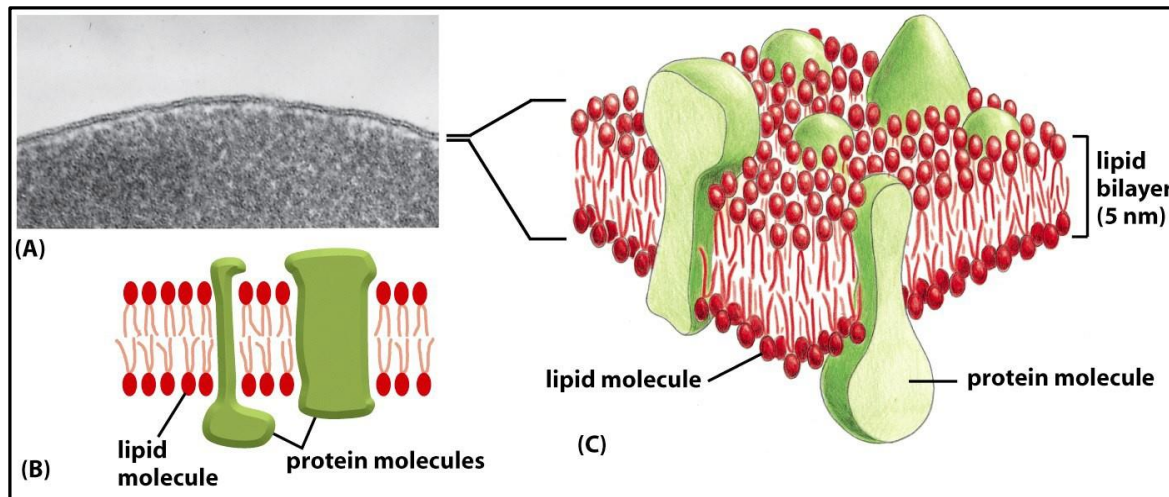
✓ Proteins

- involved in membrane functions
- transport, catalyses, structure, receptors

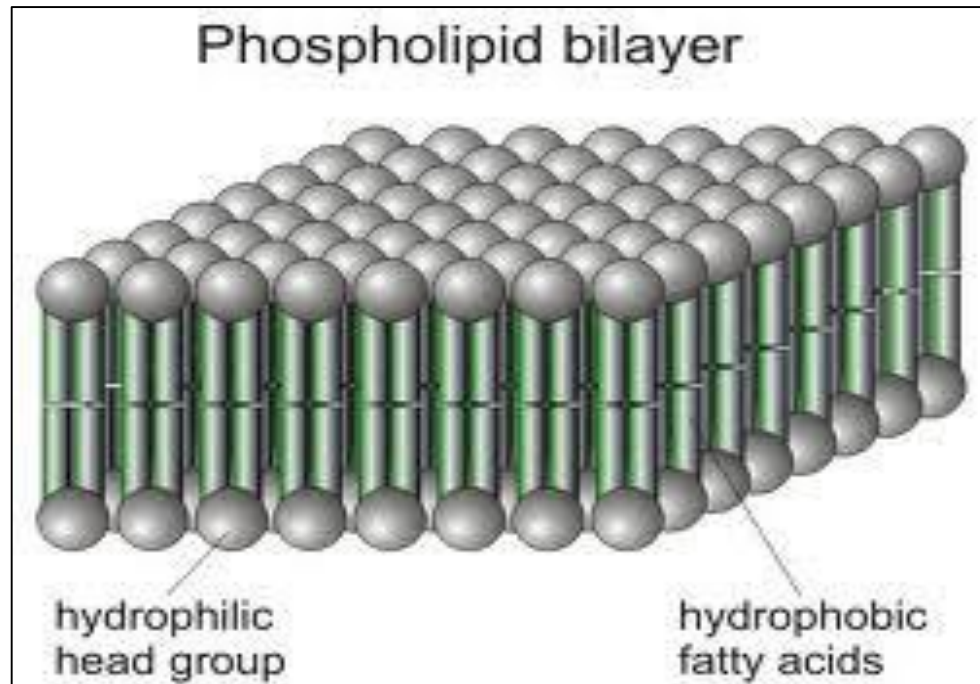
A – EM-photo erythrocyte membrane

B – 2D membrane

C – 3D membrane



Membrane lipids



- ✓ Amphipathic molecules
- ✓ Spontaneous formation of bilayer in aqueous solution

✓ Phospholipids

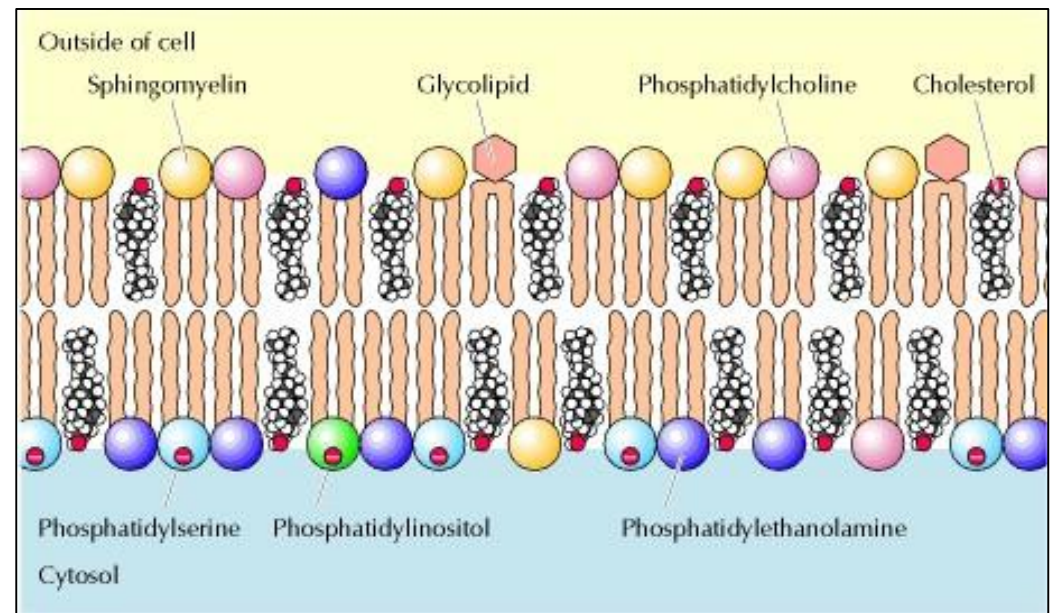
- phosphatidylcholine
- sphingomyelin
- phosphatidylethanolamine
- phosphatidylserine
- phosphatidylinositol

outer (extracellular) leaflet

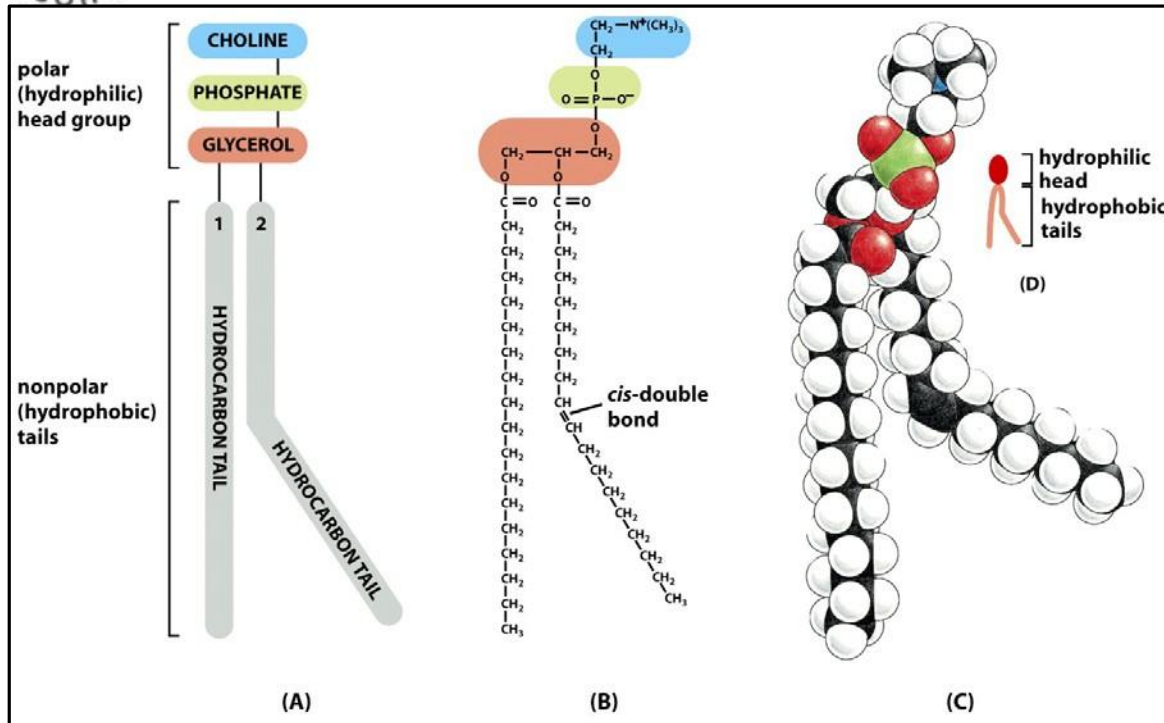
inner (cytoplasmic) leaflet

✓ Cholesterol

✓ Glycolipids

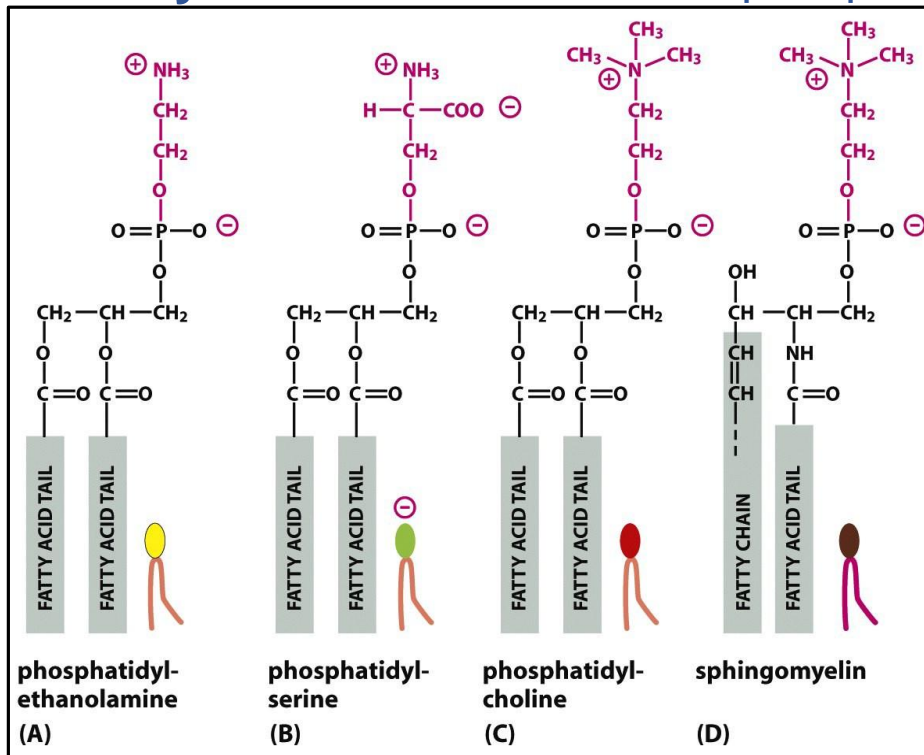


■ Phosphatidylcholin



- ✓ polar head + two hydrophobic carbohydrate chains
- ✓ tails – fatty acids (14 – 24 C atoms)
 - 1st tail – no double bonds (**saturated**)
 - 2nd tail – 1 or more *cis*-double bonds (**unsaturated**)
- ✓ differences in length and saturations → membrane fluidity

- ✓ **Bacteria** – mostly one phospholipid type; no cholesterol
- ✓ **Eucaryota** – mixture of different phospholipid types + cholesterol + glycolipids



derived from glycerol
(phosphoglycerides)

derived from
sphingosine
(sphingolipid)

outer leaflet:

- phosphatidylcholine
- sphingomyelin

inner leaflet:

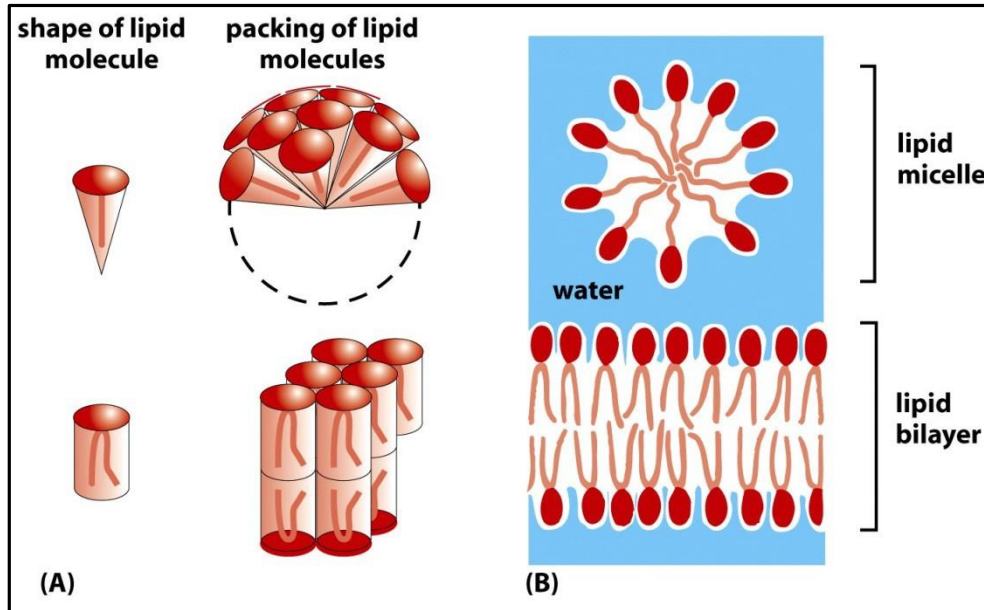
- phosphatidylethanolamine
- phosphatidylserine*
- phosphatidylinositol
- * translocation to the outer leaflet
- apoptosis

ERASMUS+

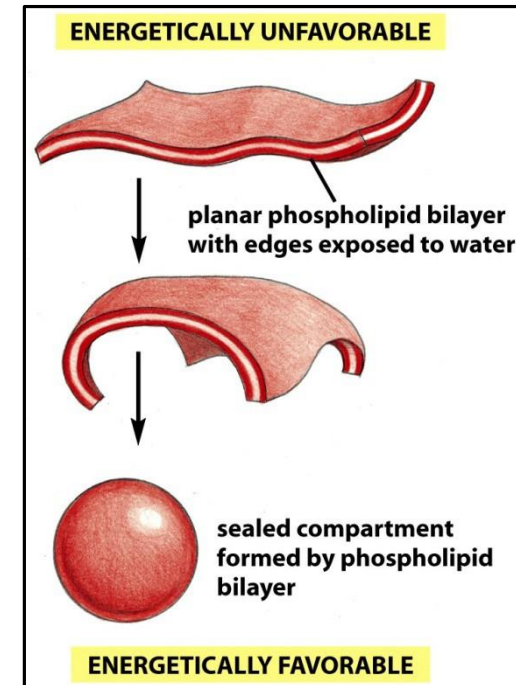
Key Action KA2 - Cooperation for innovation and the exchange of good practices

Partnerships for Digital Education Readiness

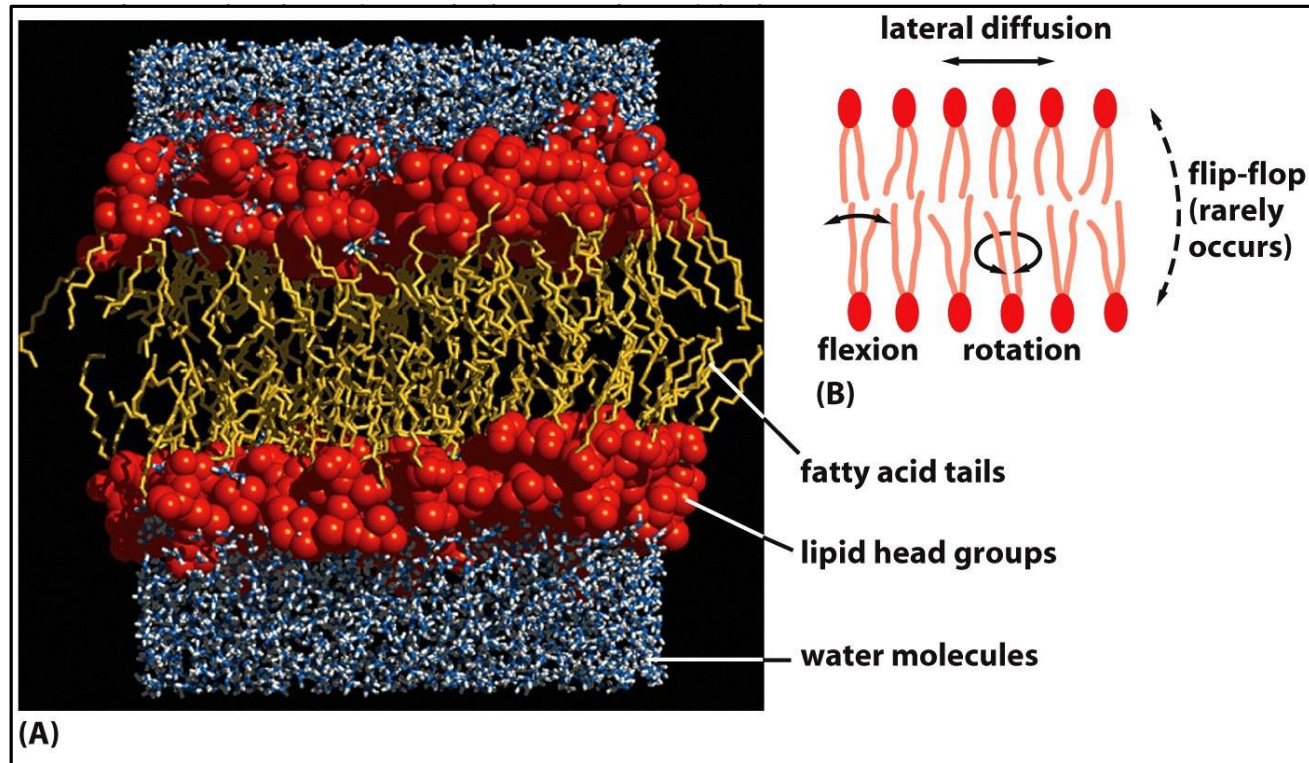
Spontaneous formation of lipid bilayer



- one tail → **micelle**
- two tails → **bilayer**
- ✓ **energetically most favored distribution**



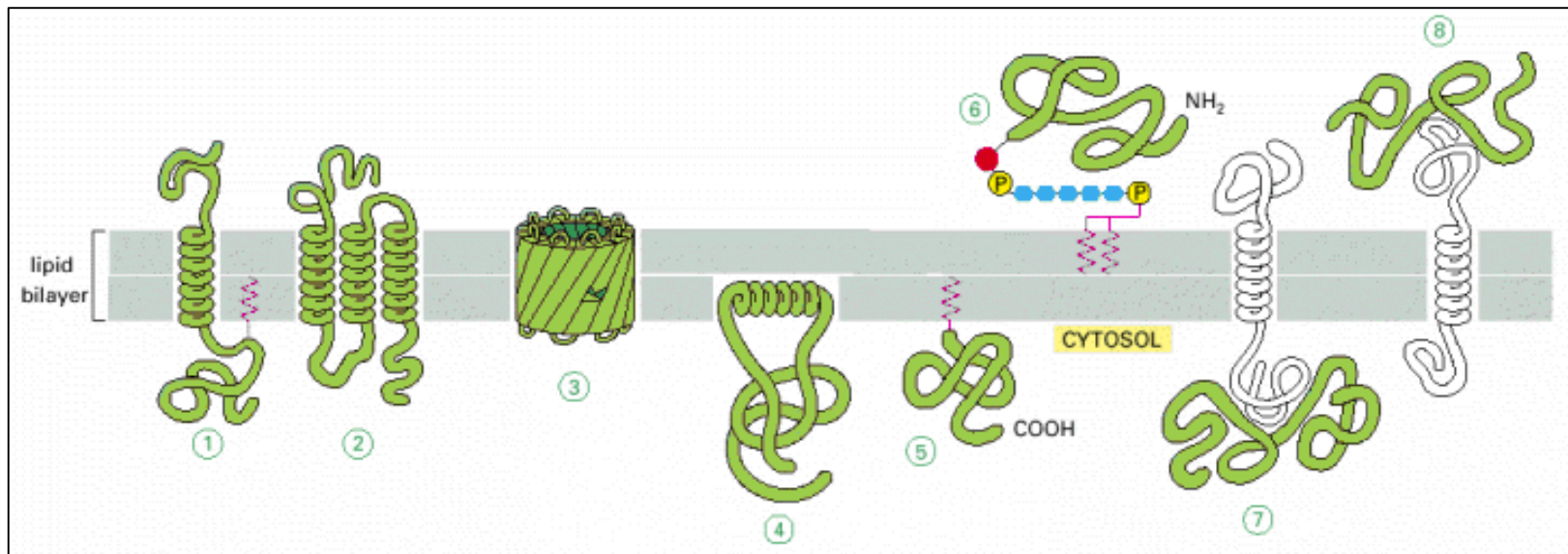
Spontaneous closure of
lipid bilayer



- ✓ **Flip-flop** – rare (< 1 per month)
- ✓ **Lateral diffusion** – frequent ($\sim 10^7$ per sec)
- ✓ **Rotation**
- ✓ **Flexion**

Membrane proteins

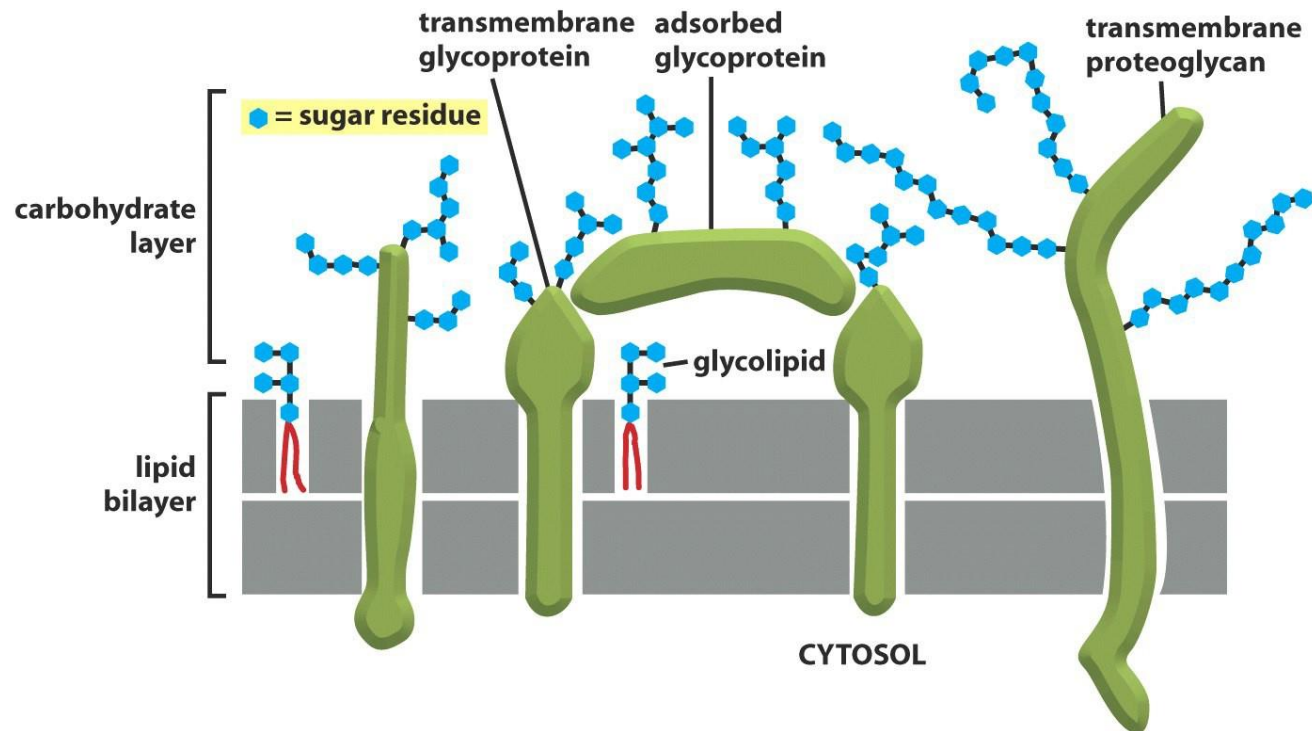
- Membrane proteins can be associated with the lipid bilayer in various ways



- ✓ **1, 2, 3** – transmembrane proteins (amphipathic)
- ✓ **4, 5, 6** – anchored proteins (exposed at only one side)
- ✓ **7, 8** – periphery proteins (noncovalent interactions with other proteins)

Membrane carbohydrates

- ✓ Bind to proteins or lipids
- ✓ Only in outer leaflet



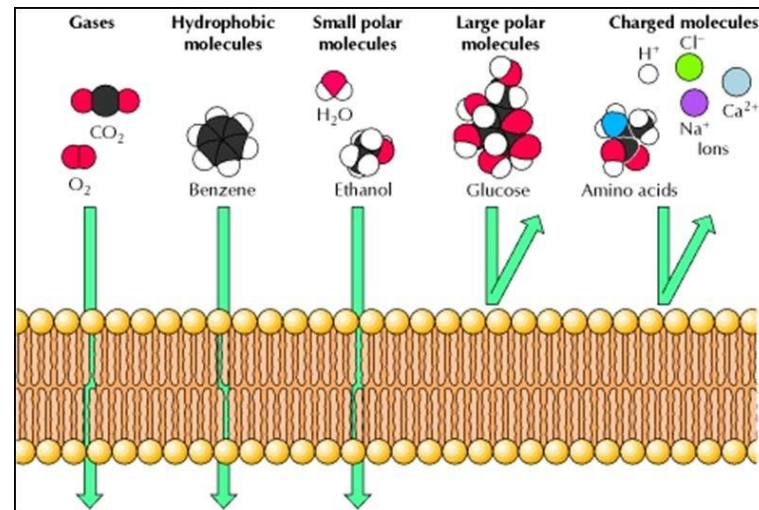
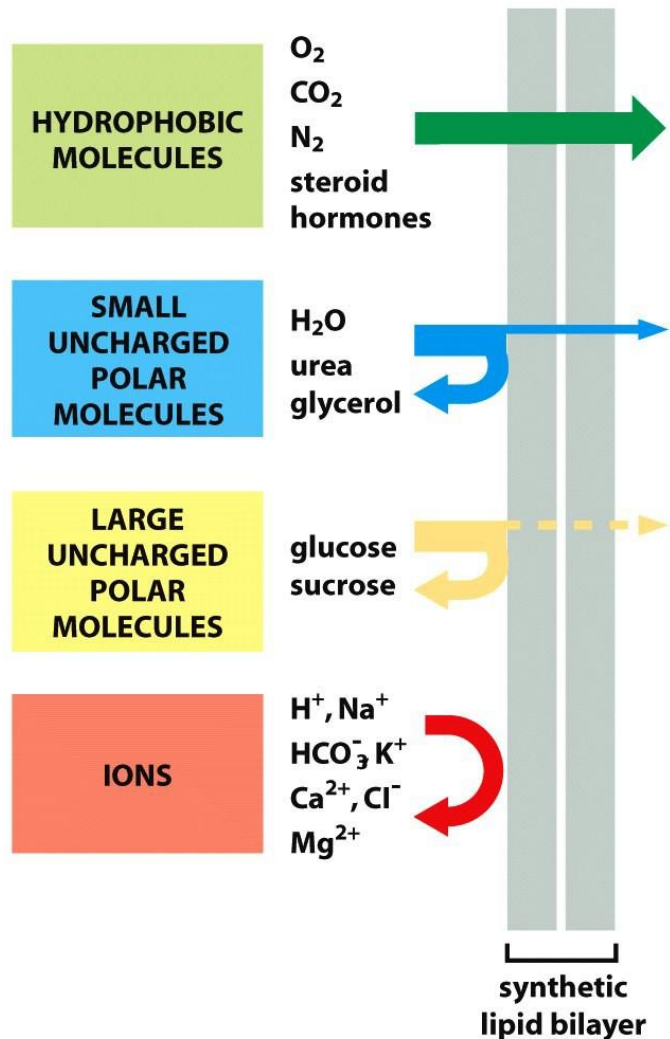
The relative permeability of a synthetic lipid bilayer to different classes of molecules

✓ small uncharged molecules can diffuse freely through a phospholipid bilayer

✓ bilayer is impermeable to:

- larger polar molecules
- (such as glucose and amino acids)
- ions

✓ the smaller the molecule and, more importantly, the less strongly it associates with water, the more rapidly the molecule diffuses across the bilayer

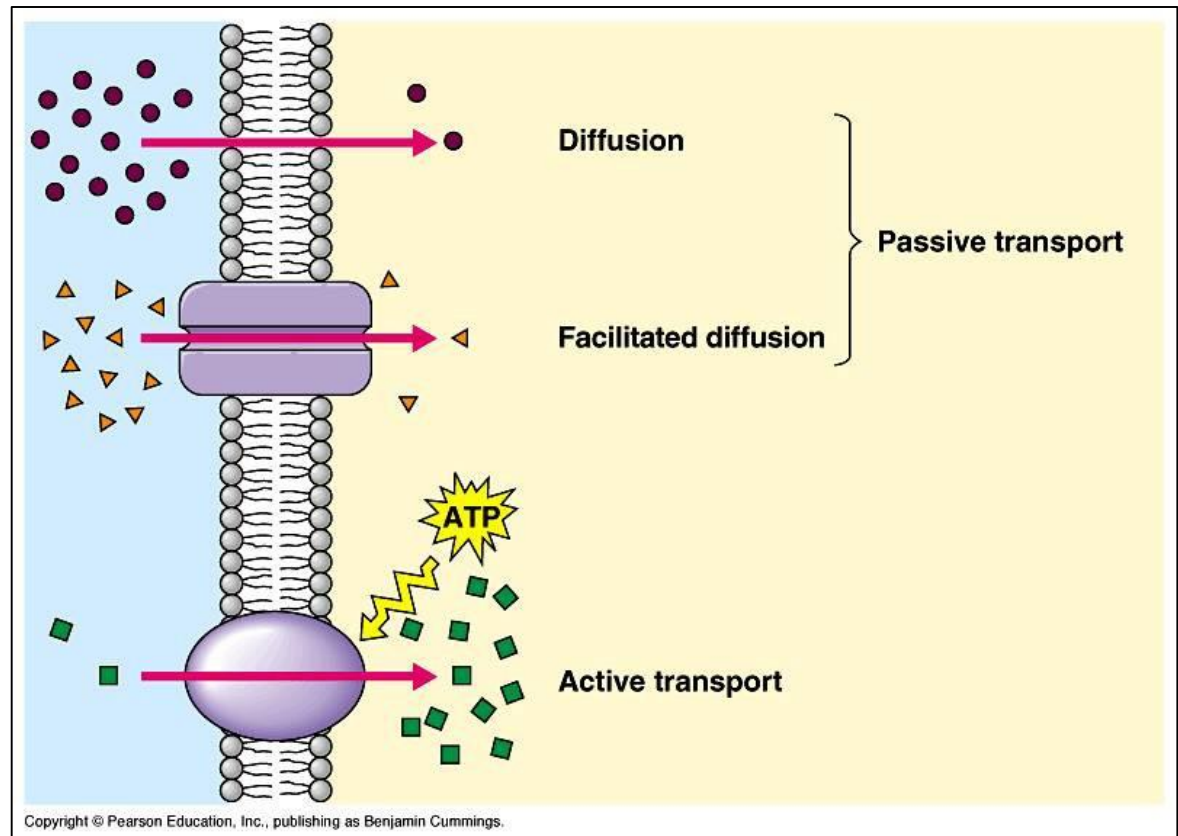


ERASMUS+

Three ways in which molecules can cross the membrane

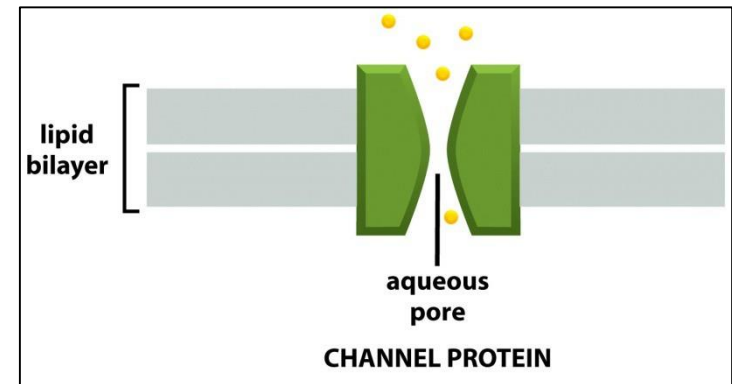
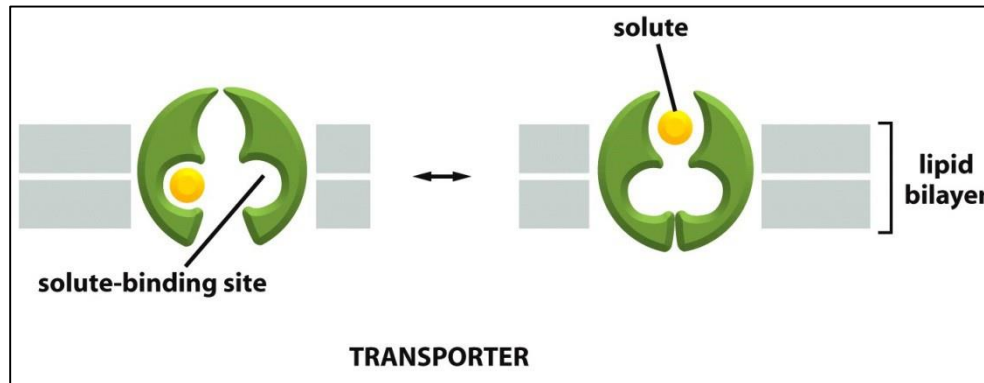
- Passive diffusion
- Facilitated diffusion
- Active transport

Passive transport - **down the concentration gradient!**



Two main classes of membrane transport proteins

- ✓ **Carriers** → bind the specific solute to be transported and undergo a series of conformational changes to transfer the bound solute across the membrane
- ✓ **Channels** → interact with the solute to be transported much more weakly

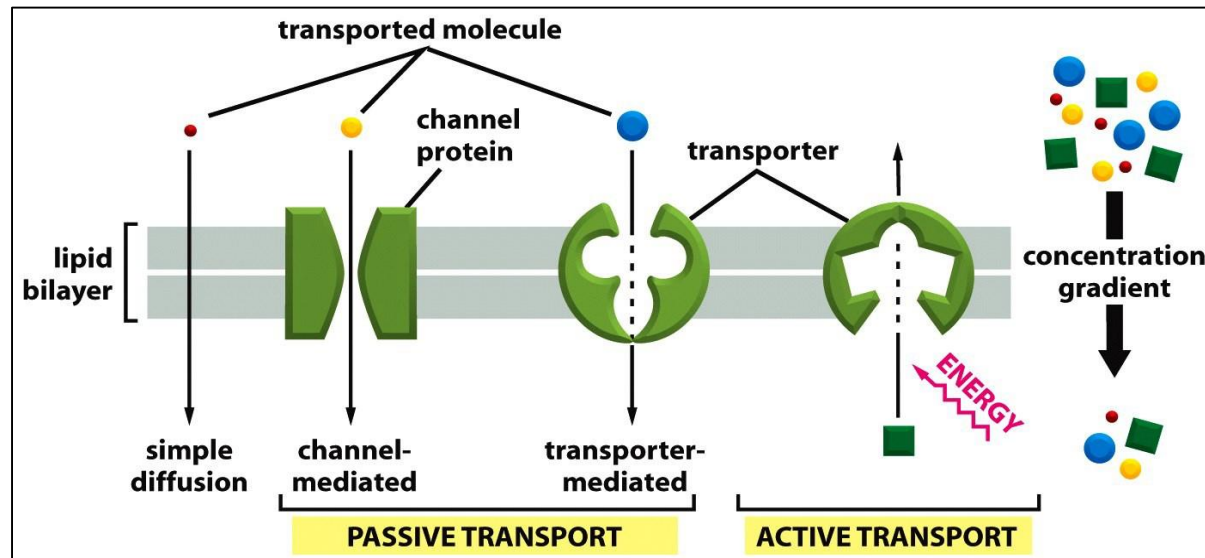


A - [carrier protein](#) alternates between two conformations,
→ [solute-binding site](#) is sequentially accessible on one side of the bilayer and then on the other

B - [channel protein](#) forms a water-filled pore across the bilayer through which specific solutes can diffuse

Passive and active transport

- ✓ **Passive transport** – all channel proteins and many carriers
- ✓ **Active transport** – only carriers; requires energy



- ✓ **Passive transport** down an electrochemical gradient occurs spontaneously
 - simple diffusion through the lipid bilayer
 - facilitated diffusion through channels and passive carriers
- ✓ **Active transport**
 - requires an input of metabolic energy
 - mediated by carriers that harvest metabolic energy to pump the solute against its electrochem.

Review Questions

- What are the functions of lipids in the cell?
- Explain the main functions in biomembranes.
- Explain the transport of ions and small molecules across cell membranes.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.

Module 1. Molecular basis of life

Topic 2. Fundamentals of Molecular Biology

Lesson 4. Cellular energy. Oxidation of glucose and fatty acids to carbon dioxide

Contents

- Introduction
- Food Molecules Are Broken Down in Three Stages to Produce ATP
- Glycolysis Is a Central ATP-producing Pathway
- Fermentations Allow ATP to Be Produced in the Absence of Oxygen
- Glycolysis Illustrates How Enzymes Couple Oxidation to Energy Storage
- Sugars and Fats Are Both Degraded to Acetyl CoA in Mitochondria
- The Citric Acid Cycle Generates NADH by Oxidizing Acetyl Groups to CO₂

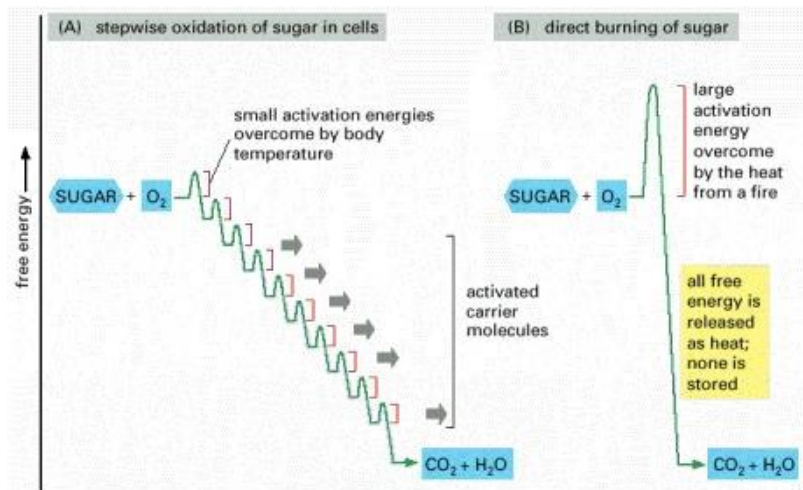


Introduction

Cells require a constant supply of energy to generate and maintain the biological order that keeps them alive. This energy is derived from the chemical bond energy in food molecules, which thereby serve as fuel for cells.

Schematic representation of the controlled stepwise oxidation of sugar in a cell, compared with ordinary burning

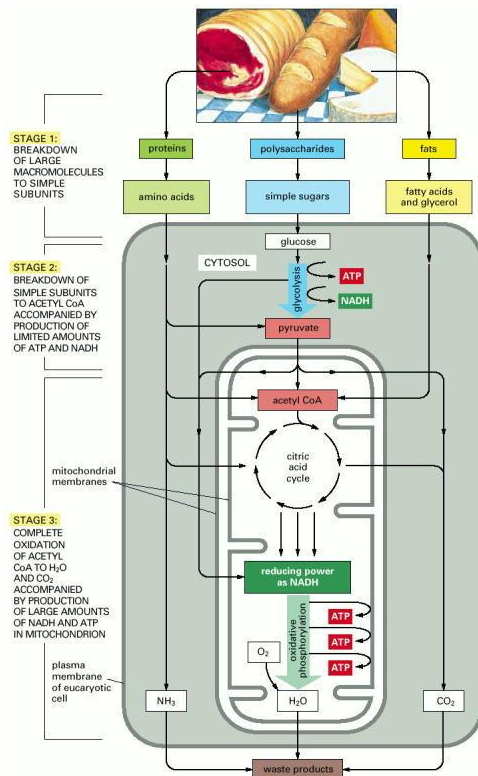
(A) In the cell, enzymes catalyze oxidation via a series of small steps in which free energy is transferred in conveniently sized packets to carrier molecules—most often ATP and NADH. At each step, an enzyme controls the reaction by reducing the activation energy barrier that has to be surmounted before the specific reaction can occur. The total free energy released is exactly the same in (A) and (B). But if the sugar was instead oxidized to CO_2 and H_2O in a single step, as in (B), it would release an amount of energy much larger than could be captured for useful purposes.



Food Molecules Are Broken Down in Three Stages to Produce ATP

Simplified diagram of the three stages of cellular metabolism that lead from food to waste products in animal cells

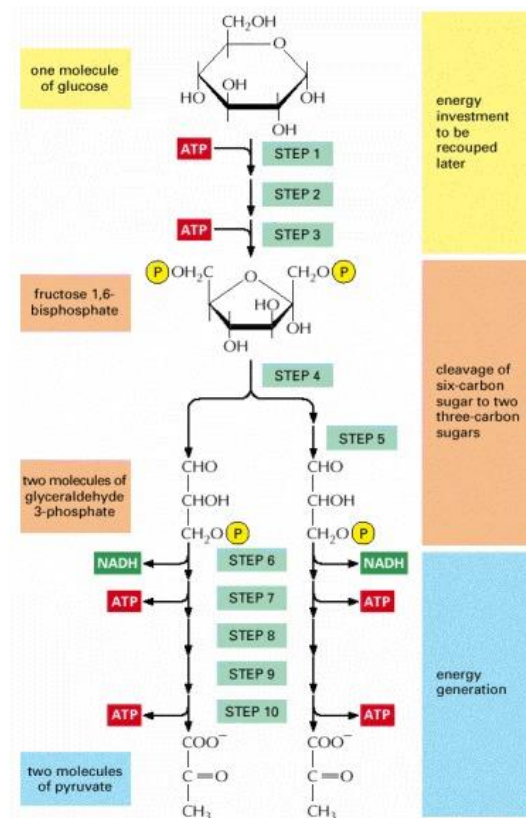
This series of reactions produces ATP, which is then used to drive biosynthetic reactions and other energy-requiring processes in the cell. Stage 1 occurs outside cells. Stage 2 occurs mainly in the cytosol, except for the final step of conversion of pyruvate to acetyl groups on acetyl CoA, which occurs in mitochondria. Stage 3 occurs in mitochondria.



Glycolysis Is a Central ATP-producing Pathway

An outline of glycolysis

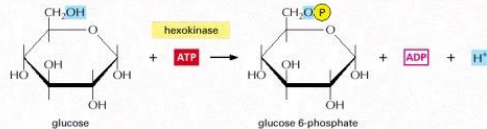
Each of the 10 steps shown is catalyzed by a different enzyme. Note that step 4 cleaves a six-carbon sugar into two three-carbon sugars, so that the number of molecules at every stage after this doubles. As indicated, step 6 begins the energy generation phase of glycolysis, which causes the net synthesis of ATP and NADH molecules.



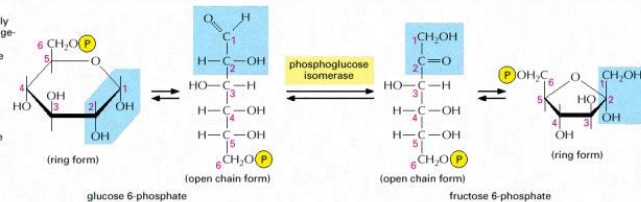
Details of the 10 Steps of Glycolysis

For each step, the part of the molecule that undergoes a change is shadowed in **blue**, and the name of the enzyme that catalyzes the reaction is in a **yellow box**.

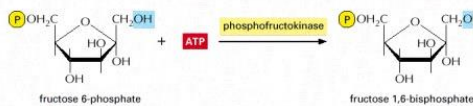
STEP 1 Glucose is phosphorylated by ATP to form a sugar phosphate. The negative charge of the phosphate prevents passage of the sugar phosphate through the plasma membrane, trapping glucose inside the cell.



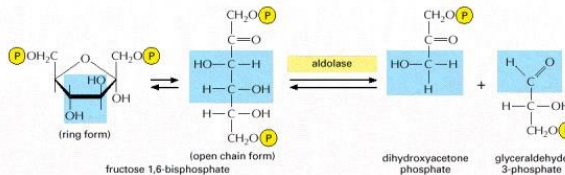
STEP 2 A readily reversible rearrangement of the chemical structure (isomerization) moves the carbonyl oxygen from carbon 1 to carbon 2, forming a ketose from an aldose sugar. (See Panel 2-4.)



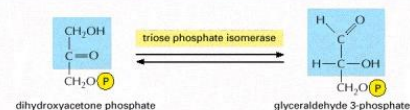
STEP 3 The new hydroxyl group on carbon 1 is phosphorylated by ATP, in preparation for the formation of two three-carbon sugar phosphates. The entry of sugars into glycolysis is controlled at this step, through regulation of the enzyme *phosphofructokinase*.



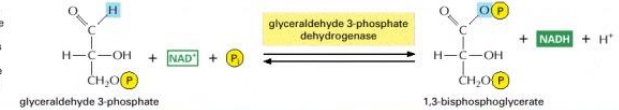
STEP 4 The six-carbon sugar is cleaved to produce two three-carbon molecules. Only the glyceraldehyde 3-phosphate can proceed immediately through glycolysis.



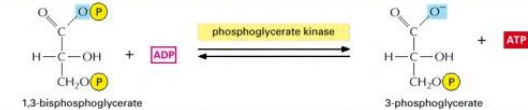
STEP 5 The other product of step 4, dihydroxyacetone phosphate, is isomerized to form glyceraldehyde 3-phosphate.



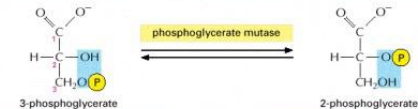
STEP 6 The two molecules of glyceraldehyde 3-phosphate are oxidized. The energy generation phase of glycolysis begins, as NADH and a new high-energy anhydride linkage to phosphate are formed (see Figure 2-73).



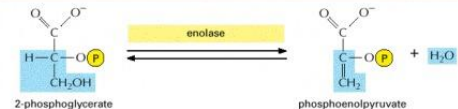
STEP 7 The transfer to ADP of the high-energy phosphate group that was generated in step 6 forms ATP.



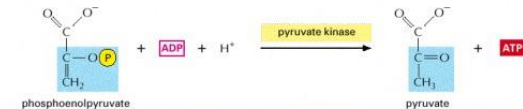
STEP 8 The remaining phosphate ester linkage in 3-phosphoglycerate, which has a relatively low free energy of hydrolysis, is moved from carbon 3 to carbon 2 to form 2-phosphoglycerate.



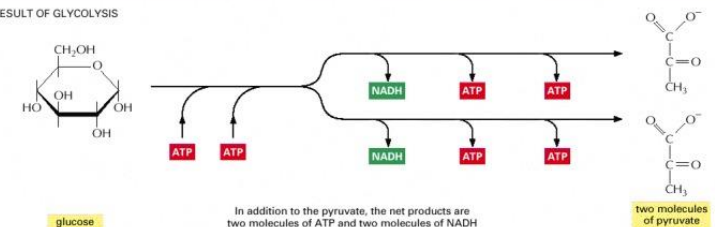
STEP 9 The removal of water from 2-phosphoglycerate creates a high-energy enol phosphate linkage.



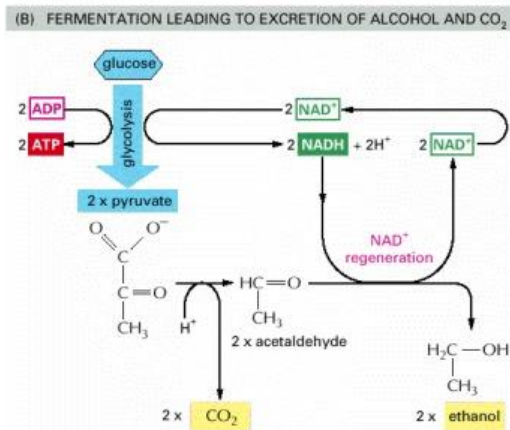
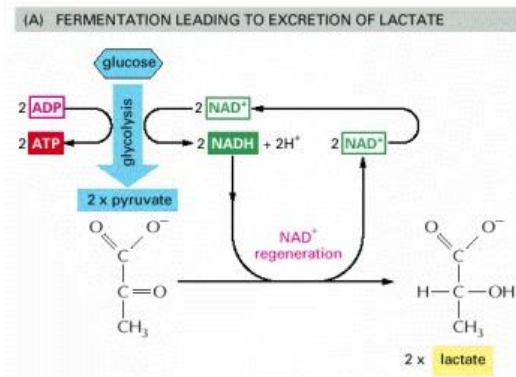
STEP 10 The transfer to ADP of the high-energy phosphate group that was generated in step 9 forms ATP, completing glycolysis.



NET RESULT OF GLYCOLYSIS



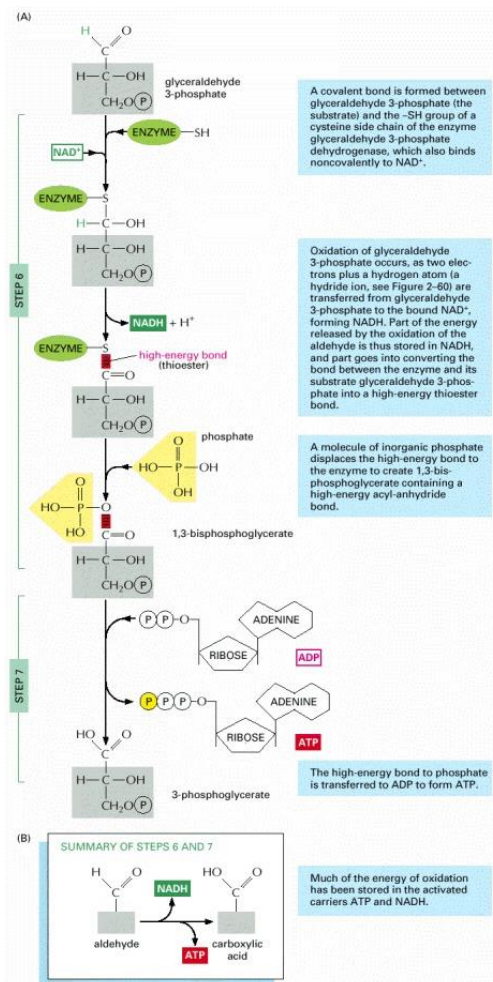
Fermentations Allow ATP to Be Produced in the Absence of Oxygen



Two pathways for the anaerobic breakdown of pyruvate

(A) When inadequate oxygen is present, for example, in a muscle cell undergoing vigorous contraction, the pyruvate produced by glycolysis is converted to lactate as shown. This reaction regenerates the NAD⁺ consumed in step 6 of glycolysis, but the whole pathway yields much less energy overall than complete oxidation. (B) In some organisms that can grow anaerobically, such as yeasts, pyruvate is converted via acetaldehyde into carbon dioxide and ethanol. Again, this pathway regenerates NAD⁺ from NADH, as required to enable glycolysis to continue. Both (A) and (B) are examples of fermentations.

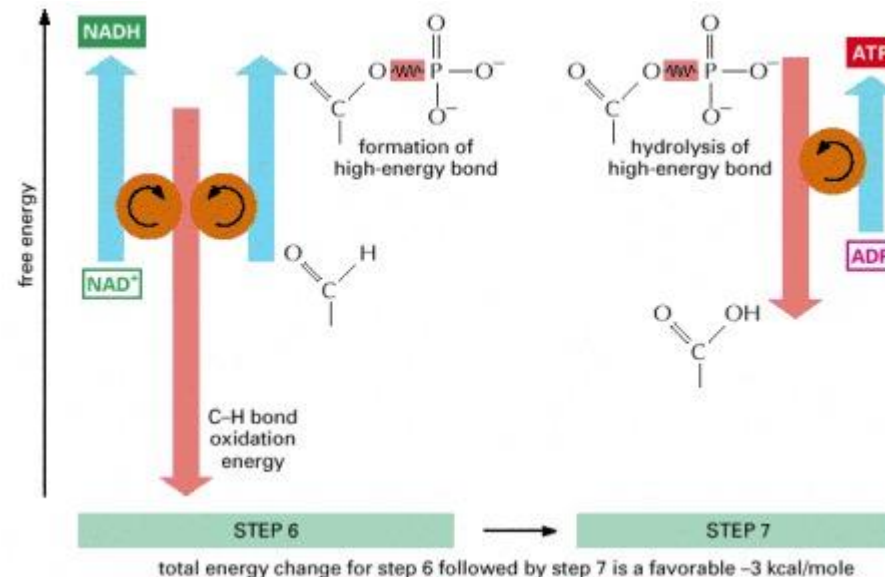
Glycolysis Illustrates How Enzymes Couple Oxidation to Energy Storage



Energy storage in steps 6 and 7 of glycolysis

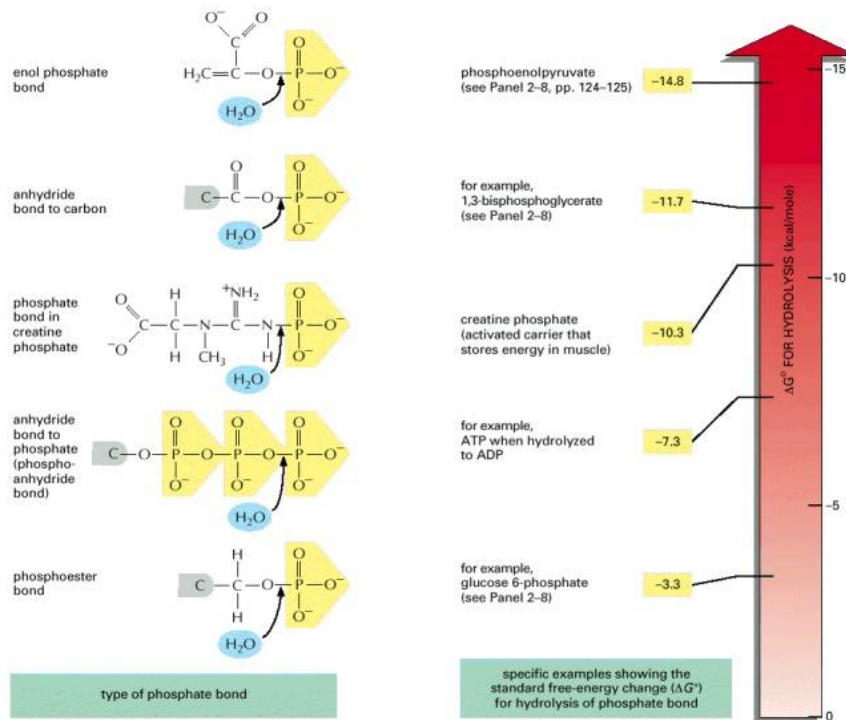
In these steps the oxidation of an aldehyde to a carboxylic acid is coupled to the formation of ATP and NADH. (A) Step 6 begins with the formation of a covalent bond between the substrate (glyceraldehyde 3-phosphate) and an -SH group exposed on the surface of the enzyme (glyceraldehyde 3-phosphate dehydrogenase). The enzyme then catalyzes transfer of hydrogen (as a hydride ion—a proton plus two electrons) from the bound glyceraldehyde 3-phosphate to a molecule of NAD^+ . Part of the energy released in this oxidation is used to form a molecule of NADH and part is used to convert the original linkage between the enzyme and its substrate to a high-energy thioester bond (shown in red). A molecule of inorganic phosphate then displaces this high-energy bond on the enzyme, creating a high-energy sugar-phosphate bond instead (red). At this point the enzyme has not only stored energy in NADH, but also coupled the energetically favorable oxidation of an aldehyde to the energetically unfavorable formation of a high-energy phosphate bond. The second reaction has been driven by the first, thereby acting like the “paddle wheel” coupler in Figure 2-56.

In reaction step 7, the high-energy sugar-phosphate intermediate just made, 1,3-bisphosphoglycerate, binds to a second enzyme, phosphoglycerate kinase. The reactive phosphate is transferred to ADP, forming a molecule of ATP and leaving a free carboxylic acid group on the oxidized sugar. (B) Summary of the overall chemical change produced by reactions 6 and 7.



Schematic view of the coupled reactions that form NADH and ATP in steps 6 and 7 of glycolysis

The C-H bond oxidation energy drives the formation of both NADH and a high-energy phosphate bond. The breakage of the high-energy bond then drives ATP formation.



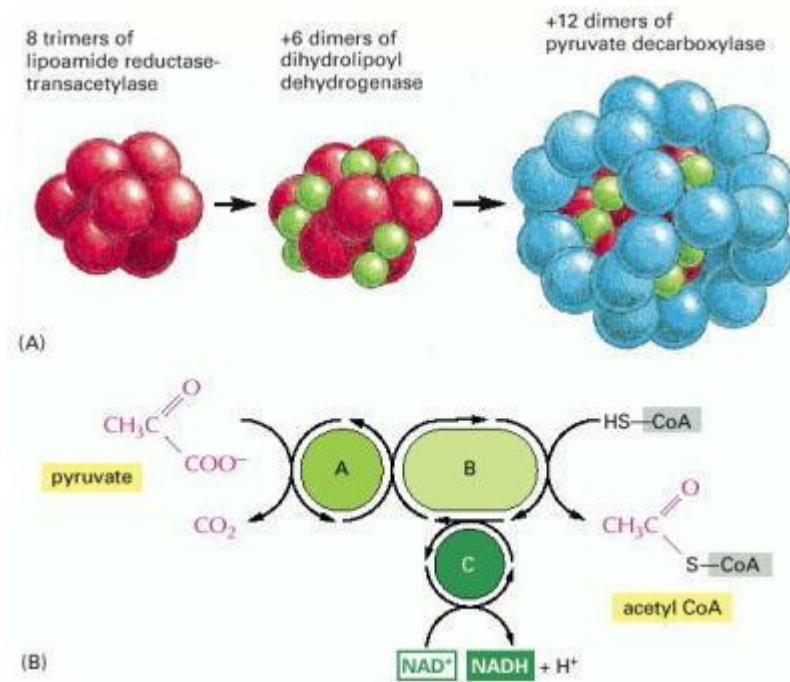
Some phosphate bond energies

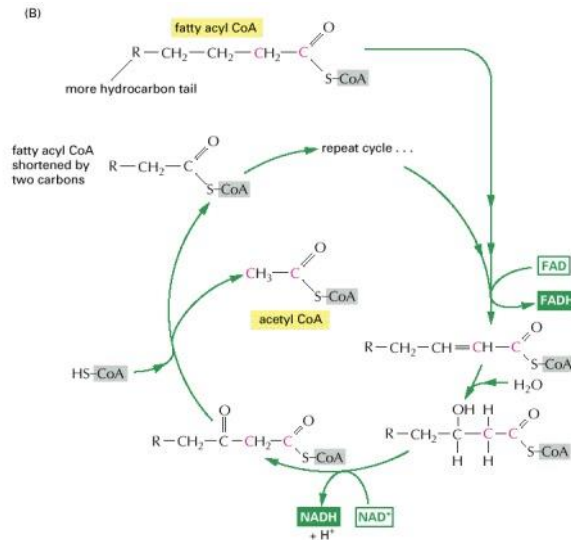
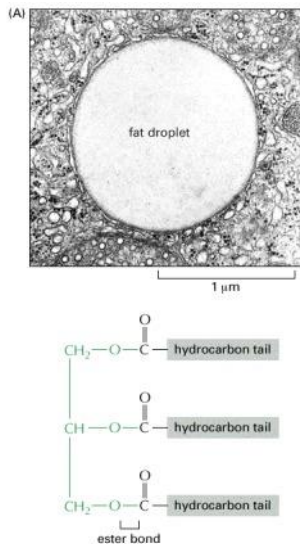
The transfer of a phosphate group from any molecule 1 to any molecule 2 is energetically favorable if the standard free-energy change (ΔG°) for the hydrolysis of the phosphate bond in molecule 1 is more negative than that for hydrolysis of the phosphate bond in molecule 2. Thus, for example, a phosphate group is readily transferred from 1,3-bisphosphoglycerate to ADP, forming ATP. Note that the hydrolysis reaction can be viewed as the transfer of the phosphate group to water.

Sugars and Fats Are Both Degraded to Acetyl CoA in Mitochondria

The oxidation of pyruvate to acetyl CoA and CO₂

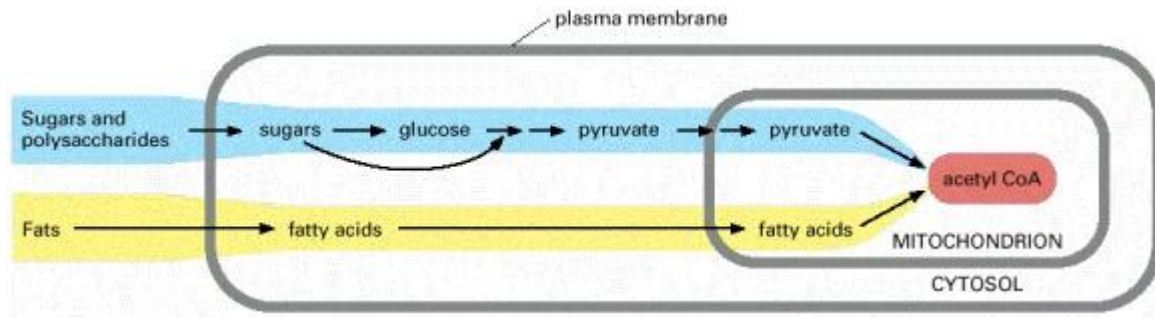
(A) The structure of the pyruvate dehydrogenase complex, which contains 60 polypeptide chains. This is an example of a large multienzyme complex in which reaction intermediates are passed directly from one enzyme to another. In eucaryotic cells it is located in the mitochondrion. (B) The reactions carried out by the pyruvate dehydrogenase complex. The complex converts pyruvate to acetyl CoA in the mitochondrial matrix; NADH is also produced in this reaction. A, B, and C are the three enzymes pyruvate decarboxylase, lipoamide reductase-transacetylase, and dihydrolipoyl dehydrogenase, respectively. These enzymes are illustrated in (A); their activities are linked as shown.





The oxidation of fatty acids to acetyl CoA

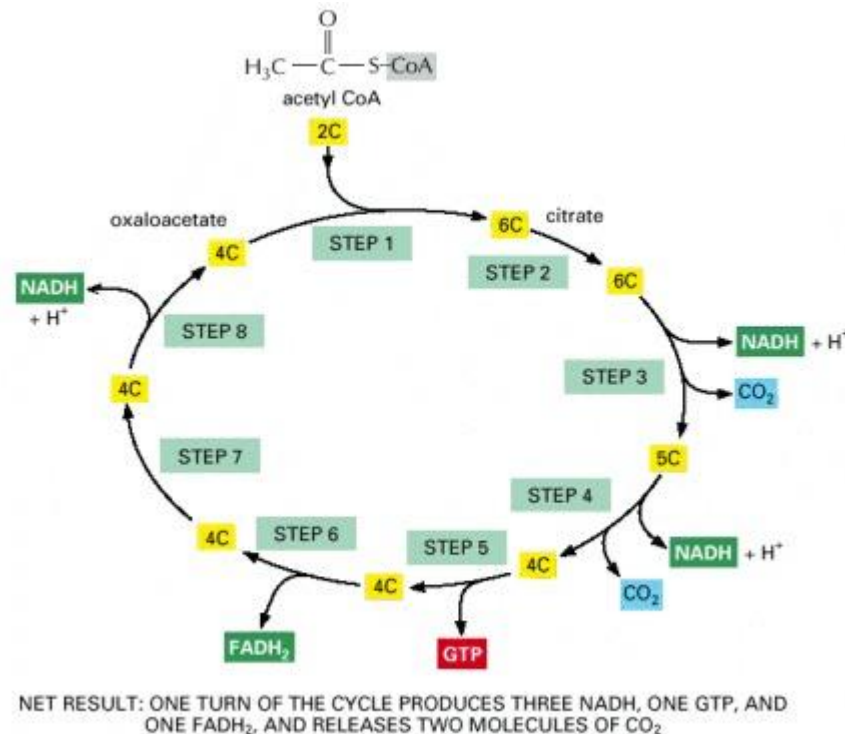
(A) Electron micrograph of a lipid droplet in the cytoplasm (top), and the structure of fats (bottom). Fats are triacylglycerols. The glycerol portion, to which three fatty acids are linked through ester bonds, is shown here in green. Fats are insoluble in water and form large lipid droplets in the specialized fat cells (called adipocytes) in which they are stored. (B) The fatty acid oxidation cycle. The cycle is catalyzed by a series of four enzymes in the mitochondrion. Each turn of the cycle shortens the fatty acid chain by two carbons (shown in red) and generates one molecule of acetyl CoA and one molecule each of NADH and FADH₂.



Pathways for the production of acetyl CoA from sugars and fats

The mitochondrion in eucaryotic cells is the place where acetyl CoA is produced from both types of major food molecules. It is therefore the place where most of the cell's oxidation reactions occur and where most of its ATP is made.

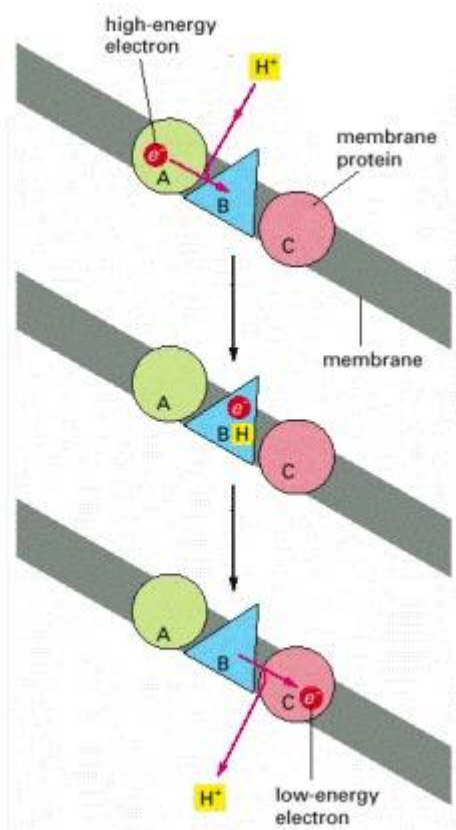
The Citric Acid Cycle Generates NADH by Oxidizing Acetyl Groups to CO₂



Simple overview of the citric acid cycle

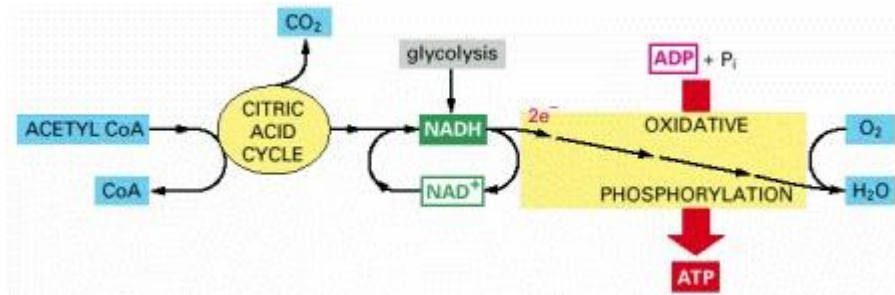
The reaction of acetyl CoA with oxaloacetate starts the cycle by producing citrate (citric acid). In each turn of the cycle, two molecules of CO₂ are produced as waste products, plus three molecules of NADH, one molecule of GTP, and one molecule of FADH₂. The number of carbon atoms in each intermediate is shown in a yellow box.

Electron Transport Drives the Synthesis of the Majority of the ATP in Most Cells



The generation of an H^+ gradient across a membrane by electron-transport reactions

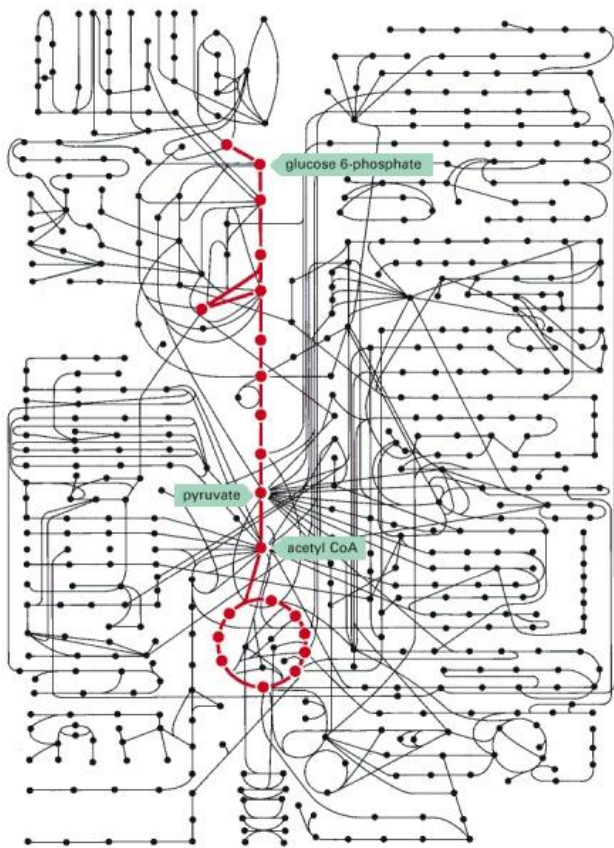
A high-energy electron (derived, for example, from the oxidation of a metabolite) is passed sequentially by carriers A, B, and C to a lower energy state. In this diagram carrier B is arranged in the membrane in such a way that it takes up H^+ from one side and releases it to the other as the electron passes. The result is an H^+ gradient. This gradient represents a form of stored energy that is harnessed by other membrane proteins to drive the formation of ATP.



The final stages of oxidation of food molecules

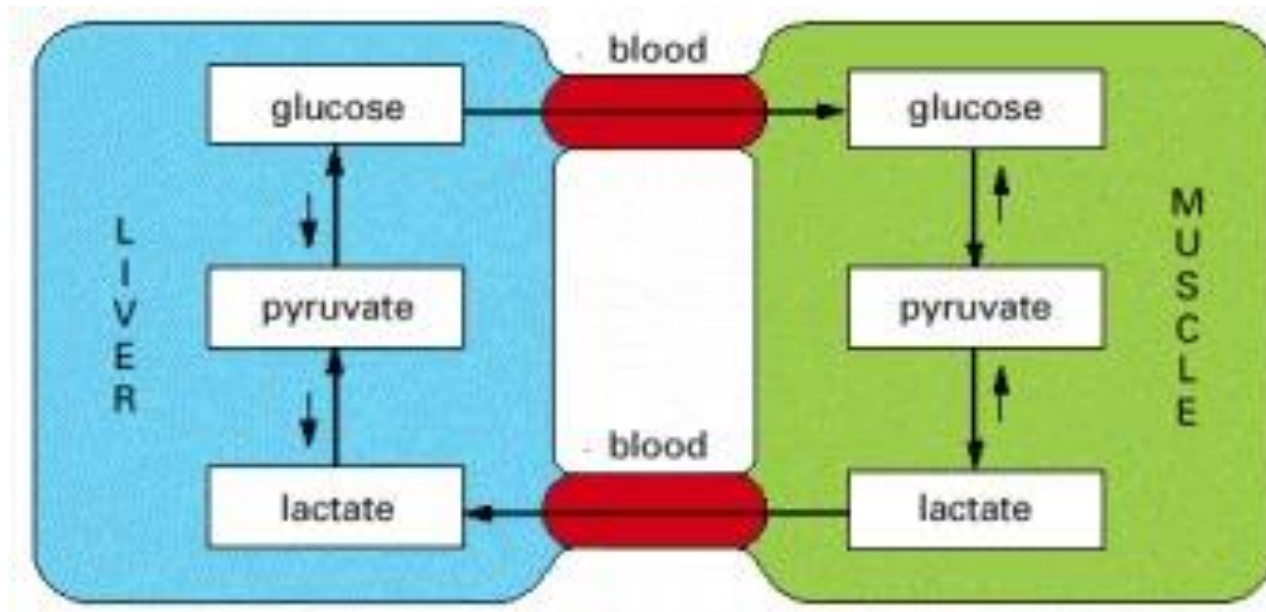
Molecules of NADH and FADH₂ (FADH₂ is not shown) are produced by the citric acid cycle. These activated carriers donate high-energy electrons that are eventually used to reduce oxygen gas to water. A major portion of the energy released during the transfer of these electrons along an electron-transfer chain in the mitochondrial inner membrane (or in the plasma membrane of bacteria) is harnessed to drive the synthesis of ATP: hence the name oxidative phosphorylation.

Metabolism Is Organized and Regulated



Glycolysis and the citric acid cycle are at the center of metabolism

Some 500 metabolic reactions of a typical cell are shown schematically with the reactions of glycolysis and the citric acid cycle in red. Other reactions either lead into these two central pathways—delivering small molecules to be catabolized with production of energy—or they lead outward and thereby supply carbon compounds for the purpose of biosynthesis



Schematic view of the metabolic cooperation between liver and muscle cells

The principal fuel of actively contracting muscle cells is glucose, much of which is supplied by liver cells. Lactic acid, the end product of anaerobic glucose breakdown by glycolysis in muscle, is converted back to glucose in the liver by the process of gluconeogenesis.

Review Questions

- What is the source of cellular energy?
- Explain the energy production pathways.
- What is the mechanism of oxidation of glucose and fatty acids to carbon dioxide?



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 2. Fundamentals of Molecular Biology

Lesson 5. Signaling the cell surface. Signal molecules and receptors. G-protein bound receptors. Signal pathways controlling gene activity



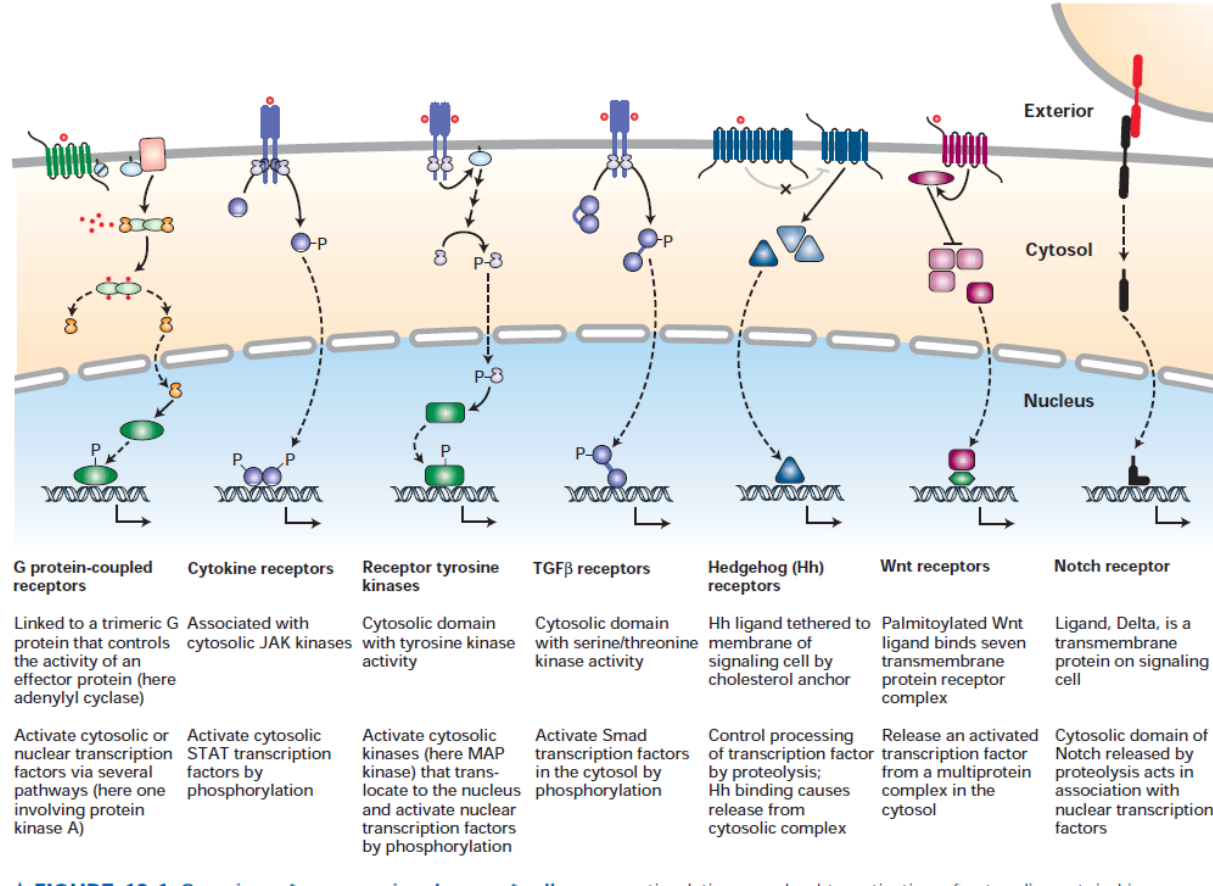
Contents

- Introduction
- Signaling the cell surface
- Signal molecules and receptors
- G-protein bound receptors
- Signal pathways controlling gene activity



Introduction

Important in plants and animals are extracellular signaling molecules that function within an organism to control metabolic processes within cells, the growth and differentiation of tissues, the synthesis and secretion of proteins, and the composition of intracellular and extracellular fluids. Adjacent cells often communicate by direct cell-cell contact. Extracellular signaling molecules are synthesized and released by signaling cells and produce a specific response only in target cells that have receptors for the signaling molecules. In multicellular organisms, an enormous variety of chemicals, including small molecules (e.g., amino acid or lipid derivatives, acetylcholine), peptides, and proteins, are used in this type of cell-to-cell communication. Some signaling molecules, especially hydrophobic molecules such as steroids, retinoids, and thyroxine, spontaneously diffuse through the plasma membrane and bind to intracellular receptors.



Overview of seven major classes of cell surface receptors

Signaling the cell surface

The signaling molecule acts as a ligand, which binds to a structurally complementary site on the extracellular or membrane-spanning domains of the receptor. Binding of a ligand to its receptor causes a conformational change in the cytosolic domain or domains of the receptor that ultimately induces specific cellular responses. The overall process of converting signals into cellular responses, as well as the individual steps in this process, is termed signal transduction. As we will see, signal transduction pathways may involve relatively few or many components.

Signaling Molecules and Cell-Surface Receptors

Communication by extracellular signals usually involves the following steps: (1) synthesis and (2) release of the signaling molecule by the signaling cell; (3) transport of the signal to the target cell; (4) binding of the signal by a specific receptor protein leading to its activation; (5) initiation of one or more intracellular signal-transduction pathways by the activated receptor; (6) specific changes in cellular function, metabolism, or development; and (7) removal of the signal, which often terminates the cellular response.

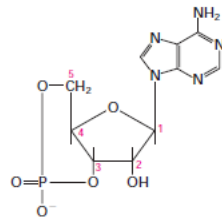
The vast majority of receptors are activated by binding of secreted or membrane-bound molecules (e.g., hormones, growth factors, neurotransmitters, and pheromones).

- Signaling Molecules in Animals Operate over Various Distances
- Receptors Activate a Limited Number of Signaling Pathways

Intracellular Signal Transduction

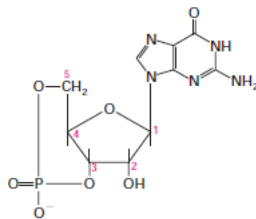
Second Messengers Carry Signals from Many Receptors

The binding of ligands (“first messengers”) to many cell surface receptors leads to a short-lived increase (or decrease) in the concentration of certain low-molecular-weight intracellular signaling molecules termed second messengers. These molecules include 3',5'-cyclic AMP (cAMP), 3',5'-cyclic GMP (cGMP), 1,2-diacylglycerol (DAG), and inositol 1,4,5-trisphosphate (IP₃), whose structures are shown in Figure. Other important second messengers are Ca²⁺ and various inositol phospholipids, also called phosphoinositides, which are embedded in cellular membranes. The major direct effect or effects of each compound are indicated below its structural formula. Calcium ion (Ca²⁺) and several membrane-bound phosphoinositides also act as second messengers.



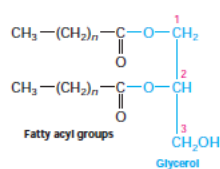
3',5'-Cyclic AMP
(cAMP)

Activates protein kinase A (PKA)



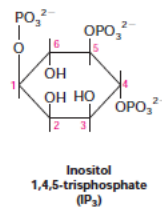
3',5'-Cyclic GMP
(cGMP)

Activates protein kinase G (PKG)
and opens cation channels in
rod cells



1,2-Diacylglycerol
(DAG)

Activates protein kinase C
(PKC)

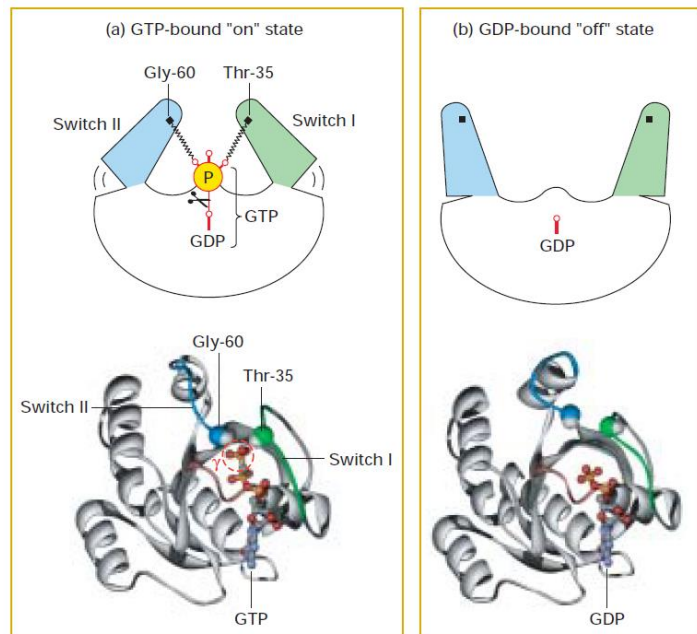


Inositol
1,4,5-trisphosphate
(IP₃)

Opens Ca²⁺ channels in
the endoplasmic reticulum

Intracellular Signal Transduction

Many Conserved Intracellular Proteins Function in Signal Transduction



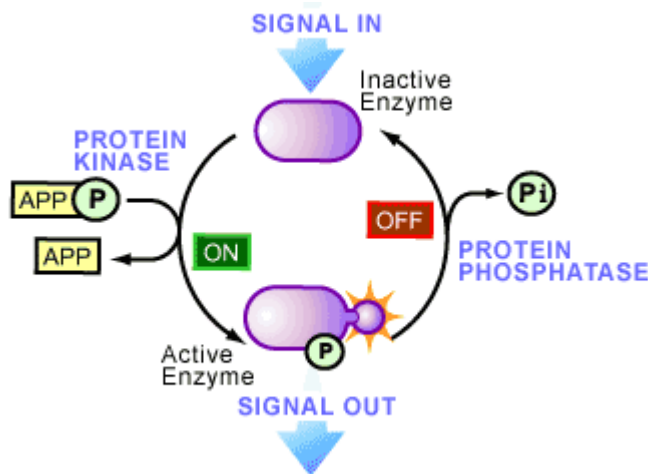
GTPase Switch Proteins are guanine nucleotide-binding proteins are turned “on” when bound to GTP and turned “off” when bound to GDP. Signal-induced conversion of the inactive to active state is mediated by a guanine nucleotide-exchange factor (GEF), which causes release of GDP from the switch protein. Subsequent binding of GTP, favored by its high intracellular concentration, induces a conformational change in two segments of the protein, termed switch I and switch II, allowing the protein to bind to and activate other downstream signaling proteins.

Intracellular Signal Transduction

Many Conserved Intracellular Proteins Function in Signal Transduction

Protein Kinases and Phosphatases

Activation of all cell surface receptors leads directly or indirectly to changes in protein phosphorylation through the activation of protein kinases or protein phosphatases. Animal cells contain two types of protein kinases: those that add phosphate to the hydroxyl group on tyrosine residues and those that add phosphate to the hydroxyl group on serine or threonine (or both) residues. Phosphatases, which remove phosphate groups, can act in concert with kinases to switch the function of various proteins on or off

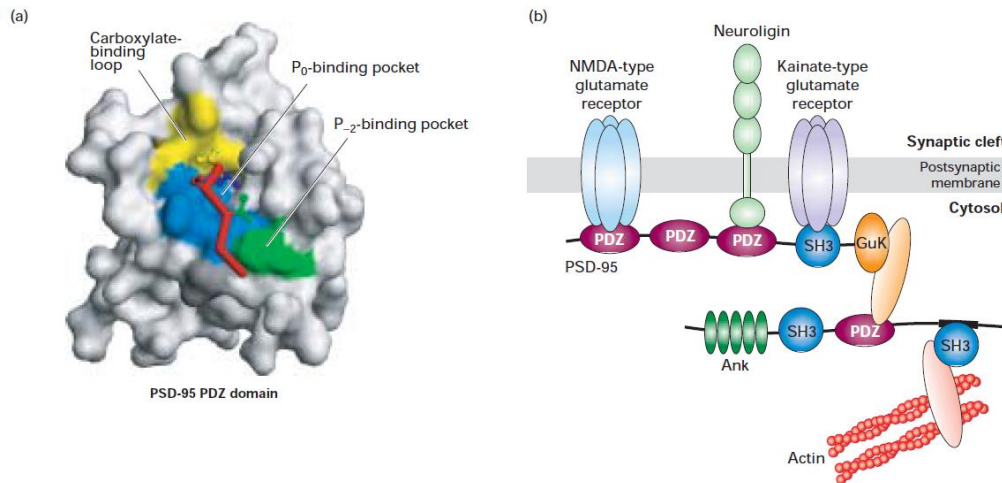


Intracellular Signal Transduction

Some Receptors and Signal-Transduction Proteins Are Localized

Clustering of Membrane Proteins Mediated by Adapter Domains

Protein Clustering in Lipid Rafts



Intracellular Signal Transduction

Appropriate Cellular Responses Depend on Interaction and Regulation of Signaling Pathways

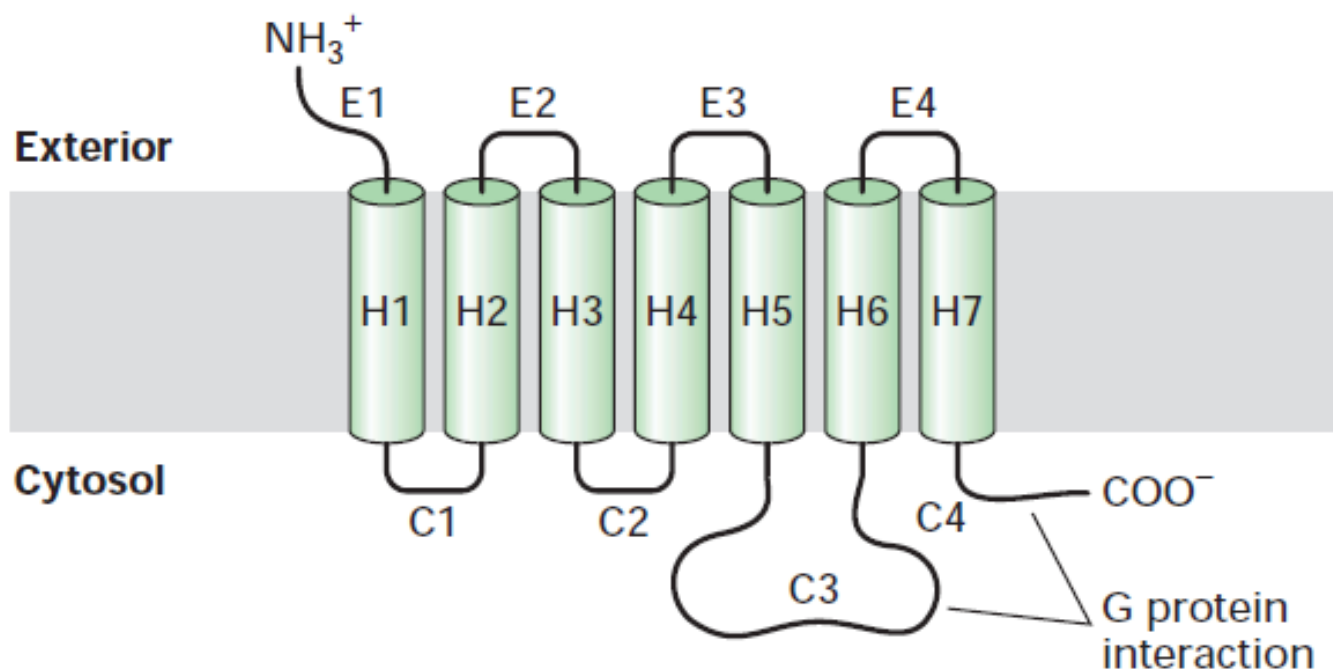
The ability of cells to respond appropriately to extracellular signals also depends on regulation of signaling pathways themselves. The sensitivity of a cell to a particular signaling molecule can be down-regulated by endocytosis of its receptors, thus decreasing the number on the cell surface, or by modifying their activity so that the receptors either cannot bind ligand or form a receptor-ligand complex that does not induce the normal cellular response. Such modulation of receptor activity often results from phosphorylation of the receptor, binding of other proteins to it, or both.

G Protein–Coupled Receptors That Activate or Inhibit Adenylyl Cyclase

| G_{α} Class | Associated Effector | 2nd Messenger | Receptor Examples |
|--------------------|---|--|--|
| $G_{s\alpha}$ | Adenylyl cyclase | cAMP (increased) | β -Adrenergic (epinephrine) receptor; receptors for glucagon, serotonin, vasopressin |
| $G_{i\alpha}$ | Adenylyl cyclase K^+ channel ($G_{\beta\gamma}$ activates effector) | cAMP (decreased) Change in membrane potential | α_1 -Adrenergic receptor Muscarinic acetylcholine receptor |
| $G_{olf\alpha}$ | Adenylyl cyclase | cAMP (increased) | Odorant receptors in nose |
| $G_{q\alpha}$ | Phospholipase C | IP_3 , DAG (increased) | α_2 -Adrenergic receptor |
| $G_{o\alpha}$ | Phospholipase C | IP_3 , DAG (increased) | Acetylcholine receptor in endothelial cells |
| $G_{t\alpha}$ | cGMP phosphodiesterase | cGMP (decreased) | Rhodopsin (light receptor) in rod cells |

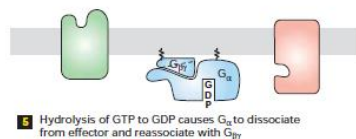
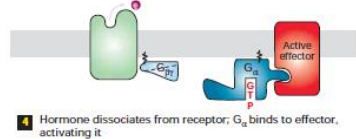
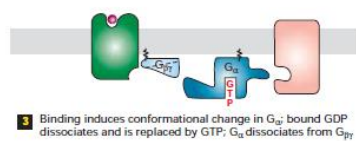
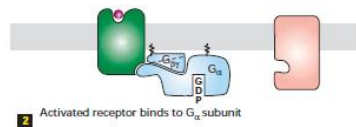
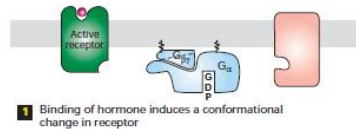
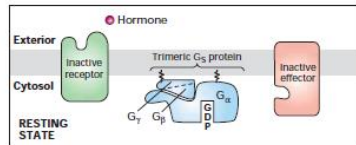
*A given G_{α} subclass may be associated with more than one effector protein. To date, only one major $G_{s\alpha}$ has been identified, but multiple $G_{q\alpha}$ and $G_{i\alpha}$ proteins have been described. Effector proteins commonly are regulated by G_{α} but in some cases by $G_{\beta\gamma}$ or the combined action of G_{α} and $G_{\beta\gamma}$. IP_3 = inositol 1,4,5-trisphosphate; DAG = 1,2-diacylglycerol.

SOURCES: See L. Birnbaumer, 1992, *Cell* 71:1069; Z. Farfel et al., 1999, *New Eng. J. Med.* 340:1012; and K. Pierce et al., 2002, *Nature Rev. Mol. Cell Biol.* 3:639.



Schematic diagram of the general structure of G protein–coupled receptors.

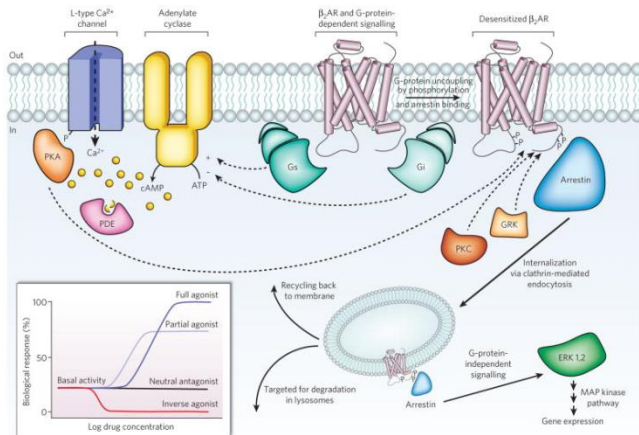
The $G\alpha$ Subunit of G Proteins Cycles Between Active and Inactive Forms



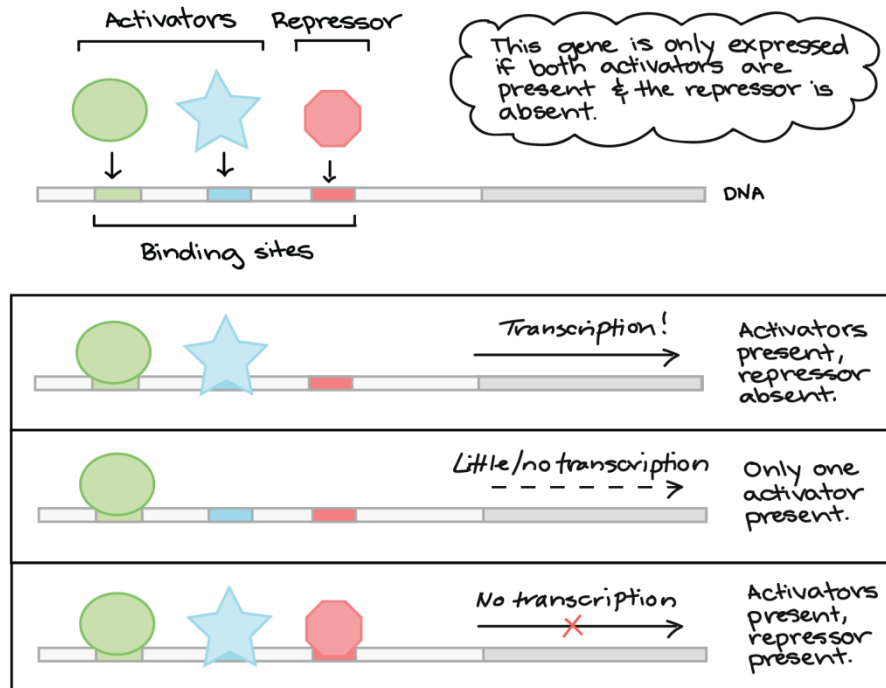
The $G\alpha$ and $G\beta\gamma$ subunits of trimeric G proteins are tethered to the membrane by covalently attached lipid molecules (wiggly black lines). Following ligand binding, dissociation of the G protein, and exchange of GDP with GTP (steps 1 – 3), the free $G\alpha$ -GTP binds to and activates an effector protein (step 4). Hydrolysis of GTP terminates signaling and leads to reassembly of the trimeric form, returning the system to the resting state (step 5). Binding of another ligand molecule causes repetition of the cycle. In some pathways, the effector protein is activated by the free $G\beta\gamma$ subunit.

Signal transduction in G-protein-coupled receptors

Diverse signalling pathways regulated by the type 2 beta adrenergic receptor (β_2 AR). The β_2 AR can activate two G proteins, $G_{\alpha s}$ and $G_{\alpha i}$ (part of the Gs and Gi heterotrimers, respectively), which differentially regulate adenylate cyclase. Adenylate cyclase generates cyclic AMP (cAMP), which activates protein kinase A (PKA), a kinase that regulates the activity of several cellular proteins including the L-type Ca^{2+} channel and the β_2 AR. cAMP second messenger levels are downregulated by specific phosphodiesterase proteins (PDEs). Activation of the β_2 AR also leads to phosphorylation by a G-protein-coupled receptor kinase (GRK) and subsequent coupling to arrestin. Arrestin is a signalling and regulatory protein that promotes the activation of extracellular signal-regulated kinases (ERK), prevents the activation of G proteins and promotes the internalization of the receptor through clathrin-coated pits. PKC, protein kinase C. The inset shows classification of ligand efficacy for GPCRs. Many GPCRs exhibit basal, agonist-independent activity. Inverse agonists inhibit this activity, and neutral antagonists have no effect. Agonists and partial agonists stimulate biological responses above the basal activity. Efficacy is not directly related to affinity; for example, a partial agonist can have a higher affinity for a GPCR than a full agonist.

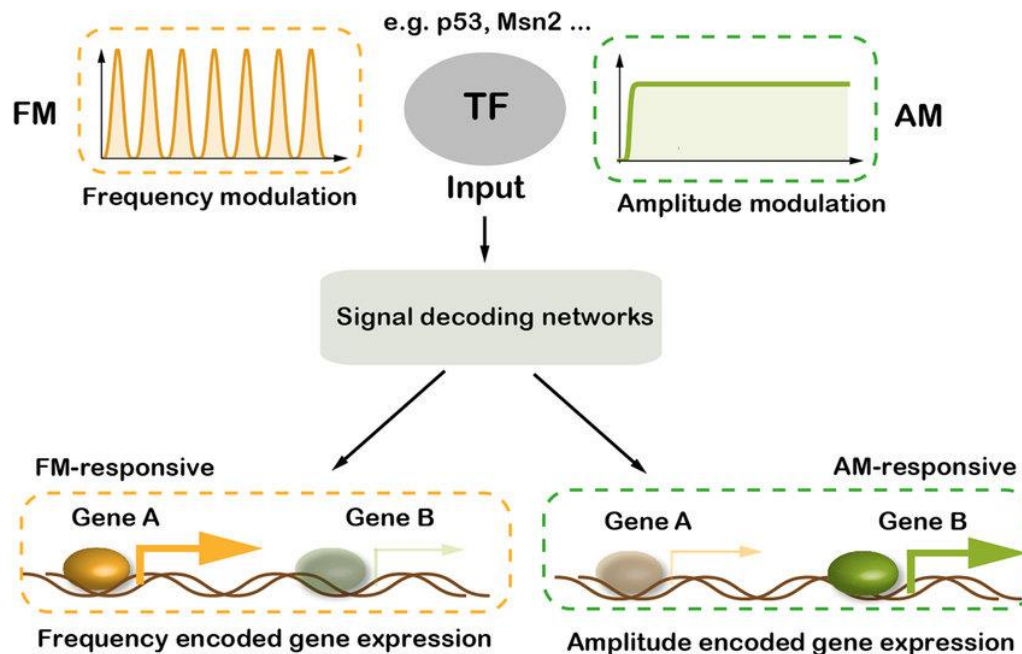


General and Gene-Specific Transcription Factors



Protein coding genes are transcribed by RNA polymerase II in the presence of half a dozen general transcription factors. These factors help to position the polymerase close to the initiation site and permit a low level of basal transcription. These factors are frequently composed of several subunits and altogether the initiation complex contains around 30 different polypeptides. Usually, each gene is regulated by a number of transcription factors binding to promoter and enhancer sequences. For genes that are expressed in a unique or a small number of cell types, one or more of these transcription factors are expressed in a single or a limited number of cell types.

Modulation of Transcription Factor Activity



The cells are exposed to soluble small molecules, to proteins in the circulation or to proteins (glycoproteins) on the surface of neighbouring cells. Changes in the concentration of such molecules are sensed by cells that harbour the corresponding receptors. We will discuss separately hormone and vitamin receptors and receptors for growth factors and cytokines. Soluble molecules such as steroid hormones, vitamins etc., diffuse through the membrane and associate with their corresponding receptors.

Review Questions

- Explain the mechanism of ligand-receptor interaction.
- Explain the mechanism of signal transduction.
- Explain the signal pathways controlling gene activity.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 1. Structure and properties of amino acids



Contents

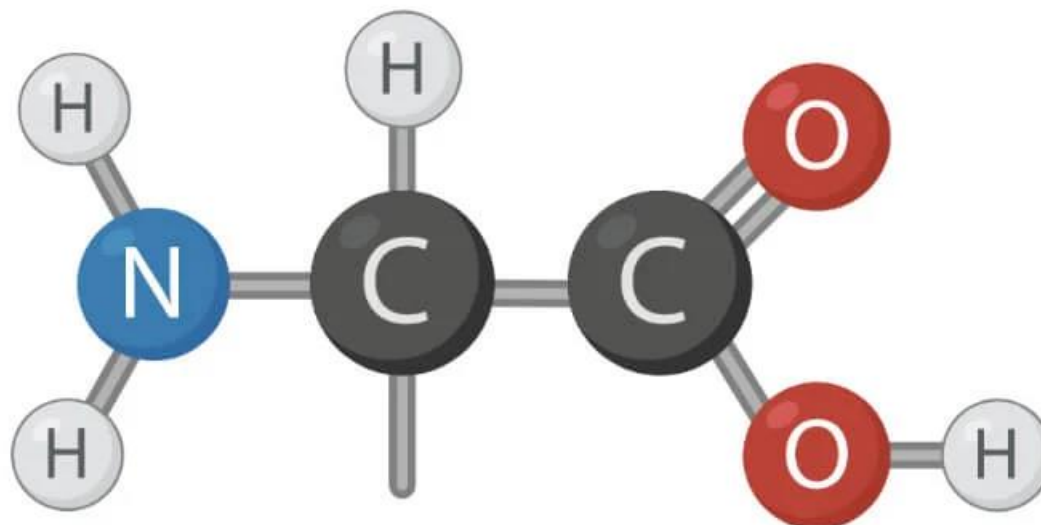
- Introduction
- Physical Properties
- Chemical Properties
- Structure of Amino acids
- Classification of amino acids
- Functions of Amino acids



Introduction

Amino acids constitute a group of neutral products clearly distinguished from other natural compounds chemically, mainly because of their ampholytic properties, and biochemically, mainly because of their role as protein constituents. An amino acid is a carboxylic acid-containing an aliphatic primary amino group in the α position to the carboxyl group and with a characteristic stereochemistry. Proteins are biosynthesized from 20 amino acids in a system involving strict genetic control. Thus, amino acids are the basic unit of proteins. More than 300 amino acids are found in nature but only 20 amino acids are standard and present in protein because they are coded by genes. Other amino acids are modified amino acids and are called non-protein amino acids. Some are residues modified after a protein has been synthesized by posttranslational modifications; others are amino acids present in living organisms but not as constituents of proteins.

Amino Acids- Properties, Structure, Classification, Functions



Physical Properties

- Amino acids are colorless, crystalline solid.
- All amino acids have a high melting point greater than 200o
- Solubility: They are soluble in water, slightly soluble in alcohol, and dissolve with difficulty in methanol, ethanol, and propanol. R-group of amino acids and pH of the solvent play important role in solubility.
- On heating to high temperatures, they decompose.
- All amino acids (except glycine) are optically active.
- Peptide bond formation: Amino acids can connect with a peptide bond involving their amino and carboxylate groups. A covalent bond formed between the alpha-amino group of one amino acid and an alpha-carboxyl group of other forming -CO-NH-linkage. Peptide bonds are planar and partially ionic.

Chemical Properties

- Zwitterionic property

A zwitterion is a molecule with functional groups, of which at least one has a positive and one has a negative electrical charge. The net charge of the entire molecule is zero. Amino acids are the best-known examples of zwitterions. They contain an amine group (basic) and a carboxylic group (acidic). The -NH_2 group is the stronger base, and so it picks up H^+ from the -COOH group to leave a zwitterion. The (neutral) zwitterion is the usual form of amino acids that exist in the solution.

- Amphoteric property

Amino acids are amphoteric in nature that is they act as both acids and base due to the two amine and carboxylic groups present.

Chemical Properties

- Ninhydrin test

When 1 ml of Ninhydrin solution is added to a 1 ml protein solution and heated, the formation of a violet color indicates the presence of α -amino acids.

- Xanthoproteic test

The xanthoproteic test is performed for the detection of aromatic amino acids (tyrosine, tryptophan, and phenylalanine) in a protein solution. The nitration of benzoid radicals present in the amino acid chain occurs due to a reaction with nitric acid, giving the solution yellow coloration.

Chemical Properties

- Reaction with Sanger's reagent

Sanger's reagent (1-fluoro-2, 4-dinitrobenzene) reacts with a free amino group in the peptide chain in a mild alkaline medium under cold conditions.

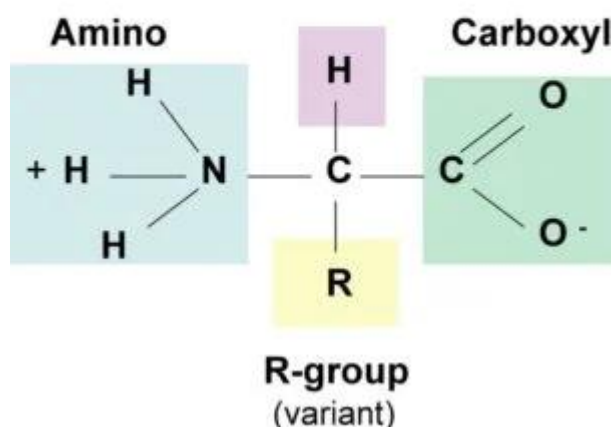
- Reaction with nitrous acid

Nitrous acid reacts with the amino group to liberate nitrogen and form the corresponding hydroxyl.

Structure of Amino acids

Amino Acid Structure

Hydrogen



All 20 of the common amino acids are alpha-amino acids. They contain a carboxyl group, an amino group, and a side chain (R group), all attached to the α -carbon.

Exceptions are:

- Glycine, which does not have a side chain. Its α -carbon contains two hydrogens.
- Proline, in which the nitrogen is part of a ring.
- Thus, each amino acid has an amine group at one end and an acid group at the other, and a distinctive side chain. The backbone is the same for all amino acids while the side chain differs from one amino acid to the next.
- All of the 20 amino acids except glycine are of the L-configuration, as for all but one amino acid the α -carbon is an asymmetric carbon. Because glycine does not contain an asymmetric carbon atom, it is not optically active and, thus, is neither D nor L.

Classification of amino acids on the basis of R-group

Amino Acid Chart

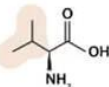
Non-polar side chains



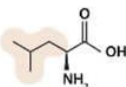
Glycine



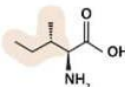
Alanine



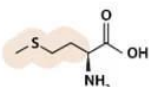
Valine



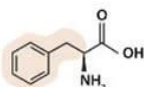
Leucine



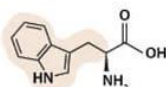
Isoleucine



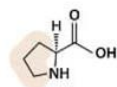
Methionine



Phenylalanine

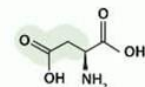


Tryptophan

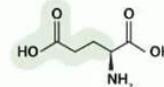


Proline

Electrically charged side chains

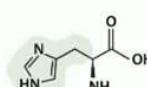


Aspartate

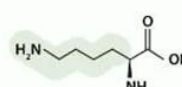


Glutamate

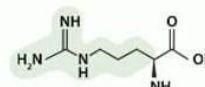
Acidic



Histidine



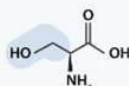
Lysine



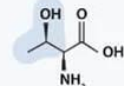
Arginine

Basic

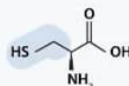
Polar side chains



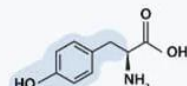
Serine



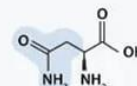
Threonine



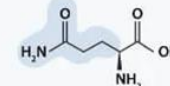
Cysteine



Tyrosine



Asparagine



Glutamine

Classification of amino acids on the basis of R-group

1. **Nonpolar, Aliphatic amino acids:** The R groups in this class of amino acids are nonpolar and hydrophobic. Glycine, Alanine, Valine, leucine, Isoleucine, Methionine, Proline.
2. **Aromatic amino acids:** Phenylalanine, tyrosine, and tryptophan, with their aromatic side chains, are relatively nonpolar (hydrophobic). All can participate in hydrophobic interactions.
3. **Polar, Uncharged amino acids:** The R groups of these amino acids are more soluble in water, or more hydrophilic, than those of the nonpolar amino acids, because they contain functional groups that form hydrogen bonds with water. This class of amino acids includes serine, threonine, cysteine, asparagine, and glutamine.
4. **Acidic amino acids:** Amino acids in which R-group is acidic or negatively charged. Glutamic acid and Aspartic acid
5. **Basic amino acids:** Amino acids in which R-group is basic or positively charged. Lysine, Arginine, Histidine

Classification of amino acids on the basis of nutrition

| Essential | Conditionally Non-Essential | Non-Essential |
|---------------|-----------------------------|---------------|
| Histidine | Arginine | Alanine |
| Isoleucine | Cystine | Asparagine |
| Leucine | Glutamine | Aspartate |
| Lysine | Glycine | Glutamate |
| Methionine | Proline | Serine |
| Phenylalanine | Tyrosine | |
| Threonine | | |
| Tryptophan | | |
| Valine | | |

Classification of amino acids on the basis of nutrition

Essential amino acids (Nine)

Nine amino acids cannot be synthesized in the body and, therefore, must be present in the diet in order for protein synthesis to occur.

These essential amino acids are histidine, isoleucine, leucine, lysine, methionine, phenylalanine, threonine, tryptophan, and valine.

Non-essential amino acids (Eleven)

These amino acids can be synthesized in the body itself and hence do not necessarily need to be acquired through diet.

These non-essential amino acids are Arginine, glutamine, tyrosine, cysteine, glycine, proline, serine, ornithine, alanine, asparagine, and aspartate.

Classification of amino acids on the basis of the metabolic fate

| Glucogenic amino acids | Glucogenic and ketogenic | Ketogenic amino acids |
|--|---|-----------------------|
| Alanine, Arginine, Asparagine, Aspartate Asparagine, Cysteine, Methionine Glutamate, Glutamine, Glycine, Histidine Proline, Serine, Threonine, Valine | Tyrosine Isoleucine Phenylalanine Tryptophan | Leucine Lysine |

Classification of amino acids on the basis of the metabolic fate

1. **Glucogenic amino acids:** These amino acids serve as precursors of gluconeogenesis for glucose formation. Glycine, alanine, serine, aspartic acid, asparagine, glutamic acid, glutamine, proline, valine, methionine, cysteine, histidine, and arginine.
2. **Ketogenic amino acids:** These amino acids break down to form ketone bodies. Leucine and Lysine.
3. **Both glucogenic and ketogenic amino acids:** These amino acids break down to form precursors for both ketone bodies and glucose. Isoleucine, Phenylalanine, Tryptophan, and tyrosine.

Functions of Amino acids

1. In particular, 20 very important amino acids are crucial for life as they contain peptides and proteins and are known to be the building blocks for all living things.
2. The linear sequence of amino acid residues in a polypeptide chain determines the three-dimensional configuration of a protein, and the structure of a protein determines its function.
3. Amino acids are imperative for sustaining the health of the human body. They largely promote the:

Production of hormones

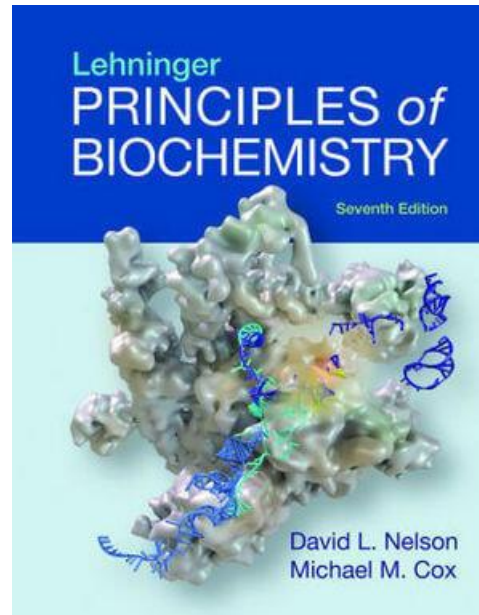
- Structure of muscles
- Human nervous system's healthy functioning
- The health of vital organs
- Normal cellular structure

Functions of Amino acids

4. The amino acids are used by various tissues to synthesize proteins and to produce nitrogen-containing compounds (e.g., purines, heme, creatine, epinephrine), or they are oxidized to produce energy.
5. The breakdown of both dietary and tissue proteins yields nitrogen-containing substrates and carbon skeletons.
6. The nitrogen-containing substrates are used in the biosynthesis of purines, pyrimidines, neurotransmitters, hormones, porphyrins, and nonessential amino acids.
7. The carbon skeletons are used as a fuel source in the citric acid cycle, used for gluconeogenesis, or used in fatty acid synthesis.

Review Questions

- Explain the physical and chemical properties of amino acids.
- What are the most popular classifications of amino acids?



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 2. Structure of proteins



Contents

- Introduction
- Primary Structure
- Secondary Structure
- Tertiary Structure
- Quaternary Structure



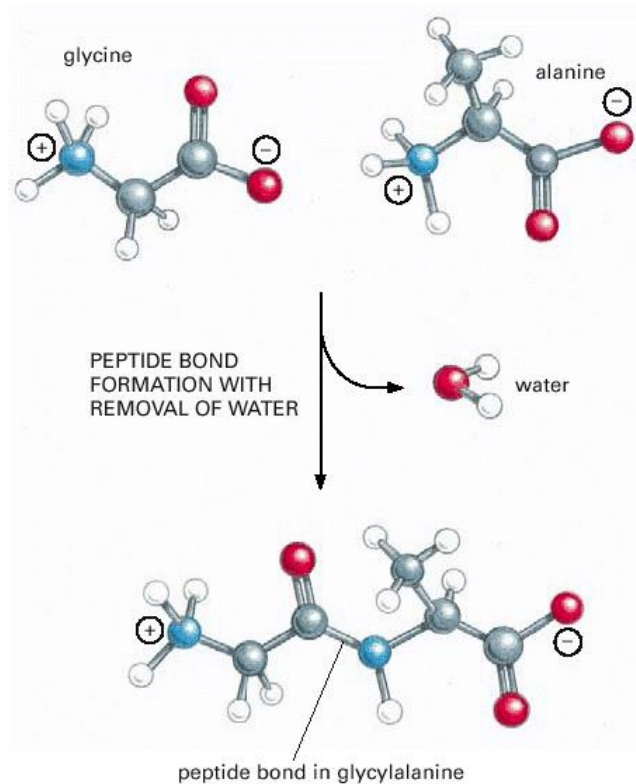
Introduction

From a chemical point of view, proteins are by far the most structurally complex and functionally sophisticated molecules known. This is perhaps not surprising, once one realizes that the structure and chemistry of each protein has been developed and fine-tuned over billions of years of evolutionary history. We start this chapter by considering how the location of each amino acid in the long string of amino acids that forms a protein determines its three-dimensional shape. We will then use this understanding of protein structure at the atomic level to describe how the precise shape of each protein molecule determines its function in a cell.

The Shape of a Protein Is Specified by Its Amino Acid Sequence

A peptide bond

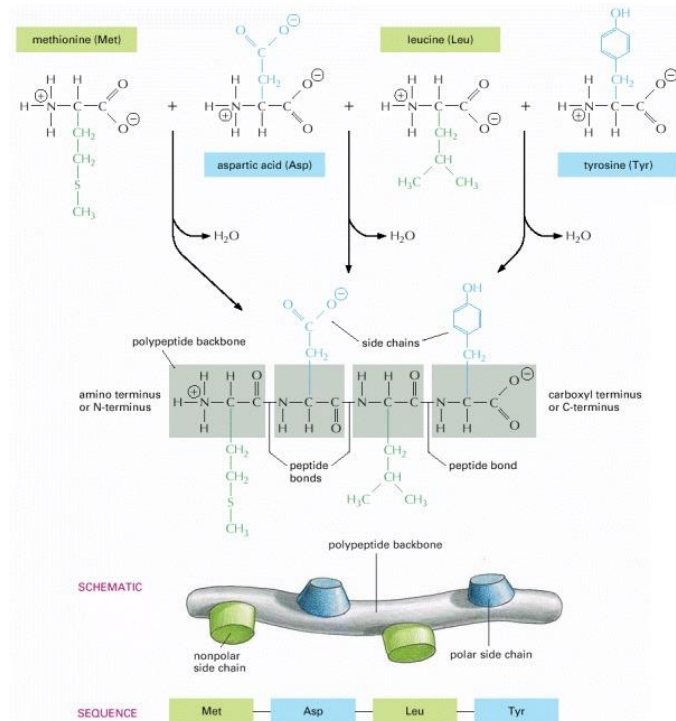
This covalent bond forms when the carbon atom from the carboxyl group of one amino acid shares electrons with the nitrogen atom (blue) from the amino group of a second amino acid. As indicated, a molecule of water is lost in this condensation reaction.



The Shape of a Protein Is Specified by Its Amino Acid Sequence

The structural components of a protein

A protein consists of a polypeptide backbone with attached side chains. Each type of protein differs in its sequence and number of amino acids; therefore, it is the sequence of the chemically different side chains that makes each protein distinct. The two ends of a polypeptide chain are chemically different: the end carrying the free amino group (NH_3^+ , also written NH_2) is the amino terminus, or N-terminus, and that carrying the free carboxyl group (COO^- , also written COOH) is the carboxyl terminus or C-terminus. The amino acid sequence of a protein is always presented in the N-to-C direction, reading from left to right..



The Shape of a Protein Is Specified by Its Amino Acid Sequence

| AMINO ACID | | | SIDE CHAIN |
|---------------|-----|---|-----------------|
| Aspartic acid | Asp | D | negative |
| Glutamic acid | Glu | E | negative |
| Arginine | Arg | R | positive |
| Lysine | Lys | K | positive |
| Histidine | His | H | positive |
| Asparagine | Asn | N | uncharged polar |
| Glutamine | Gln | Q | uncharged polar |
| Serine | Ser | S | uncharged polar |
| Threonine | Thr | T | uncharged polar |
| Tyrosine | Tyr | Y | uncharged polar |

———— POLAR AMINO ACIDS ————

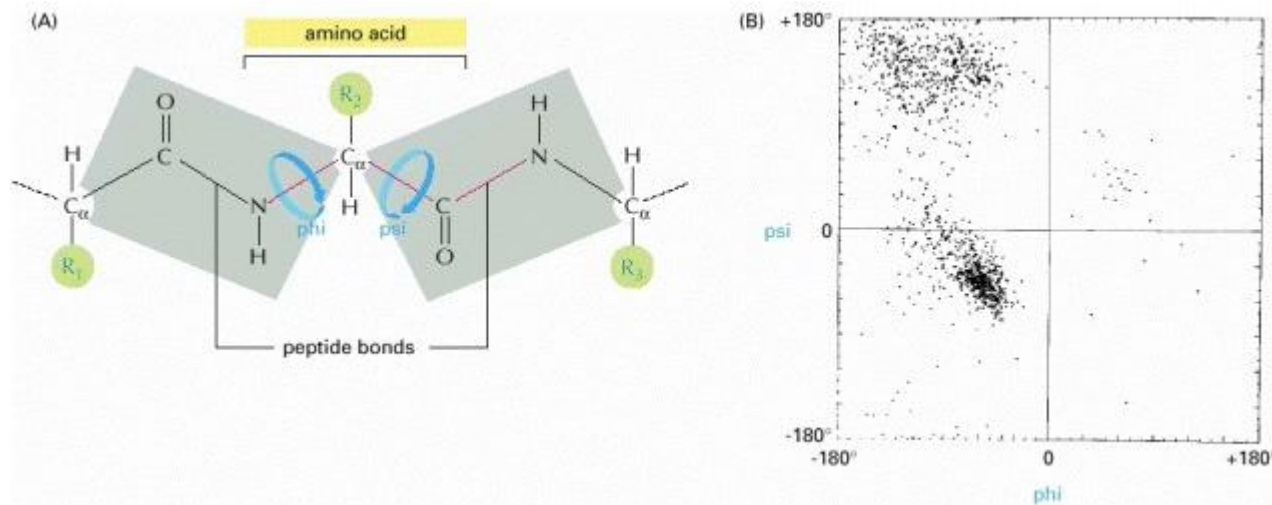
| AMINO ACID | | | SIDE CHAIN |
|---------------|-----|---|------------|
| Alanine | Ala | A | nonpolar |
| Glycine | Gly | G | nonpolar |
| Valine | Val | V | nonpolar |
| Leucine | Leu | L | nonpolar |
| Isoleucine | Ile | I | nonpolar |
| Proline | Pro | P | nonpolar |
| Phenylalanine | Phe | F | nonpolar |
| Methionine | Met | M | nonpolar |
| Tryptophan | Trp | W | nonpolar |
| Cysteine | Cys | C | nonpolar |

———— NONPOLAR AMINO ACIDS ————

The 20 amino acids found in proteins

Both three-letter and one-letter abbreviations are listed. As shown, there are equal numbers of polar and nonpolar side chains. For their atomic structures,

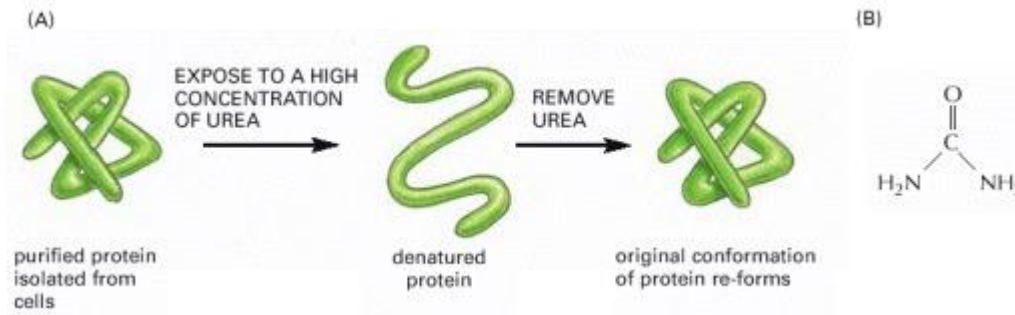
The Shape of a Protein Is Specified by Its Amino Acid Sequence



Steric limitations on the bond angles in a polypeptide chain

(A) Each amino acid contributes three bonds (red) to the backbone of the chain. The peptide bond is planar (gray shading) and does not permit rotation. By contrast, rotation can occur about the $C_\alpha-C$ bond, whose angle of rotation is called psi (ψ), and about the $N-C_\alpha$ bond, whose angle of rotation is called phi (ϕ). By convention, an R group is often used to denote an amino acid side chain (green circles). (B) The conformation of the main-chain atoms in a protein is determined by one pair of ϕ and ψ angles for each amino acid; because of steric collisions between atoms within each amino acid, most pairs of ϕ and ψ angles do not occur. In this so-called Ramachandran plot, each dot represents an observed pair of angles in a protein.

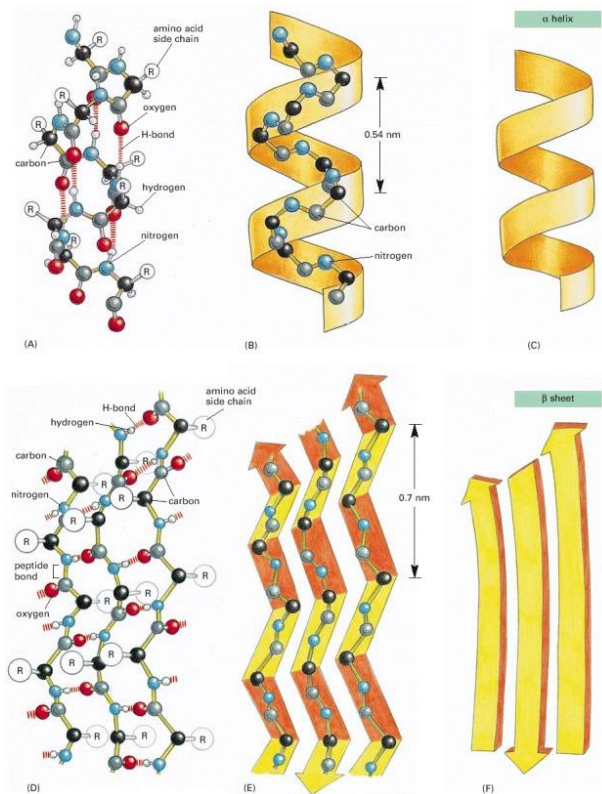
Proteins Fold into a Conformation of Lowest Energy



The refolding of a denatured protein

(A) This experiment demonstrates that the conformation of a protein is determined solely by its amino acid sequence. (B) The structure of urea. Urea is very soluble in water and unfolds proteins at high concentrations, where there is about one urea molecule for every six water molecules.

The α Helix and the β Sheet Are Common Folding Patterns



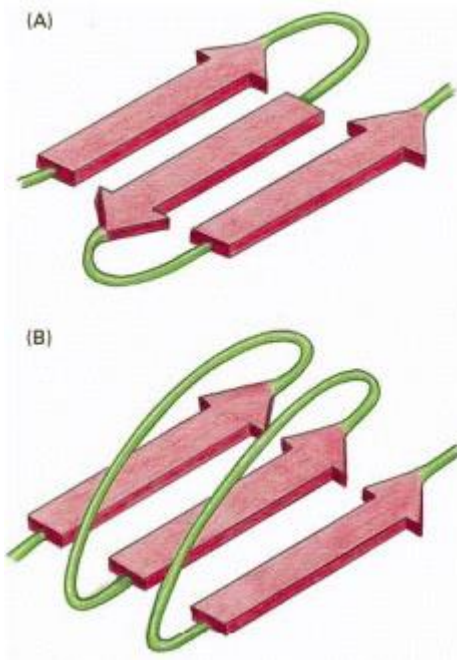
The regular conformation of the polypeptide backbone observed in the α helix and the β sheet

(A, B, and C) The α helix. The N–H of every peptide bond is hydrogen-bonded to the C=O of a neighboring peptide bond located four peptide bonds away in the same chain. (D, E, and F) The β sheet. In this example, adjacent peptide chains run in opposite (antiparallel) directions. The individual polypeptide chains (strands) in a β sheet are held together by hydrogen-bonding between peptide bonds in different strands, and the amino acid side chains in each strand alternately project above and below the plane of the sheet. (A) and (D) show all the atoms in the polypeptide backbone, but the amino acid side chains are truncated and denoted by R. In contrast, (B) and (E) show the backbone atoms only, while (C) and (F) display the shorthand symbols that are used to represent the α helix and the β sheet in ribbon drawings of proteins

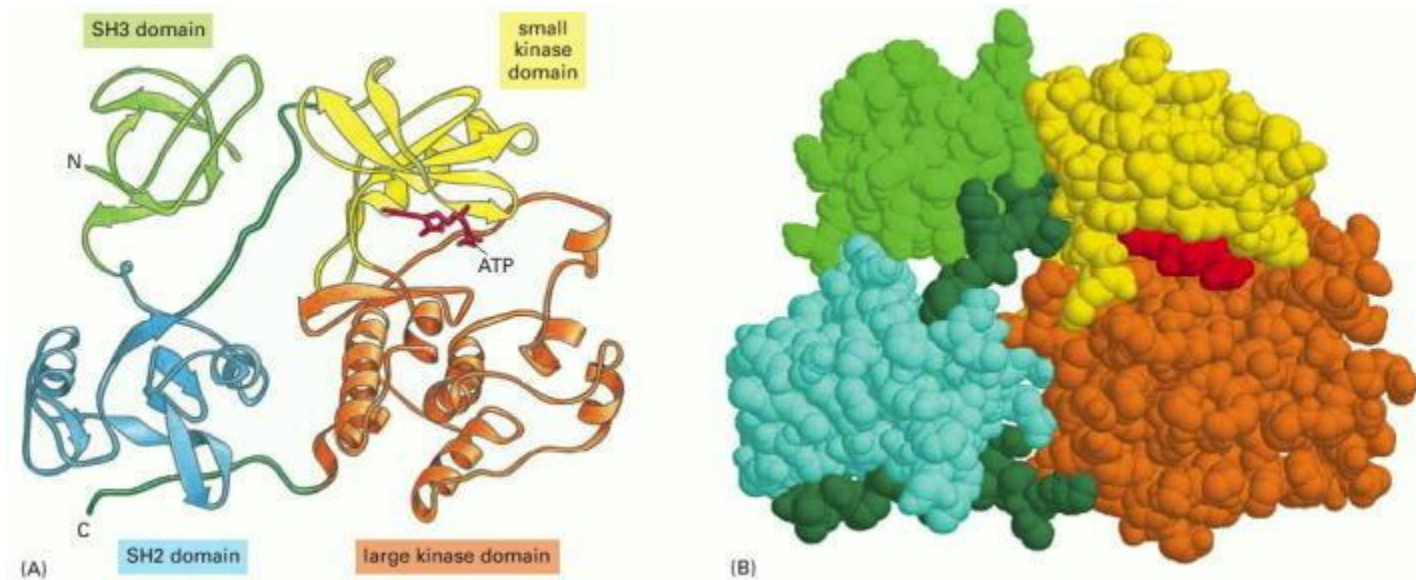
The α Helix and the β Sheet Are Common Folding Patterns

Two types of β sheet structures

(A) An antiparallel β sheet (see Figure 3-9D). (B) A parallel β sheet. Both of these structures are common in proteins.



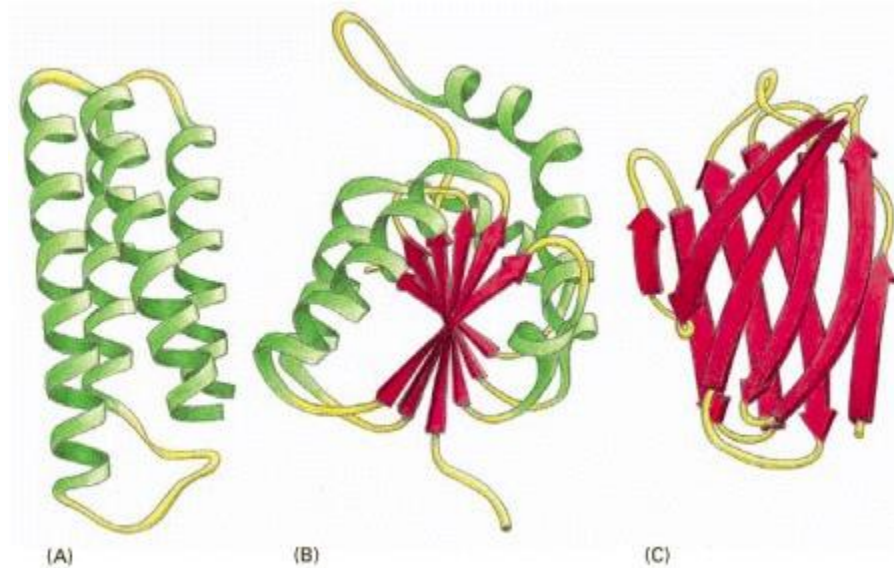
The Protein Domain Is a Fundamental Unit of Organization



A protein formed from four domains

In the Src protein shown, two of the domains form a protein kinase enzyme, while the SH2 and SH3 domains perform regulatory functions. (A) A ribbon model, with ATP substrate in red. (B) A spacing-filling model, with ATP substrate in red. Note that the site that binds ATP is positioned at the interface of the two domains that form the kinase.

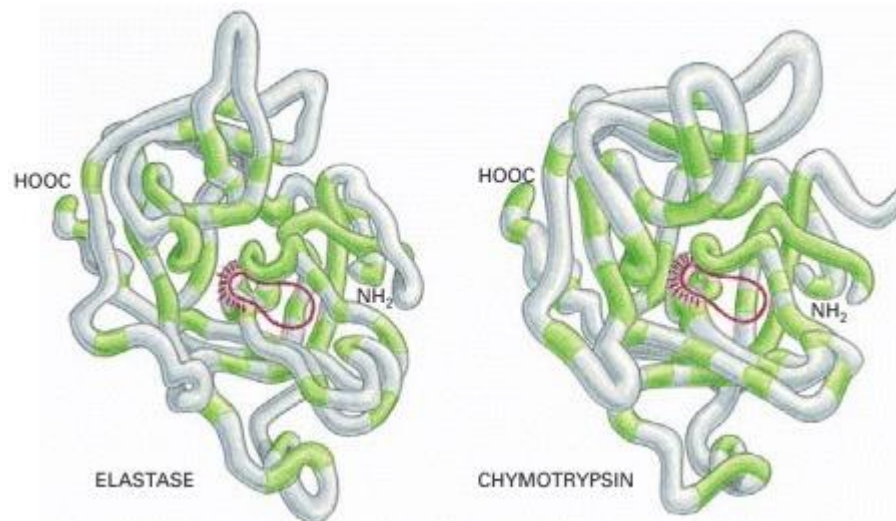
The Protein Domain Is a Fundamental Unit of Organization



Ribbon models of three different protein domains

(A) Cytochrome b562, a single-domain protein involved in electron transport in mitochondria. This protein is composed almost entirely of α helices. (B) The NAD-binding domain of the enzyme lactic dehydrogenase, which is composed of a mixture of α helices and β sheets. (C) The variable domain of an immunoglobulin (antibody) light chain, composed of a sandwich of two β sheets. In these examples, the α helices are shown in green, while strands organized as β sheets are denoted by red arrows.

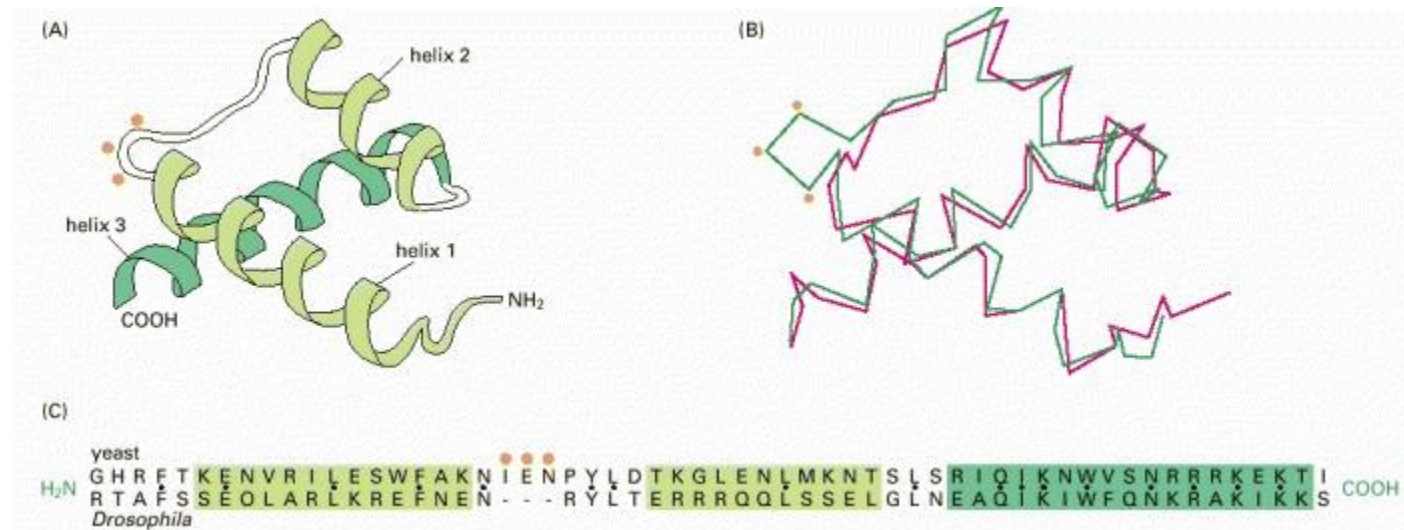
Proteins Can Be Classified into Many Families



The conformations of two serine proteases compared

The backbone conformations of elastase and chymotrypsin. Although only those amino acids in the polypeptide chain shaded in green are the same in the two proteins, the two conformations are very similar nearly everywhere. The active site of each enzyme is circled in red; this is where the peptide bonds of the proteins that serve as substrates are bound and cleaved by hydrolysis. The serine proteases derive their name from the amino acid serine, whose side chain is part of the active site of each enzyme and directly participates in the cleavage reaction.

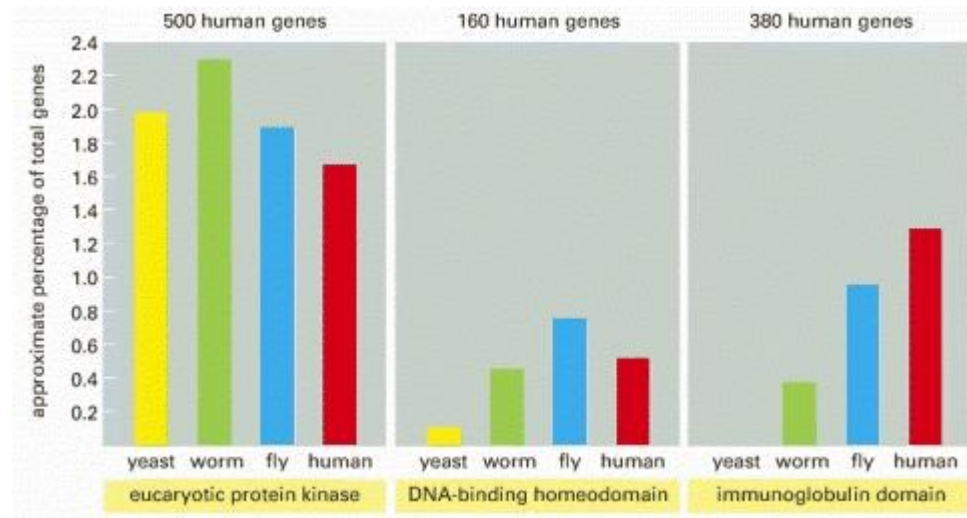
Proteins Can Be Classified into Many Families



A comparison of a class of DNA-binding domains, called homeodomains, in a pair of proteins from two organisms separated by more than a billion years of evolution

(A) A ribbon model of the structure common to both proteins. (B) A trace of the α -carbon positions. The three-dimensional structures shown were determined by x-ray crystallography for the yeast $\alpha 2$ protein (green) and the *Drosophila* engrailed protein (red). (C) A comparison of amino acid sequences for the region of the proteins shown in (A) and (B). Black dots mark sites with identical amino acids. Orange dots indicate the position of a three amino acid insert in the $\alpha 2$ protein.

Proteins Can Be Classified into Many Families



Percentage of total genes containing one or more copies of the indicated protein domain, as derived from complete genome sequences

Note that one of the three domains selected, the immunoglobulin domain, has been a relatively late addition, and its relative abundance has increased in the vertebrate lineage. The estimates of human gene numbers are approximate.

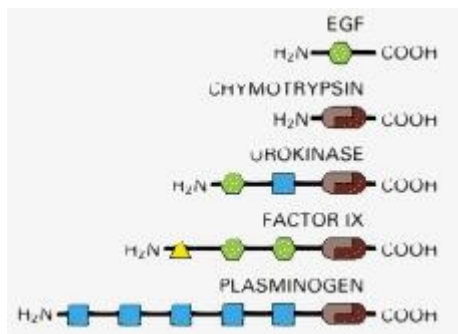
Sequence Homology Searches Can Identify Close Relatives



The use of short signature sequences to find homologous protein domains

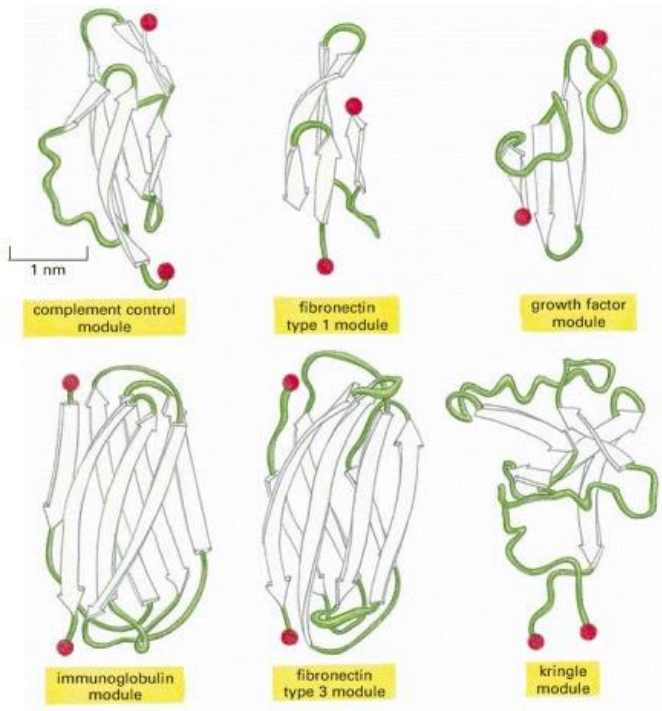
The two short sequences of 15 and 9 amino acids shown (green) can be used to search large databases for a protein domain that is found in many proteins, the SH2 domain. Here, the first 50 amino acids of the SH2 domain of 100 amino acids is compared for the human and Drosophila Src protein. In the computer-generated sequence comparison (yellow row), exact matches between the human and Drosophila proteins are noted by the one-letter abbreviation for the amino acid; the positions with a similar but nonidentical amino acid are denoted by +, and nonmatches are blank. In this diagram, wherever one or both proteins contain an exact match to a position in the green sequences, both aligned sequences are colored red.

Some Protein Domains, Called Modules, Form Parts of Many Different Proteins



An extensive shuffling of blocks of protein sequence (protein domains) has occurred during protein evolution. Those portions of a protein denoted by the same shape and color in this diagram are evolutionarily related. Serine proteases like chymotrypsin are formed from two domains (brown). In the three other proteases shown, which are highly regulated and more specialized, these two protease domains are connected to one or more domains homologous to domains found in epidermal growth factor (EGF; green), to a calcium-binding protein (yellow), or to a “kringle” domain (blue) that contains three internal disulfide bridges.

Some Protein Domains, Called Modules, Form Parts of Many Different Proteins



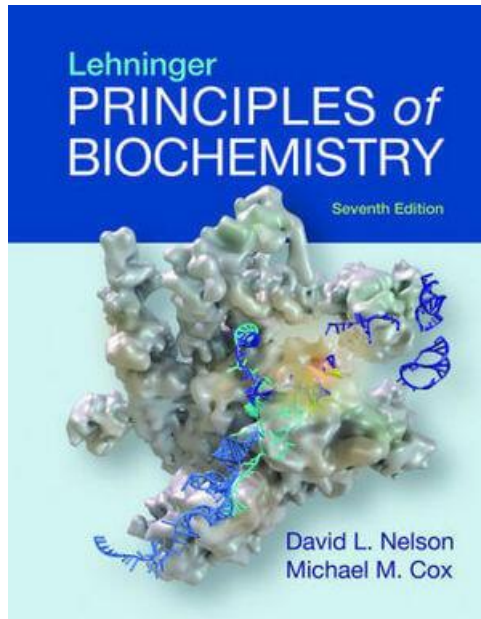
The three-dimensional structures of some protein modules

In these ribbon diagrams, β -sheet strands are shown as arrows, and the N- and C-termini are indicated by red spheres



Review Questions

- Explain structural organization of peptides.
- What are the main characteristics of each level of organization of peptide molecules?



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 3. Chemical bonds in the protein molecule



Contents

- Introduction
- Peptide bonds
- Ionic Bonds
- Disulfide Bonds
- Hydrogen bonds
- Hydrophobic bonds



Introduction

Proteins are the polymers of amino acids. Amino acids are joined together by a special type of covalent bond (peptide bond) to form linear structures called polypeptides. The polypeptides are then folded into specific structures to form the functional conformation of the protein. The folding of proteins into specific shapes and conformations are assisted and stabilized by many types of bonds in them. Some of these bonds are strong bonds whereas others are weak interactions. Important types of bonds involved in protein structure and conformation are Peptide bonds, Ionic bonds, Disulfide bonds, Hydrogen bonds and Hydrophobic Interactions.

Types of bonds

There are 5 types of chemical bonds that play important roles in determining and stabilizing 3-D protein structure.

They are:

- 1) Peptide bonds
- 2) Ionic Bonds
- 3) Disulfide Bonds
- 4) Hydrogen bonds
- 5) Hydrophobic bonds

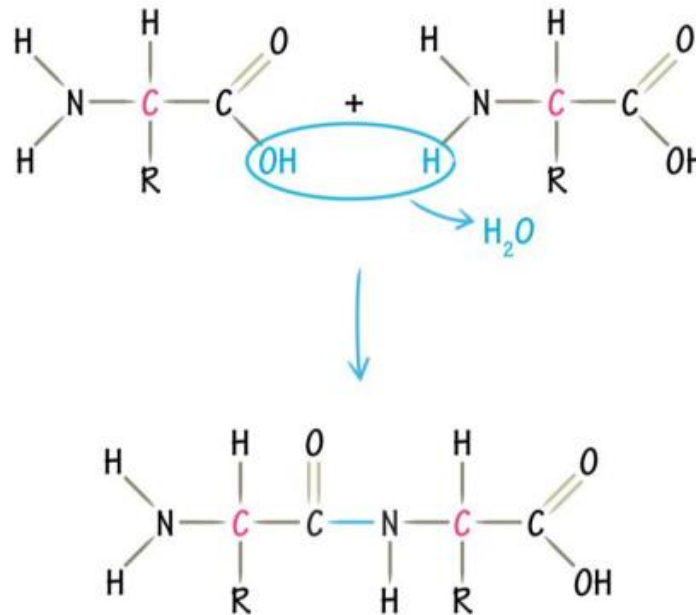
Peptide Bonds

- Peptide bond definition: a covalent bond formed between the carboxylic group of one amino acid and the group of another amino acid
- Peptide bond is a strong covalent bond with high bond dissociation energy.
- It is formed by the joining of two amino acid residues during protein synthesis.
- The carboxylic group ($-\text{COOH}$) of one amino acid combine with the amino group ($-\text{NH}_2$) of another amino acid to form the peptide bond.
- Peptide bond formation is an example for a condensation or elimination reaction.
- One molecule of water is eliminated during the formation of peptide bond by the condensation reaction of two amino acids.
- The resulting compound after the peptide bond formation is called a dipeptide.

Peptide Bonds

- A dipeptide has a free amino group at one end and a carboxylic group at the other end.
- The free amino group or carboxyl group of a dipeptide can form another peptide bond with another peptide bond with a third amino acid and so on.
- Many amino acids join together in this manner to form a polypeptide.
- Peptide bonds formation is facilitated by the enzyme Peptidyl transferase during the translation process of protein synthesis.
- Peptidyl transferase enzyme is a ribozyme; it is a part of the ribosomal RNA (rRNA) of large subunit of Ribosome.
- In prokaryotes the 23S rRNA and in eukaryotes the 28S rRNA acts as the Peptidyl transferase enzyme.

The primary structure of the protein is stabilized by the peptide bond



Ionic Bond

Ionic Bond definition: a chemical bond formed between the two ions of opposite charges.

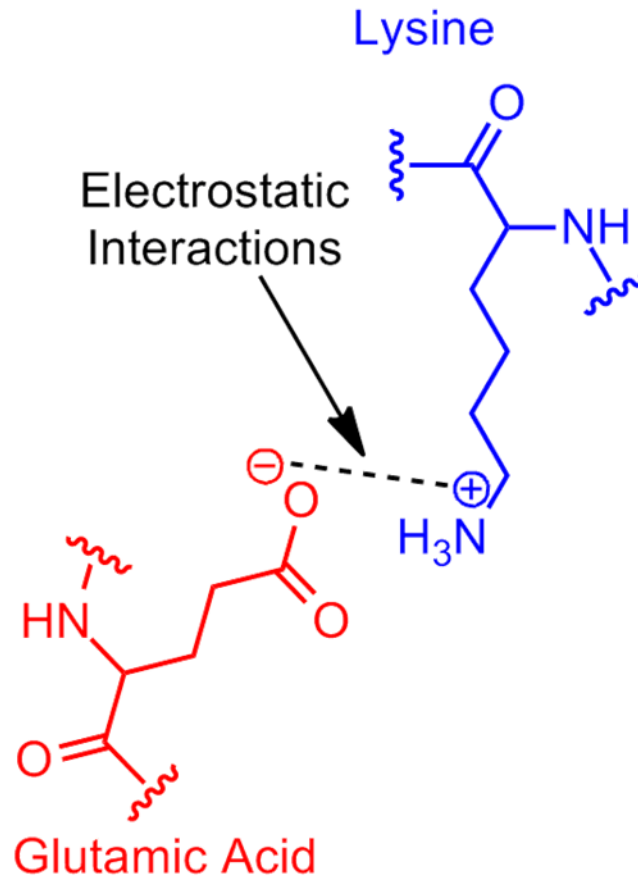
In proteins, the ionic bonds are formed between the ionized acidic or basic groups of amino acids.

The R groups (side chains) of certain amino acids contain additional acidic ($-\text{COO}^-$) or basic ($-\text{NH}_3^+$) groups.

These R groups can ionize to produce charged groups at certain pH.

Acidic R groups will be negatively charged since they release the H^+ ions

Basic R groups will be positively charged since they accept H^+ ions from the medium



Basic R groups will be positively charged since they accept H^+ ions from the medium

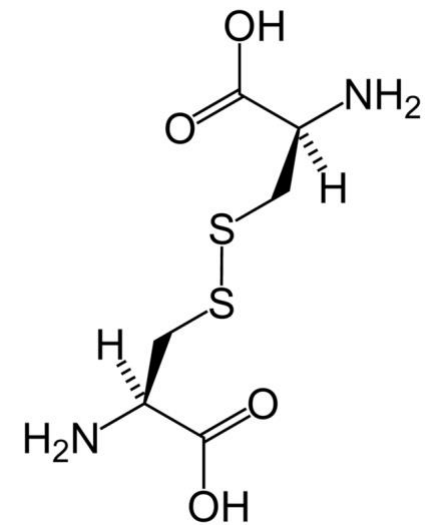
- After the ionization of side chain as mentioned above, the amino acids in the protein chain can attract/repel each other based on their charges. {This is also called “Electrostatic Interactions”}
- The attraction of oppositely charged R groups results in the formation of ionic bonds.
- Ionic bonds are weak bonds and they are very fragile in an aqueous medium.
- Even a change in pH may break down the ionic bonds.
- This is the reason for the denaturation of proteins in acidic or basic medium.
- Tertiary and Quaternary structures of proteins are stabilized by ionic bonds.

Disulfide bonds

- Disulfide bond definition: A covalent bond formed from two thiol groups of two cysteine residues in a protein.
- The cysteine (Cys or C, a sulfur containing amino acid) contain a highly reactive sulfhydryl group (-SH) in its side chain (R group).
- The sulfhydryl is highly polar and highly reactive.
- If two molecules of a cysteine line up alongside each other, the neighboring sulfhydryl groups can be oxidized.
- This reaction results in the formation of a permanent covalent connection between two cysteine residues called disulfide bond.
- Disulfide bond in protein chemistry is better known as the disulfide bridge or S-S bond.

Disulfide bonds

- They are very strong bonds that are not easy to break.
- A disulfide bond may be formed between the cysteine residues of same polypeptide chain or different polypeptide chain of a functional protein.
- Disulfide bonds stabilize the tertiary structures of the protein.



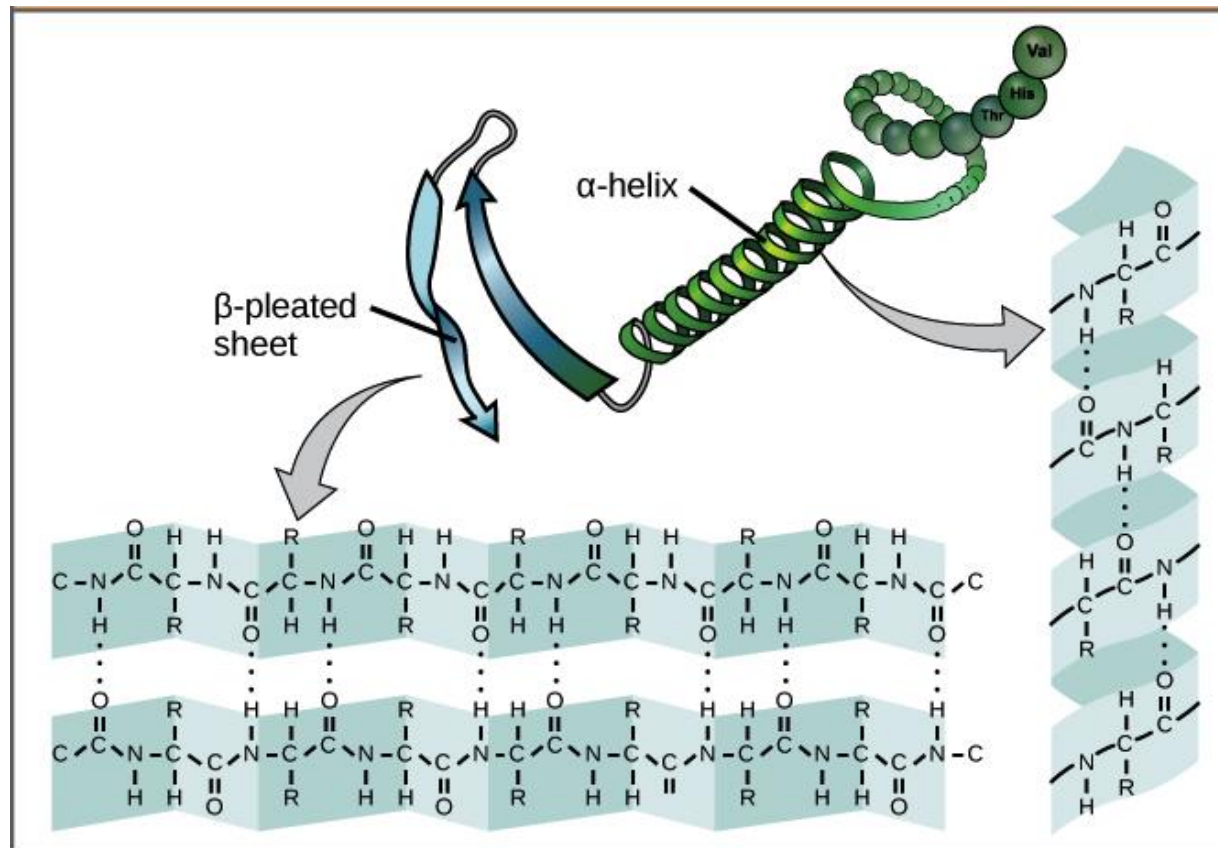
Hydrogen bonds

- Hydrogen bond definition: Hydrogen bond is an electrostatic attraction between a hydrogen atom, which is covalently bound to a high electronegative atom (such as Oxygen and Nitrogen), to another electronegative atom of same or different molecules of their close neighborhood.
- Hydrogen present in the-OH group of -NH_2 of amino acids become slightly electropositive.
- This is due to the high electronegativity of O and N when compared to hydrogen.
- Due to the high electronegativity, Oxygen and Nitrogen attract the shared electron of hydrogen more towards them.
- Thus hydrogen attached to these high electronegative atoms will get a partial positive charge called δ positive whereas the electronegative atoms will get a partial negative charge called δ negative.

Hydrogen bonds

- Consequently, the slightly positive H is then attracted towards the neighboring electronegative oxygen of --C=O or nitrogen atom --NH_2 group.
- These --C=O and NH_2 groups occur along the length of the polypeptide chain in regular sequence.
- Thus the formation of hydrogen bonds gives a regular shape to the polypeptide chain such as alpha helix and beta plates.
- Hydrogen bonds are very weak bonds.
- Occurrence of hydrogen bonds in high frequency makes a considerable contribution towards the molecular stability of proteins.
- Hydrogen bonds are involved in stabilizing the secondary, tertiary and quarternary structure of proteins.

Hydrogen bonds



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Hydrophobic Interactions

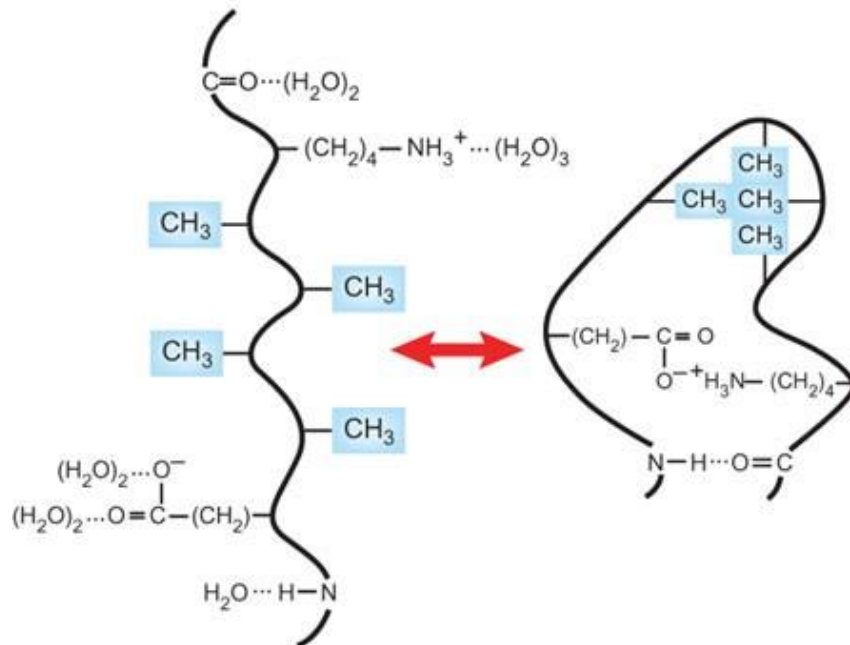
- Some R groups in the amino acids are non-polar.
- Example: Alanine, valine, isoleucine, leucine and methionine.
- The non-polar R groups are hydrophobic and they try stay away from water.
- In a long polypeptide chain there may be many such non-polar amino acids which may be adjacent to each other or separated by polar R groups.
- In an aqueous environment (inside the cell) the linear polypeptide will fold into such a shape that the hydrophobic amino acids come in close contact with each other and they try to exclude the water due to its hydrophobicity.
- By this method, the peptide chain of a globular protein will fold into a spherical shape in the aqueous environment.

Hydrophobic Interactions

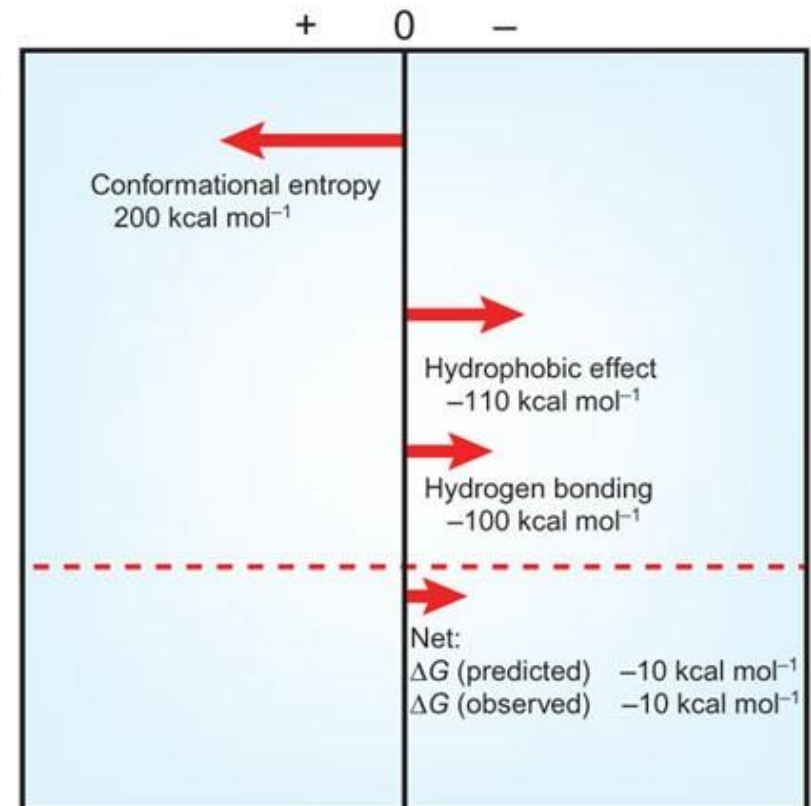
- In a folded protein the hydrophobic groups tend to orient towards the inner side of the protein.
- The hydrophilic residues will form a shell over the hydrophobic residues.
- The hydrophilic shell makes the protein soluble in the aqueous environment.
- Similarly, in the unit membrane, the orientations of membrane proteins are also affected by the hydrophilic and hydrophobic interactions.
- The hydrophobic domain of membrane protein orient towards the exterior of the membrane whereas the hydrophilic domain will orient towards the interior (to the lipid portion).

Hydrophobic Interactions

a

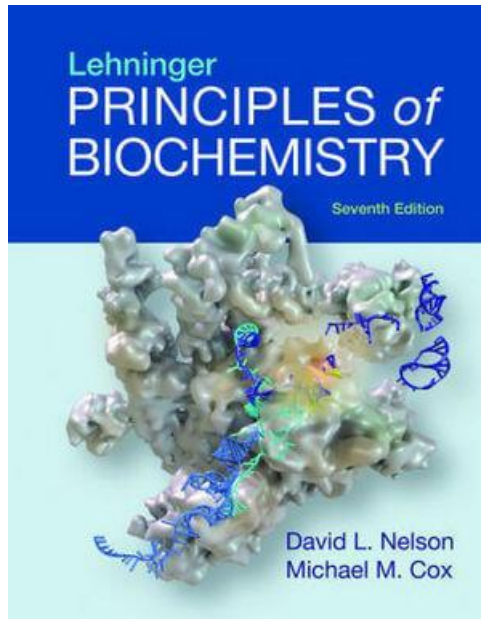


b



Review Questions

- Explain the role of different interactions of structure formation of protein molecule.
- Which type is more important for the formation of tertiary structure?



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 4. Protein functions



Contents

- Introduction
- Structure
- Enzymes
- Hormones
- Transport
- Defense

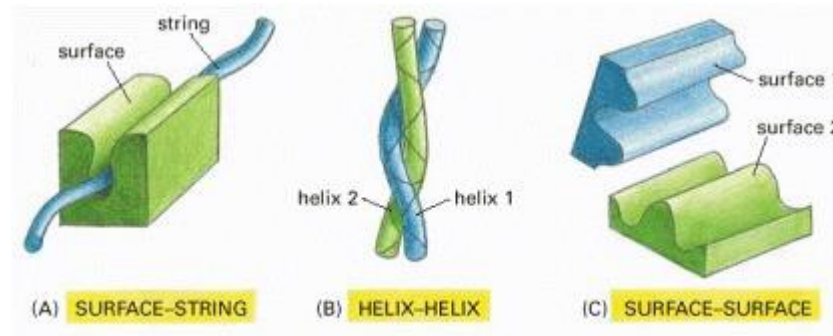


Introduction

Proteins are the “workhorses” of the body and participate in many bodily functions. As we’ve already discussed, proteins come in all sizes and shapes, and each is specifically structured for its particular function. This page describes some of the important functions of proteins. As you read through them, keep in mind that synthesis of all of these different proteins requires adequate amounts of amino acids. As you can imagine, consuming a diet that is deficient in protein and essential amino acids can impair many of the body’s functions.

| Protein Types and Functions | | |
|------------------------------------|---|--|
| Type | Examples | Functions |
| Structure | Actin, myosin, collagen, elastin, keratin | Give tissues (bone, tendons, ligaments, cartilage, skin, muscles) strength and structure |
| Enzymes | Amylase, lipase, pepsin, lactase | Digest macronutrients into smaller monomers that can be absorbed; performs steps in metabolic pathways to allow for nutrient utilization |
| Hormones | Insulin, glucagon, thyroxine | Chemical messengers that travel in blood and coordinate processes around the body |
| Fluid and acid-base balance | Albumin, hemoglobin | Maintains appropriate balance of fluids and pH in different body compartments |
| Transport | Hemoglobin, albumin, protein channels, carrier proteins | Carry substances around the body in the blood or lymph; help molecules cross cell membranes |
| Defense | Collagen, lysozyme, antibodies | Protect the body from foreign pathogens |

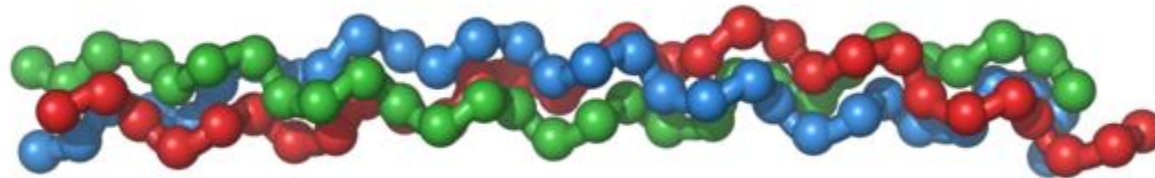
Proteins Bind to Other Proteins Through Several Types of Interfaces



Three ways in which two proteins can bind to each other

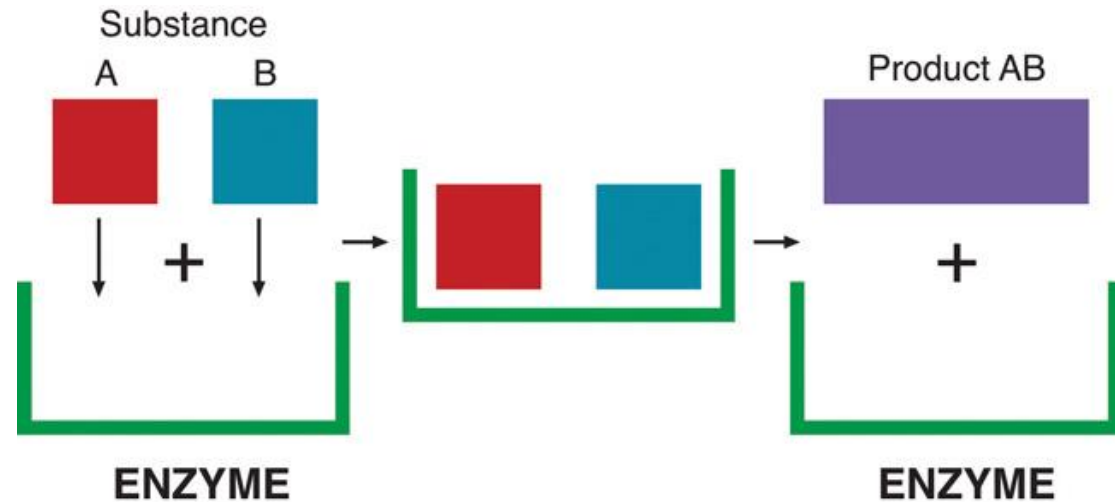
Only the interacting parts of the two proteins are shown. (A) A rigid surface on one protein can bind to an extended loop of polypeptide chain (a “string”) on a second protein. (B) Two α helices can bind together to form a coiled-coil. (C) Two complementary rigid surfaces often link two proteins together.

STRUCTURE



More than one hundred different structural proteins have been discovered in the human body, but the most abundant by far is collagen, which makes up about 6 percent of total body weight. Collagen makes up 30 percent of bone tissue and comprises large amounts of tendons, ligaments, cartilage, skin, and muscle. Collagen is a strong, fibrous protein made up of mostly glycine and proline amino acids. Within its quaternary structure, three protein strands twist around each other like a rope and then these collagen ropes overlap with others.

Enzymes



Some proteins function as enzymes. Enzymes are proteins that conduct specific chemical reactions. An enzyme's job is to provide a site for a chemical reaction and to lower the amount of energy and time it takes for that chemical reaction to happen. This is why enzymes are sometimes called catalysts. On average, more than one hundred chemical reactions occur in cells every single second and most of them require enzymes. The liver alone contains over one thousand enzyme systems. Enzymes are specific and will use only particular substrates that fit into their active site, similar to the way a lock can be opened only with a specific key. Nearly every chemical reaction requires a specific enzyme. Fortunately, an enzyme can fulfill its role as a catalyst over and over again, although eventually it is destroyed and rebuilt. All bodily functions, including the breakdown of nutrients in the stomach and small intestine, the transformation of nutrients into molecules a cell can use, and building all macromolecules, including protein itself, involve enzymes.

Hormones


Proteins are responsible for hormone synthesis. Hormones are chemical messengers produced in one part of the body and then transported in the blood to a different part of the body. When the hormone gets to the target tissue/part of the body, it communicates a message to initiate a specific reaction or cellular process. For instance, after you eat a meal, your blood glucose levels rise. In response to the increased blood glucose, the pancreas releases the hormone insulin. Insulin tells the cells of the body that glucose is available and to take it up from the blood and store it or use it for making energy or building macromolecules. A major function of hormones is to turn enzymes on and off, so some proteins can even regulate the actions of other proteins. While not all hormones are made from proteins, many of them are. Other examples of hormones made from proteins include glucagon, melatonin, and thyroid hormone.

Hormones of the thyroid gland

Thyroglobulin, the active groups of which are two molecules of the iodine-containing compound thyroxine, has a molecular weight of 670,000. Thyroglobulin also contains thyroxine with two and three iodine atoms instead of four and tyrosine with one and two iodine atoms. Injection of the hormone causes an increase in metabolism; lack of it results in a slowdown.

Another hormone, calcitonin, which lowers the calcium level of the blood, occurs in the thyroid gland. The amino acid sequences of calcitonin from pig, beef, and salmon differ from human calcitonin in some amino acids. All of them, however, have the half-cystines (C) and the prolinamide (P) in the same position.

C.S.N.L.S.T.C.V.L.S.A.Y.W.K.D.L.N.N.Y.H.R.F.S.G.M.G.F.G.P.E.T.P(CONH₂)

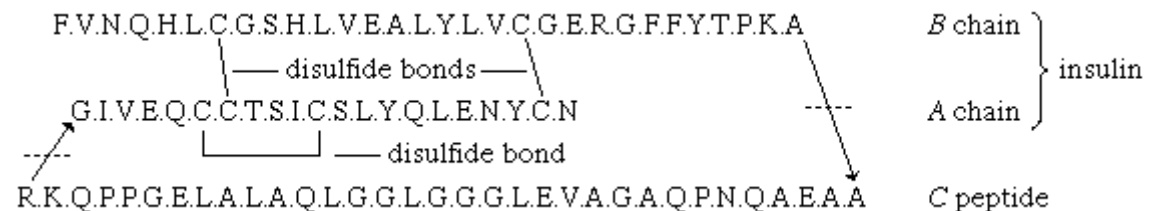
A horizontal line with a vertical tick at each end, representing a disulfide bond between the two cysteine (C) residues in the amino acid sequence.

The amino acid sequence of human calcitonin. At the left end the line represents the disulfide bond. At the right end (CONH₂) indicates that the C terminal proline is present as prolinamide.

Parathyroid hormone (parathormone), produced in small glands that are embedded in or lie behind the thyroid gland, is essential for maintaining the calcium level of the blood. A decrease in its production results in hypocalcemia (a reduction of calcium levels in the bloodstream below the normal range). Bovine parathormone has a molecular weight of 8,500; it contains no cystine or cysteine and is rich in aspartic acid, glutamic acid, or their amides.

Hormones of the pancreas

Although the amino acid structure of insulin has been known since 1949, repeated attempts to synthesize it gave very poor yields because of the failure of the two peptide chains to combine forming the correct disulfide bridge. The ease of the biosynthesis of insulin is explained by the discovery in the pancreas of proinsulin, from which insulin is formed. The single peptide chain of proinsulin loses a peptide consisting of 33 amino acids and called the connecting peptide, or C peptide, during its conversion to insulin. The disulfide bridges of proinsulin connect the A and B chains.





Fluid and Electrolyte Balance

Proper protein intake enables the basic biological processes of the body to maintain the status quo in a changing environment. Fluid balance refers to maintaining the distribution of water in the body. If too much water in the blood suddenly moves into a tissue, the results are swelling and, potentially, cell death. Water always flows from an area of high concentration to one of a low concentration. As a result, water moves toward areas that have higher concentrations of other solutes, such as proteins and glucose. Proteins attract fluid, so to keep the water evenly distributed between blood and cells, proteins continuously circulate at high concentrations in the blood. When protein intake is deficient it can cause edema (swelling). The most abundant protein in the blood is albumin. Albumin's presence in the blood makes the protein concentration in the blood similar to that in cells. Therefore, fluid exchange between the blood and cells is not in the extreme, but rather is minimized to preserve the status quo. Transport proteins (discussed below) in the cell membrane help to maintain the proper balance of electrolytes (like sodium and potassium) inside and outside the cell.

Acid-Base (pH) Balance

Protein is also essential in maintaining proper pH balance (the measure of how acidic or basic a substance is) in the blood. The pH scale ranges from 0 (strongly acidic) to 14 (strongly basic/alkaline). Blood pH is maintained between 7.35 and 7.45, which is slightly basic. If the blood becomes too acidic (a condition known as acidosis) it means that the level of hydrogen (H^+) in the blood is excessive. If the blood becomes too basic/alkaline (a condition known as alkalosis) it means that the level of H^+ in the blood is deficient. Even a slight change in blood pH can affect body functions. Two examples of this include:

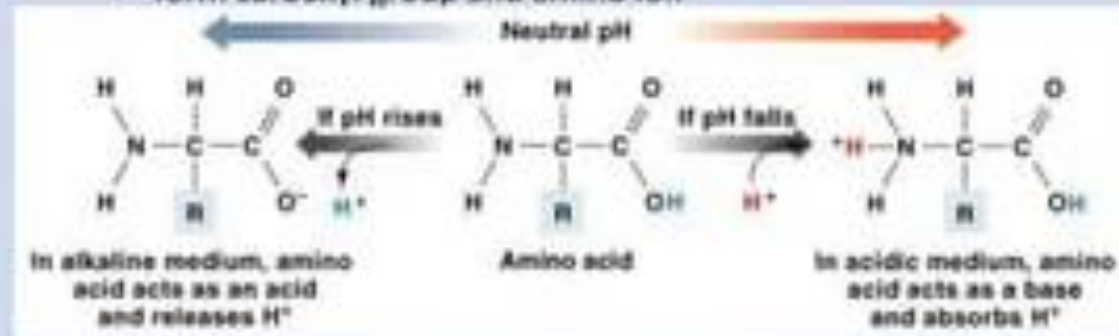
When proteins are exposed to acids or bases the proteins change shape and stop functioning as intended. This process of proteins uncoiling and losing their shape and function is known as denaturation. Denaturation of proteins also occurs with exposure to heat, heavy metals, alcohol, and other damaging substances.

Acidic blood (from ketoacidosis) can lead to coma and/or death in extreme cases.

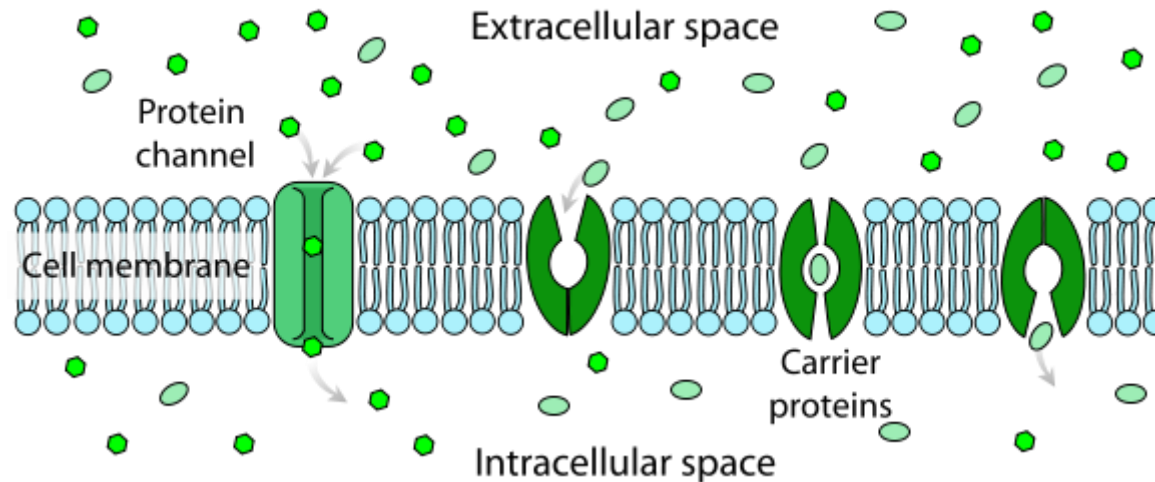
The body has several systems that hold the blood pH within the normal range to prevent issues. Some proteins act as buffers and release hydrogen (H^+) into the blood if it gets too basic. Proteins can also take hydrogen from the blood if it gets too acidic. By releasing and taking hydrogen when needed, proteins maintain acid-base balance and keep blood pH within a normal range.

Acid-Base (pH) Balance

- Amino acids in protein buffer systems
- Depend on free and terminal amino acids
- Respond to pH changes by accepting or releasing H^+
- If pH rises:
 - carboxyl group of amino acid dissociates, acting as weak acid, releasing a hydrogen ion
- If pH drops:
 - carboxylate ion and amino group act as weak bases
 - accept H^+
 - form carboxyl group and amino ion

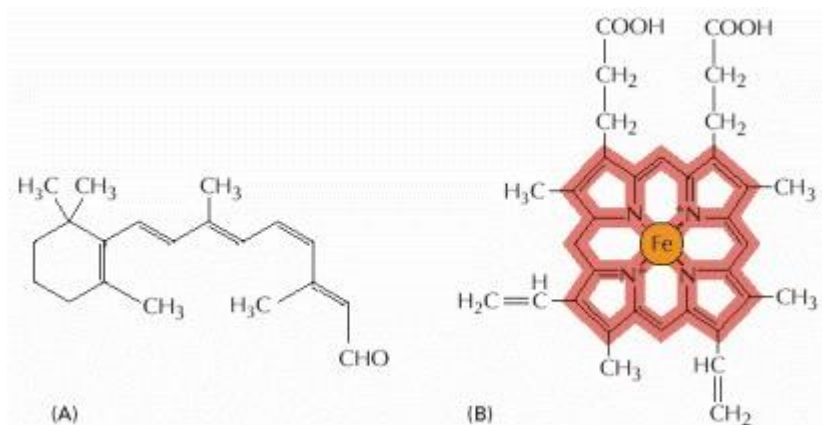


Transport



Proteins also play a role in nutrient transport. A cell's membrane is usually not permeable to large molecules. To get the required nutrients and molecules into the cell, many transport proteins exist in the cell membrane. Some of these proteins act as channels that allow particular molecules to move in and out of cells. Others act as one-way taxis and require energy to function.

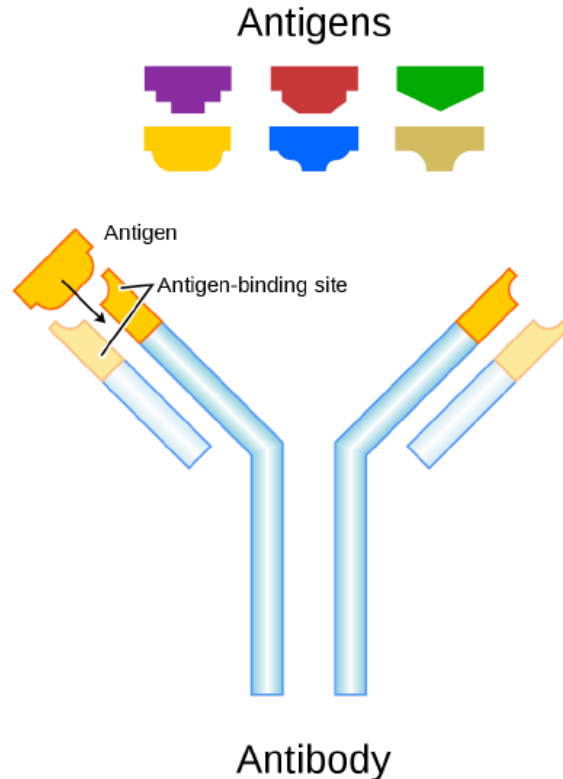
Tightly Bound Small Molecules Add Extra Functions to Proteins



Retinal and heme

(A) The structure of retinal, the light-sensitive molecule attached to rhodopsin in the eye. (B) The structure of a heme group. The carbon-containing heme ring is red and the iron atom at its center is orange. A heme group is tightly bound to each of the four polypeptide chains in hemoglobin, the oxygen-carrying protein

Antibodies



Our immune system is designed to attack and destroy foreign substances. When a foreign substance attacks the body, the immune system produces antibodies to defend against it. Antibodies are special proteins that recognize a unique molecule on harmful bacteria and viruses known as an antigen. Antibodies bind to the antigen and destroy it. Antibodies also trigger other factors in the immune system to seek and destroy unwanted intruders.



Wound Healing, Tissue Regeneration, and Nerve Function

Proteins are involved in all aspects of wound healing, a process that takes place in three phases: inflammation, proliferation, and remodeling. For example, if you get a small cut, your skin will turn red and become inflamed. The healing process begins with proteins, such as bradykinin, which dilate blood vessels at the site of injury. An additional protein called fibrin helps to secure platelets that form a clot to stop the bleeding. Next, in the proliferation phase, cells move in and mend the injured tissue by installing newly made collagen (protein) fibers. The collagen fibers help pull the wound edges together. In the remodeling phase, more collagen is deposited, forming a scar. Scar tissue is only about 80 percent as functional as normal uninjured tissue. If a diet is insufficient in protein, the process of wound healing is markedly slowed.



Wound Healing, Tissue Regeneration, and Nerve Function

While wound healing takes place only after an injury is sustained, a different process called tissue regeneration is ongoing in the body. During tissue regeneration, an exact structural and functional copy of the old tissue is created. Ultimately, the old tissue is replaced with brand new, fully functional tissue. Cells are constantly being broken down, repaired, and replaced. When proteins in the cells are broken down, the amino acids are recycled into new proteins. Some cells (such as skin, hair, nails, and intestinal cells) have a very high rate of regeneration, while others, (such as heart-muscle cells and nerve cells) do not regenerate at any appreciable levels. Tissue regeneration is the creation of new cells (cell division), which requires many different proteins including enzymes, transport proteins, hormones, and collagen. The cells lining the intestine regenerate every three to five days. Protein-inadequate diets impair tissue regeneration, causing many health problems including impairment of nutrient digestion and absorption.

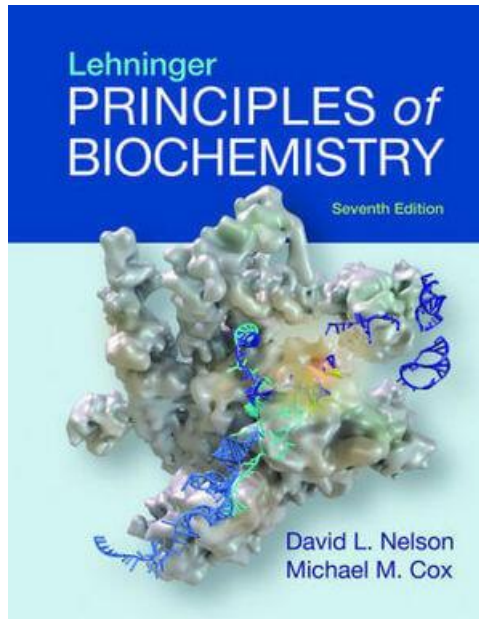
Amino acids can be used to make neurotransmitters (e.g., epinephrine) that transmit messages from one nerve cell to another.

Energy Source

Some of the amino acids in proteins can be disassembled and used to make energy. In healthy people, protein contributes little to energy needs. If a person's diet does not contain enough carbohydrates and fats their body will use amino acids to make energy. When proteins are needed for energy, they are taken from the blood and body tissues (e.g., muscle). To use proteins for energy, deamination is required. Deamination is a process where the amine group is removed from the amino acid and the nitrogen is transported to the kidney for excretion. The remaining components are metabolized for energy. To protect our body tissues from being broken down for energy, it is important to eat an adequate amount of fat and carbohydrate. It's also important to note that our body cannot store excess protein. Excess protein intake results in nitrogen excretion; the remaining components are used for energy or converted to fat for later use.

Review Questions

- Explain the main functions of proteins.
- Give some examples of structural proteins, some hormones.



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 5. Mechanism of enzyme catalysis



Contents

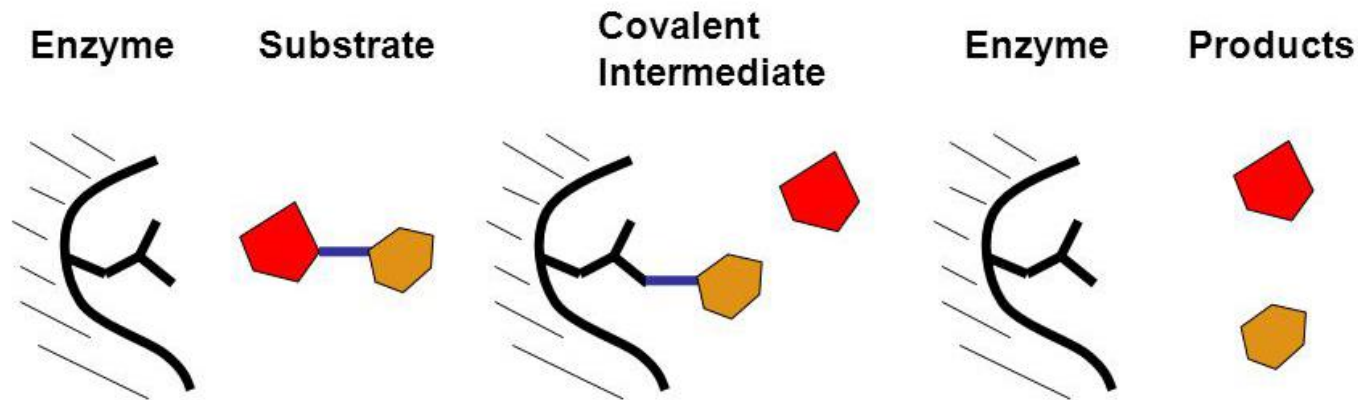
- Introduction
- Definition of an Enzyme
- Covalent Catalysis
- Acid-Base Catalysis
- Electrostatic Catalysis
- Hydrophobic bonds



Introduction

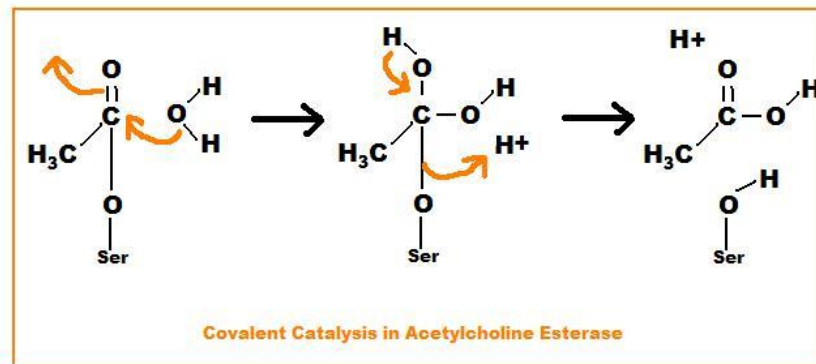
The enzymatic catalysis of reactions is essential to living systems. Under biologically relevant conditions, uncatalyzed reactions tend to be slow—most biological molecules are quite stable in the neutral-pH, mild-temperature, aqueous environment inside cells. Furthermore, many common chemical processes are unfavorable or unlikely in the cellular environment, such as the transient formation of unstable charged intermediates or the collision of two or more molecules in the precise orientation required for reaction.

Covalent Catalysis



Covalent catalysis involves the formation of a covalent bond between the enzyme and at least one of the substrates involved in the reaction. Often times this involves nucleophilic catalysis which is a subclass of covalent catalysis. Several amino acid R-groups can serve as a nucleophile and are often found at the active site of enzymes. Nucleophilic side chains are often activated by deprotonation caused by neighboring side chains, such as histidine that can act as a base. Alternatively, water can also activate the nucleophile. The intermediate covalent bond formation between the enzyme and the substrate enables bond cleavage and the removal of a leaving group

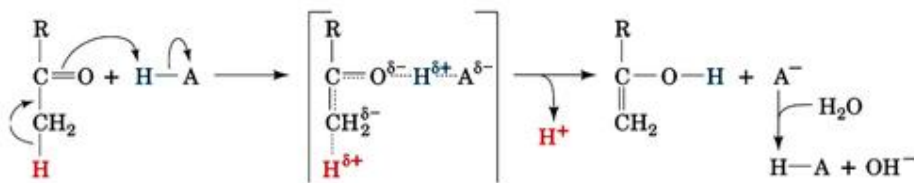
Examples of Enzymes that Participate in Covalent Catalysis



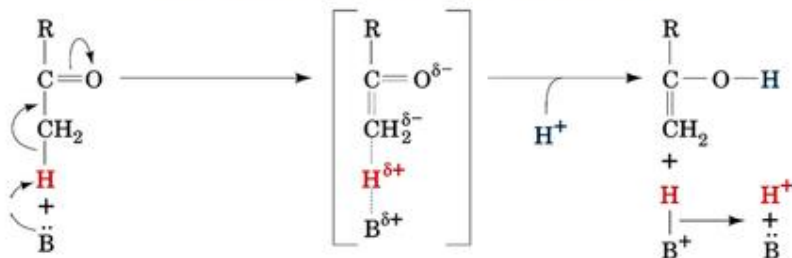
Examples of enzymes that participate in covalent catalysis include the proteolytic enzyme Chymotrypsin and trypsin in which the nucleophile is the hydroxyl group on the serine. Chymotrypsin is a degradative protease of the digestive system. It catalyzes the cleavage of peptide bonds that are adjacent to large aromatic or nonpolar residues. It cleaves the peptide bond on the carboxyl terminus side of the protein. The chymotrypsin has three main catalytic residues termed as the catalytic triad. These are His 57, Asp 102 and Ser 195. Upon deprotonation the serine residue becomes a powerful nucleophile due to its alkoxide that will attack the relatively unreactive carbon of the carbonyl in the protein. Typical residues used in covalent catalysis are Lys, His, Cys, Asp, Glu, and Ser and some other coenzymes.

Acid-Base Catalysis

General Acid Catalysis

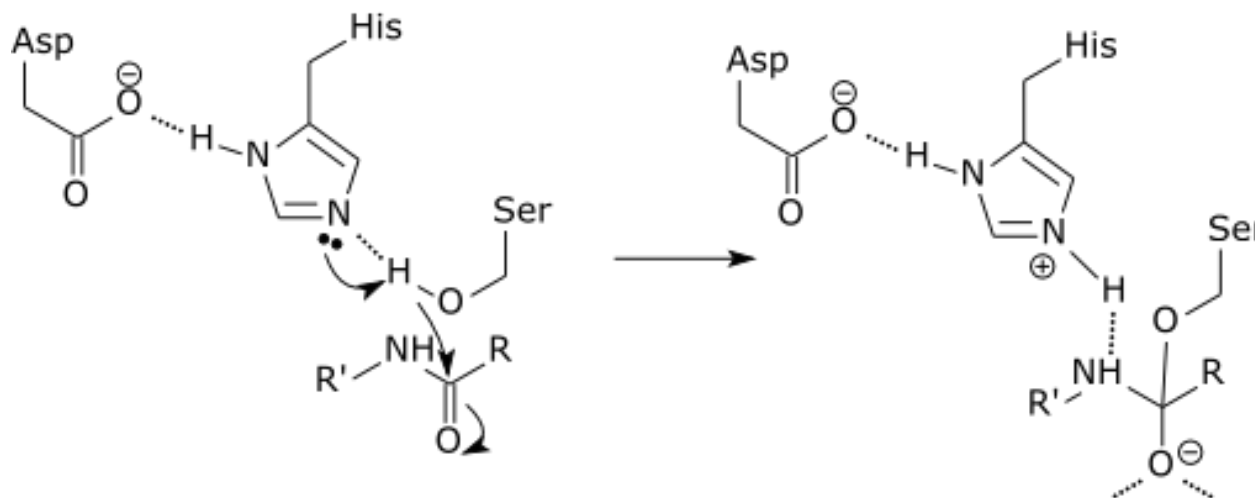


General Base Catalysis



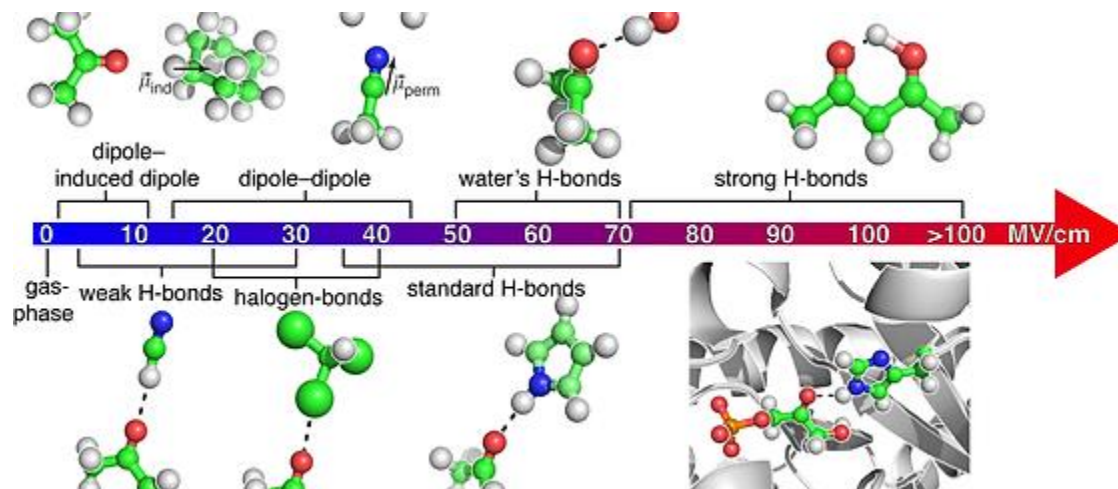
Acid-Base Catalysis is involved in any reaction mechanism that requires the transfer of a proton from one molecule to another. It is very common to see this mechanism combined with Covalent Catalysis as many nucleophiles are activated by the removal of a proton, including alcohol, thiol, and amine functional groups. Enzymes that utilize Acid-Base Catalysis can be subgrouped further into either specific acid-base or general acid-base reactions. Specific acid or specific base catalysis occur if a hydronium ion (H₃O⁺) or a hydroxide ion (OH⁻), respectively, are utilized directly in the reaction mechanism, and the pH of the solution affects the rate of catalysis. General acid and general base reactions occur when molecules other than hydronium ion (H₃O⁺) or a hydroxide ion (OH⁻) are the source of proton donation or acceptance. Most commonly, an active site amino acid residue is used to accept or donate a proton within the reaction mechanism. In general acid-base reactions the pH is usually held constant within a buffered system.

Acid-Base Catalysis



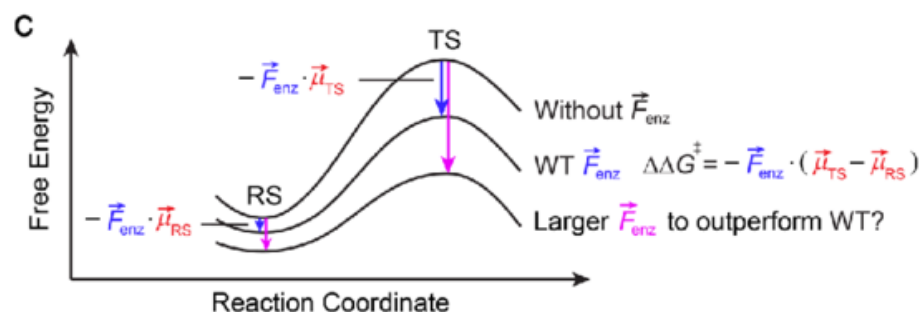
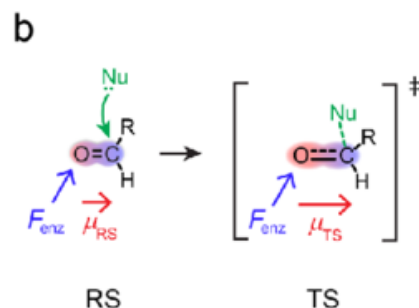
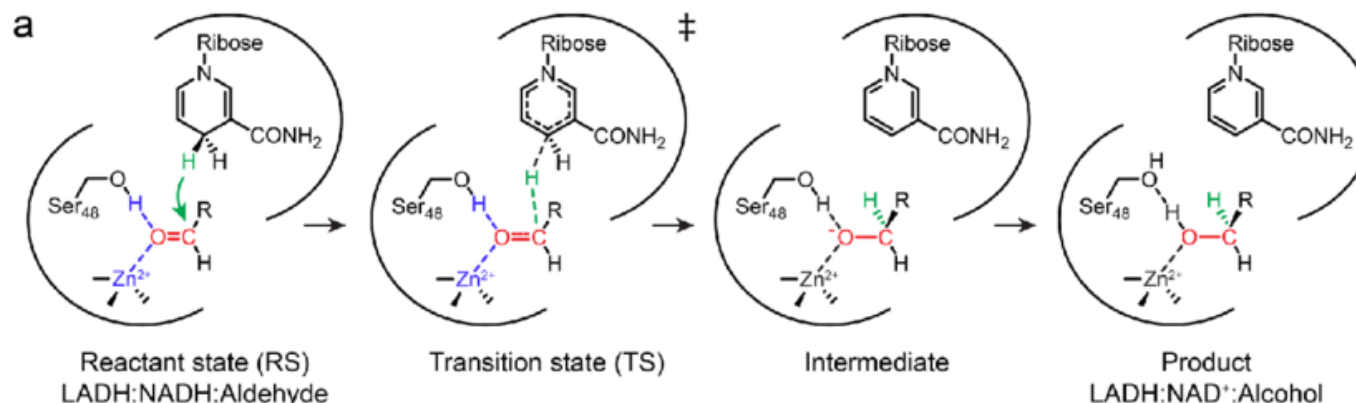
The initial step of the serine protease catalytic mechanism involves the histidine of the active site accepting a proton from the serine residue. This prepares the serine as a nucleophile to attack the amide bond of the substrate. This mechanism includes donation of a proton from serine (a base, pKa 14) to histidine (an acid, pKa 6), made possible due to the local environment of the bases.

Electrostatic Catalysis

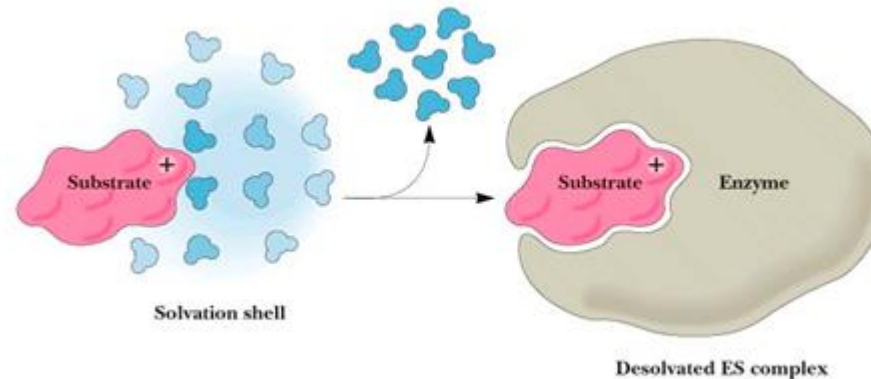


Electrostatic catalysis occurs when the enzyme active site stabilizes the transition state of the reaction by forming electrostatic interactions with the substrate. The electrostatic interactions can be ionic, ionic-dipole, dipole-dipole, or hydrophobic interactions. Hydrogen bonding is one of the most common electrostatic interactions formed in the active site.

Electrostatic Catalysis



Desolvation



Enzyme active sites can become devoid of water and mimic the reaction characteristics of the gas phase. This can destabilize the polarized state of charged groups such as acids and bases. Thus, the neutral form of these types of residues becomes the favored state. This is due to significant alterations in the pKa of the active site residues within the nonpolar environment. This can cause normally acidic residues such as glutamate to abstract a proton from histidine and behave as a base, for example.

Catalysis by Approximation

In catalysis by approximation, the enzyme enhances the reaction rate by binding with multiple substrates and positioning them favorably so that the reaction can proceed. Binding with the enzyme reduces the rotational entropy of the substrates that would otherwise be randomly free floating in solution, and enables the correct positioning of substrates for the reaction. The loss of entropy, which is not favorable, is offset by the binding energy released with the substrate-enzyme interaction.

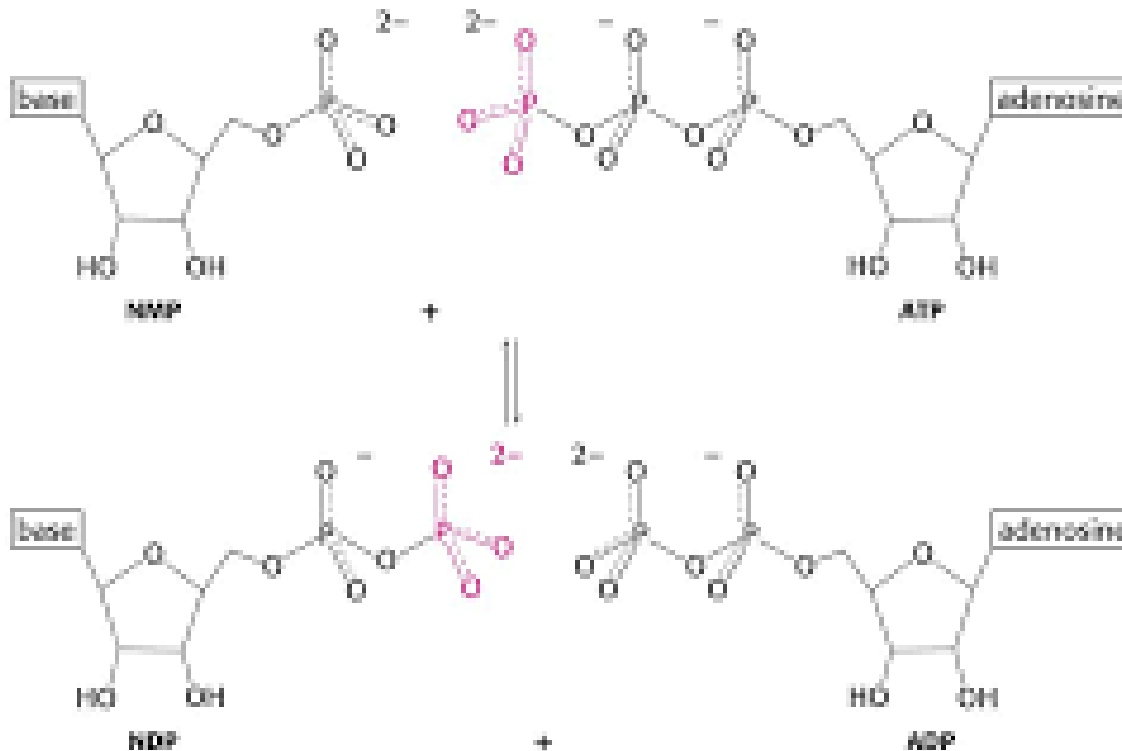
In addition to correctly positioning the substrates to interact with one another, catalysis by approximation converts a reaction that would have been second order, with substrates that are free floating in solution, to a first order reaction, where all of the substrates are held in place by the enzyme and behave as a single molecule. This can dramatically improve the catalytic rate of the reaction from 10^5 to 10^7 times faster, depending on the enzyme system.

Catalysis by Approximation

In catalysis by approximation, the enzyme enhances the reaction rate by binding with multiple substrates and positioning them favorably so that the reaction can proceed. Binding with the enzyme reduces the rotational entropy of the substrates that would otherwise be randomly free floating in solution, and enables the correct positioning of substrates for the reaction. The loss of entropy, which is not favorable, is offset by the binding energy released with the substrate-enzyme interaction.

In addition to correctly positioning the substrates to interact with one another, catalysis by approximation converts a reaction that would have been second order, with substrates that are free floating in solution, to a first order reaction, where all of the substrates are held in place by the enzyme and behave as a single molecule. This can dramatically improve the catalytic rate of the reaction from 10^5 to 10^7 times faster, depending on the enzyme system.

Catalysis by Approximation



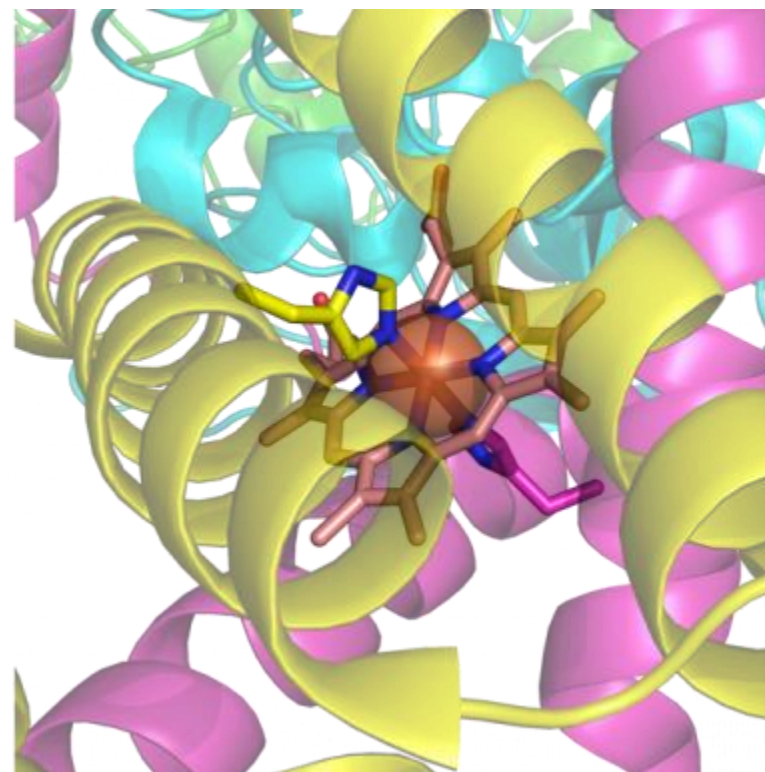
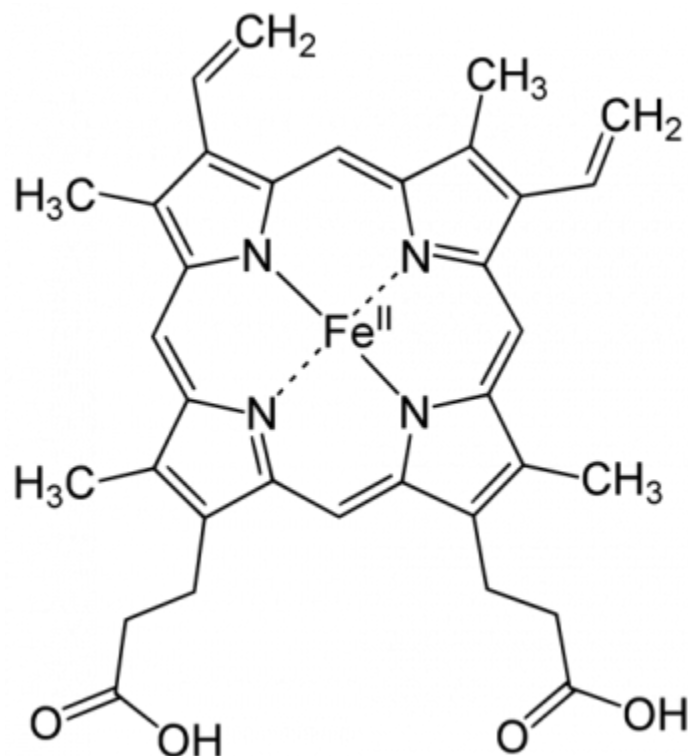
Strain Distortion

In organic chemistry, you learned that certain structures such as three-membered and four-membered ring structures, such as epoxides were highly reactive due to the strain distortion inherent to the unfavored bond angles inherent to the ring. Enzyme active sites can also utilize strain distortion within a bound substrate to increase the reactivity of the molecule and favor the formation of the transition state. Many enzymes that function by the induced fit model also utilize strain distortion within their catalytic mechanism. Within the unbound state they remain in a low catalytic state, however the interaction with the substrate induces the destabilization of the enzyme active site or may induce strain within the substrate causing the initiation of the catalytic activity of the enzyme.

Cofactor Catalysis

Cofactors are molecules that bind to enzymes and are required for the catalytic activity of the enzyme. They can be divided into two major categories: metals and coenzymes. Metal cofactors that are commonly found in human enzymes include: iron, magnesium, manganese, cobalt, copper, zinc, and molybdenum. Coenzymes are small organic molecules that are often derived from vitamins, which are essential organic nutrients consumed within the diet. Coenzymes can bind loosely with the enzyme and have the ability to bind and release from the active site, or they may be tight binding and lack the ability to release easily from the enzyme. Tight binding coenzymes are referred to as prosthetic groups. Enzymes that are not yet associated with a required cofactor are called apoenzymes, whereas enzymes that are bound with their required cofactors are called holoenzymes. Sometimes organic molecules and metals combine to form coenzymes, such as in the case of the heme cofactor. Coordination of heme cofactors with their enzyme counterparts often involves electrostatic interactions with histidine residues as shown in the succinate dehydrogenase enzyme.

Cofactor Catalysis

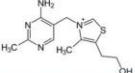
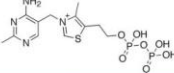
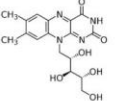
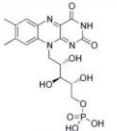
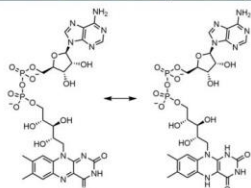

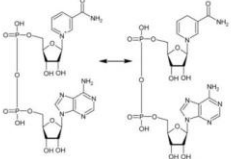

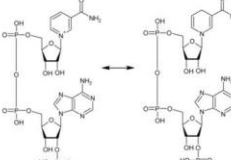


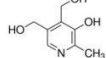
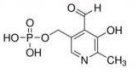
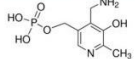
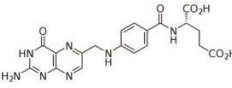
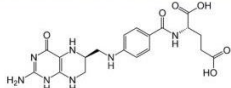
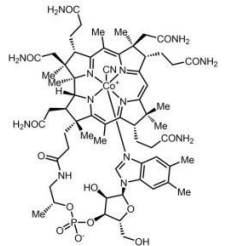
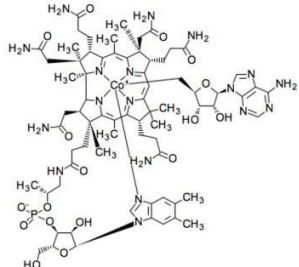
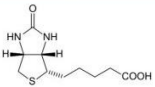
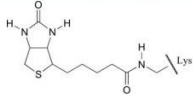
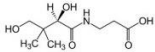
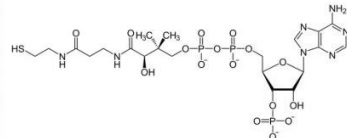
Cofactor Catalysis

As an example of how vitamins can be utilized as cofactors, shows the common B vitamins and the coenzymes derived from their structures. Many vitamin deficiencies cause disease states due to the inactivity of apoenzymes that are unable to function without the correctly bound coenzyme.

Cofactors can help to mediate enzymatic reactions through the use of any of the different catalytic strategies listed above. They can serve as nucleophiles and mediate covalent catalysis or form electrostatic interactions with the substrate and stabilize the transition state. They can also cause strain distortion or facilitate acid-base catalysis. Metal-aided catalysis can often use homolytic reaction mechanisms that involve radical intermediates. This can be important in reactions such as those occurring in the electron transport chain that require the safe movement of single electrons.

Essential B-Vitamins and their Modified Enzyme Cofactors

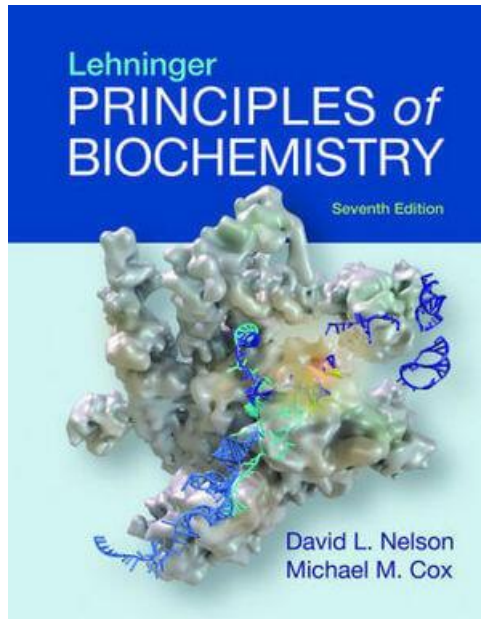
| B Vitamins | Modified Enzyme Cofactors |
|---|--|
| Vitamin B1 (Thymine) | Thymine Diphosphate (TPP) |
|  |  |
| Vitamin B2 (Riboflavin) | Flavin Mononucleotide (FMN) |
|  |  |
| | Flavin Adenine Dinucleotide (FAD \leftrightarrow FADH₂) |
| |  |
| Vitamin B3 (Niacinamide) | Nicotinamide Adenine Dinucleotide (NAD⁺ \leftrightarrow NADH) |
|  |  |
| Vitamin B3 (Niacin - carboxylic acid form) | Nicotinamide Dinucleotide Phosphate (NADP⁺ \leftrightarrow NADPH) |
|  |  |

| | |
|---|---|
| Vitamin B6 (Pyridoxine) | Pyridoxyl 5'-Phosphate |
|  |  |
| | Pyridoxamine 5'-Phosphate |
| |  |
| Vitamin B9 (Folic Acid) | Tetrahydrofolate |
|  |  |
| Vitamin B12 (Cyanocobalamin) | Adenosylcobalamin |
|  |  |
| Biotin | Biotin-Enzyme Complex |
|  |  |
| Pantothenic Acid | Coenzyme A |
|  |  |



Review Questions

- What is the definition of enzyme?
- Explain the mechanism of catalytic reactions.



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 6. Types of enzyme reactions



Contents

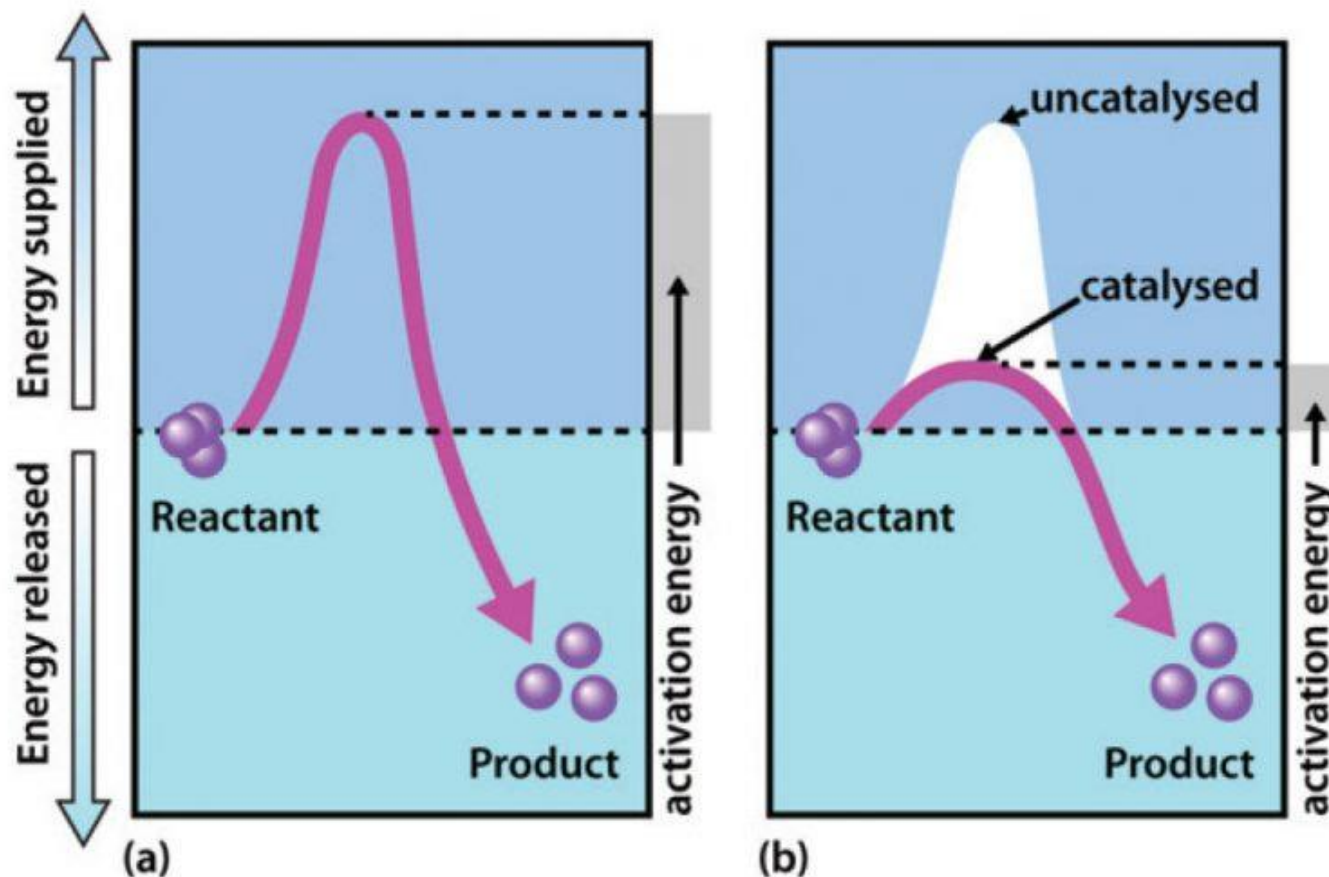
- Introduction
- Oxidation-Reduction Reactions
- Group Transfer Reactions
- Hydrolysis Reactions
- The Formation/Removal of Carbon-Carbon Double Bonds
- Isomerization Reactions
- Ligation Reactions



Introduction

Enzymes are biological catalysts that reduce the activation energy required for a reaction to proceed in the forward direction. They facilitate the formation of the transition state species within the reaction and speed up the rate of the reaction by a million-fold in comparison to non catalyzed reactions. Note that enzymes do NOT alter the ΔG of the reaction and do NOT have any affect on the spontaneity or equilibrium position of the reaction. Enzymes, like other catalysts, are also not used up during the reaction. Enzymes have high substrate specificity, and can even show regiospecificity that leads to the generation of stereospecific products.

Introduction



ERASMUS+

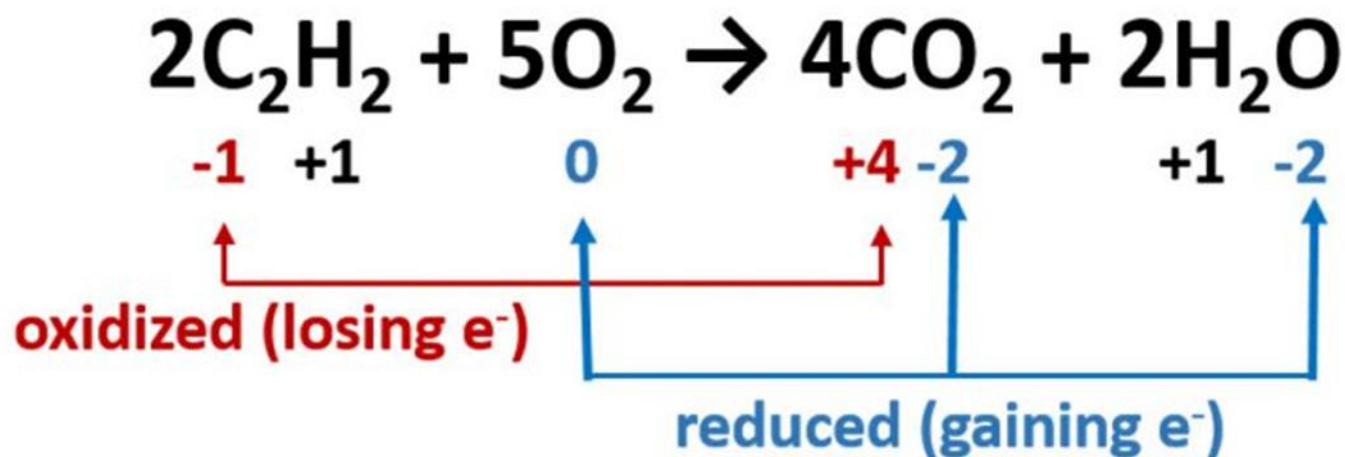
Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Oxidation-Reduction Reactions

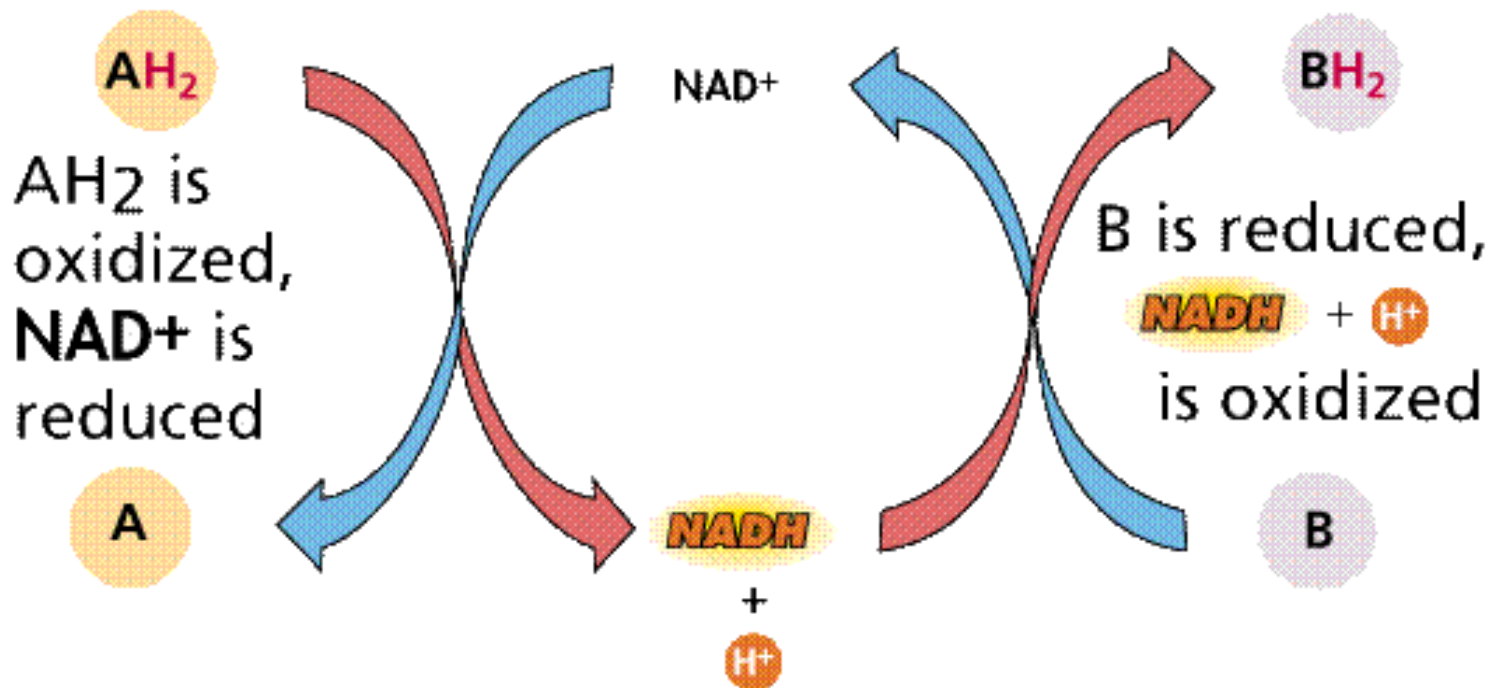
An oxidation-reduction (redox) reaction is a type of chemical reaction that involves a transfer of electrons between two atoms or compounds. The substance that loses the electrons is said to be oxidized, while the substance that gains the electrons is said to be reduced. Redox reactions always have to occur together. If one molecule is oxidized, then another molecule has to be reduced (ie. electrons don't appear out of thin air to be added to a compound, they always have to come from somewhere!).

The change in electron composition can be evaluated in the change of the oxidation state (or number) of an atom. Therefore, an oxidation-reduction reaction is any chemical reaction in which the oxidation state (number) of a molecule, atom, or ion changes by gaining or losing an electron. We will learn how to evaluate the oxidation state of a molecule within this section. Overall, redox reactions are common and vital to some of the basic functions of life, including photosynthesis, respiration, combustion, and corrosion or rusting.

Oxidation-Reduction Reactions



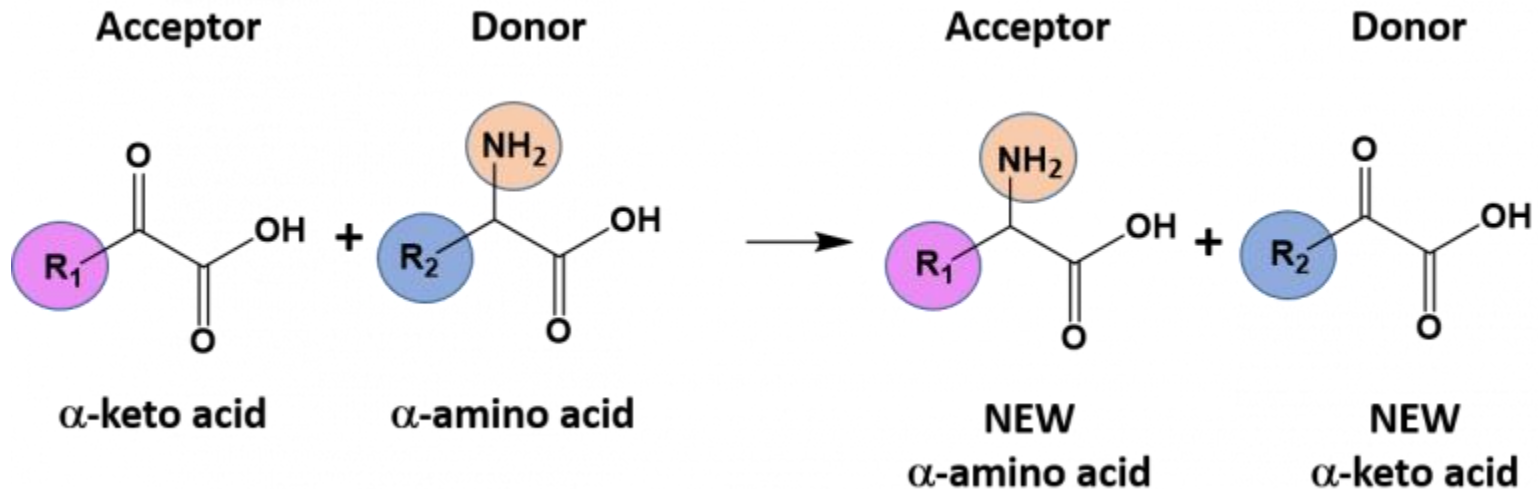
Oxidation-Reduction Reactions



Group Transfer Reactions

In group transfer reactions, a functional group will be transferred from one molecule that serves as the donor molecule to another molecule that will be the acceptor molecule. The transfer of an amine functional group from one molecule to another is common example of this type of reaction.

Group Transfer Reactions

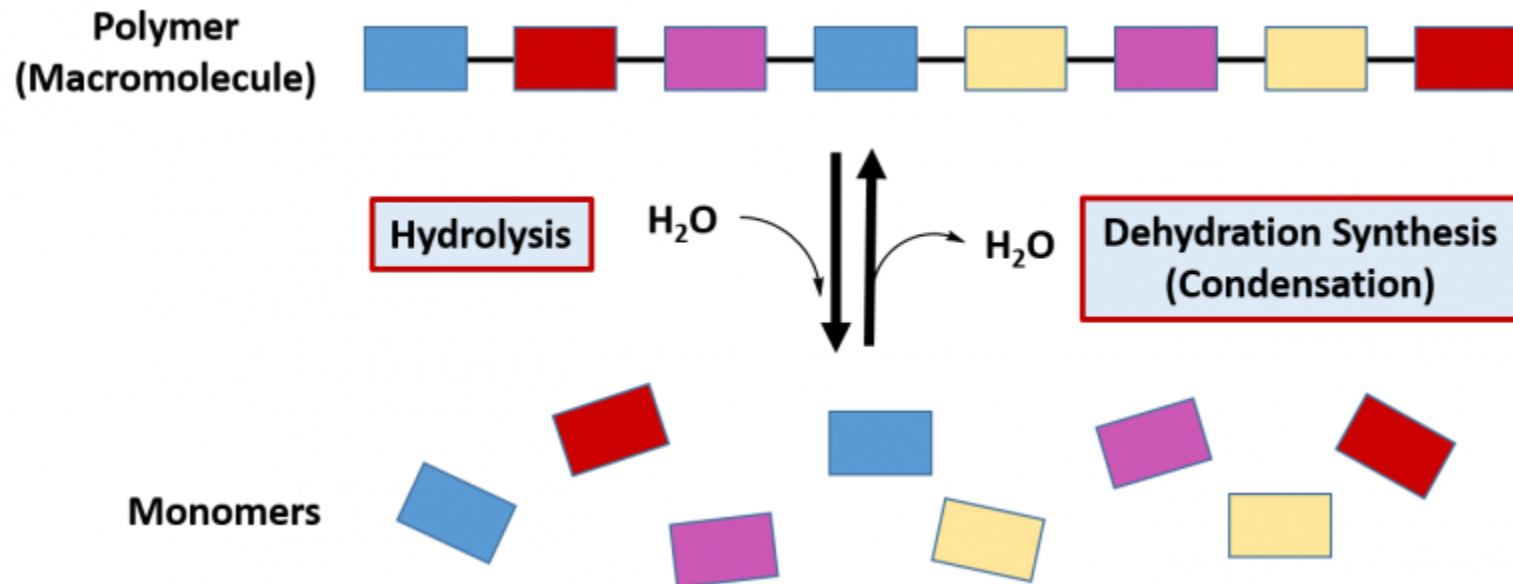


Transfer of an Amine Functional Group. A common group transfer reaction in biological systems is one that is used to produce α -amino acids that can then be used for protein synthesis. In this reaction, one α -amino acid serves as the donor molecule and an α -keto acid (these molecules contain a carboxylic acid functional group and a ketone functional group separated by one α -carbon) serves as the acceptor. In the acceptor molecule, the carbonyl oxygen is replaced with the amine functional group, whereas in the donor molecule, the amine functional group is replaced by an oxygen forming a new ketone functional group.

Hydrolysis Reactions

The classification of hydrolysis reactions include both the forward reactions that involve the addition of water to a molecule to break it apart or the reverse reaction involving the removal of water to join molecules together, termed dehydration synthesis (or condensation) (Figure 7.7). When water is added to a molecule to break it apart into two molecules this reaction is called hydrolysis. The term 'lysis' means to break apart, and the term 'hydro' refers to water. Thus, the term hydrolysis means to break apart with water. The reverse of that reaction involves the removal of water from two molecules to join them together into a larger molecule. Since the two molecules are losing water, they are being dehydrated. Thus, the formation of molecules through the removal of water is known as dehydration synthesis. Since water is also a by-product of these reactions, they are also commonly referred to as condensation reactions. As we have seen in Chapter 6, the formation of the major classes of macromolecules in the body (proteins, carbohydrates, lipids, and nucleic acids) are formed through dehydration synthesis where water is removed from the molecules. During normal digestion of our food molecules, the major macromolecules are broken down into their building blocks through the process of hydrolysi

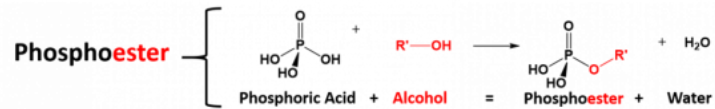
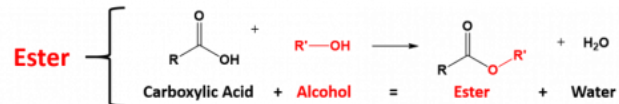
Hydrolysis Reactions



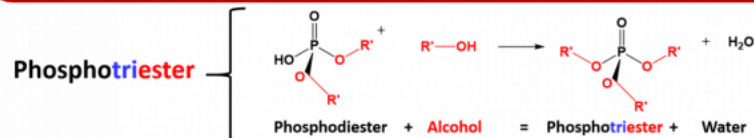
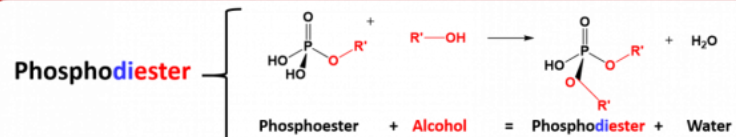
Hydrolysis and Dehydration Synthesis. The reactions of hydrolysis mediate the breakdown of larger polymers into their monomeric building blocks by the addition of water to the molecules. The reverse of the reaction is dehydration synthesis, where water is removed from the monomer building blocks to create the larger polymer structure.

Hydrolysis Reactions

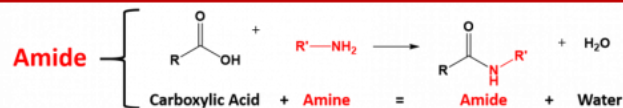
LIPIDS



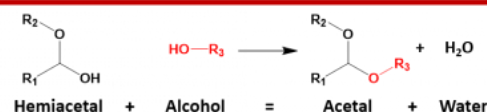
DNA/RNA



PROTEINS



CARBOHYDRATES

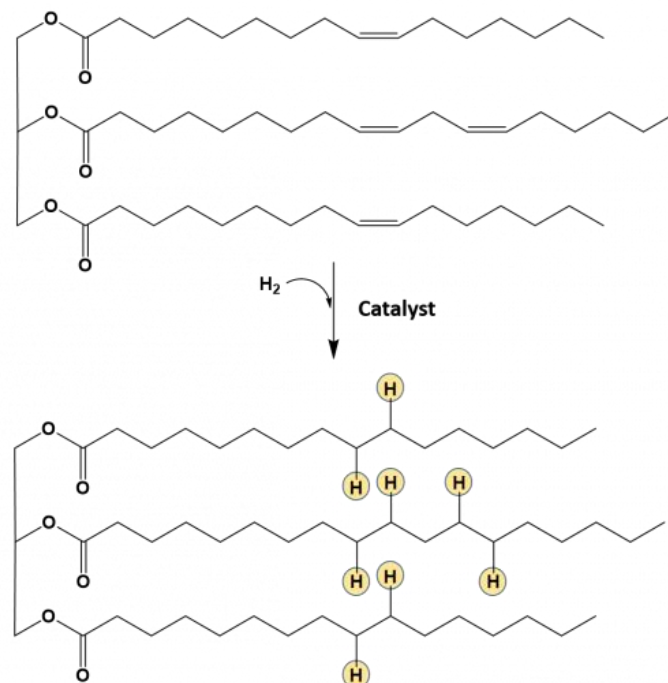


Dehydration Synthesis Reactions Involved in Macromolecule Formation. The major organic reactions required for the biosynthesis of lipids, nucleic acids (DNA/RNA), proteins, and carbohydrates are shown. Note that in all of the reactions, there is a functional group that contains two electron withdrawing groups (the carboxylic acid, phosphoric acid and the hemiacetal each have two oxygen atoms attached to a central carbon or phosphorus atom). This forms a reactive partially positive center atom (carbon in the case of the carboxylic acid and hemiacetal, or phosphorus in the case of the phosphoric acid) that can be attacked by the electronegative oxygen or nitrogen from an alcohol or amine functional group.

The Formation/Removal of Carbon-Carbon Double Bonds

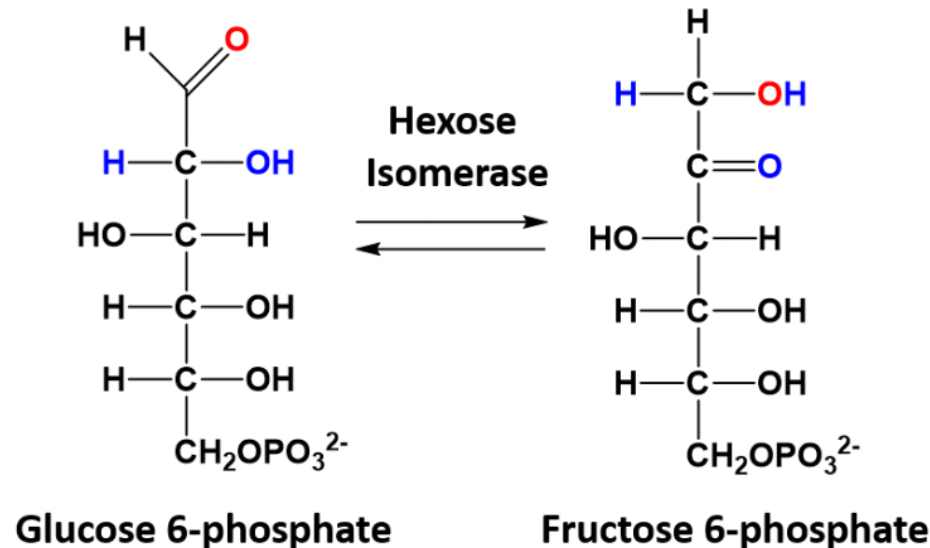
Reactions that mediate the formation and removal of carbon-carbon double bonds are also common in biological systems and are catalyzed by a class of enzymes called lyases. The formation or removal of carbon-carbon double bonds is also used in synthetic organic chemistry reactions to create desired organic molecules. One of these types of reactions is called a hydrogenation reaction, where a molecule of hydrogen (H_2) is added across a C-C double bond, reducing it to a C-C single bond. If this is done using unsaturated oils, the unsaturated fats can be converted into saturated fats. This type of reaction is commonly done to produce partially hydrogenated oils converting them from liquids at room temperature into solids. Margarines made from vegetable oil are made in this manner. Unfortunately, a by-product of this reaction can be the formation of TAGS containing trans double bonds. Once the health hazards of consuming trans fats was recognized, the Food and Drug Administration (FDA) placed a ban on the inclusion of trans fats in food products. This ban was enacted in the summer of 2015 and gave food-makers three years to eliminate them from the food supply, with a deadline of June 18, 2018.

The Formation/Removal of Carbon-Carbon Double Bonds



Hydrogenation of Oils to Produce Margarine. Unsaturated oils can be partially or fully hydrogenated to produce the saturated fatty acids to produce margarines that will remain solid at room temperature. The addition of the new hydrogen atoms to create the saturated hydrocarbons are shown in yellow in the final product.

Isomerization Reactions

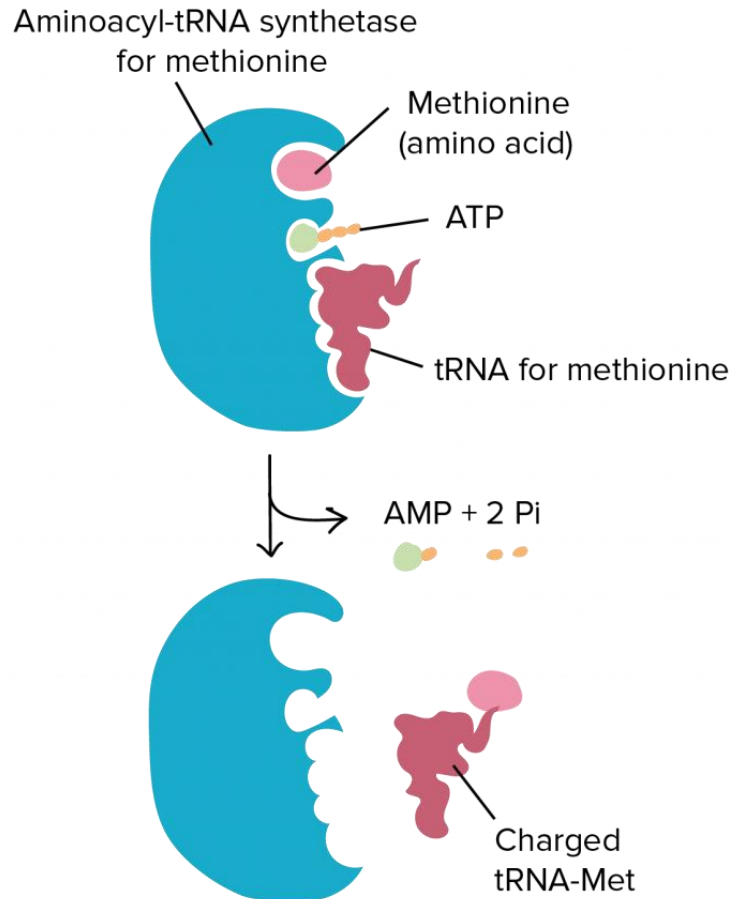


In isomerization reactions a single molecule is rearranged such that it retains the same molecular formula but now has a different bonding order of the atoms forming a structural or stereoisomer. The conversion of glucose 6-phosphate to fructose 6-phosphate is a good example of an isomerization reaction

Ligation Reactions

Ligation reactions use the energy of ATP to join two molecules together. An example of this kind of reaction is the joining of the amino acid with the transfer RNA (tRNA) molecule during protein synthesis. During protein synthesis the tRNA molecules bring each of the amino acids to the ribosome where they can be incorporated into the newly growing protein sequence. To do this, the tRNA molecules must first be attached to the appropriate amino acid. Specific enzymes are available called amino acyl – tRNA synthetases that mediate this reaction. The synthetase enzymes use the energy of ATP to covalently attach the amino acid to the tRNA molecule. For each of the 20 amino acids, there is a specific tRNA molecule and a specific synthetase enzyme that will ensure the correct attachment of the correct amino acid with its tRNA molecule.

Ligation Reactions

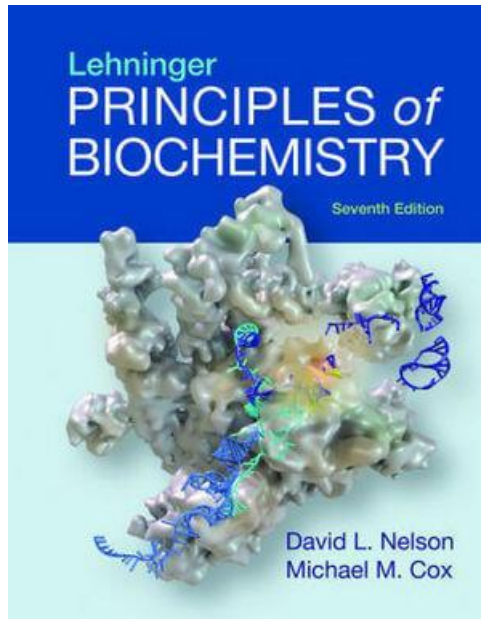


Ligation Reaction Covalently Attaching Methionine with the Appropriate tRNA. The aminoacyl tRNA synthetase enzyme for methionine (shown in blue) covalently attaches methionine (light pink) with the methionine tRNA molecule (dark pink). This reaction requires the energy provided from the breakdown of the ATP molecule into AMP, releasing energy with the breakdown of the phosphate bonds into two inorganic phosphate ions (2 Pi).



Review Questions

- What is the main characteristics of enzyme reactions?
- Explain the different type of catalytic reactions.



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 7. Kinetics of enzyme reactions



Contents

- Introduction
- Equilibrium constant
- Factors influencing enzyme activity
- Michaelis Constant



Introduction

Velocity or rate of enzyme reaction is assessed by the rate of change of substrate to product per unit time.

In practice, initial velocity is determined. If much time is allowed to lapse, the velocity may tend to fall due to decrease in substrate concentration below a critical level.

Catalysts increase the rate of reaction, but do not alter the equilibrium.

The velocity is proportional to the concentration of reacting molecules.

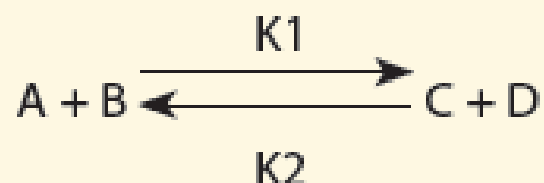


If concentration of A or B is doubled, the rate of reaction is also doubled. If concentrations of A and B are doubled together, the velocity becomes 4-fold.

Derivation of equilibrium constant

$$V \propto [A] [B]$$

At equilibrium, forward reaction and backward reaction are equal, so that



$$\text{Forward reaction } R1 = K1 [A] [B]$$

$$\text{and backward reaction } R2 = K2 [C] [D]$$

$$\text{At equilibrium, } R1 = R2$$

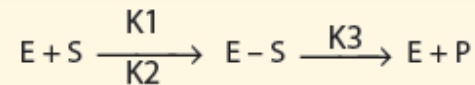
$$\text{Or, } K1 [A] [B] = K2 [C] [D]$$

$$\text{Or, } K1 = \frac{[C] [D]}{[A] [B]} = K_{eq} \text{ or}$$

$$K2 = \frac{[A] [B]}{[C] [D]} \text{ Equilibrium constant.}$$

1. The equilibrium constant of the reaction is the ratio of reaction rate constants of forward and backward reactions.
2. At equilibrium, forward and backward reactions are equal. Equilibrium is a dynamic state. Even though no net change in concentrations of substrate and product occurs, molecules are always interconverted.
3. Numerical value of the constant can be calculated by finding the concentrations of substrates and products.
4. If K_{eq} is more than 1, the forward reaction is favored. In such instances, the reaction is spontaneous and exothermic.
5. Concentration of enzyme does not affect the K_{eq} . Concentration of enzyme certainly increases the rate of reaction; but not the K_{eq} or the ultimate state. In other words, enzyme makes it quicker to reach the equilibrium.

Derivation of Michaelis constant (K_m)



If concentration of substrate is increased, the forward reaction K_1 is increased, and so K_3 as well as total velocity is correspondingly enhanced. The three different constants may be made into one equation,

$$K_m = \frac{K_2 + K_3}{K_1}$$

K_m is called as Michaelis Constant

It is further shown that

$$\text{Velocity (v)} = \frac{V_{\max} [S]}{K_m + [S]}$$

When concentration of substrate is made equal to K_m , i.e.

When $[S] = K_m$

$$\text{Velocity (v)} = \frac{V_{\max} [S]}{[S] + [S]} = \frac{V_{\max} [S]}{2[S]} = \frac{V_{\max}}{2}$$

or $v = \frac{1}{2} V_{\max}$

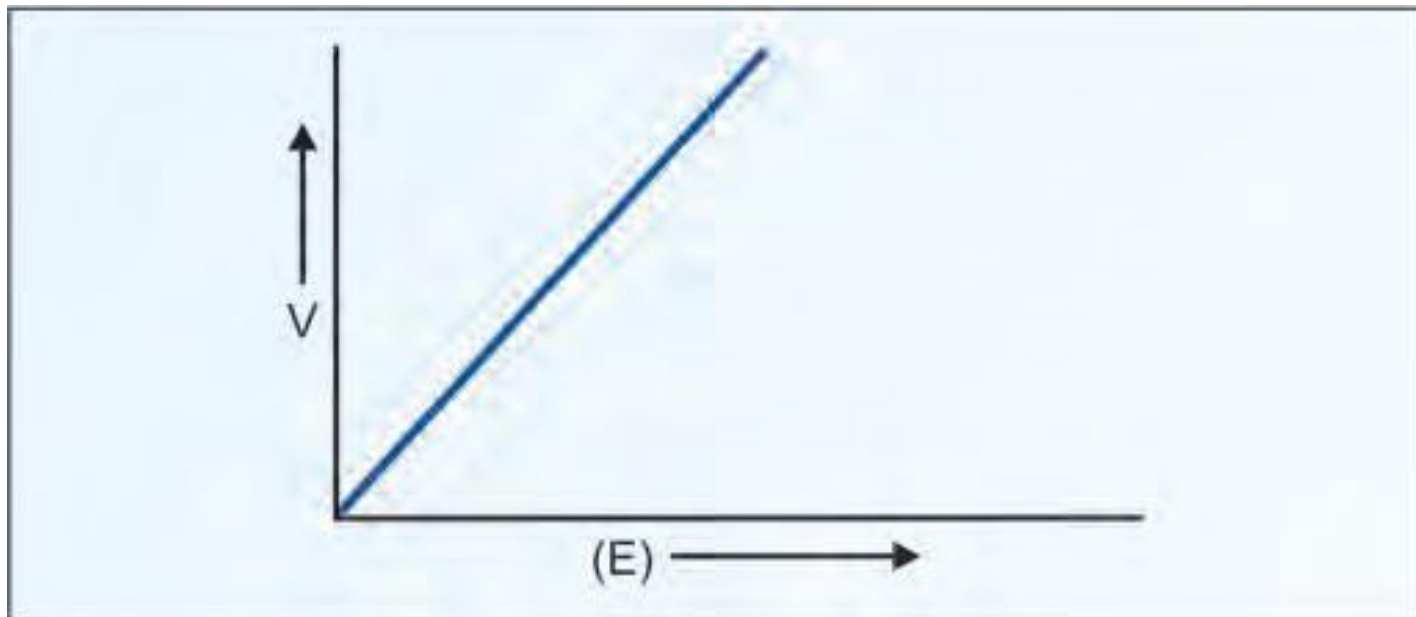
Factors influencing enzyme activity

1. Enzyme concentration
2. Substrate concentration
3. Product concentration
4. Temperature
5. Hydrogen ion concentration (pH)
6. Presence of activators
7. Presence of inhibitors
8. Presence of repressor or derepressor
9. Covalent modification.

Enzyme Concentration

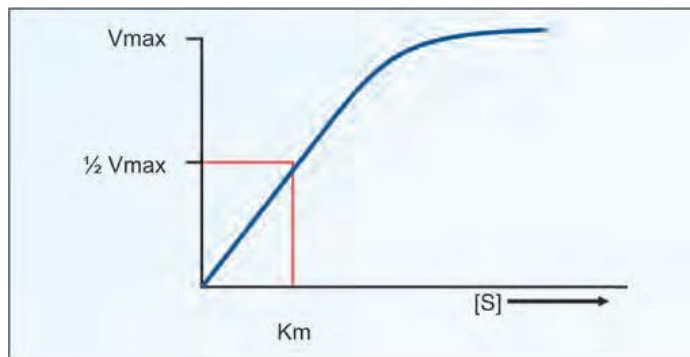
- i. Rate of a reaction or velocity (V) is directly proportional to the enzyme concentration, when sufficient substrate is present. Velocity of reaction is increased proportionately with the concentration of enzyme, provided substrate concentration is unlimited.
- ii. Hence, this property is made use of determining the level of particular enzyme in plasma, serum or tissues.
- iii. Known volume of serum is incubated with substrate for a fixed time, then reaction is stopped and product is quantitated (end point method). Since the product formed will be proportional to the enzyme concentration, the latter could be assayed.

Enzyme Concentration



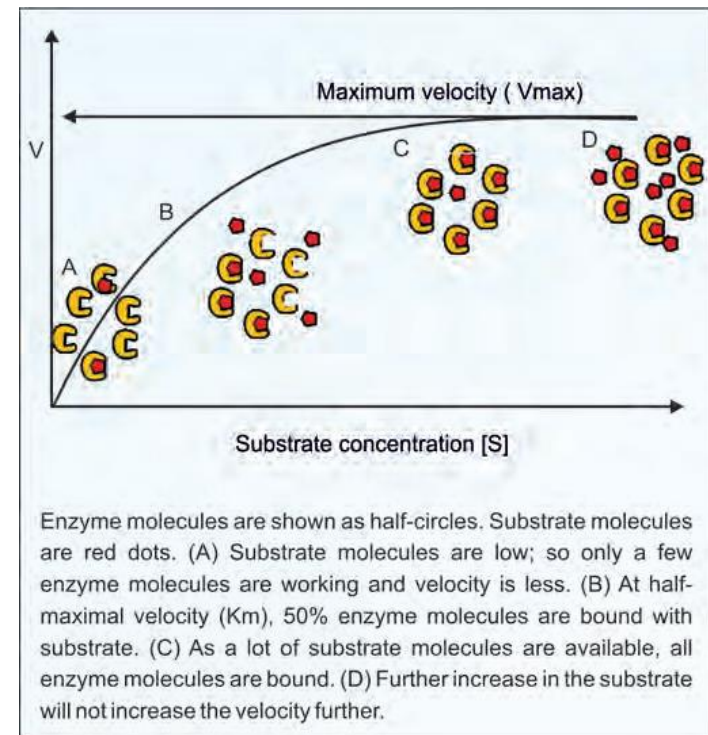
Effect of enzyme concentration

Effect of Substrate Concentration

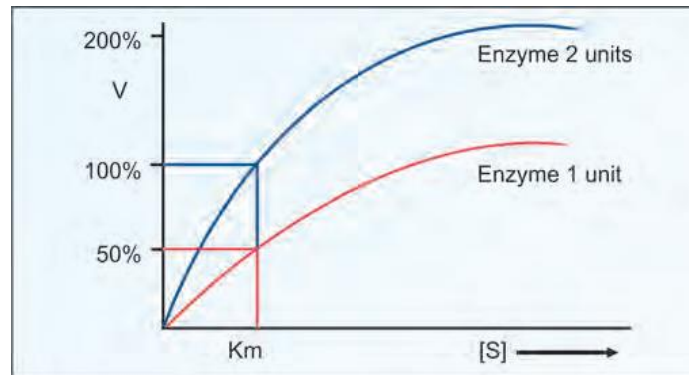


Effect of substrate concentration (substrate saturation curve)

The maximum velocity obtained is called V_{max}

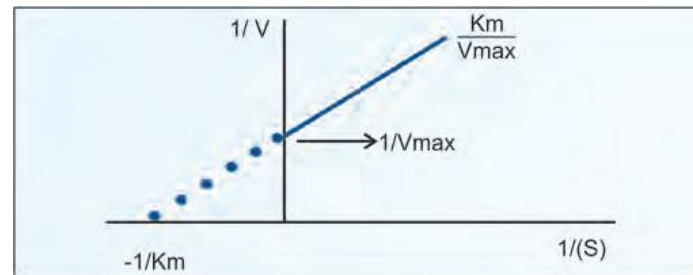


Michaelis Constant



The lesser the numerical value of K_m , the affinity of the enzyme for the substrate is more. To cite an example, K_m of glucokinase is 10 mmol/L and that of hexokinase is 0.05 mmol/L. Therefore, 50% molecules of hexokinase are saturated even at a lower concentration of glucose. In other words, hexokinase has more affinity for glucose than glucokinase.

Double Reciprocal Plot

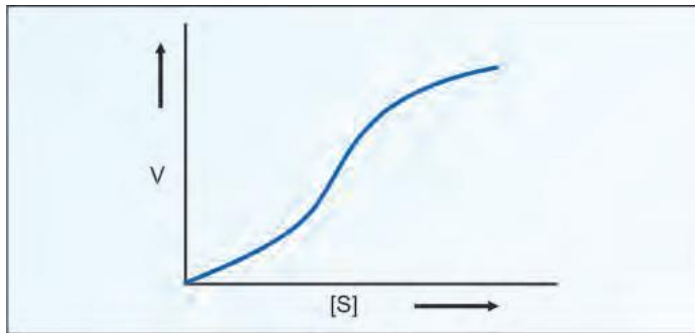


Sometimes it is impractical to achieve high substrate concentrations to reach the maximal velocity conditions. So, $\frac{1}{2}V_{max}$ or K_m may be difficult to determine. Then, the experimental data at lower concentrations is plotted as reciprocals. The straight line thus obtained is extrapolated to get the reciprocal of K_m . This is called Lineweaver-Burk Plot or Double Reciprocal Plot which can be derived from the Michaelis-Menten equation (For Lineweaver-Burk equation, see Box 5.9). If we plot $1/v$ against $1/[S]$, it will give a straight line graph. Intercept in X axis is minus $1/K_m$, from which the K_m can be calculated.

Dixon Plot

The velocity (V) is measured at several concentrations of inhibitor (I), when the substrate (S) concentration is kept constant. It is used for determining inhibition constants. A plot of $1/V$ versus $[I]$ yields a straight line. The experiment is repeated at different concentrations of substrates.

Cooperative Binding

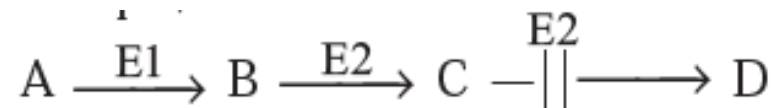


Some enzymes may not strictly follow the Michaelis-Menten kinetics. When the enzyme has many subunits, and binding of substrate to one unit enhances the affinity for binding to other subunits (cooperative binding), a sigmoid shaped saturation curve is obtained. In such cases, determination of K_m value, as shown in the previous paragraph, will be invalid. Instead the Hill equation, originally described for explaining the oxygen binding to Hemoglobin, is employed.

Effect of Concentration of Products

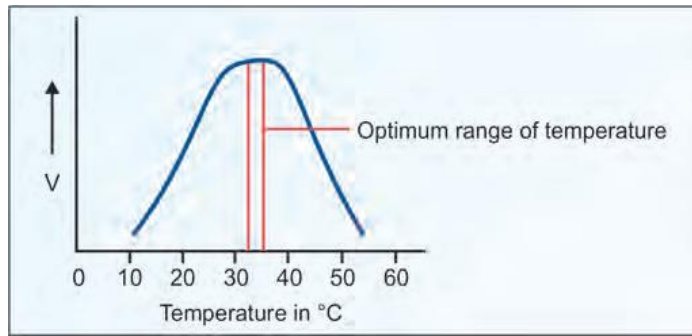
In a reversible reaction, S D P, when equilibrium is reached, as per the law of mass action, the reaction rate is slowed down. So when product concentration is increased, the reaction is slowed, stopped or even reversed. In inborn errors of metabolism, one enzyme of a metabolic pathway is blocked.

For example,



If E3 enzyme is absent, C will accumulate, which in turn, will inhibit E2. Consequently, in course of time, the whole pathway is blocked.

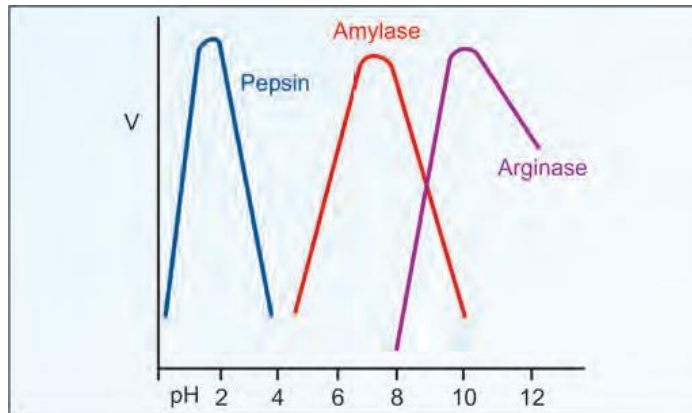
Effect of Temperature



But when temperature is more than 50°C, heat denaturation and consequent loss of tertiary structure of protein occurs. So activity of the enzyme is decreased. Most human enzymes have the optimum temperature around 37°C. Certain bacteria living in hot springs will have enzymes with optimum temperature near 100°C.

The velocity of enzyme reaction increases when temperature of the medium is increased; reaches a maximum and then falls (Bell shaped curve). The temperature at which maximum amount of the substrate is converted to the product per unit time is called the optimum temperature. As temperature is increased, more molecules get activation energy, or molecules are at increased rate of motion. So their collision probabilities are increased and so the reaction velocity is enhanced. The temperature coefficient (Q₁₀) is the factor by which the rate of catalysis is increased by a rise in 10°C. Generally, the rate of reaction of most enzymes will double by a rise in 10°C.

Effect of pH

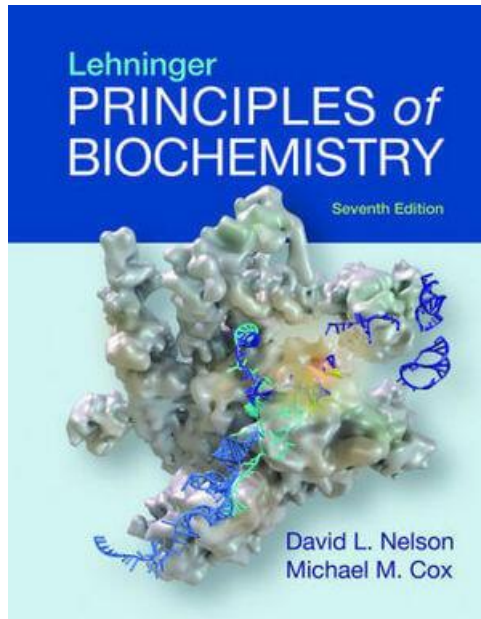


Each enzyme has an optimum pH, on both sides of which the velocity will be drastically reduced. The graph will show a bell shaped curve. The pH decides the charge on the amino acid residues at the active site. The net charge on the enzyme protein would influence substrate binding and catalytic activity. Optimum pH may vary depending on the temperature, concentration of substrate, presence of ions, etc. Usually enzymes have the optimum pH between 6 and 8. Some important exceptions are pepsin (with optimum pH 1–2); alkaline phosphatase (optimum pH 9–10) and acid phosphatase (4–5).



Review Questions

- Explain the kinetics of the enzymatic reactions.
- What are the important constants describing the enzymatic reaction?
- What does Michaelis constant mean?



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 8. Regulation of enzyme reactions



Contents

- Introduction
- Enzyme Activation
- Enzyme Inhibition
- Allosteric Regulation



Introduction

Enzymes can be regulated by other molecules that either increase or reduce their activity. Molecules that increase the activity of an enzyme are called activators, while molecules that decrease the activity of an enzyme are called inhibitors.

There are many kinds of molecules that block or promote enzyme function, and that affect enzyme function by different routes.



Enzyme Activation

In presence of certain inorganic ions, some enzymes show higher activity. Thus, chloride ions activate salivary amylase and calcium ions activate lipase.

Enzyme Activation

Another type of activation is the conversion of an inactive pro-enzyme or zymogen to the active enzyme. i. By splitting a single peptide bond, and removal of

a small polypeptide from trypsinogen, the active trypsin is formed. This results in unmasking of the active center.

ii. Similarly trypsin activates chymotrypsinogen, so that A peptide (1–13 amino acids), B peptide (16–46) and C peptide (149–245) are formed from chymotrypsinogen. These 3 segments align in such a manner that histidine (57) and aspartate (102) and serine (195) residues form the active site.

iii. All the gastrointestinal enzymes are synthesized in the form of pro-enzymes, and only after secretion into the alimentary canal, they are activated. This prevents autolysis of cellular structural proteins.

iv. Coagulation factors are seen in blood as zymogen form.

v. Similar activation of precursor protein is taking place in the case of complement components. These activities are needed only occasionally; but when needed, a large number of molecules are to be produced instantaneously.

Competitive Inhibition

- Here inhibitor molecules are competing with the normal substrate molecules for attaching with the active site of the enzyme.



- Since E-I (enzyme-inhibitor complex) can react only to reform the enzyme and inhibitor, the number of enzyme molecules available for E-S formation is reduced. Suppose 100 molecules of substrate and 100 molecules of inhibitor are competing for 100 molecules of the enzyme. So, half the enzyme molecules are trapped by the inhibitor and only half the molecules are available for catalysis to form the product.
- iii. Since effective concentration of enzyme is reduced, the reaction velocity is decreased.

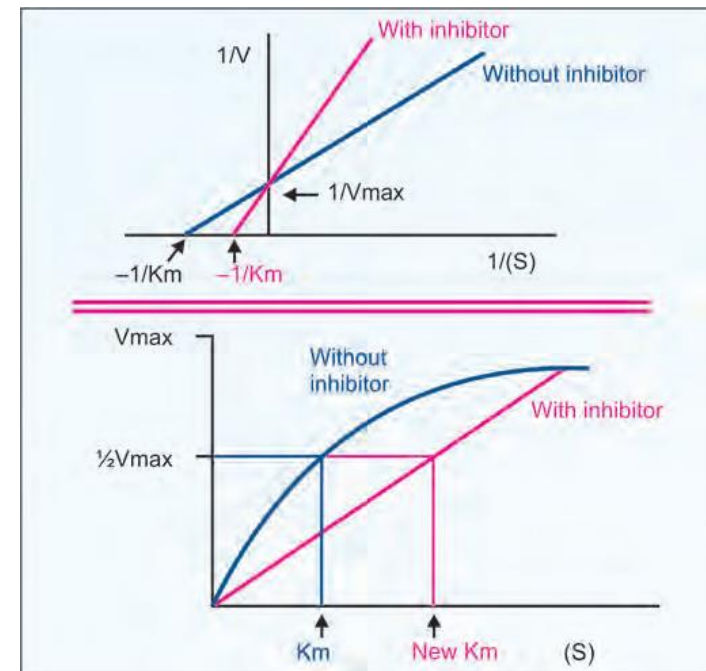
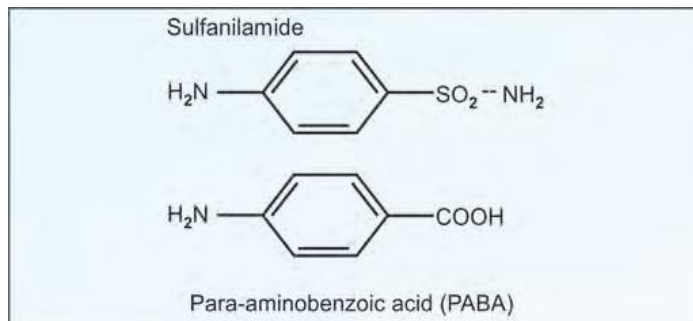
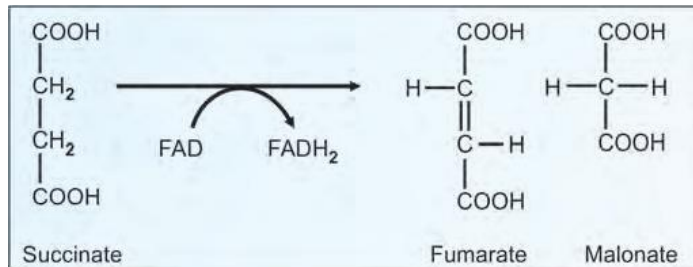
Competitive Inhibition

- In competitive inhibition, the inhibitor will be a structural analog of the substrate. There will be similarity in three-dimensional structure between substrate (S) and inhibitor (I). For example, the succinate dehydrogenase reaction is inhibited by malonate.
- Competitive inhibition is usually reversible. Or, excess substrate abolishes the inhibition. In the previous example of 100 moles of E and 100 moles of I, if 900 moles of S are added, only 1/10th of enzyme molecules are attached to inhibitor and 90% are working with substrate. Thus 50% inhibition in the first example is now decreased to 10% inhibition.

Competitive Inhibition

- From the graphs, it is obvious that in the case of competitive inhibition, the K_m is increased in presence of competitive inhibitor. Thus competitive inhibitor apparently increases the K_m . In other words, the affinity of the enzyme towards substrate is apparently decreased in presence of the inhibitor.
- But V_{max} is not changed.

Competitive Inhibition



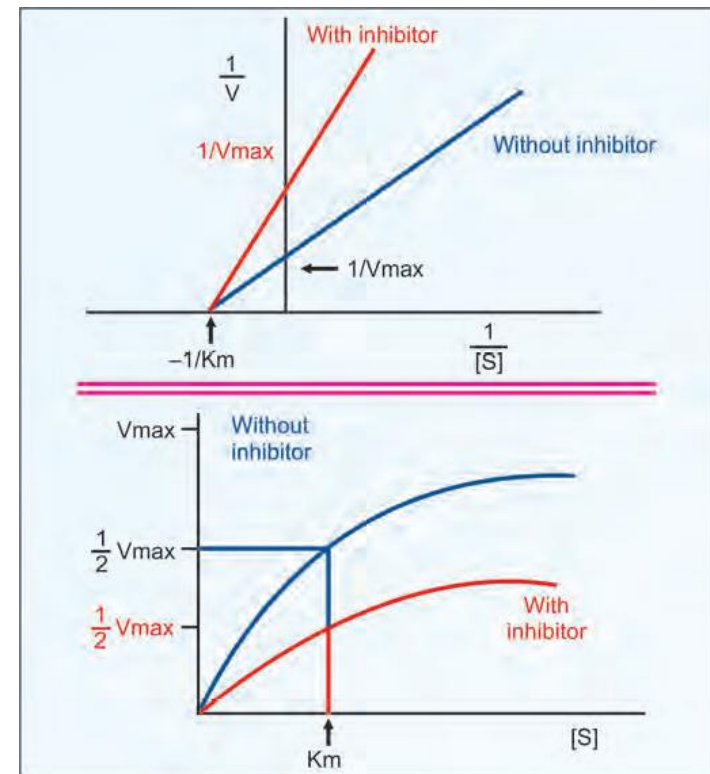
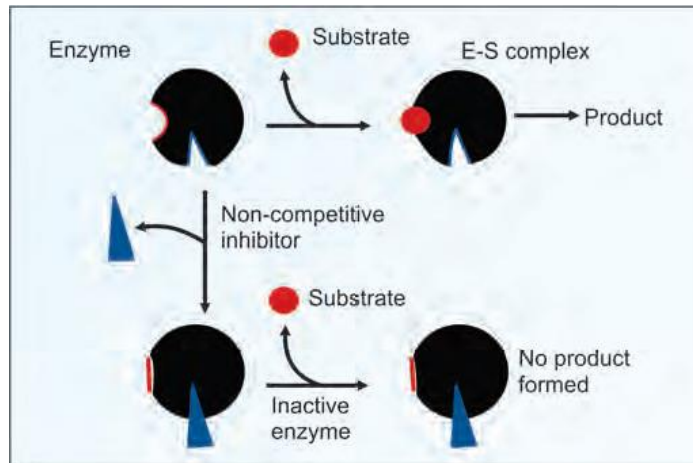
Non-competitive Inhibition (Irreversible)

- A variety of poisons, such as iodoacetate, heavy metal ions (lead, mercury) and oxidizing agents act as irreversible non-competitive inhibitors. There is no competition between substrate and inhibitor.
- The inhibitor usually binds to a different domain on the enzyme, other than the substrate binding site. Since these inhibitors have no structural resemblance to the substrate, an increase in the substrate concentration generally does not relieve this inhibition. Examples are:

Non-competitive Inhibition (Irreversible)

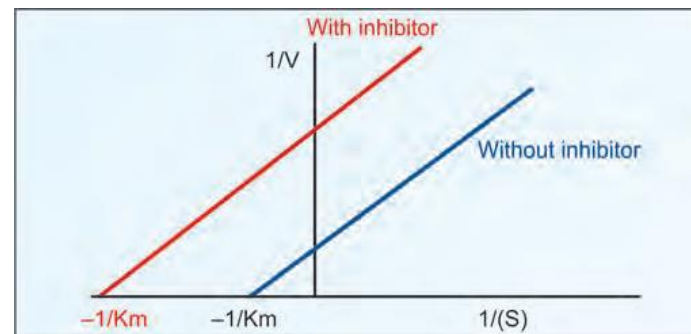
Examples are:

- Cyanide inhibits cytochrome oxidase.
- Fluoride will remove magnesium and manganese ions and so will inhibit the enzyme, enolase, and consequently the glycolysis.
- Iodoacetate would inhibit enzymes having –SH group in their active centers.
- BAL (British Anti-Lewisite; dimercaprol) is used as an antidote for heavy metal poisoning. The heavy metals act as enzyme poisons by reacting with the SH group. BAL has several SH groups with which the heavy metal ions can react and thereby their poisonous effects are reduced.
- Acetylcholinesterase enzyme cleaves acetylcholine to form acetate and choline and therefore terminates the action of acetylcholine. Certain chemicals e.g. diisopropyl fluorophosphates (DFP) binds to the active site, serine of acetylcholinesterase. As a result acetylcholine accumulates and over-stimulates autonomous nervous system including heart, blood vessels and glands. This leads to vomiting, salivation, sweating, and in worst cases even death. DFP forms an irreversible covalent bond with acetylcholinesterase, and activity can be regained only if new enzyme is synthesized.



Uncompetitive Inhibition

Here inhibitor does not have any affinity for free enzyme. Inhibitor binds to enzyme–substrate complex; but not to the free enzyme. In such cases both V_{max} and K_m are decreased. Inhibition of placental alkaline phosphatase (Regan iso-enzyme) by phenylalanine is an example of uncompetitive inhibition.



Suicide Inhibition

- i. It is a special type of irreversible inhibition of enzyme activity. It is also known as mechanism based inactivation. The inhibitor makes use of the enzyme's own reaction mechanism to inactivate it (mechanism based inactivation).
- ii. In suicide inhibition, the structural analog is converted to a more effective inhibitor with the help of the enzyme to be inhibited. The substrate-like compound initially binds with the enzyme and the first few steps of the pathway are catalyzed.
- iii. This new product irreversibly binds to the enzyme and inhibits further reactions.

Suicide Inhibition

iv. For example, ornithine decarboxylase (ODC) catalyzes the conversion of ornithine to putrescine which is necessary for polyamine synthesis. When the ODC in trypanosoma is inhibited multiplication of the parasite is arrested. Therefore inhibitors of ODC enzyme such as difluoromethylornithine (DFMO) has been found to be effective against trypanosomiasis (sleeping sickness). DFMO is initially inert, but on binding with the enzyme, forms irreversible covalent complex with the co-enzyme (pyridoxal phosphate) and the amino acid residues of the enzyme. In mammalian cells, the turnover rate of ODC is very high, and so the inhibition by DFMO is only transient. So DFMO kills the parasites with no side effects to the patient.

v. A similar mechanism is observed in the case of Allopurinol that is oxidized by xanthine oxidase to alloxanthine which is a more potent inhibitor of xanthine oxidase.

vi. The anti-inflammatory action of Aspirin is also based on suicide inhibition. Arachidonic acid is converted to prostaglandin by the enzyme Cyclo-oxygenase. Aspirin acetylates a serine residue in the active center of cyclo-oxygenase, thus prostaglandin synthesis is inhibited, and so inflammation subsides.

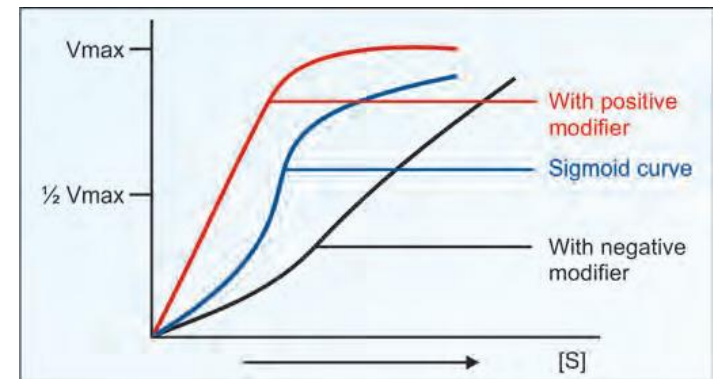
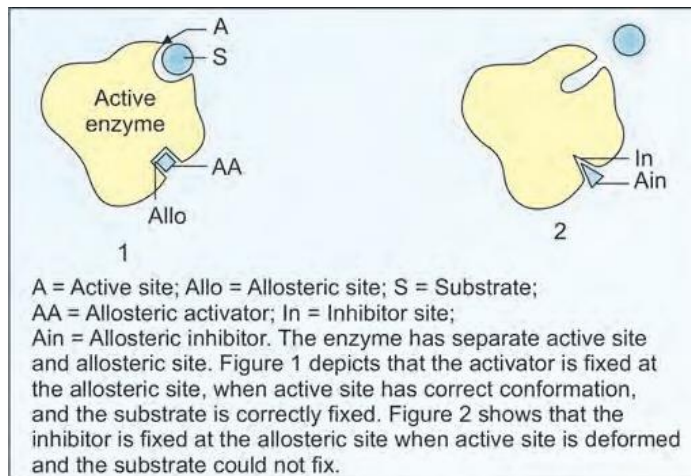
Allosteric Regulation

- i. Allosteric enzyme has one catalytic site where the substrate binds and another separate allosteric site where the modifier binds (allo = other).
- ii. Allosteric and substrate binding sites may or may not be physically adjacent.
- iii. The binding of the regulatory molecule can either enhance the activity of the enzyme (allosteric activation), or inhibit the activity of the enzyme (allosteric inhibition).
- iv. In the former case, the regulatory molecule is known as the positive modifier and in the latter case as the negative modifier.

Allosteric Regulation

- v. The binding of substrate to one of the subunits of the enzyme may enhance substrate binding by other subunits. This effect is said to be positive cooperativity. If the binding of substrate to one of the subunits decreases the avidity of substrate binding by other sites, the effect is called negative cooperativity.
- vi. In most cases, a combination is observed, resulting in a sigmoid shaped curve.

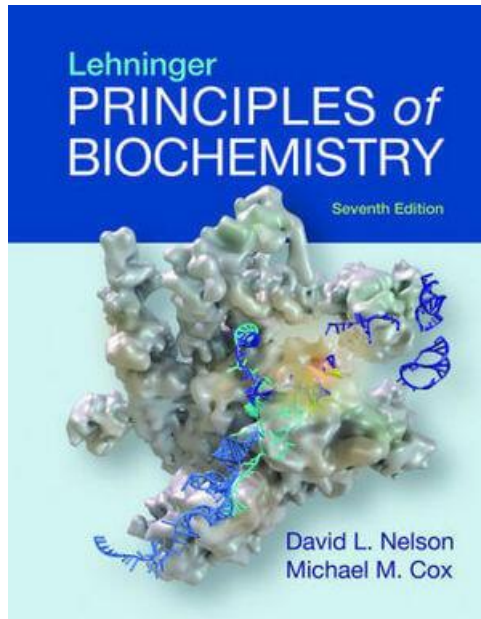
Allosteric Regulation





Review Questions

- What type of regulations of enzyme action exist?
- Explain the activation mechanism.
- Explain the inhibition mechanism.
- Explain the allosterism.



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 3. Proteins and enzymes

Lesson 9. Enzyme classification



Contents

- Introduction
- IUBMB System of Classification



Introduction

When early workers isolated certain enzymes, whimsical names were given. Some of these, such as Pepsin, Trypsin, Chymotrypsin, etc. are still used. Later, it was agreed to call the enzymes by adding the suffix "-ase" to the substrate. Thus, enzyme Lactase acts on the substrate lactose, and the products glucose and galactose are formed. Enzymes that hydrolyse starch (amylose) are termed as amylases; those that dehydrogenate the substrates are called dehydrogenases. These are known as the trivial names of the enzymes.

IUBMB System of Classification

International Union of Biochemistry and Molecular Biology (IUBMB) in 1964, (modified in 1972 and 1978), suggested the IUBMB system of nomenclature of enzymes. It is complex and cumbersome; but unambiguous. As per this system, the name starts with EC (enzyme class) followed by 4 digits.

First digit represents the class

Second digit stands for the subclass

Third digit is the sub-sub class or subgroup

Fourth digit gives the number of the particular enzyme in the list.

The enzymes are grouped into following six major classes

Classification of enzymes

Class 1: Oxidoreductases: Transfer of hydrogen or addition of oxygen; e.g. Lactate dehydrogenase (NAD); Glucose-6-phosphate dehydrogenase (NADP); Succinate dehydrogenase (FAD); dioxygenases.

Class 2: Transferases: Transfer of groups other than hydrogen. Example, Aminotransferase. (Subclass: Kinase, transfer of phosphoryl group from ATP; e.g. Hexokinase).

Class 3: Hydrolases: Cleave bond and add water; e.g. Acetylcholine esterase; Trypsin.

Class 4: Lyases: Cleave without adding water, e.g. Aldolase; HMG CoA lyase; ATP Citrate lyase. (Subclass: Hydratase; add water to a double bond).

Class 5: Isomerases: Intramolecular transfers. They include racemases and epimerases. Example, Triose phosphate isomerase.

Class 6: Ligases: ATP dependent condensation of two molecules, e.g. Acetyl CoA carboxylase; Glutamine synthetase; PRPP synthetase.

Class 1: Oxidoreductases

This group of enzymes will catalyze oxidation of one substrate with simultaneous reduction of another substrate or co-enzyme. This may be represented as

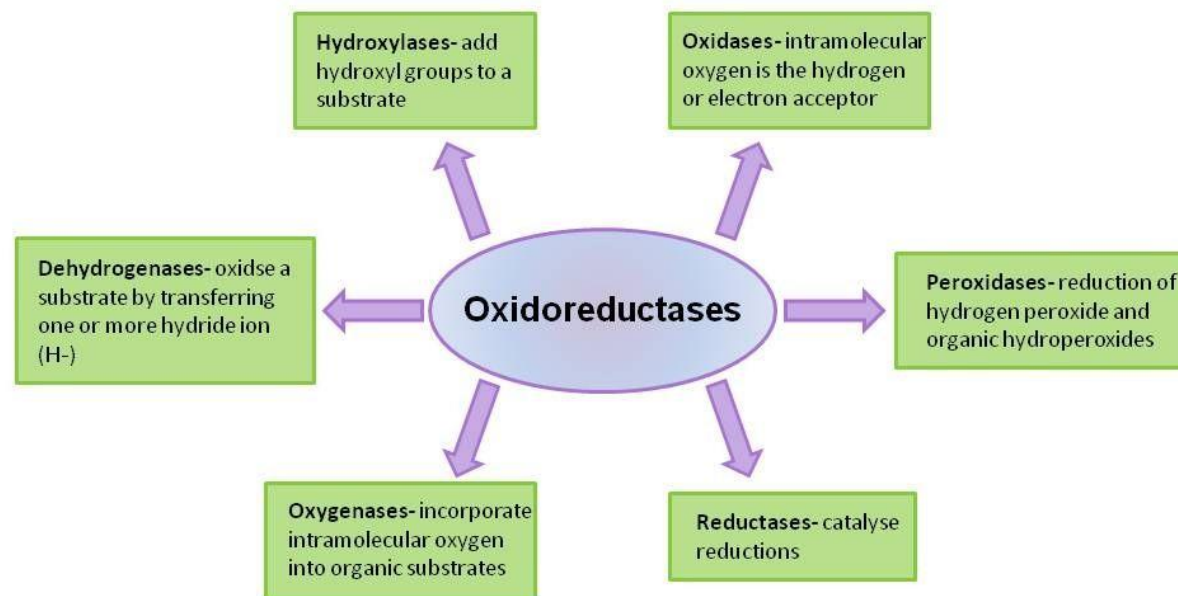


for example,



The enzyme is Alcohol dehydrogenase; IUB name is Alcohol-NAD-oxidoreductase; Code number is EC.1.1.1.1. Oxidoreductases may also oxidize substrates by adding oxygen, e.g. oxidases, oxygenases and dehydrogenases

Class 1: Oxidoreductases



Class 2: Transferases

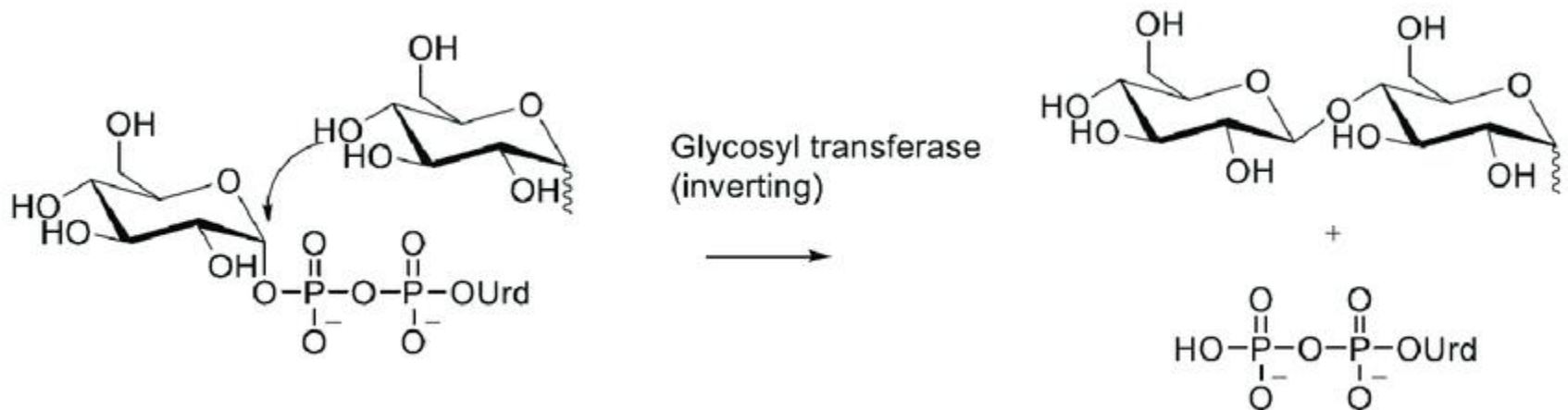
This class of enzymes transfers one group (other than hydrogen) from the substrate to another substrate. This may be represented as

$A-R + B \rightarrow A + B-R$, For example,

$\text{Hexose} + \text{ATP} \rightarrow \text{Hexose-6-phosphate} + \text{ADP}$

The name of enzyme is Hexokinase and systematic name is ATP-Hexose–6-phosphate-transferase

Class 2: Transferases



Class 3: Hydrolases

This class of enzymes can hydrolyze ester, ether, peptide or glycosidic bonds by adding water and then breaking the bond.

Acetylcholine + H₂O → Choline + acetate

The enzyme is Acetylcholine esterase or Acetylcholine hydrolase (systematic). All digestive enzymes are hydrolases.

Class 3: Hydrolases



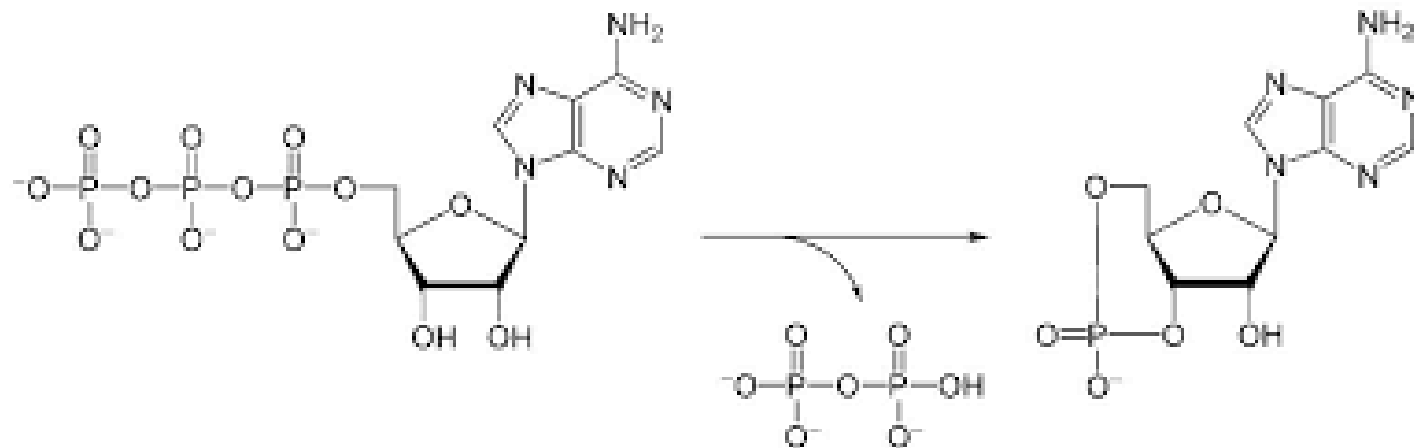


Class 4: Lyases

These enzymes can remove groups from substrates or break bonds by mechanisms other than hydrolysis. For example,

Fructose-1,6-bisphosphate → Glyceraldehyde-3-phosphate + dihydroxy acetone phosphate

Class 4: Lyases





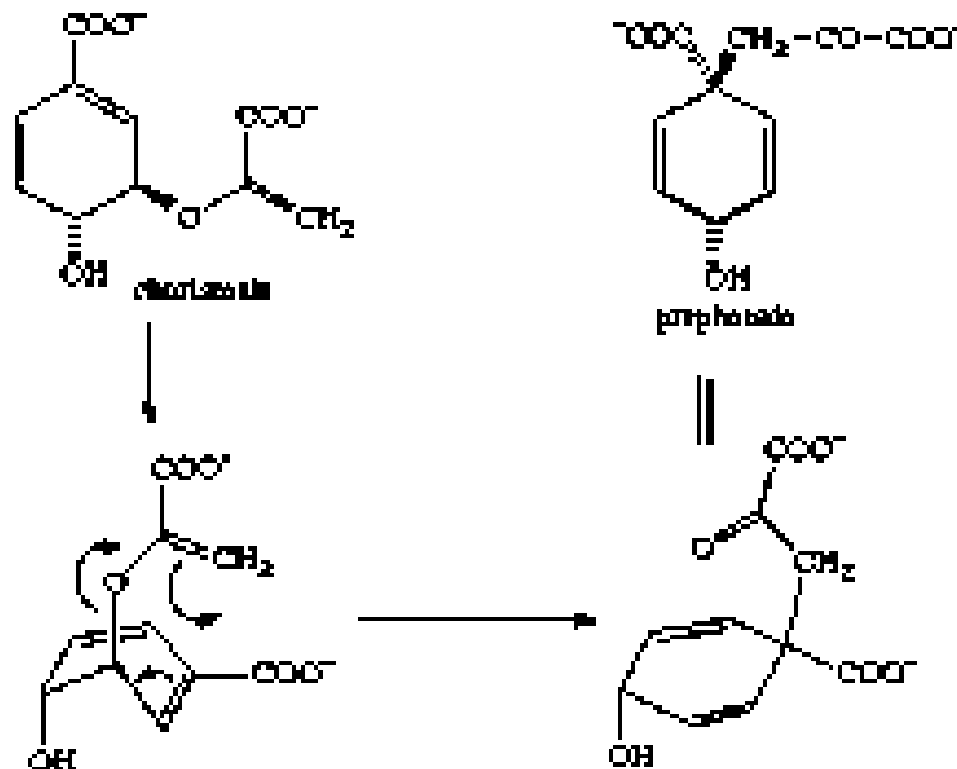
Class 5: Isomerases

These enzymes can produce optical, geometric or positional isomers of substrates. Racemases, epimerases, cistrans isomerases are examples.

Glyceraldehyde-3-phosphate → Dihydroxy acetone phosphate

Enzyme is Triose phosphate isomerase.

Class 5: Isomerases





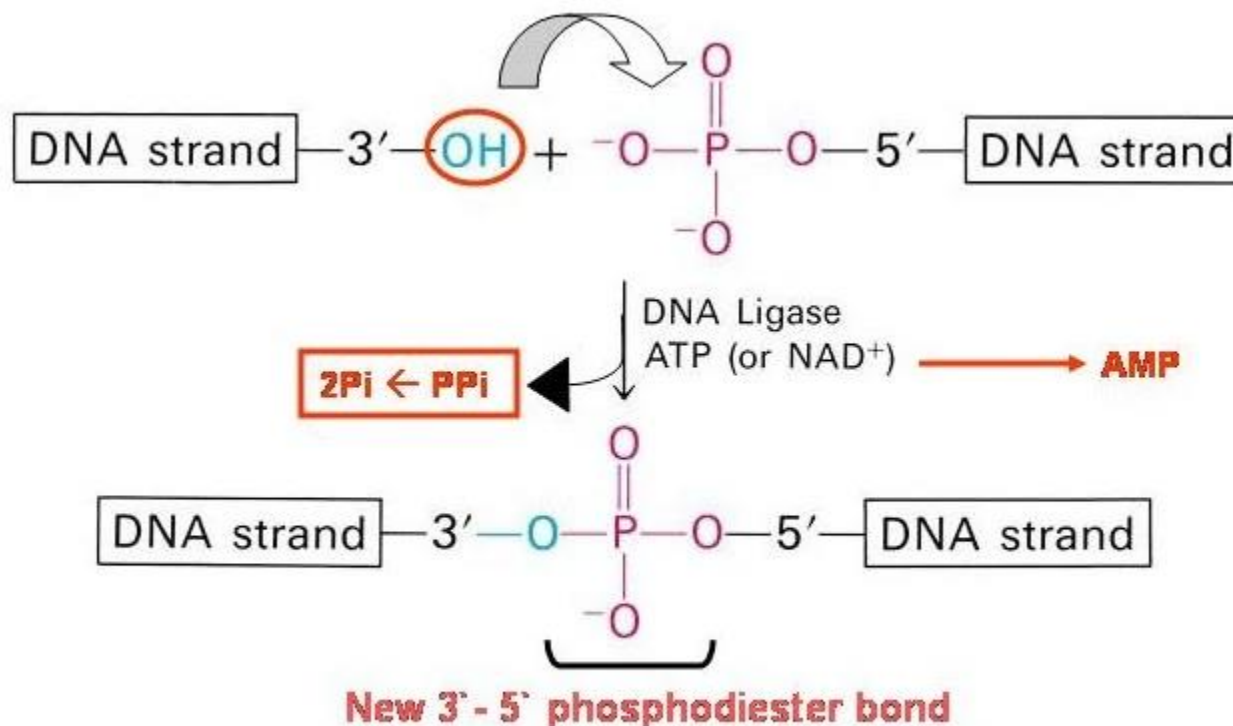
Class 6: Ligases

These enzymes link two substrates together, usually with the simultaneous hydrolysis of ATP, (Latin, Ligare = to bind). For example,



Enzyme is Acetyl CoA carboxylase.

Class 6: Ligases



Synthetase and synthase are different

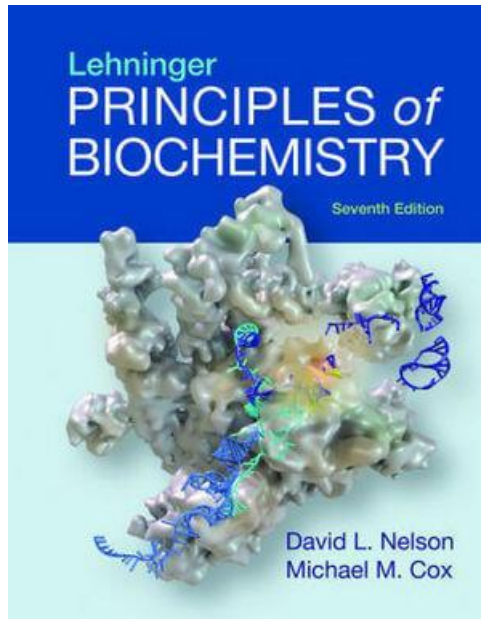
Synthetases are ATP-dependent enzymes catalyzing biosynthetic reactions; they belong to Ligases (class 6). Examples are Carbamoyl phosphate synthetase; Arginino succinate synthetase; PRPP synthetase and Glutamine synthetase.

Synthases are enzymes catalyzing biosynthetic reactions; but they do not require ATP directly; they belong to classes other than Ligases. Examples are Glycogen synthase and ALA synthase.



Review Questions

- What is the main criterion by which enzymes are classified?
- Give some examples from each class of enzyme and explain their action.



Lehninger, A. L., Nelson, D. L., & Cox, M. M.
(2000). Lehninger principles of biochemistry.
New York: Worth Publishers.



Molecular Biology of the Cell. 4th edition.
Alberts B, Johnson A, Lewis J, et al.
New York: Garland Science; 2002.



Module 1. Molecular basis of life

Topic 4. Biophysical Problems

Lesson 1. Thermodynamics



Contents

- Introduction
- Basic Concepts and Definitions
- The Properties of a Pure Substance
- Work and Heat
- The First Law of Thermodynamics
- The Second and Third Laws of Thermodynamics



Introduction

Thermodynamics is the science that deals with heat and work and these properties of substances that bear a relation to heat and work. Like all sciences, the basis of thermodynamics is experimental observation.

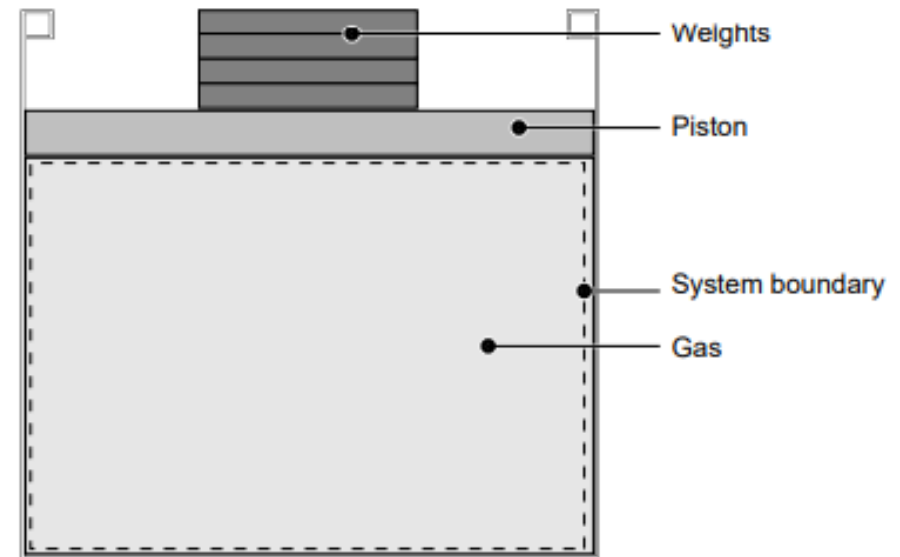
In what follows, we will present the basic thermodynamics laws, and apply them to a number of representative examples.

Basic Concepts and Definitions

Thermodynamic system

A **thermodynamic system** is defined as a quantity of matter of fixed mass and identity on which attention is focused for study. Everything external to the system is the surroundings, and the system is separated from the surroundings by the system boundaries. These boundaries may be either movable or fixed.

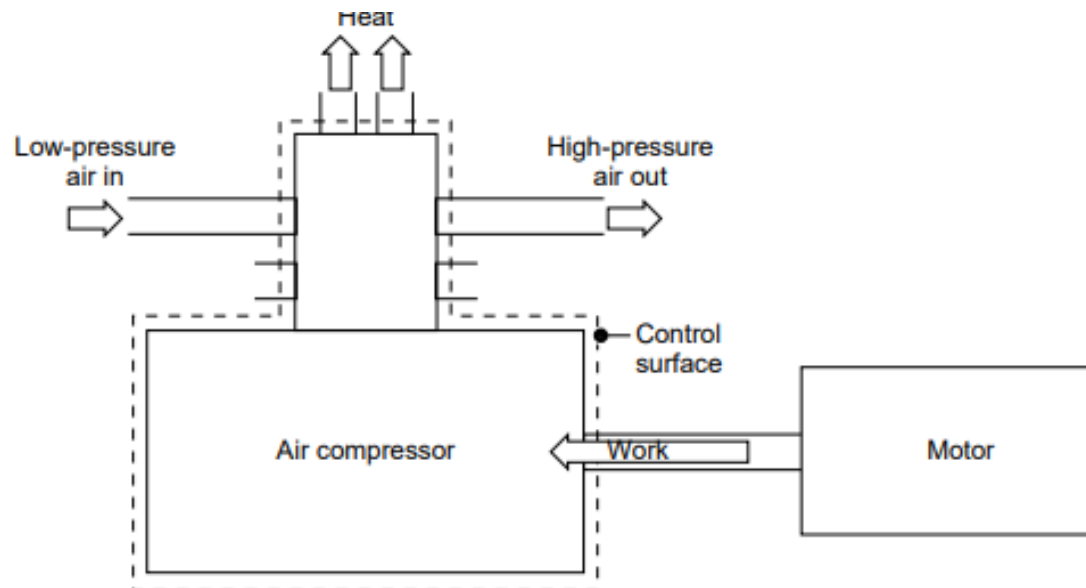
Example of a thermodynamic system.



Basic Concepts and Definitions

Control volume

Mass, as well as heat and work (and momentum), can flow across the control surface



Basic Concepts and Definitions

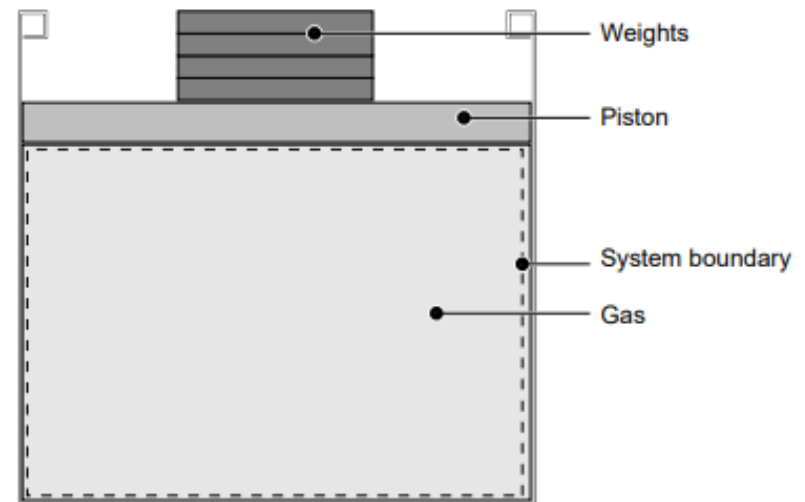
Properties and state of a substance

A given mass of water can exist in various forms. If it is liquid, it may become vapour when heated or solid when cooled. Thus, we talk about the different Phases of a substance. A phase is defined as a quantity of matter that is homogeneous throughout. When more than one phase is present, the phases are separated from each other by the phases boundaries. In each phase the substance may exist at various pressures and temperatures or, to use the thermodynamic term, in various states. The State may be identified or described by certain observable, macroscopic properties; some familiar ones are temperature, pressure, and density.

Basic Concepts and Definitions

Process and cycles

Whenever one or more of the properties of a system change, a Change in State has occurred. The path of the succession of states through which the system passes is called the process. When a system in a given initial state goes through a number of different changes of state or processes and finally returns to its initial state, the system has undergone a Cycle. Therefore, at the conclusion of a cycle, all the properties have the same value they had at the beginning. Steam (water) that circulates through a steam power plant (like the conventional side of a nuclear reactor) undergoes a cycle.



Basic Concepts and Definitions

Energy

One of the very important concepts in a study of thermodynamics is the concept of energy. Energy is a fundamental concept, such as mass or force and, as is often the case with such concepts, is very difficult to define. Energy is defined as the capability to produce an effect. It is important to note that energy can be stored within a system and can be transferred (as heat, for example) from one system to another.

Basic Concepts and Definitions

Specific volume

The specific volume of a substance is defined as the volume per unit mass, and is given the symbol v .

$$v = \delta V / \delta m [\text{m}^3/\text{kg}]$$

The density of a substance (ρ) is defined as the mass per unit volume, and is therefore the reciprocal of the specific volume.

$$\rho = 1/v [\text{kg}/\text{m}^3]$$

Basic Concepts and Definitions

Pressure

Pressure is defined as the normal component of force per unit area.

$$p = \frac{\delta F_n}{\delta A} \quad \left[\frac{N}{m^2} \text{ or } Pa \right]$$

Basic Concepts and Definitions

Temperature

Although temperature is a familiar property, an exact definition of it is difficult. Thus, we define equality of temperatures. Two bodies have equality of temperatures if, when they are in thermal communication, no change in observable property occurs.



Basic Concepts and Definitions

The zeroth law of thermodynamics

The zeroth law of thermodynamics states that when two bodies have equality of temperatures with a third body, they in turn have equality of temperatures with each other.

The Properties of a Pure Substance

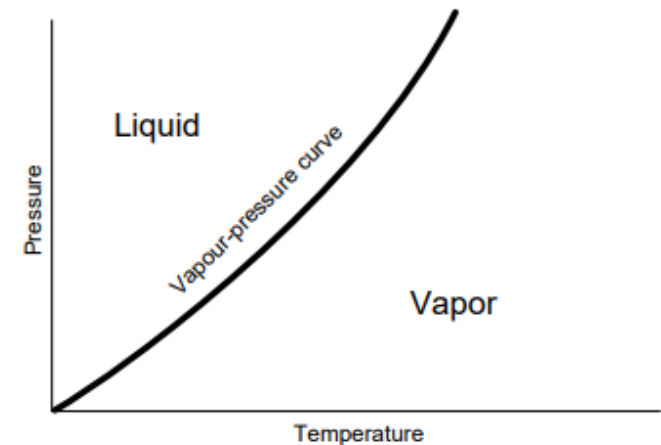
The pure substance

A pure substance is one that has a homogeneous and invariable chemical composition. It may exist in more than one phase, but the chemical composition is the same in all phases. Thus, liquid water, a mixture of liquid water and water vapour (steam), and a mixture of ice and liquid water are all pure substances; every phase has the same chemical composition.

The Properties of a Pure Substance

The term **Saturation Temperature** designates the temperature at which vaporization takes place at a given pressure. This pressure is called the **Saturation Pressure** for the given temperature. Thus, for water at 99.6 °C the saturation pressure is 0.1 MPa, and for water at 0.1 MPa the saturation temperature is 99.6 °C. For a pure substance there is a definite relation between saturation pressure and saturation temperature.

Vapour-pressure curve of a pure substance.



The Properties of a Pure Substance

Equation of state for the vapour phase of a simple compressible substance

From experimental observations it has been established that the P-v-T behaviour of gases at low density is closely given by the following equation of state:

$$P v = RT$$

where

$$R = 8.3144 \text{ kN m/kmol K}$$

is the universal gas constant.

Work and Heat

Definition of work

Work is defined as a force F acting through a displacement x , the displacement being in the direction of the force.

$$W = \int_1^2 F dx$$

Thermodynamics def:

Thermodynamics defines work as follows: work is done by a system if the sole effect on the surroundings (everything external to the system) could be the raising of a weight. Work done by a system is considered positive and work done to a system is considered negative

Work and Heat

Definition of heat

Heat is defined as the form of energy that is transferred across the boundary of a system at a given temperature to another system (or the surroundings) at a lower temperature by virtue of the temperature difference between the two systems. Another aspect of this definition of heat is that a body never contains heat. Rather, heat can be identified only as it crosses the boundary. Thus, heat is a transient phenomenon.

First law of thermodynamics

The change in internal energy of a system is equal to the heat added to the system minus the work done by the system.

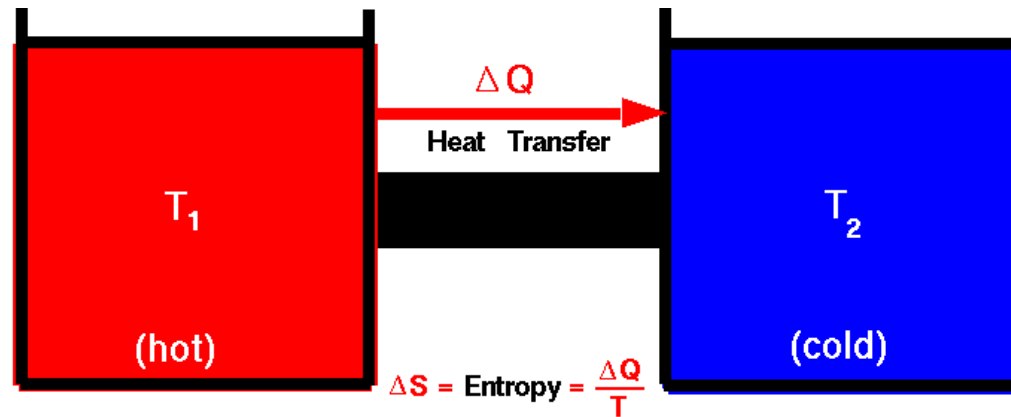
$$\Delta U = Q - W$$

Change in
internal
energy

Heat added
to the system

Work done
by the system

Second law of thermodynamics



There exists a useful thermodynamic variable called entropy (S).
A natural process that starts in one equilibrium state and ends
in another will go in the direction that causes the entropy of the
system plus the environment to increase for an irreversible
process and to remain constant for a reversible process.

$$S_f = S_i \text{ (reversible)}$$

$$S_f > S_i \text{ (irreversible)}$$

Gibbs energy

- A process is spontaneous if the overall entropy change for the system is positive.

So the total entropy change, ΔS_{tot} , at constant temperature and pressure is given by:

$$\Delta S_{\text{tot}} = \Delta S - \frac{\Delta H}{T}$$

$$\Delta S_{\text{sur}} = -\frac{\Delta H}{T}$$

Multiplying both sides by $-T$ yields:

$$-T\Delta S_{\text{tot}} = -T\Delta S + \Delta H$$

Since the temperature is always a positive number, the reaction is spontaneous if the term $-T\Delta S_{\text{tot}}$ is negative. If the process is in equilibrium then this term is equal to zero. The product of temperature and entropy has units of energy and is related to the amount of energy available to do work. This term, $-T\Delta S_{\text{tot}}$, is usually called the Gibbs energy difference, ΔG , and is written as:

$$\Delta G = \Delta H - T\Delta S$$

- if ΔG is a positive then the reaction is unfavorable and the initial state is favored,
- if ΔG is zero the reaction is in equilibrium, and
- only if ΔG is negative will the reaction occur spontaneously.

Relationship between the gibbs energy and the equilibrium constant

- For any given reaction $A \leftrightarrow B$ with an equilibrium constant K , the value of the equilibrium constant can be written in terms of the change in the Gibbs energy:

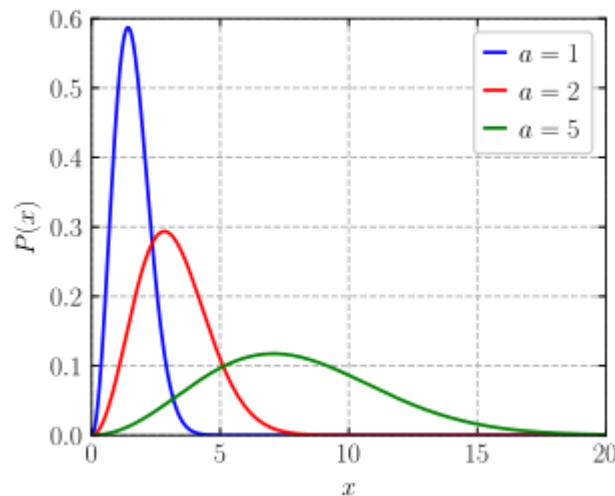
$$K = e^{-\Delta G/RT}$$

Relationships among the equilibrium constant, K , Gibbs energy change, ΔG , and direction of a chemical reaction*.

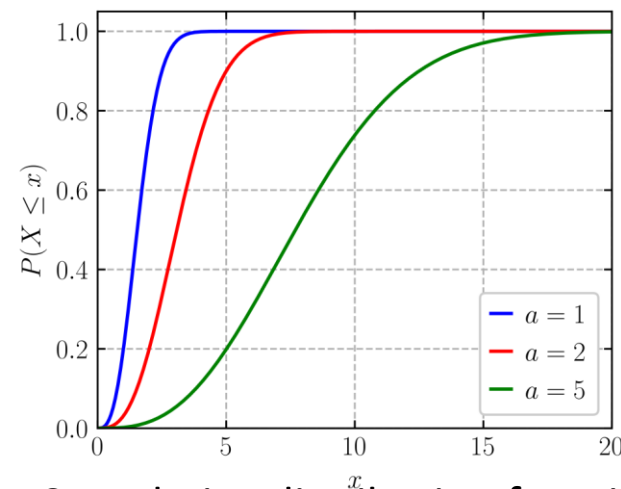
| K | ΔG | Direction |
|--------|------------|---------------------|
| >1.0 | Negative | Proceeds forward |
| 1.0 | Zero | At equilibrium |
| <1.0 | Positive | Proceeds in reverse |

The Boltzmann distribution and statistical thermodynamics

In statistical mechanics and mathematics, a Boltzmann distribution (also called Gibbs distribution) is a probability distribution or probability measure that gives the probability that a system will be in a certain state as a function of that state's energy and the temperature of the system.



Probability density function

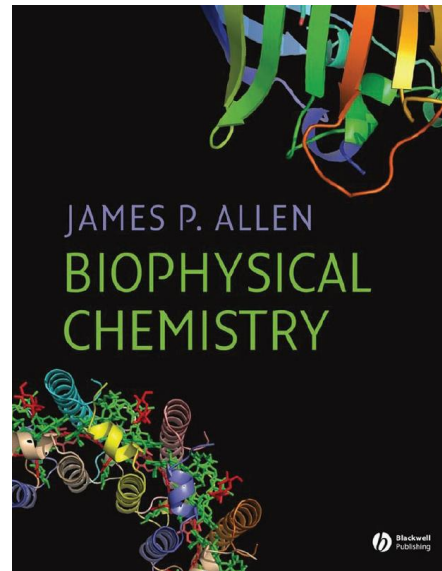


Cumulative distribution function

Review Questions

- Explain the state of equilibrium. Also discuss thermal, chemical and mechanical equilibrium with suitable examples
- Explain Zeroth law of Thermodynamics
- Explain First law of thermodynamics
- Explain Second Law of Thermodynamics

References



Biophysical Chemistry,
James P. Allen
2008 by Blackwell Publishing



Module 1. Molecular basis of life

Topic 4. Biophysical Problems

Lesson 2. Kinetics



Contents

- Introduction
- Order and molecularity
- Kinetics of first and second order reaction
- Pseudo unimolecular reaction
- Arrhenius equation

Introduction

Chemical kinetics is the branch of physical chemistry which deals with a study of the speed of chemical reactions. Such studies also enable us to understand the mechanism by which the reaction occurs. Thus, in chemical kinetics we can also determine the rate of chemical reaction. From the kinetic stand point the reactions are classified into two groups: a) homogeneous reactions which occur entirely in one phase b) heterogeneous reactions where the transformation takes place on the surface of a catalyst or the walls of a container.

Rate of reaction

The rate of reaction i.e. the velocity of a reaction is the amount of a chemical change occurring per unit time. The rate is generally expressed as the decrease in concentration of a reactant or as the increase in concentration of the product. If C the concentration of a reactant at any time t is, the rate is $-dC/dt$ or if the concentration of a product be x at any time t , the rate would be dx/dt .

Rate of reaction

Factors influencing the rate of reaction Rate of a chemical reaction is influenced by the following factors

- (i) Temperature
- (ii) Concentration of the reactants
- (iii) Nature of reactants
- (iv) Catalysts
- (v) Radiation

Rate of reaction

- (i) **Temperature** In most cases, the rate of a reaction in a homogeneous reaction is approximately doubled or tripled by an increase in temperature of only 100 C. In some cases the rise in reaction rates are even higher.
- (ii) **Concentration of the reactants** At a fixed temperature and in the absence of catalyst, the rate of given reaction increases with increased concentration of reactants. With increasing concentration of the reactant 4 the number of molecules per unit volume is increased, thus the collision frequency is increased, which ultimately causes increased reaction rate.
- (iii) **Nature of reactants** A chemical reaction involves the rearrangement of atoms between the reacting molecules to the product. Old bonds are broken and new bonds are formed. Consequently, the nature and the strength of the bonds in reactant molecules greatly influence the rate of its transformation into products. The reaction in which involve lesser bond rearrangement proceeds much faster than those which involve larger bond rearrangement.



Rate of reaction

(iv) **Catalysts** The rate of a chemical reaction is increased in presence of a catalyst which ultimately enhanced the speed of a chemical reaction.

(v) **Radiation** The rate of a number of chemical reactions increases when radiations of specific wave length are absorbed by the reacting molecules. Such reactions are called photochemical reactions. For example, chlorine may be mixed safely with hydrogen in dark, since the reaction between the two is very slow. However when the mixture is exposed to light, the reaction is explosive.

Order of reaction

The order is the number of concentration terms on which reaction rates depends. Thus, if the rate of a reaction depends on the first power of the concentration of reactant, i.e. $\text{Rate} = kC_1$ Thus the reaction is said to be **of the first order**. When the rate is proportional to the product of two reactant concentrations or the square of the concentration of a reactant, the reaction is **of the second order**.

Order of reaction

The molecularity of a reaction is defined as the number of molecules or atoms which take part in the process of a chemical change. The reaction is said to be **unimolecular**, **bimolecular**, **termolecular** according to one, two, or three molecules are involved in the process of a chemical change. The term unimolecular was used for all first order reactions, the term bimolecular for 2nd order reactions etc.

First Order Kinetics

A reaction of the first order is represented as



where X is the reactant and Y the product. The rate of the reaction will be directly proportional to the concentration i.e.,

$$(1.1) - dc/dt = KC$$

in which C is the concentration of the reactant at any time t and K is a constant, called the velocity constant or specific reaction rate. Thus– $dc/C = K dt$

Second Order Kinetics

A reaction will be of the second order when the reaction rate would depend upon the product of two concentrations. Second order reactions are of two types

- (i) The rate is proportional to the square of the same reactant concentration
- (ii) The rate is proportional to the product of the two reactant concentration
- (iii) The rate is proportional to the square of the same reactant concentration

Pseudo unimolecular Reactions

There are a number of reactions, which follow the first order kinetics though more than one kind of reactants is involved in the reaction. Common examples are the inversion of cane sugar or the hydrolysis of an ester in an acid medium.



These are truly second order reactions. Two substances water and cane sugar, or water and ester participate, but the rate of reaction is experimentally observed to depend only on the concentration of cane sugar or ester.

Catalytic Reactions

Definition Catalysis is the process of enhancing the rate of the reaction by means of a foreign substance which remains unchanged in mass and chemical composition. Classification

- i) **Homogeneous Catalysis** Homogeneous Catalysis is where the catalyst and the reactant constitute a single phase.
- ii) **Heterogeneous Catalysis** Heterogeneous Catalysis is where the catalyst and the reactant constitute separate phase.

Catalytic Reactions

Characteristics

- i) The catalyst remains unchanged in mass and in chemical composition at the end of the reaction.
- ii) A very minute quantity of a catalyst can produce an appreciable effect on the speed of a reaction.
- iii) A catalyst can not start a reaction but only increases its speed. The catalyst really provides an alternative path for the transformation in which the required activation energy is less. Lesser activation energy permits larger amounts of reaction in a given time.
- iv) The catalyst does not affect the final state of equilibrium.



Temperature and Reaction Rates

The Arrhenius Equation Temperature has a profound influence on the reaction velocity. In homogenous thermal reactions, for every ten degree rise in temperature, the velocity of reaction is doubled or trebled. The ratio is called **temperature coefficient**.

Temperature and Reaction Rates

Arrhenius (1889) showed that the velocity constant (K) of a chemical process increases exponentially with temperature for a large number of reactions. It was observed that the plot of $\log K$ against $1/T$ gives a linear relation.

He, therefore, suggested empirically the relation as

$$d\ln K/dT = E/RT^2 \text{ or}$$

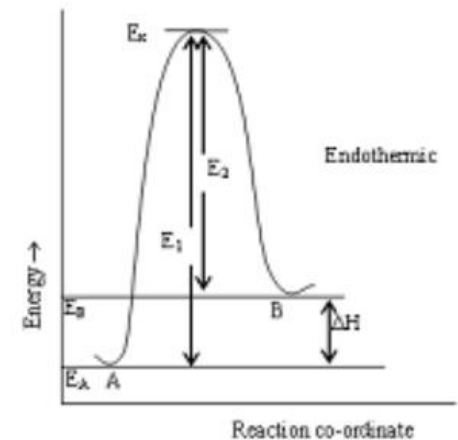
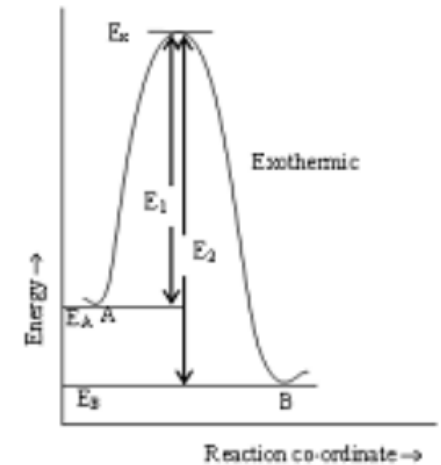
$$K = Ae^{-E/RT} \text{ (A = constant)}$$

A is called the frequency factor or pre exponential factor.

E is called the activation energy of the reaction.

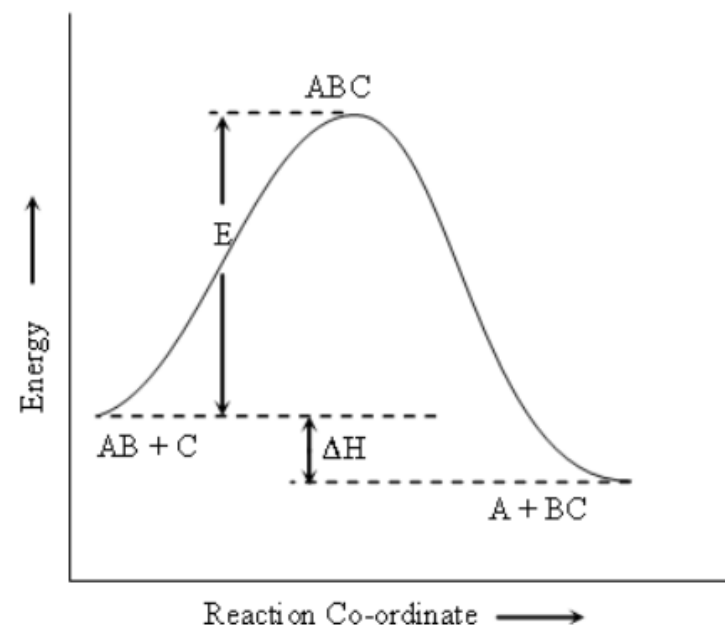
Activation energy of a reaction: its significance

The average energy of the reactant is represented by E_A and that of the resultant by E_B . But if E_A is greater than E_B the reactant A will not be straight way transformed to the product B. There is a minimum energy level for the reaction denoted by E_x to which the reactant molecule must be raised in order to enable it to undergo the chemical change. The excess or additional energy ($E_x - E_A$) which the reactant must acquire in order to undergo transformation is the activation energy E_1 .



Transition state theory

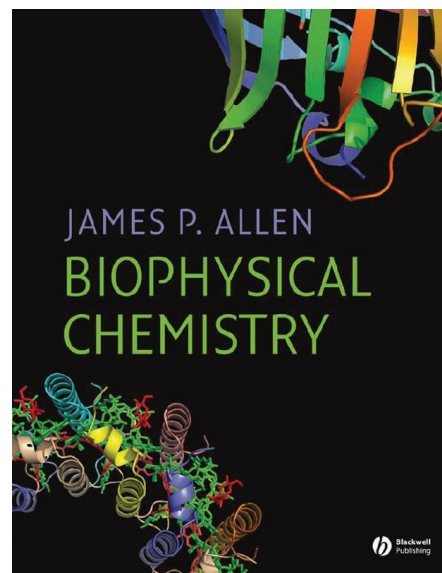
According to this theory, the two reactants say, AB and C first form a transition complex $A\cdots B\cdots C$ which then decomposes into A and BC. The energy necessary to push C to approach B to form this transition state or activated complex is the energy of activation. The observed rate of reaction would be equal to the net rate of formation of the activated complex.



Review Questions

- What is rate of a reaction?
- What is rate constant?
- What is order of a reaction? Give an example of fractional order reaction.
- What is molecularity of a reaction?
- What is the difference between order and molecularity of a chemical reaction?

References



Biophysical Chemistry,
James P. Allen
2008 by Blackwell Publishing



Module 1. Molecular basis of life

Topic 4. Biophysical Problems

Lesson 3. Quantum mechanics



Contents

- Introduction
- Classical concepts
- Principles of quantum theory
- The Schrodinger equation
- Born interpretation



Introduction

To understand the properties of cell membranes, proteins, and nucleic acids, it is necessary to have knowledge of how the molecules forming these biological components respond to the different interactions in their environment..

Classical concepts

classical particle is described by a number of parameters: mass, m , position, r , velocity, v , and charge, q . For simplicity, these are regarded as simple variables rather than as vectors that are needed to describe three-dimensional motion. The kinetic energy of the particle, KE , and the total energy, E , are related to these parameters and the potential energy, V , by:

$$KE = \frac{1}{2}mv^2$$

$$E = KE + V$$

According to classical mechanics, the energy of the system is always conserved. Also conserved is the linear momentum, p , which is equal to:

$$p = mv$$

Classical concepts

Various possible interactions between particles, such as gravitation or electrostatic interactions, are described by well-defined relationships involving the parameters that describe the particles. For example, the electrostatic force, F , between two charged particles, q_1 and q_2 , separated by the distance, r_{12} , is given by:

$$F = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r_{12}^2}$$

where ϵ_0 is a constant, the vacuum permittivity.

Classical concepts

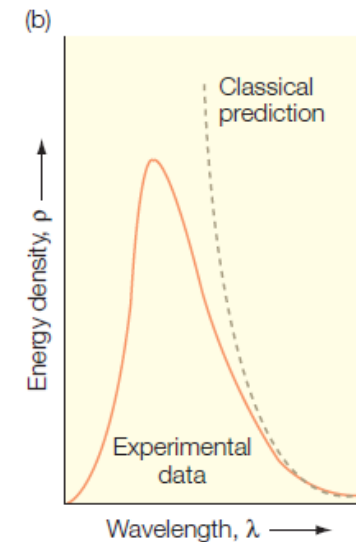
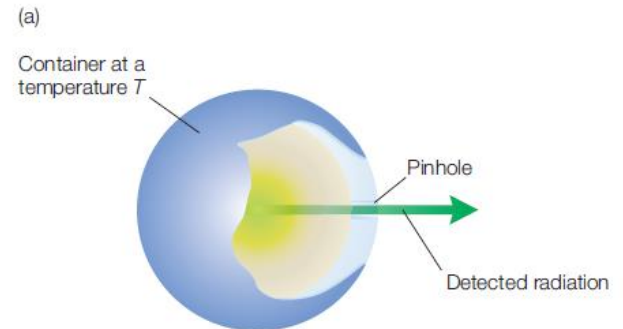
According to classical mechanics, once all of the interactions and the initial conditions have been established, then the time evolution of the system can be predicted for all times using the principle of the conservation of energy, which states that the total energy of the system always remains the same, and using the laws of mechanics, such as the relationship between force, F , mass, m , and position, x :

$$F = ma = m \frac{d^2x}{dt^2}$$

where α is a constant, the vacuum permittivity.

Blackbody radiation

When objects get hot they emit radiation. The type of radiation emitted depends upon the temperature. For example, when you heat something the color can be a red or blue or even white. In most cases, the color emitted at different parts of the object will vary due to differences in the temperature. For an ideal emitter called a blackbody, the emitted radiation is in thermal equilibrium with the object, resulting in a uniform emission of radiation. A blackbody can be modeled as a sphere in which the light emitted by the interior walls is trapped – that is, absorbed and re-emitted – except for a small portion that can escape through a pinhole.



Blackbody radiation

The dependence of ρ on the wavelength was determined for many materials and was found to always follow the same general dependence. The dependence of the energy distribution can be modeled using statistical arguments for classical thermodynamics. Assuming that the radiated energy follows the classical dependence on the amplitude and is independent of the wavelength, the long-wavelength part of the dependence is predicted but fails for short wavelengths. Instead of predicting that the density will drop to zero, the theory predicts the so-called ultraviolet catastrophe: that the energy density will become infinite:

$$\rho = \frac{8\pi kT}{\lambda^4}$$

Blackbody radiation

Using statistical arguments but with the different dependence for the energy, Planck derived a new dependence for the energy density that agreed with the experimental data:

$$\rho = \frac{8\pi hc}{\lambda^5} \left(\frac{1}{e^{hc/\lambda kT} - 1} \right)$$

At long wavelengths, this equation agrees with the classical prediction. At long wavelengths the exponential term is very small and the exponential can be written approximately as:

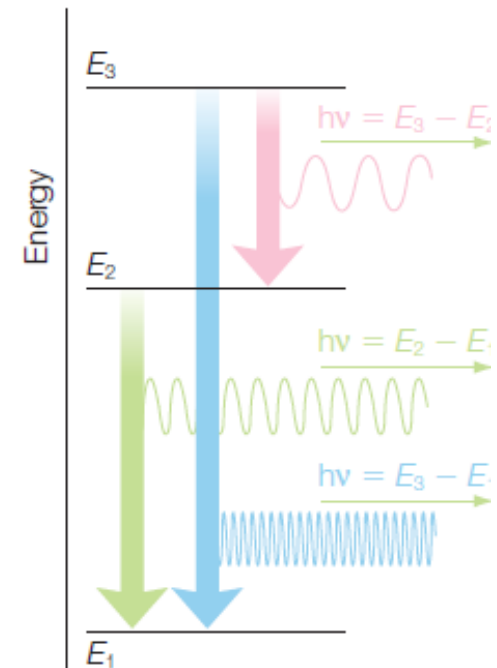
$$e^x - 1 \approx x \text{ when } x \ll 1$$

Using this approximation, the energy density can be written as:

$$\rho \approx \frac{8\pi hc}{\lambda^5} \left(\frac{hc}{\lambda kT} \right)^{-1} = \frac{8\pi kT}{\lambda^4}$$

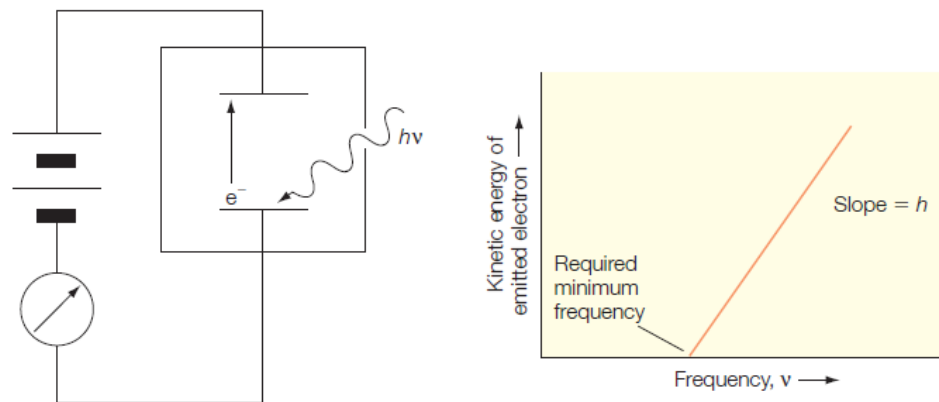
Photoelectric effect

When light of certain wavelengths strikes a metal surface, an electron is ejected from the metal in the photoelectric effect. Classically, the kinetic energy of the ejected electron should be related to the light intensity, or amplitude squared, and is independent of the frequency used. Experimentally it is possible to measure this property for different metals and different light conditions (Figure 9.3). It is found that no electrons are ejected regardless of the intensity if the light frequency is below a certain value that is characteristic of the metal. For light above this critical frequency, the kinetic energy is linearly dependent upon the frequency, and the intensity only changes the number of electrons ejected.



Atomic spectra

When objects are heated they emit light that is characteristic for each element. Experimentally, the emitted light is observed to be not continuous but discrete. This feature suggests that the energy associated with the atoms is discrete or quantized. In classical physics, there is no reason for this property of quantized light emission.

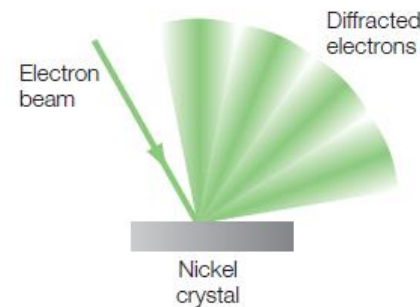


Principles of quantum theory

Wave–particle duality

Classically, particles and waves have distinct parameters:

- Particles: mass m , position r , velocity v , charge q
- Waves: wavelength λ , frequency ν , velocity v , amplitude A



Principles of quantum theory

Schrödinger's equation

Classically, the energy of a particle is given by the sum of the kinetic and potential energies:

$$E = \frac{1}{2}mv^2 + V(x) = \frac{p^2}{2m} + V(x)$$

In quantum mechanics this expression is modified with the introduction of operators, as provided in Table. Some of these operators involve \hbar and i , which are defined as:

$$\hbar = \frac{h}{2\pi} \quad \text{and} \quad i = \sqrt{-1}$$

Physical variables and the corresponding quantum operators.

| Variable | Operator |
|----------|---------------------------------------|
| X | x |
| V | V |
| p_x | $-i\hbar \frac{\partial}{\partial x}$ |
| t | t |
| E | $i\hbar \frac{\partial}{\partial t}$ |

$$E\psi(r) = -\frac{\hbar^2}{2m}\nabla^2\psi(r) + V(r)\psi(r)$$

Principles of quantum theory

Born interpretation

The physical interpretation of quantum mechanics, in particular the interpretation of the wavefunction, was developed by many scientists, most notably Max Born. Since all particles are also waves, particles are always distributed in space. The wavefunction in Schrödinger's equation has no direct physical meaning and can be a complex function rather than a real function. In the Born interpretation, the probability of finding any particle at a particular location is not given by the wavefunction itself but rather by the square of the wavefunction. These ideas led to several fundamental postulates of quantum mechanics, as follows.

Principles of quantum theory

Born interpretation

1. A particle is never at a specific location but only has a probability of being there. The probability of finding a particle at a specific position is given by the square of the wavefunction times the volume $d\tau$ as:

$$\psi^*(r)\psi(r)d\tau$$

where $d\tau = dx dy dz = r^2 \sin \theta dr d\vartheta d\phi$

In this equation ψ^* is the complex conjugate. Since any wavefunction ψ can be written in terms of two real functions, A and B, the conjugate can be defined as:

$$\psi = A + iB \text{ and } \psi^* = A - iB$$

The probability of finding a particle within a volume V is then:

$$\int_V \psi^*(r)\psi(r) d\tau$$

where $\psi^*(x)$ is the complex conjugate of the wavefunction

Principles of quantum theory

Born interpretation

2. The wavefunction may be a complex function but the probability is

always real, since:

$$\psi(x) = A(x) + iB(x)$$

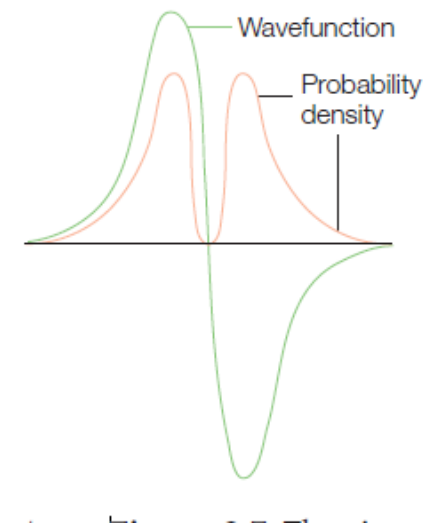
$$\psi^*(x) = A(x) - iB(x)$$

$$\text{so } \psi^*(x)\psi(x) = [A(x) + iB(x)][A(x) - iB(x)] = A^2(x) + B^2(x) \geq 0$$

Principles of quantum theory

Born interpretation

3. The probability is always a positive and real number. The total probability of finding an object anywhere in space must be equal to one. The sum of all of the probabilities is mathematically written as the integral of the probability. Therefore, the integral of the probability over all space must be equal to one:



$$1 = \int_0^{\infty} \psi^*(r) \psi(r) d\tau$$

Principles of quantum theory

Born interpretation

4. Particles do not have a specific position or momentum but rather there is a distribution of values that reflect the distribution of the particle. The physically relevant quantity is the average, or expectation, value. Every physical observable p has an associated operator and the average, or expectation, value of the observable is given by:

$$\langle p \rangle = \int \psi^*(r) \hat{p} \psi(r) d\tau$$

where V is the operator.

For example, the average values of the position and momentum of a particle can be calculated by substituting the operators for position r and x component of the momentum:

$$\langle r \rangle = \int \psi^*(r) r \psi(r) d\tau$$

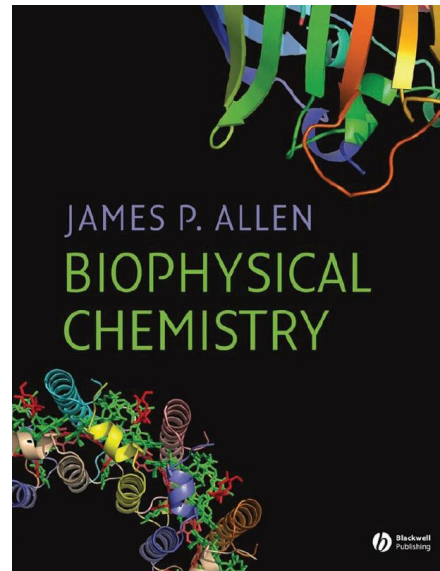
$$\langle x^2 \rangle = \int \psi^*(x) x^2 \psi(x) dx$$

$$\langle p_x \rangle = \int \psi^*(r) \left(-i\hbar \frac{\partial}{\partial x} \right) \psi(r) d\tau$$

Review Questions

- Why is the observed blackbody radiation in conflict with classical physics?
- Why does the presence of discrete lines in the emission spectra of atoms support the ideas of quantum theory?
- Why is there a minimal frequency needed to cause the photoelectric effect?
- What is the probability of finding a particle within a volume V_0 ?

References



Biophysical Chemistry,
James P. Allen
2008 by Blackwell Publishing



Module 1. Molecular basis of life

Topic 4. Biophysical Problems

Lesson 4. Spectroscopy



Contents

- Introduction
- Infrared spectroscopy
- Mass spectrometry
- Nuclear magnetic resonance spectroscopy
- Ultraviolet spectroscopy

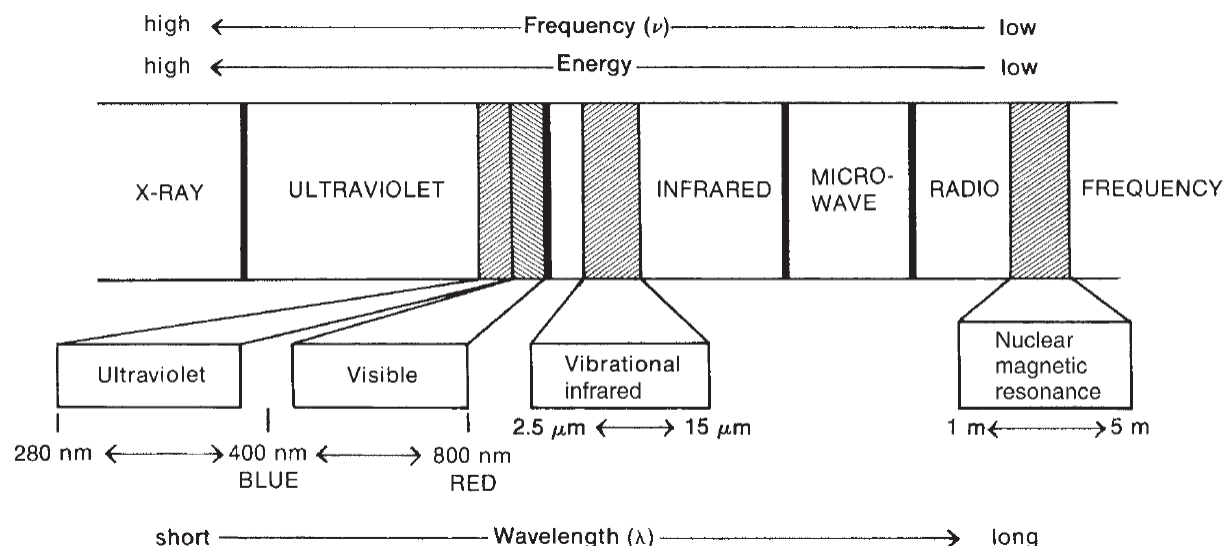
Introduction

Spectroscopy is basically an experimental subject and is concerned with the absorption, emission or scattering of electromagnetic radiation by atoms or molecules. Electromagnetic radiation covers a wide wavelength range, from radio waves to γ -rays, and the atoms or molecules may be in the gas, liquid or solid phase or, of great importance in surface chemistry, adsorbed on a solid surface.

Experimental methods of spectroscopy began in the more accessible visible region of the electromagnetic spectrum where the eye could be used as the detector.

Infrared spectroscopy

Almost any compound having covalent bonds, whether organic or inorganic, absorbs various frequencies of electromagnetic radiation in the infrared region of the electromagnetic spectrum. This region lies at wavelengths longer than those associated with visible light, which range from approximately 400 to 800 nm ($1 \text{ nm} = 10^{-9} \text{ m}$), but lies at wavelengths shorter than those associated with microwaves, which are longer than 1 mm. For chemical purposes, we are interested in the vibrational portion of the infrared region. It includes radiation with wavelengths (λ) between 2.5 μm and 25 μm ($1 \text{ mm} = 10^{-3} \text{ m}$). Although the more technically correct unit for wavelength in the infrared region of the spectrum is the micrometer (μm), you will often see the micron (m) used on infrared spectra. Figure illustrates the relationship of the infrared region to others included in the electromagnetic spectrum.



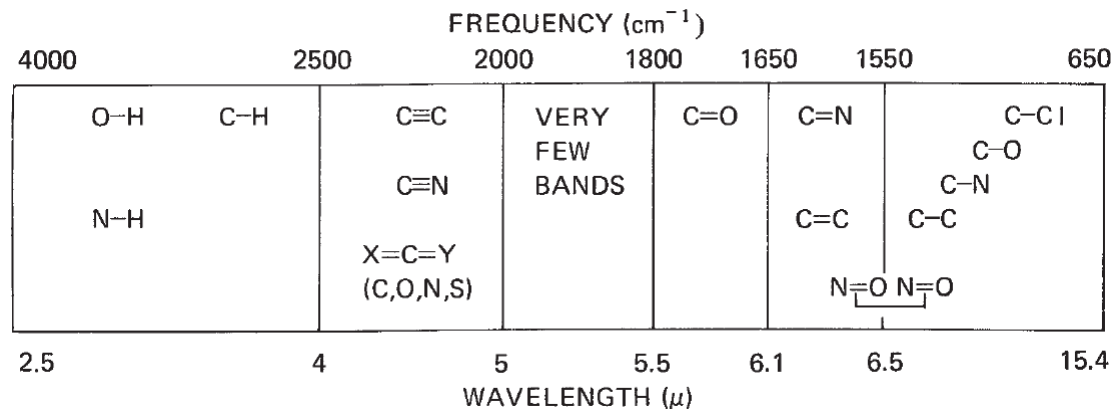
Infrared spectroscopy

The absorption of infrared radiation corresponds to energy changes on the order of 8 to 40 kJ/mole. Radiation in this energy range corresponds to the range encompassing the stretching and bending vibrational frequencies of the bonds in most covalent molecules. In the absorption process, those frequencies of infrared radiation that match the natural vibrational frequencies of the molecule in question are absorbed, and the energy absorbed serves to increase the amplitude of the vibrational motions of the bonds in the molecule. Note, however, that not all bonds in a molecule are capable of absorbing infrared energy, even if the frequency of the radiation exactly matches that of the bond motion. Only those bonds that have a dipole moment that changes as a function of time are capable of absorbing infrared radiation. Symmetric bonds, such as those of H₂ or Cl₂, do not absorb infrared radiation. A bond must present an electrical dipole that is changing at the same frequency as the incoming radiation for energy to be transferred.

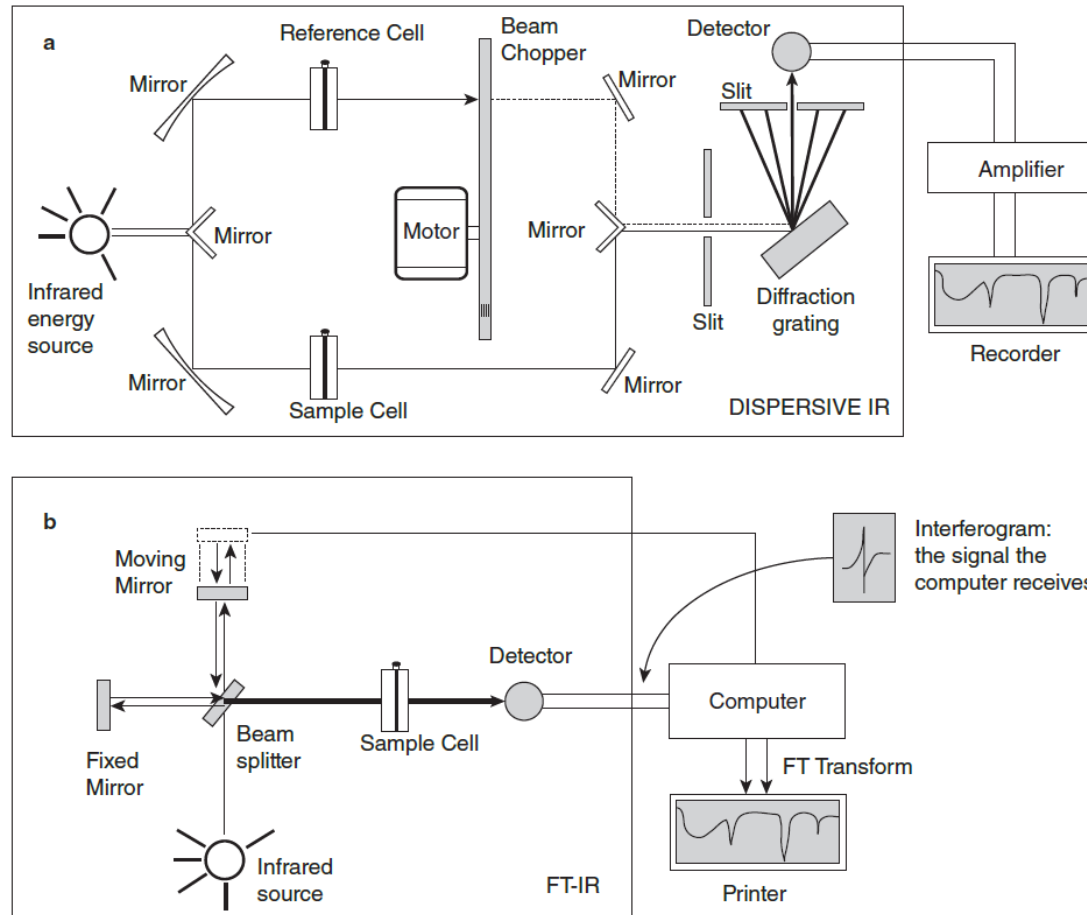
| Region of Spectrum | Energy Transitions |
|---------------------|--|
| X-rays | Bond breaking |
| Ultraviolet/visible | Electronic |
| Infrared | Vibrational |
| Microwave | Rotational |
| Radiofrequencies | Nuclear spin (nuclear magnetic resonance) Electronic spin (electron spin resonance) |

Uses of the infrared spectrum

- The infrared spectrum can be used for molecules much as a fingerprint can be used for humans.
- More important use of the infrared spectrum is to determine structural information about a molecule.

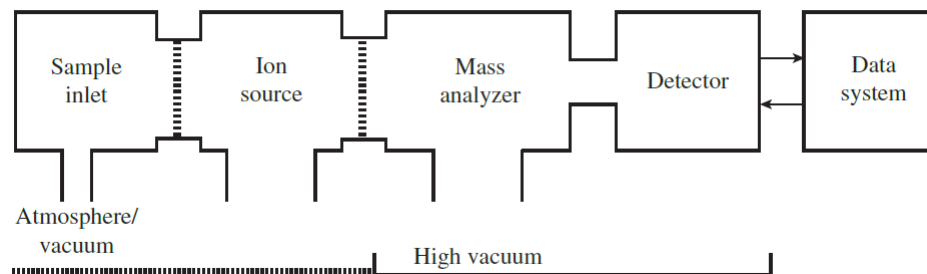


Schematic diagrams of (a) dispersive and (b) Fourier transform infrared spectrophotometers



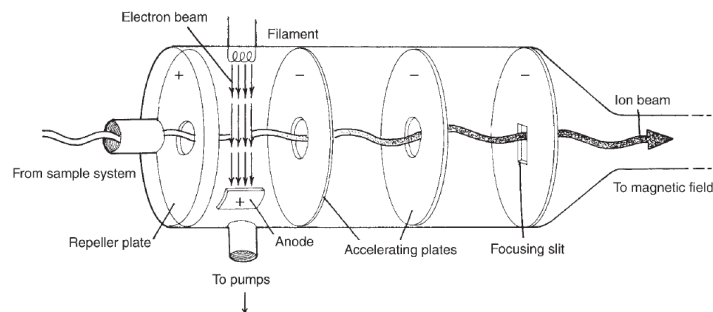
Mass spectrometry

In its simplest form, the mass spectrometer has five components. The first component of the mass spectrometer is the sample inlet, which brings the sample from the laboratory environment (1 atm) to the lower pressure of the mass spectrometer. Pressures inside the mass spectrometer range from a few millimeters of mercury in a chemical ionization source to a few micrometers of mercury in the mass analyzer and detector regions of the instrument. The sample inlet leads to the ion source, where the sample molecules are transformed into gas phase ions. Some instruments have been developed recently that combine the sample inlet and ion source under ambient conditions, thereby greatly simplifying sample preparation. The ions are then accelerated by an electromagnetic field. Next, the mass analyzer separates the sample ions based on their mass-to-charge (m/z) ratio. The ions then are counted by the detector, and the signal is recorded and processed by the data system, typically a personal computer (PC). The output from the data system is the mass spectrum—a graph of the number of ions detected as a function of their m/z ratio.



Ionization methods

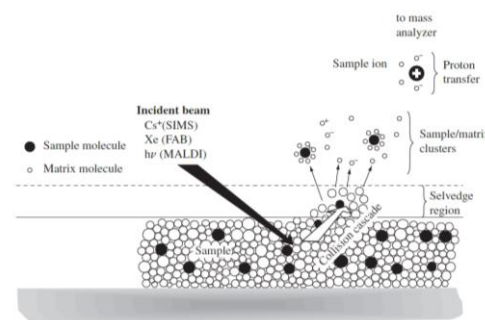
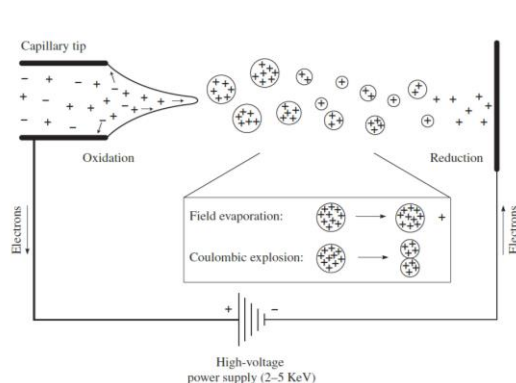
Electron Ionization (EI)



Chemical Ionization (CI)

Desorption Ionization Techniques (SIMS, FAB, and MALDI)

Electrospray Ionization (ESI)



Mass analysis

The Magnetic Sector Mass Analyzer

Double-Focusing Mass Analyzers

Quadrupole Mass Analyzers

Time-of-Flight Mass Analyzers

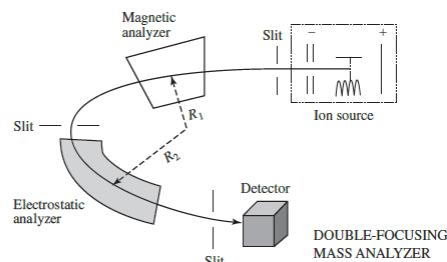
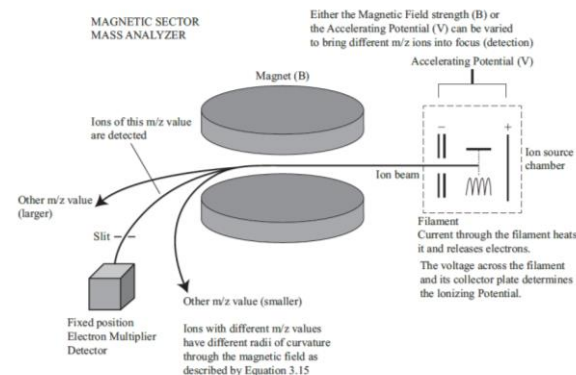
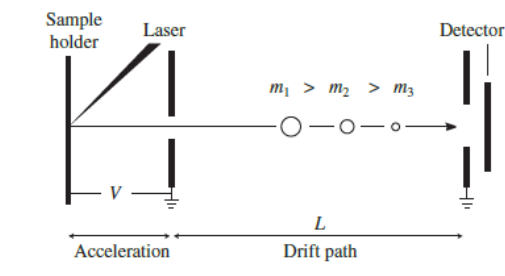
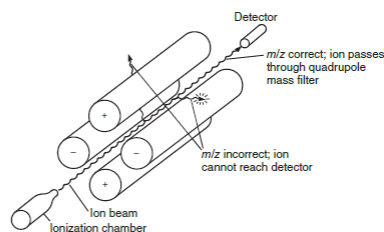
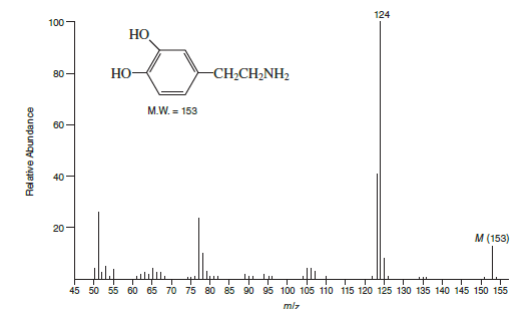
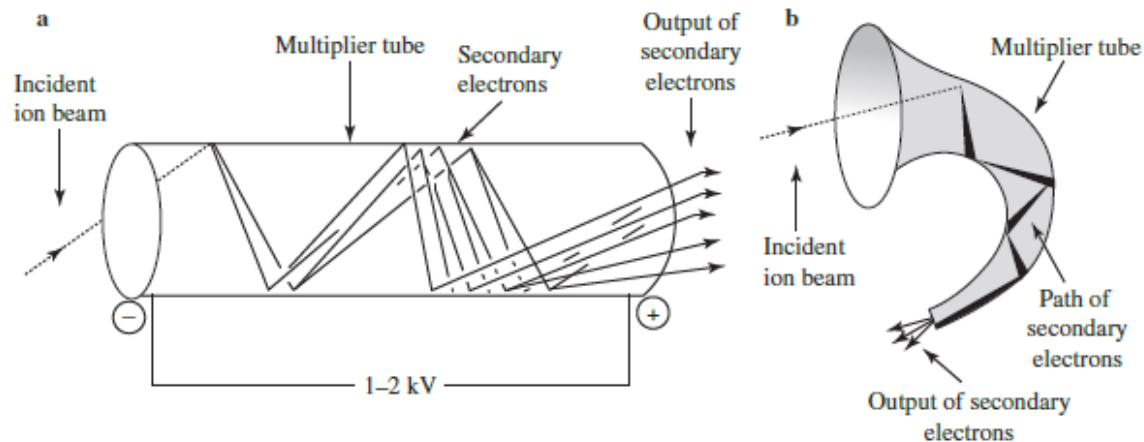


FIGURE 3.12 Schematic of a double-focusing mass analyzer.



Detection and quantitation: the mass spectrum



ERASMUS+

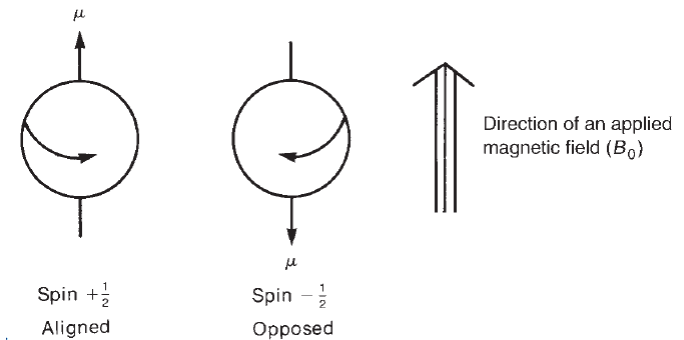
Nuclear magnetic resonance spectroscopy

Nuclear magnetic resonance (NMR) is a spectroscopic method that is even more important to the organic chemist than infrared spectroscopy. Many nuclei may be studied by NMR techniques, but hydrogen and carbon are most commonly available. Whereas infrared (IR) spectroscopy reveals the types of functional groups present in a molecule, NMR gives information about the number of magnetically distinct atoms of the type being studied. When hydrogen nuclei (protons) are studied, for instance, one can determine the number of each of the distinct types of hydrogen nuclei as well as obtain information regarding the nature of the immediate environment of each type. Similar information can be determined for the carbon nuclei. The combination of IR and NMR data is often sufficient to determine completely the structure of an unknown molecule.

| Element | ^1_1H | ^2_1H | $^{12}_6\text{C}$ | $^{13}_6\text{C}$ | $^{14}_7\text{N}$ | $^{16}_8\text{O}$ | $^{17}_8\text{O}$ | $^{19}_9\text{F}$ | $^{31}_{15}\text{P}$ | $^{35}_{17}\text{Cl}$ |
|-----------------------------|----------------|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|----------------------|-----------------------|
| Nuclear spin quantum number | $\frac{1}{2}$ | 1 | 0 | $\frac{1}{2}$ | 1 | 0 | $\frac{5}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{3}{2}$ |
| Number of spin states | 2 | 3 | 0 | 2 | 3 | 0 | 6 | 2 | 2 | 4 |

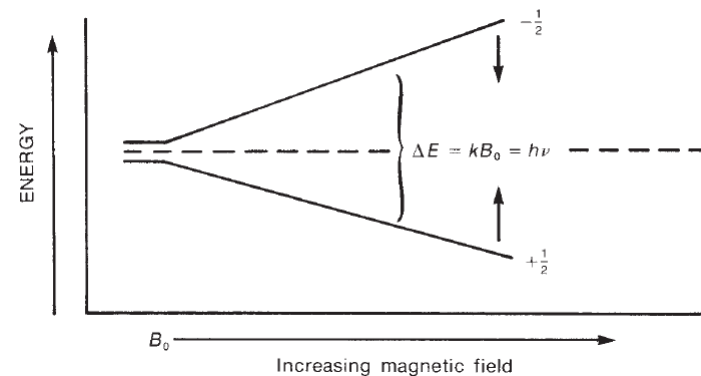
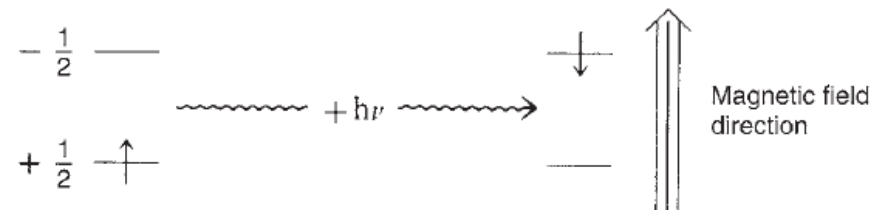
Nuclear magnetic moments

Spin states are not of equivalent energy in an applied magnetic field because the nucleus is a charged particle, and any moving charge generates a magnetic field of its own. Thus, the nucleus has a magnetic moment μ generated by its charge and spin. A hydrogen nucleus may have a clockwise or counterclockwise spin, and the nuclear magnetic moments (μ) in the two cases are pointed in opposite directions.



Absorption of energy

The nuclear magnetic resonance phenomenon occurs when nuclei aligned with an applied field are induced to absorb energy and change their spin orientation with respect to the applied field. The energy absorption is a quantized process, and the energy absorbed must equal the energy difference between the two states involved.



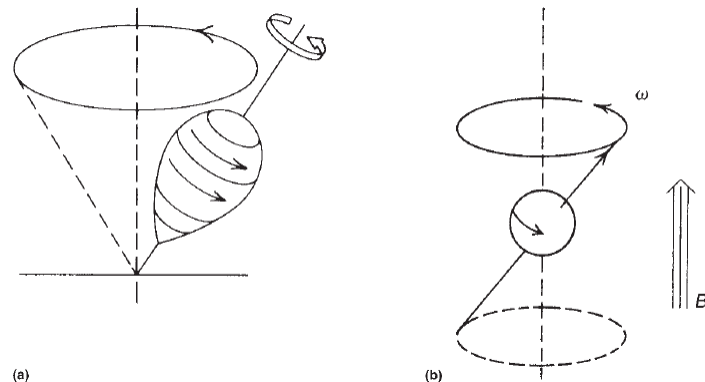
Frequencies and field strengths at which selected nuclei have their nuclear resonances

| Isotope | Natural Abundance (%) | Field Strength, B_0 (Tesla ^a) | Frequency, ν (MHz) | Magnetogyric Ratio, γ (radians/Tesla) |
|-----------------|-----------------------|---|------------------------|--|
| ¹ H | 99.98 | 1.00 | 42.6 | 267.53 |
| | | 1.41 | 60.0 | |
| | | 2.35 | 100.0 | |
| | | 4.70 | 200.0 | |
| | | 7.05 | 300.0 | |
| ² H | 0.0156 | 1.00 | 6.5 | 41.1 |
| ¹³ C | 1.108 | 1.00 | 10.7 | 67.28 |
| | | 1.41 | 15.1 | |
| | | 2.35 | 25.0 | |
| | | 4.70 | 50.0 | |
| | | 7.05 | 75.0 | |
| ¹⁹ F | 100.0 | 1.00 | 40.0 | 251.7 |
| ³¹ P | 100.0 | 1.00 | 17.2 | 108.3 |

^a 1 Tesla = 10,000 Gauss.

The mechanism of absorption (resonance)

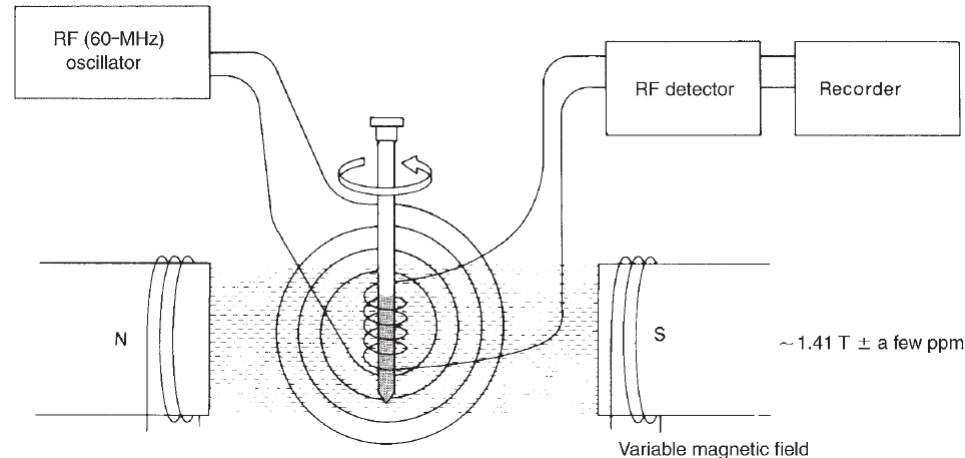
To understand the nature of a nuclear spin transition, the analogy of a child's spinning top is useful. Protons absorb energy because they begin to precess in an applied magnetic field. The phenomenon of precession is similar to that of a spinning top. Owing to the influence of the earth's gravitational field, the top begins to “wobble,” or precess, about its axis. A spinning nucleus behaves in a similar fashion under the influence of an applied magnetic field.



The nuclear magnetic resonance spectrometer

The Continuous-Wave (CW) Instrument

The Pulsed Fourier Transform (FT) Instrument

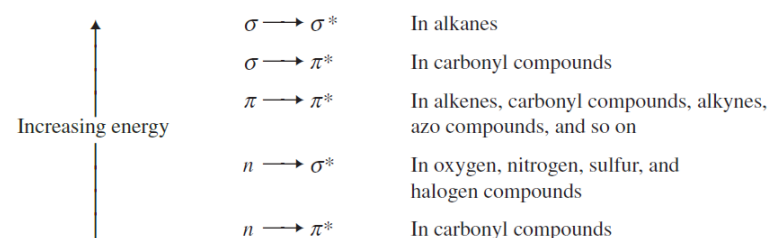
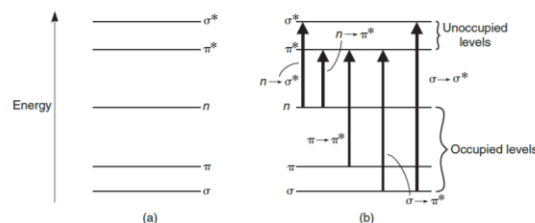
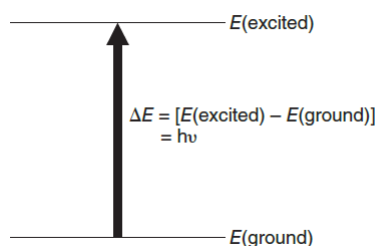


Ultraviolet spectroscopy

Most organic molecules and functional groups are transparent in the portions of the electromagnetic spectrum that we call the ultraviolet (UV) and visible (VIS) regions—that is, the regions where wavelengths range from 190 nm to 800 nm. Consequently, absorption spectroscopy is of limited utility in this range of wavelengths. However, in some cases we can derive useful information from these regions of the spectrum. That information, when combined with the details provided by infrared and nuclear magnetic resonance (NMR) spectra, can lead to valuable structural proposals.

The nature of electronic excitations

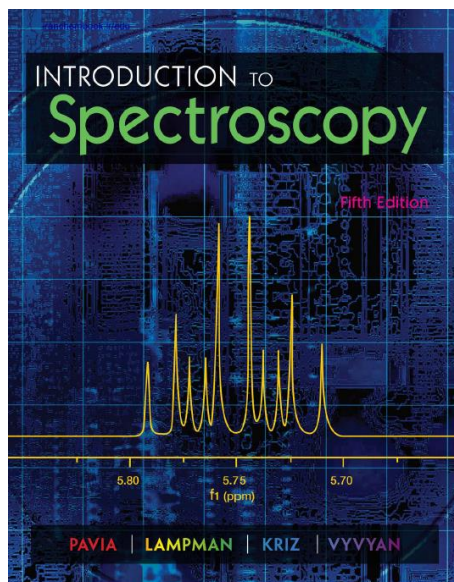
When continuous radiation passes through a transparent material, a portion of the radiation may be absorbed. If that occurs, the residual radiation, when it is passed through a prism, yields a spectrum with gaps in it, called an absorption spectrum. As a result of energy absorption, atoms or molecules pass from a state of low energy (the initial, or ground state) to a state of higher energy (the excited state). Figure depicts this excitation process, which is quantized. The electromagnetic radiation that is absorbed has energy exactly equal to the energy difference between the excited and ground states.



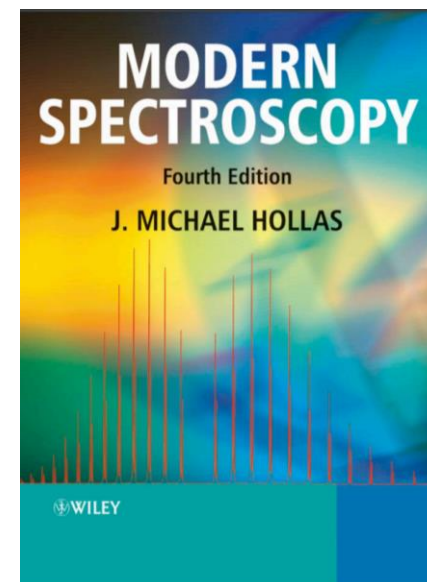
Review Questions

- Why is the observed blackbody radiation in conflict with classical physics?
- Why does the presence of discrete lines in the emission spectra of atoms support the ideas of quantum theory?
- Why is there a minimal frequency needed to cause the photoelectric effect?
- What is the probability of finding a particle within a volume V_0 ?

References



Introduction to Spectroscopy,
Fifth Edition
Donald L. Pavia, Gary M. Lampman,
George S. Kriz, and James R. Vyvyan
2015, 2009 Cengage Learning



Modern spectroscopy
Fourth Edition
J. Michael Hollas
2004 by John Wiley & Sons Ltd, The Atrium,
Southern Gate



Module 1. Molecular basis of life

Topic 4. Biophysical Problems

Lesson 5. Understanding biological systems using physical chemistry



Contents

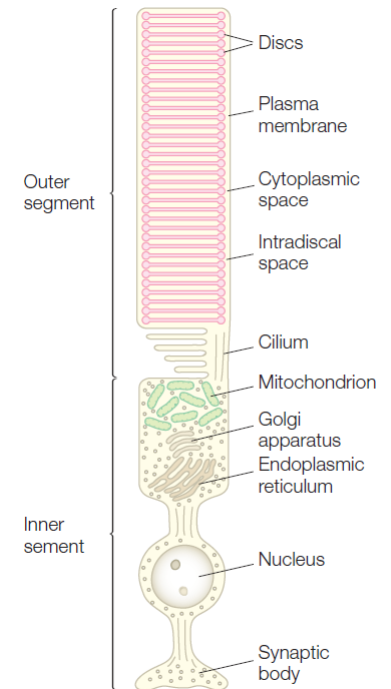
- Introduction
- Signal transduction
- Membrane potentials, transporters, and channels
- Molecular imaging
- Photosynthesis

Introduction

Biophysical chemistry is an interdisciplinary field, combining elements of biology, chemistry, and physics. It is aimed at the collection and analysis of quantitative data for predictive physical models of biological phenomena at the molecular level. It is also used to describe techniques that study the physical properties of important biological molecules at the chemical level.

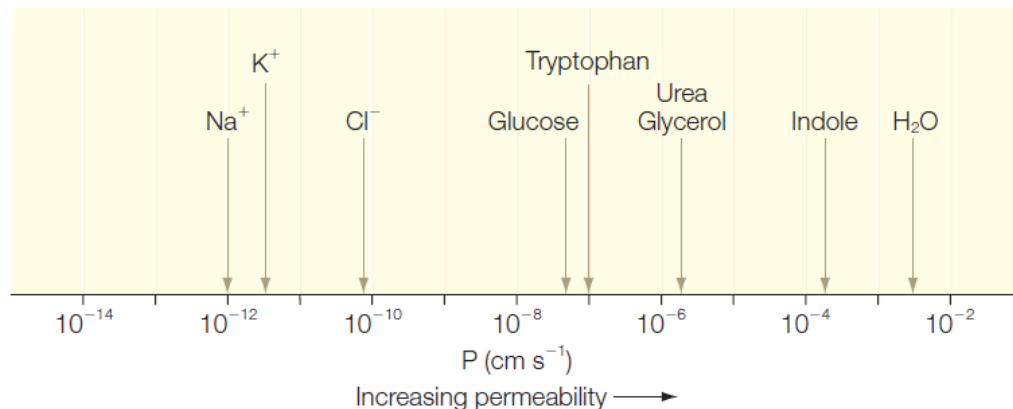
Signal transduction

The visual process can be divided into four steps: recognition, conversion, amplification, and processing. The recognition step in vision is unusual for biological signal transduction as it involves the absorption of light by a pigment buried inside of a protein called rhodopsin rather than the binding of a signal molecule to a protein receptor, as found in other signal-transduction processes. After recognition, there is a signal conversion, which requires a structural change of the protein and an associated molecule called retinal in response to light absorption. Once the signal has activated rhodopsin, it is amplified by many orders of magnitude, allowing the signal to be processed into a change in membrane potential and consequently a signal to the brain.



Membrane potential

The cell membrane is a lipid bilayer. The composition of cell membranes varies among cell types and growth conditions, but in general the major component is phospholipids. Phospholipids are amphipathic molecules; that is, they have both a polar head group and a hydrophobic fatty acid portion. Because of the central hydrophobic core of the bilayer, the cell membrane is largely impermeable to ions and polar molecules. In contrast, water traverses membranes much more readily. The permeability of small molecules ranges over several orders of magnitude and is correlated with the solubility of each molecule in nonpolar solvents relative to water. The cell makes use of this ability to control the relative concentrations of ions to build ion gradients across the cell membrane for many metabolic processes.

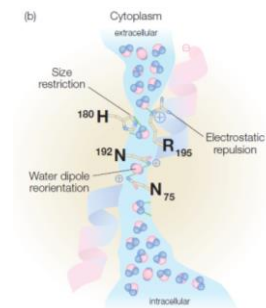
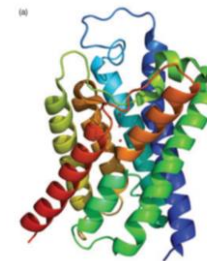
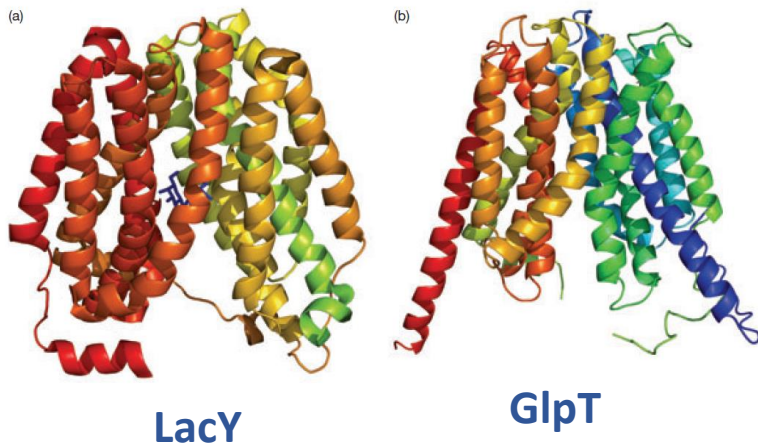


Energetics of transport across membranes

The transport of ions depends upon the change of free energy of the transported ion. Active transport requires a coupled input of free energy. This process is energetically unfavorable and will not occur spontaneously. The transport could be coupled to ATP hydrolysis but alternatively it could be coupled to the transport of another molecule across the membrane. This often occurs in cell membranes, in proteins known as antiporters, which can simultaneously transport two different molecules in opposite directions.

Transporters

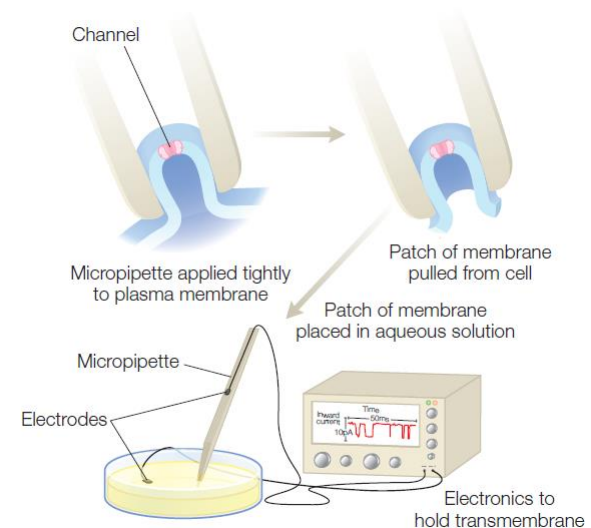
Transport proteins selectively mediate the passage of molecules across the membrane, which is otherwise impermeable. More than 360 transporter families have been identified, highlighting the critical role of transport processes in cells.



The aquaporins. (a) The threedimensional structure of aquaporin. (b) A schematic representation of the water channel of aquaporin.

Ion channels

Cell membranes possess ion channels that are proteins designed to transport specific ions across the cell membrane. Ion channels can be distinguished from ion transporters by certain characteristics. Channels can transport ions at a significantly faster rate than transporters. Also, the rate of ion transport in a channel is usually gated; that is, the channel is open or closed depending on another factor. The controlling factors are the binding of a specific ligand for ligand-gated channels and the membrane potential for voltage-gated channels.



The patch-clamp technique for the measurement of an individual ion channel.

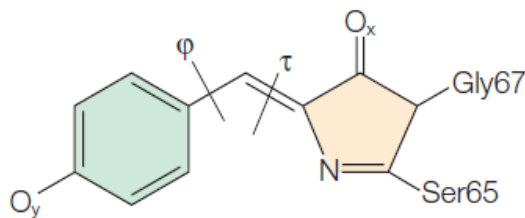
Molecular imaging

Fluorescence has been used for many years to visualize cellular components. Originally, small organic dyes were attached through the use of antibodies and by the use of fluorophores that recognized organelles. More recently, fluorescent proteins have provided the opportunity to probe gene expression, protein trafficking, and responses to signals. Spectroscopy using fluorescence has expanded rapidly as these tools have become more commonplace, allowing scientists to target specific sites even at the singlemolecule level.

Green fluorescent protein

Green fluorescent protein (GFP) was discovered as a companion protein to aequorin, a chemiluminescent protein from the jellyfish *Aequorea*, and was found to be composed of a single polypeptide with 238 amino acid residues. GFP emits a green fluorescence when irradiated with UV light. The proper expression of the chromophore of GFP was found to require molecular oxygen but to be independent of any enzymes. Biosynthesis of the chromophore was found to be an exception compared to most chromophore-containing proteins as the protein was found to catalyze the synthesis of the pigment from the polypeptide chain.

In most cases, the gene for GFP can be attached to a gene of interest and the resulting protein will be fluorescent, allowing tracking of genes using molecular biology.



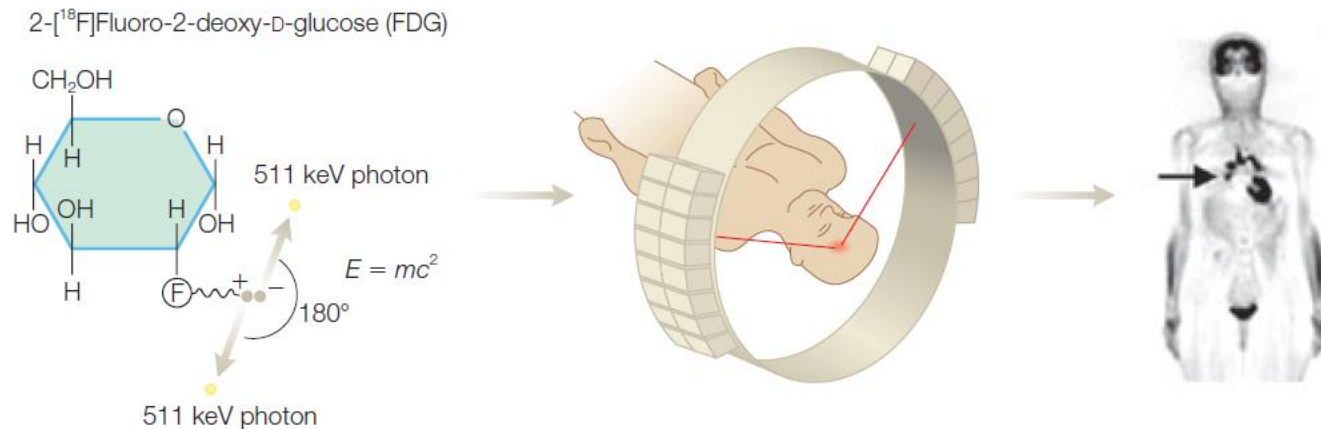
Structure of the chromophore of GFP formed after cyclization of the amino acid residues Ser-65, Tyr-66, and Gly-67.

Imaging in organisms

Many of the spectroscopic techniques, such as MRI and fluorescence, are being used in clinical settings to visualize metabolic processes in the human body. Another imaging technique, positron emission tomography (PET), is being considered increasingly for patient treatment in the examination of molecular processes and their failure in disease. In this technique, a certain process is targeted and a suitable probe designed. The probes sensed in PET are drugs or analogs that have been labeled with radioisotopes. The measurements are sensitive so that very low doses can be applied to minimize unforeseen side effects. As the name implies, this technique is tomography-based, which means that the resulting information is measured in three dimensions so that it can be mapped onto the body.

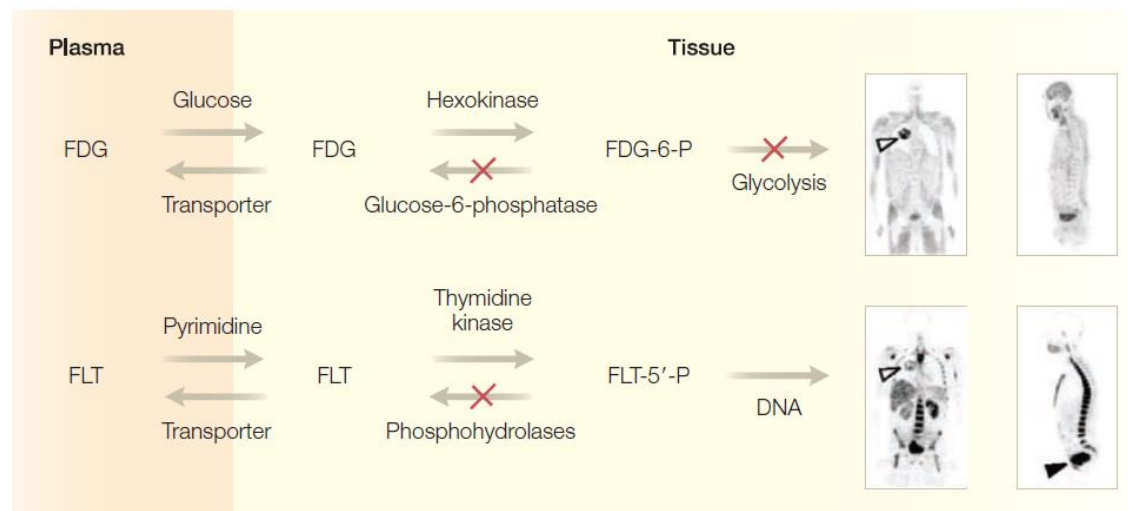
Radioactive decay

There are several different radioactive processes that certain isotopes of elements can undergo. To understand these processes it is necessary to keep track of the specific isotopes involved. For the decay processes each isotope is denoted by the mass number, which equals the number of protons and neutrons, and the atomic number, which equals the number of protons.

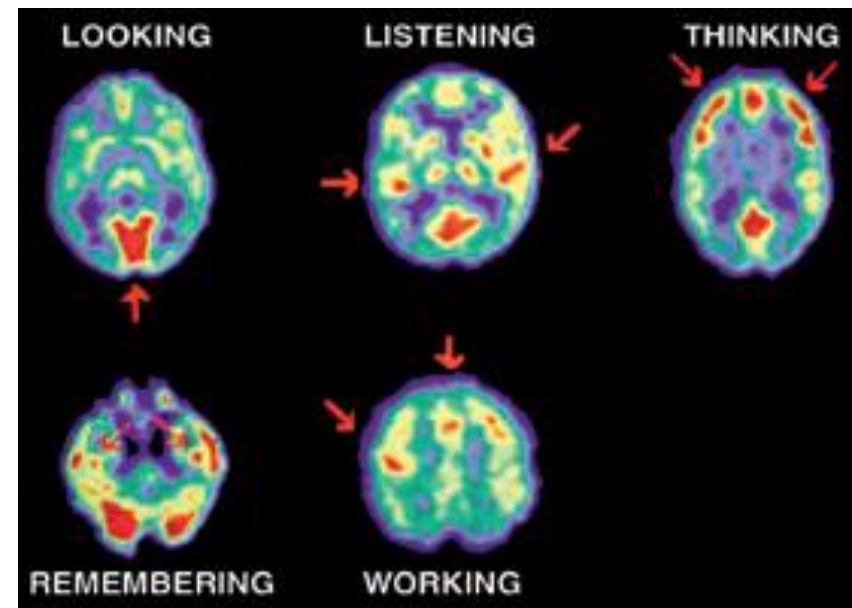


PET

The technique of PET is an imaging technique that makes use of positronemitting isotopes as probes of specific biochemical processes *in vivo*. In PET, the subject is injected with a molecule containing a label with an isotope that will emit positrons.

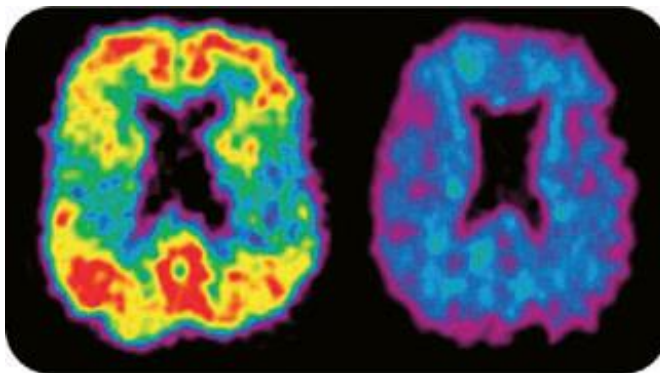


In addition to these clinical uses, PET can be used to characterize the stimulation responses of the brain. For example, specific areas of the brain are found to have high glucose metabolism in response to different sensory stimulations. PET provides a relatively benign approach for studying brain activity and to examine potential therapies. For example, by linking a PET reporter gene to a potential therapeutic gene, it is possible to track where the therapeutic gene is being translated and to correlate the gene activity with a physiological response.



Parkinson's disease

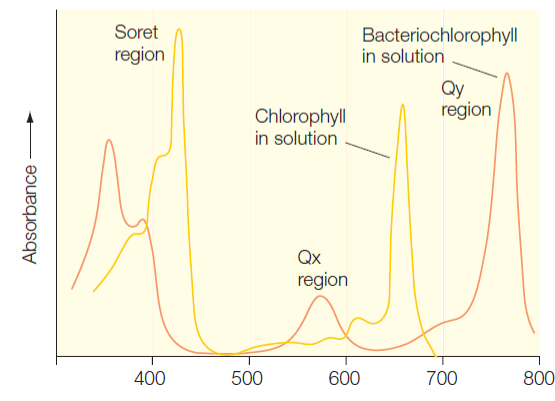
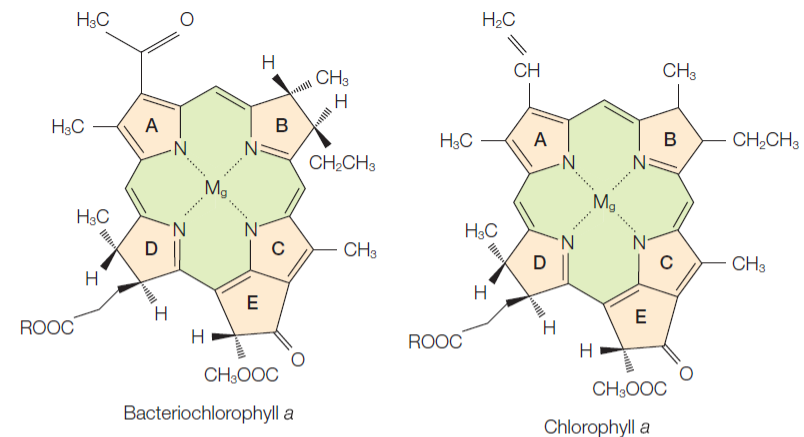
Parkinson's disease is the second most common neurodegenerative disease after Alzheimer's disease. The first warning sign is development of a tremor in one arm. As the disease progresses, voluntary movement becomes slower with some patients experiencing temporary loss of the ability to move.



Compared to a healthy person (right), a patient with Parkinson's disease (left) will have a deterioration in dopamine-transporter activity, which is evident in PET scans.

Photosynthesis

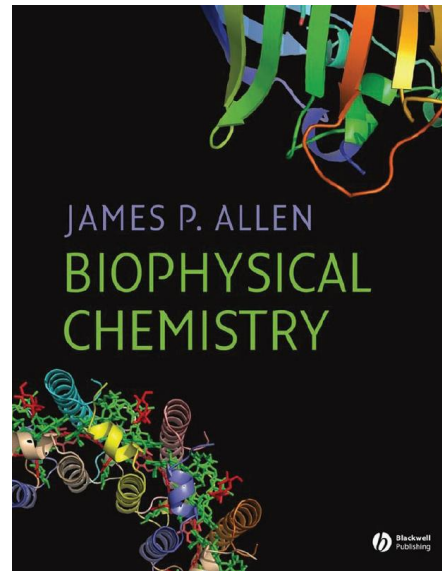
Photosynthesis is the biological process by which the energy of the sun is converted into energy-rich compounds that are used to drive cellular processes.



Review Questions

- The lipid bilayer is permeable to what type of small molecules?
- What is the biological role of transporters?
- How can PET be used to monitor glucose metabolism?
- How can PET be used with Alzheimer's and related diseases?

References



Biophysical Chemistry,
James P. Allen
2008 by Blackwell Publishing

Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 1. DNA: The Genetic Material, Experimental Proof of the Genetic, Function of DNA, Genetic Role of DNA in Bacteriophage



Contents

- Introduction
- Experimental Proof of the Genetic Function of DNA
- Genetic Role of DNA in Bacteriophage



Introduction

Each species of living organism has a unique set of inherited characteristics that makes it different from other species. Each species has its own developmental plan—often described as a sort of “blueprint” for building the organism—which is encoded in the DNA molecules present in its cells. This developmental plan determines the characteristics that are inherited.

That the cell nucleus plays a key role in inheritance was recognized in the 1870s by the observation that the nuclei of male and female reproductive cells undergo fusion in the process of fertilization. Soon thereafter, **chromosomes** were first observed inside the nucleus as thread-like objects that become visible in the light microscope when the cell is stained with certain dyes. Chromosomes were found to exhibit a characteristic “splitting” behavior in which each daughter cell formed by cell division receives an identical complement of chromosomes

Further evidence for the importance of chromosomes was provided by the observation that, whereas the number of chromosomes in each cell may differ among biological species, the number of chromosomes is nearly always constant within the cells of any particular species. These features of chromosomes were well understood by about 1900, and they made it seem likely that chromosomes were the carriers of the genes.

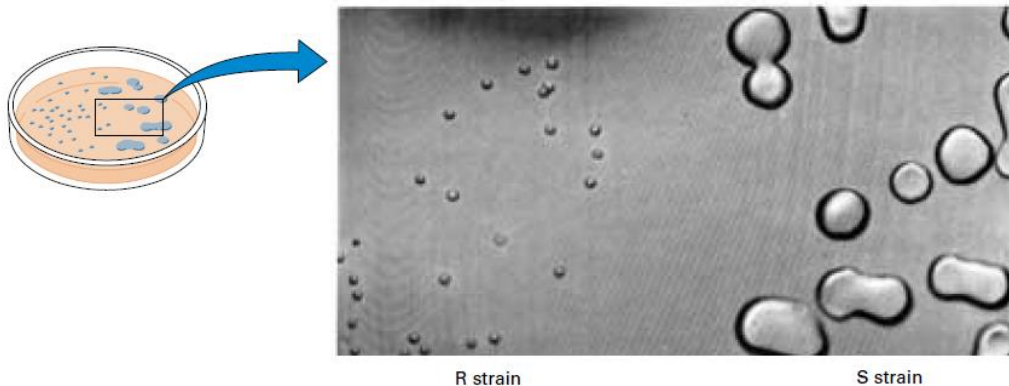
Microscopic studies with special stains showed that DNA is present in chromosomes. Chromosomes also contain various types of proteins, but the amount and kinds of chromosomal proteins differ greatly from one cell type to another, whereas the amount of DNA per cell is constant. Furthermore, nearly all of the DNA present in cells of higher organisms is present in the chromosomes. These arguments for DNA as the genetic material were unconvincing, however, because crude chemical analyses had suggested (erroneously, as it turned out) that DNA lacks the chemical diversity needed in a genetic substance.



The favored candidate for the genetic material was protein, because proteins were known to be an exceedingly diverse collection of molecules. Proteins therefore became widely accepted as the genetic material, and DNA was assumed to function merely as the structural framework of the chromosomes. The experiments described below finally demonstrated that DNA is the genetic material.

Experimental Proof of the Genetic Function of DNA

Colonies of rough (R, the small colonies) and smooth (S, the large colonies) strains of *Streptococcus pneumoniae*. The S colonies are larger because of the gelatinous capsule on the S cells.



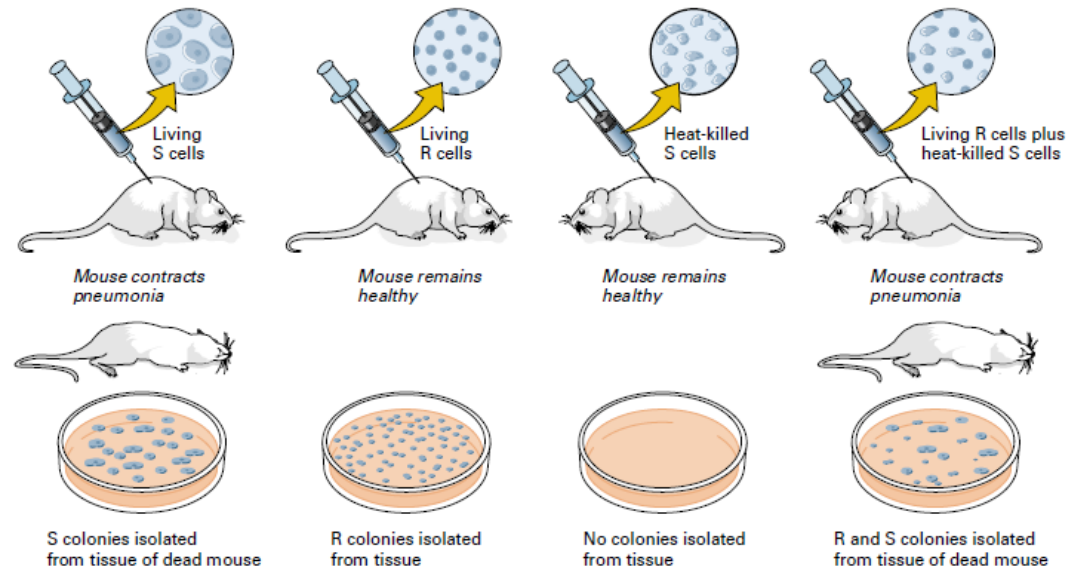
An important first step was taken by Frederick Griffith in 1928 when he demonstrated that a physical trait can be passed from one cell to another. He was working with two strains of the bacterium *Streptococcus pneumoniae* identified as S and R.

Experimental Proof of the Genetic Function of DNA

When a bacterial cell is grown on solid medium, it undergoes repeated cell divisions to form a visible clump of cells called a colony. The S type of *S. pneumoniae* synthesizes a gelatinous capsule composed of complex carbohydrate (polysaccharide). The enveloping capsule makes each colony large and gives it a glistening or smooth (S) appearance. This capsule also enables the bacterium to cause pneumonia by protecting it from the defense mechanisms of an infected animal.

Experimental Proof of the Genetic Function of DNA

The R strains of *S. pneumoniae* are unable to synthesize the capsular polysaccharide; they form small colonies that have a rough (R) surface. This strain of the bacterium does not cause pneumonia, because without the capsule the bacteria are inactivated by the immune system of the host. Both types of bacteria “breed true” in the sense that the progeny formed by cell division have the capsular type of the parent, either S or R.



The Griffith's experiment demonstrating bacterial transformation. A mouse remains healthy if injected with either the nonvirulent R strain of *S. pneumoniae* or heat-killed cell fragments of the usually virulent S strain. R cells in the presence of heat-killed S cells are transformed into the virulent S strain, causing pneumonia in the mouse.

Experimental Proof of the Genetic Function of DNA

Mice injected with living S cells get pneumonia. Mice injected either with living R cells or with heat-killed S cells remain healthy. Here is Griffith's critical finding:

mice injected with a mixture of living R cells and heat-killed S cells contract the disease—they often die of pneumonia.

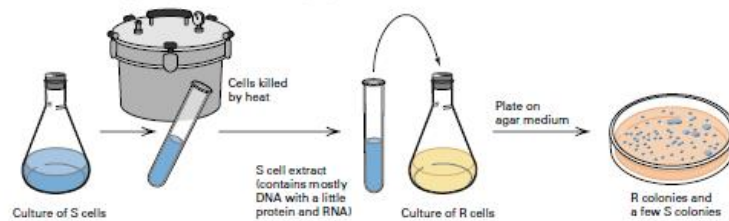
Experimental Proof of the Genetic Function of DNA

Transformation in *Streptococcus* was originally discovered in 1928, but it was not until 1944 that the chemical substance responsible for changing the R cells into S cells was identified. In a milestone experiment, Oswald Avery, Colin MacLeod, and Maclyn McCarty showed that the substance causing the transformation of R cells into S cells was DNA. In doing these experiments, they first had to develop chemical procedures for isolating almost pure DNA from cells, which had never been done before.

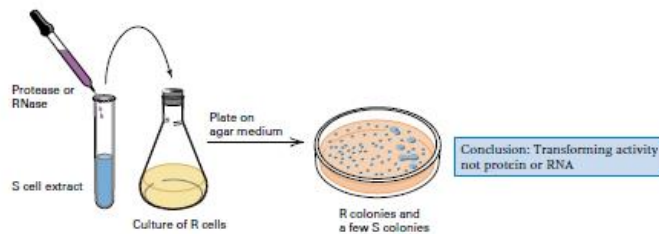
Experimental Proof of the Genetic Function of DNA

When they added DNA isolated from S cells to growing cultures of R cells, they observed transformation: A few cells of type S cells were produced. Although the DNA preparations contained traces of protein and RNA (ribonucleic acid, an abundant cellular macromolecule chemically related to DNA), the transforming activity was not altered by treatments that destroyed either protein or RNA. However, treatments that destroyed DNA eliminated the transforming activity

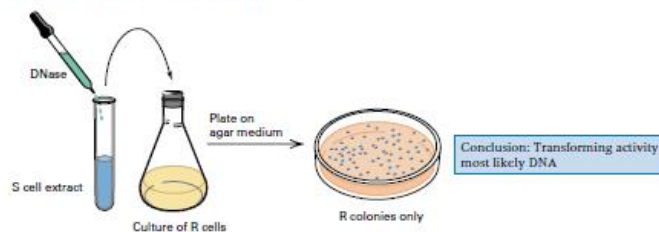
(A) The transforming activity in S cells is not destroyed by heat.



(B) The transforming activity is not destroyed by either protease or RNase.

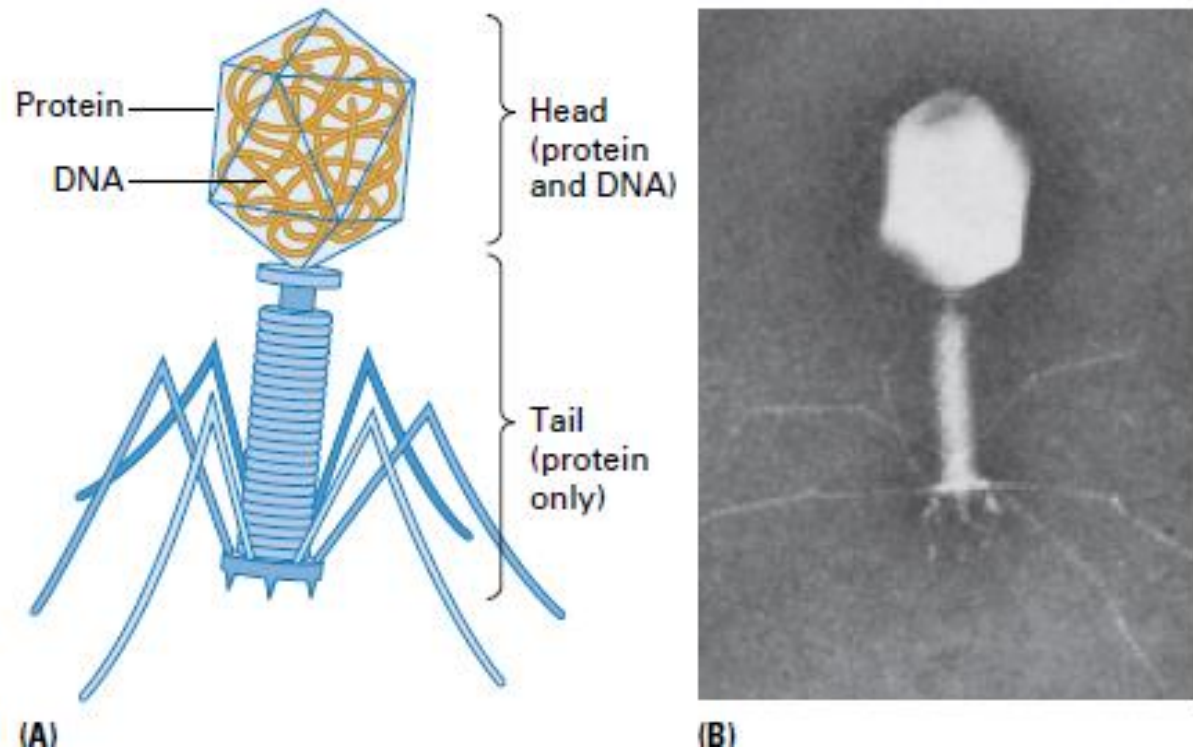


(C) The transforming activity is destroyed by DNase.

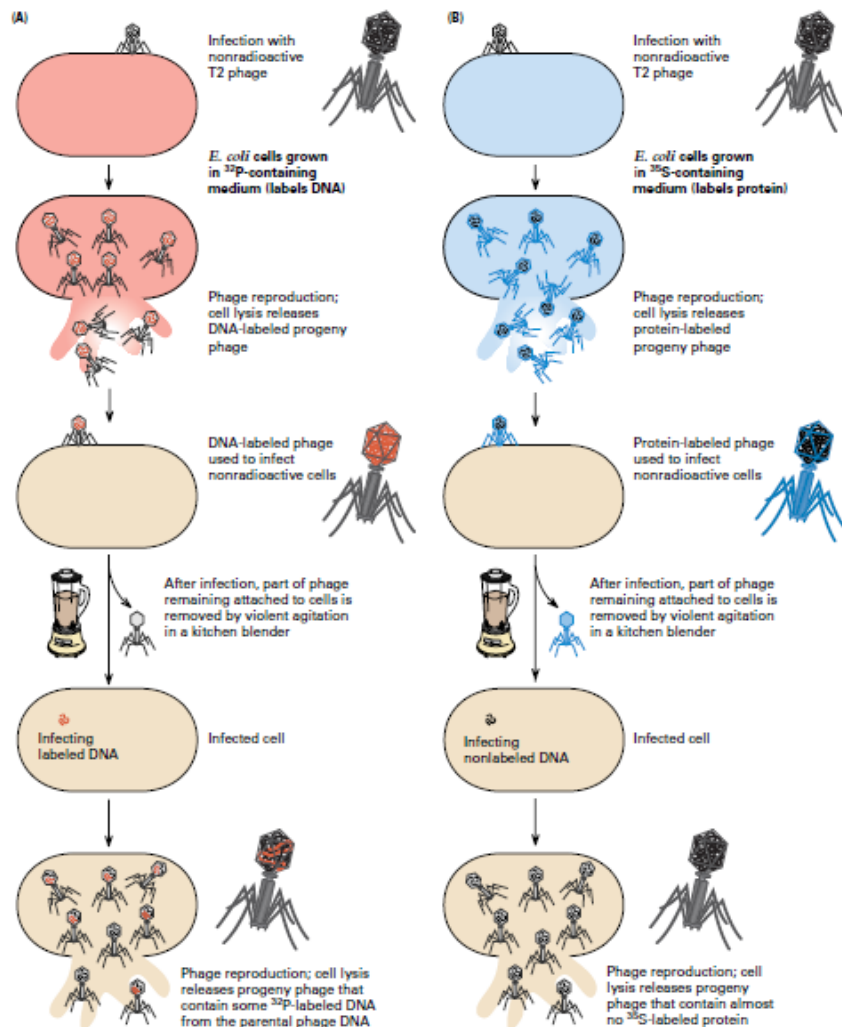


A diagram of the Avery–MacLeod–McCarty experiment that demonstrated that DNA is the active material in bacterial transformation. (A) Purified DNA extracted from heat-killed S cells can convert some living R cells into S cells, but the material may still contain undetectable traces of protein and/or RNA. (B) The transforming activity is not destroyed by either protease or RNase. (C) The transforming activity is destroyed by DNase and so probably consists of DNA.

Genetic Role of DNA in Bacteriophage



A) Drawing of E. coli phage T2, showing various components. The DNA is confined to the interior of the head. (B) An electron micrograph of phage T4, a closely related phage.



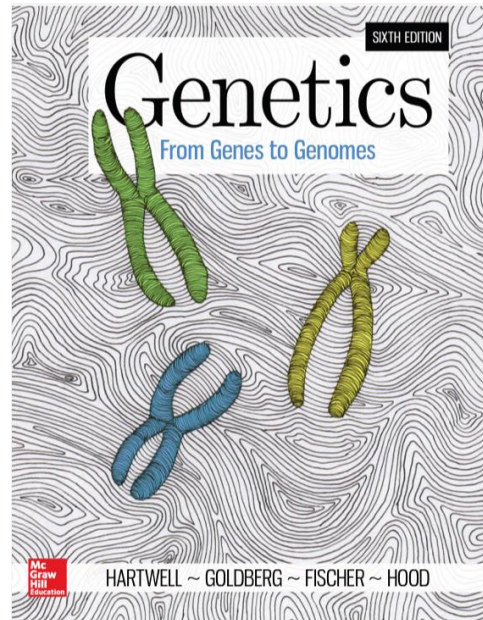
Conclusion: DNA from an infecting parental phage is inherited in the progeny phage

The Hershey–Chase (“blender”) experiment demonstrating that DNA, not protein, is responsible for directing the reproduction of phage T2 in infected *E. coli* cells. (A) Radioactive DNA is transmitted to progeny phage in substantial amounts. (B) Radioactive protein is transmitted to progeny phage in negligible amounts.



Review Questions

- What is the role of DNA?
- How the genetic material is kept in the cell?
- How do they proof the role of DNA?



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 2. DNA Structure: The Double Helix



Contents

- Introduction
- The Structure of DNA
- How DNA Is Arranged in the Cell



Introduction

In the 1950s, Francis Crick and James Watson worked together at the University of Cambridge, England, to determine the structure of DNA. Other scientists, such as Linus Pauling and Maurice Wilkins, were also actively exploring this field. Pauling had discovered the secondary structure of proteins using X-ray crystallography. X-ray crystallography is a method for investigating molecular structure by observing the patterns formed by X-rays shot through a crystal of the substance. The patterns give important information about the structure of the molecule of interest. In Wilkins' lab, researcher Rosalind Franklin was using X-ray crystallography to understand the structure of DNA. Watson and Crick were able to piece together the puzzle of the DNA molecule using Franklin's data.

Introduction



Pioneering scientists (a) James Watson and Francis Crick are pictured here with American geneticist Maclyn McCarty. Scientist Rosalind Franklin discovered (b) the X-ray diffraction pattern of DNA, which helped to elucidate its double helix structure. (credit a: modification of work by Marjorie McCarty; b: modification of work by NIH)

ERASMUS+

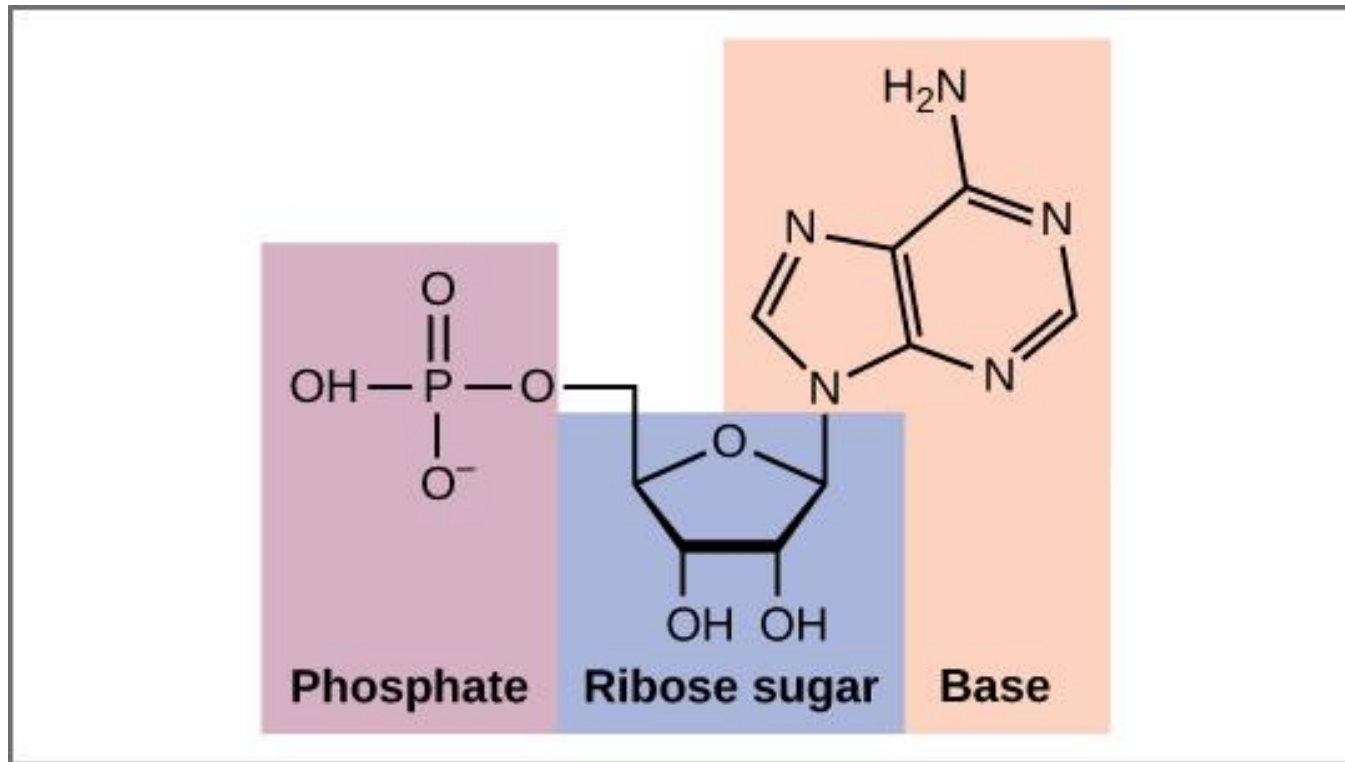
Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Watson and Crick also had key pieces of information available from other researchers such as Chargaff's rules. Chargaff had shown that of the four kinds of monomers (nucleotides) present in a DNA molecule, two types were always present in equal amounts and the remaining two types were also always present in equal amounts. This meant they were always paired in some way. In 1962, James Watson, Francis Crick, and Maurice Wilkins were awarded the Nobel Prize in Medicine for their work in determining the structure of DNA.

The Structure of DNA

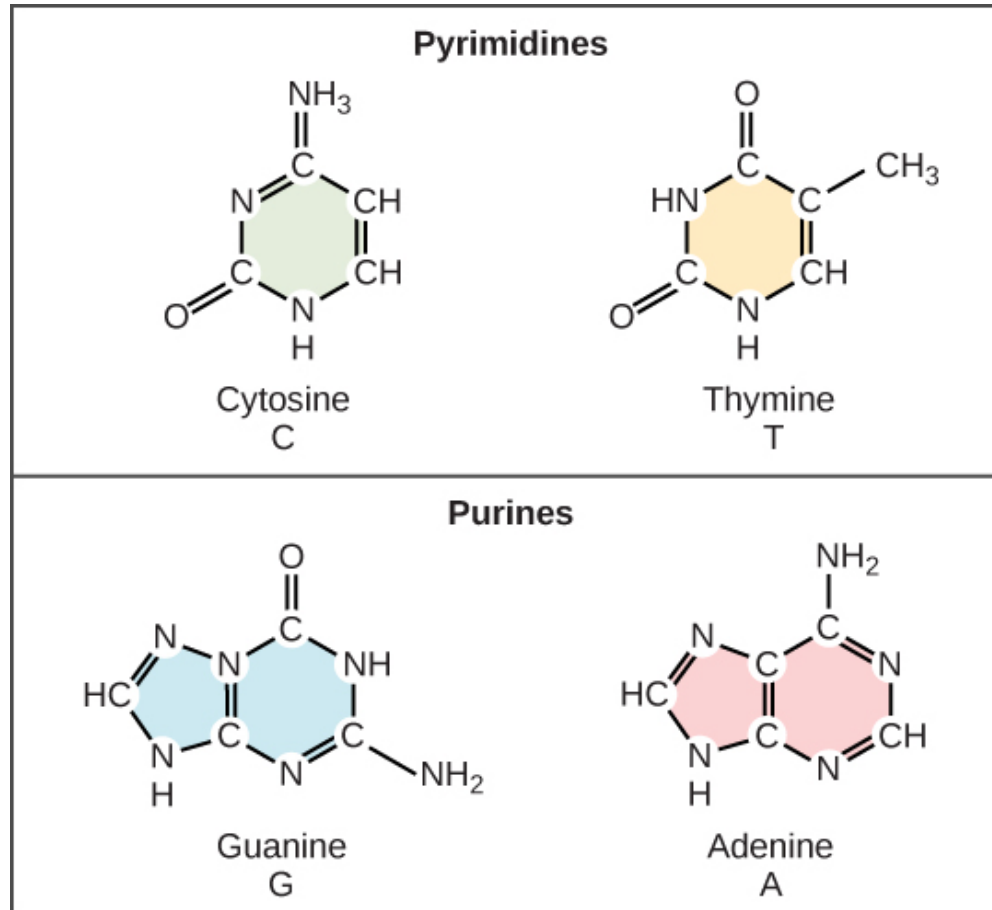
There are two types of nucleic acids, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The building blocks of DNA are nucleotides, which are made up of three parts: a deoxyribose (5-carbon sugar), a phosphate group, and a nitrogenous base. There are four types of nitrogenous bases in DNA. Adenine (A) and guanine (G) are double-ringed purines, and cytosine (C) and thymine (T) are smaller, single-ringed pyrimidines. The nucleotide is named according to the nitrogenous base it contains.

The Structure of DNA



Each DNA nucleotide is made up of a sugar, a phosphate group, and a base.

Structure



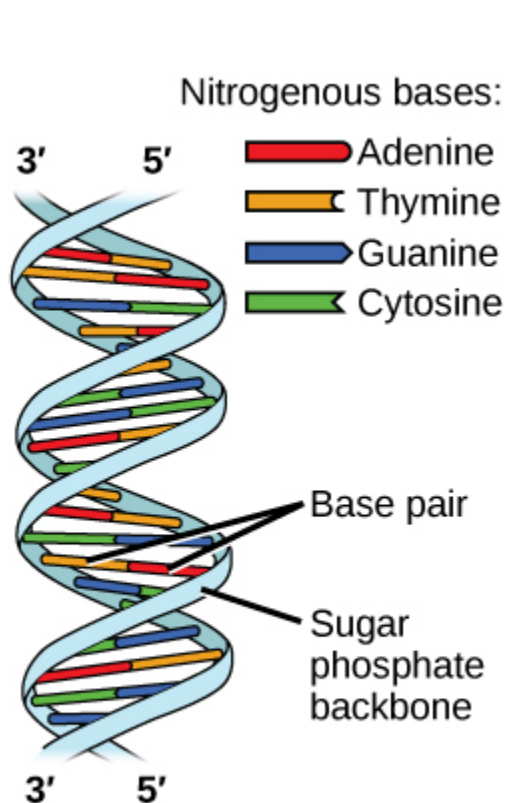
Cytosine and thymine are pyrimidines. Guanine and adenine are purines

The phosphate group of one nucleotide bonds covalently with the sugar molecule of the next nucleotide, and so on, forming a long polymer of nucleotide monomers. The sugar–phosphate groups line up in a “backbone” for each single strand of DNA, and the nucleotide bases stick out from this backbone. The carbon atoms of the five-carbon sugar are numbered clockwise from the oxygen as 1′, 2′, 3′, 4′, and 5′ (1′ is read as “one prime”). The phosphate group is attached to the 5′ carbon of one nucleotide and the 3′ carbon of the next nucleotide. In its natural state, each DNA molecule is actually composed of two single strands held together along their length with hydrogen bonds between the bases.

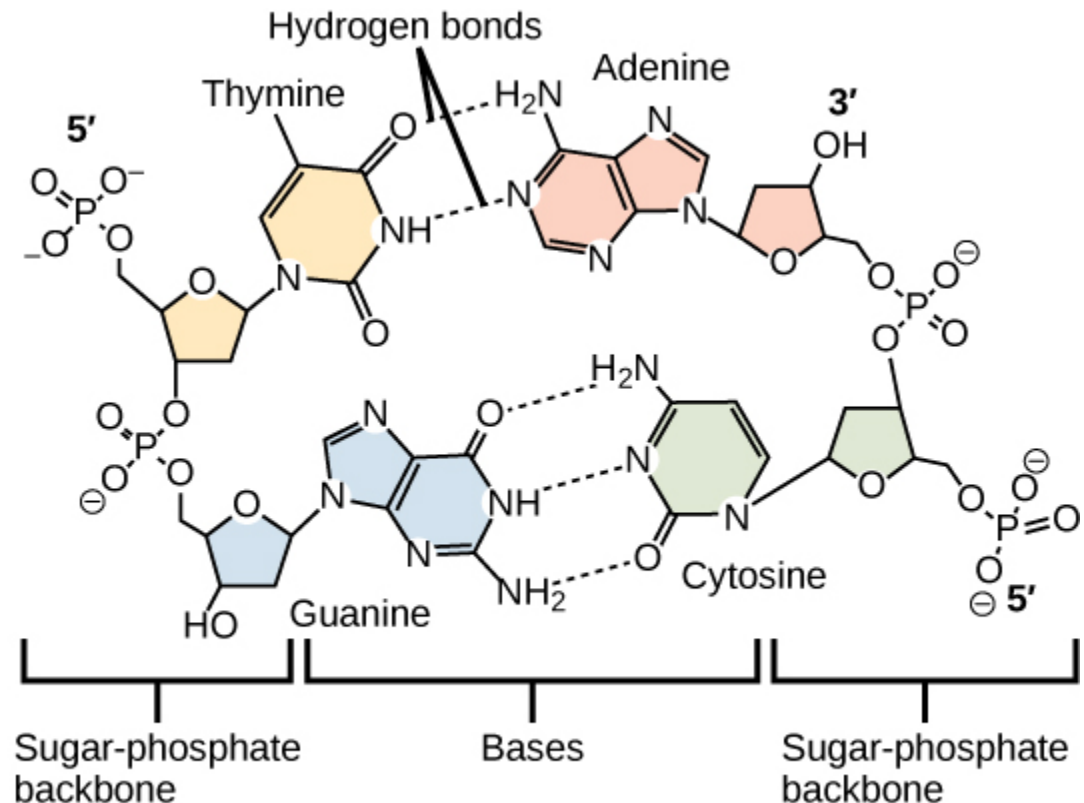
Watson and Crick proposed that the DNA is made up of two strands that are twisted around each other to form a right-handed helix, called a double helix. Base-pairing takes place between a purine and pyrimidine: namely, A pairs with T, and G pairs with C. In other words, adenine and thymine are complementary base pairs, and cytosine and guanine are also complementary base pairs. This is the basis for Chargaff's rule; because of their complementarity, there is as much adenine as thymine in a DNA molecule and as much guanine as cytosine.



Adenine and thymine are connected by two hydrogen bonds, and cytosine and guanine are connected by three hydrogen bonds. The two strands are anti-parallel in nature; that is, one strand will have the 3' carbon of the sugar in the “upward” position, whereas the other strand will have the 5' carbon in the upward position. The diameter of the DNA double helix is uniform throughout because a purine (two rings) always pairs with a pyrimidine (one ring) and their combined lengths are always equal.



(a)



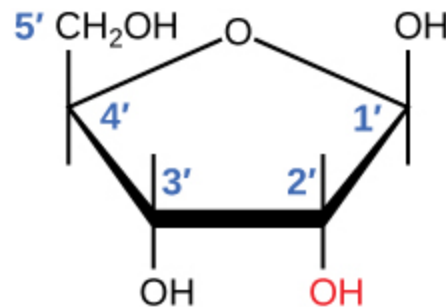
(b)

DNA (a) forms a double stranded helix, and (b) adenine pairs with thymine and cytosine pairs with guanine. (credit a: modification of work by Jerome Walker, Dennis Myts)

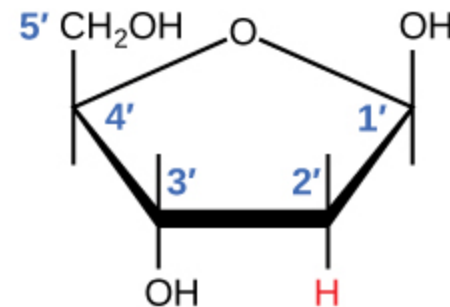
The Structure of RNA

There is a second nucleic acid in all cells called ribonucleic acid, or RNA. Like DNA, RNA is a polymer of nucleotides. Each of the nucleotides in RNA is made up of a nitrogenous base, a five-carbon sugar, and a phosphate group. In the case of RNA, the five-carbon sugar is ribose, not deoxyribose. Ribose has a hydroxyl group at the 2' carbon, unlike deoxyribose, which has only a hydrogen atom.

The Structure of RNA



Ribose



Deoxyribose

The difference between the ribose found in RNA and the deoxyribose found in DNA is that ribose has a hydroxyl group at the 2' carbon.

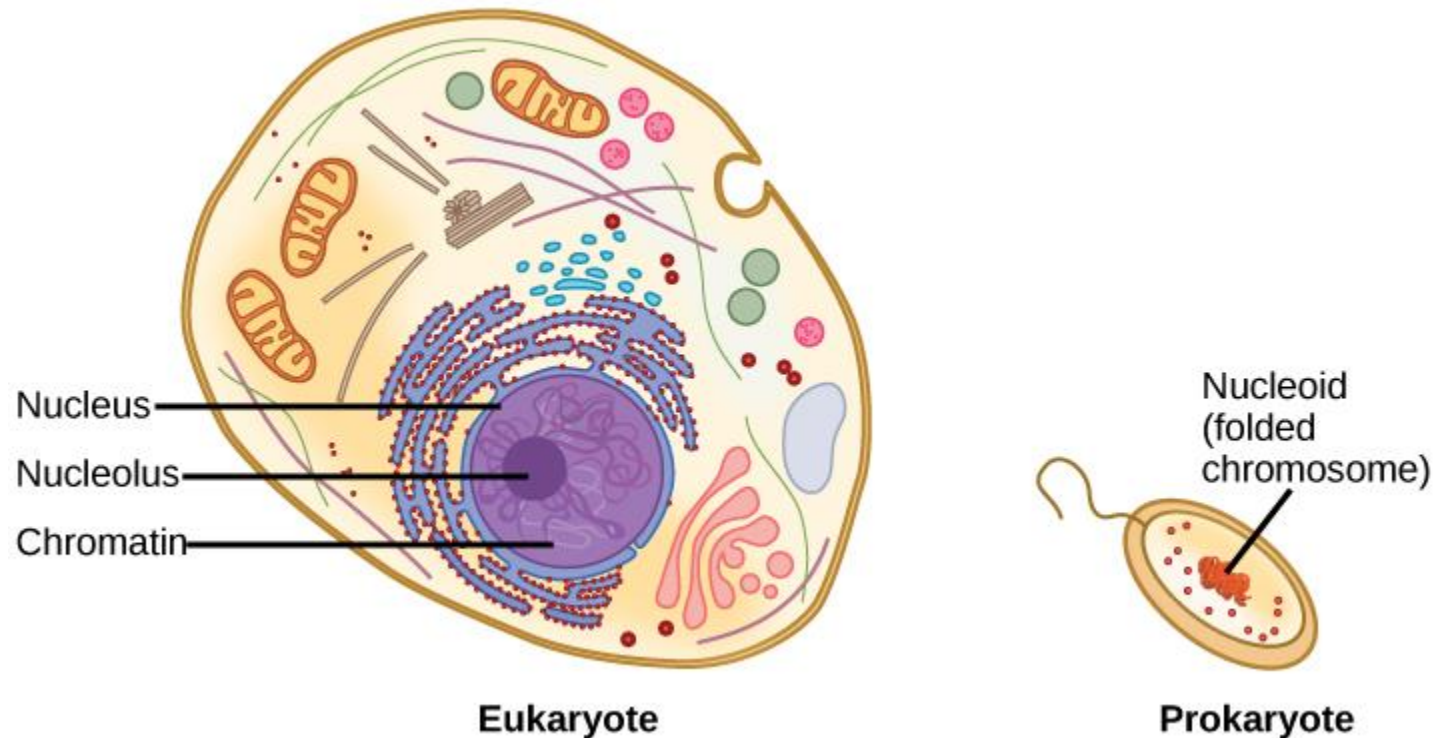
The Structure of RNA

RNA nucleotides contain the nitrogenous bases adenine, cytosine, and guanine. However, they do not contain thymine, which is instead replaced by uracil, symbolized by a “U.” RNA exists as a single-stranded molecule rather than a double-stranded helix. Molecular biologists have named several kinds of RNA on the basis of their function. These include messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA)—molecules that are involved in the production of proteins from the DNA code.

How DNA Is Arranged in the Cell

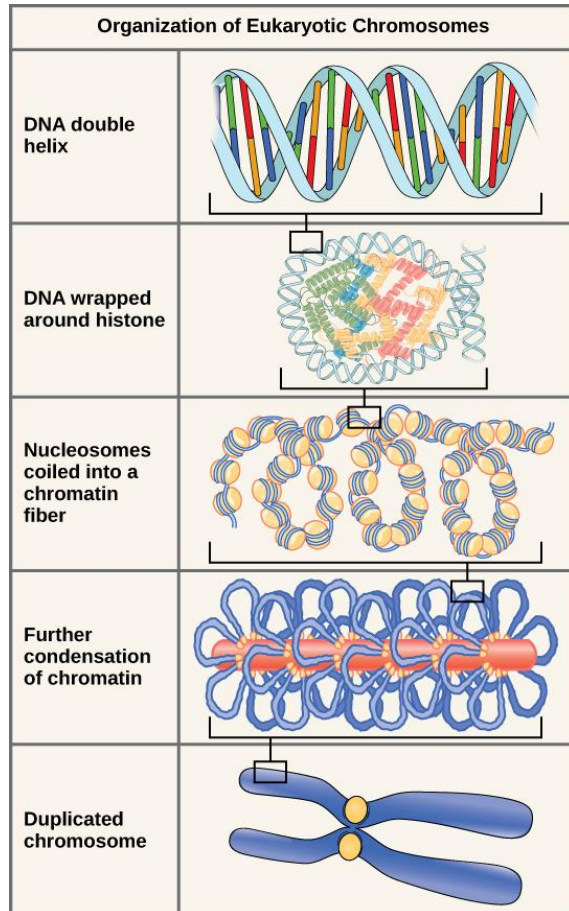
DNA is a working molecule; it must be replicated when a cell is ready to divide, and it must be “read” to produce the molecules, such as proteins, to carry out the functions of the cell. For this reason, the DNA is protected and packaged in very specific ways. In addition, DNA molecules can be very long. Stretched end-to-end, the DNA molecules in a single human cell would come to a length of about 2 meters. Thus, the DNA for a cell must be packaged in a very ordered way to fit and function within a structure (the cell) that is not visible to the naked eye. The chromosomes of prokaryotes are much simpler than those of eukaryotes in many of their features. Most prokaryotes contain a single, circular chromosome that is found in an area in the cytoplasm called the nucleoid.

How DNA Is Arranged in the Cell



A eukaryote contains a well-defined nucleus, whereas in prokaryotes, the chromosome lies in the cytoplasm in an area called the nucleoid.

The size of the genome in one of the most well-studied prokaryotes, *Escherichia coli*, is 4.6 million base pairs, which would extend a distance of about 1.6 mm if stretched out. So how does this fit inside a small bacterial cell? The DNA is twisted beyond the double helix in what is known as supercoiling. Some proteins are known to be involved in the supercoiling; other proteins and enzymes help in maintaining the supercoiled structure.



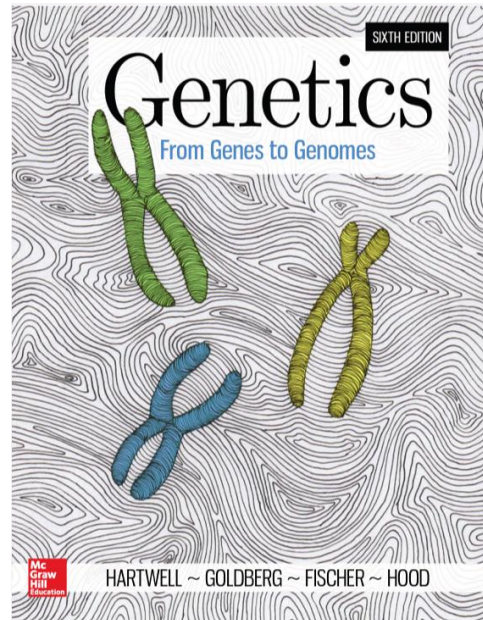
These figures illustrate the compaction of the eukaryotic chromosome.

Eukaryotes, whose chromosomes each consist of a linear DNA molecule, employ a different type of packing strategy to fit their DNA inside the nucleus. At the most basic level, DNA is wrapped around proteins known as histones to form structures called nucleosomes. The DNA is wrapped tightly around the histone core. This nucleosome is linked to the next one by a short strand of DNA that is free of histones. This is also known as the “beads on a string” structure; the nucleosomes are the “beads” and the short lengths of DNA between them are the “string.” The nucleosomes, with their DNA coiled around them, stack compactly onto each other to form a 30-nm-wide fiber. This fiber is further coiled into a thicker and more compact structure. At the metaphase stage of mitosis, when the chromosomes are lined up in the center of the cell, the chromosomes are at their most compacted. They are approximately 700 nm in width, and are found in association with scaffold proteins.



Review Questions

- Explain the structure of DNA.
- How the genetic material is organized in the nucleus?



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 3. An Overview of DNA Replication



Contents

- Introduction
- The basic idea
- DNA polymerase
- DNA replication in eukaryotes



Introduction

DNA replication, or the copying of a cell's DNA, is no simple task! There are about 3 billion base pairs of DNA in your genome, all of which must be accurately copied when any one of your trillions of cells divides.

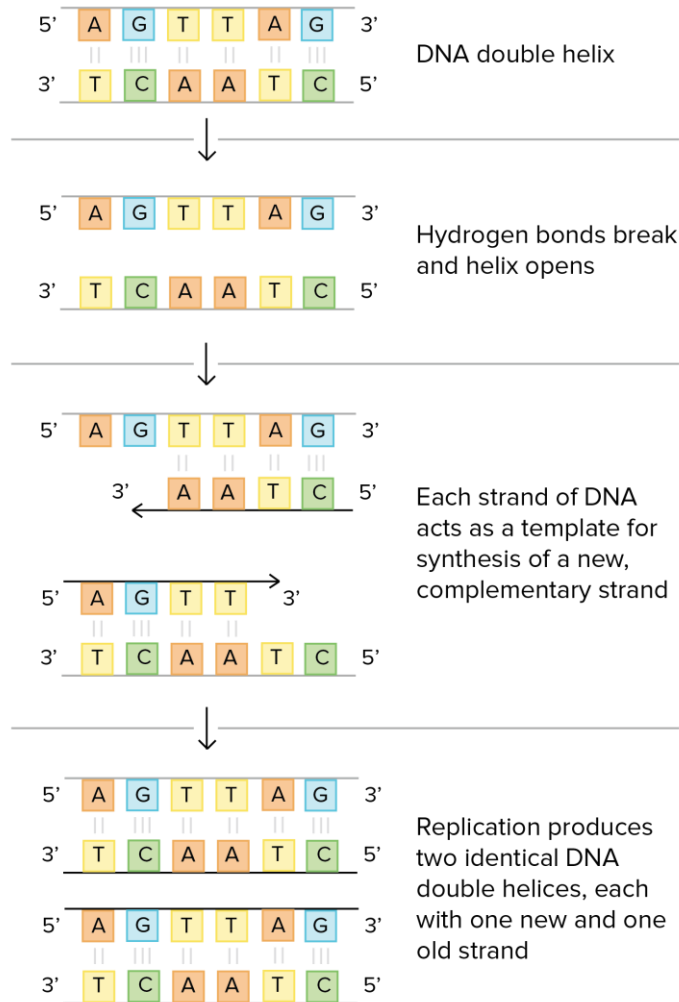
The basic mechanisms of DNA replication are similar across organisms. In this article, we'll focus on DNA replication as it takes place in the bacterium *E. coli*, but the mechanisms of replication are similar in humans and other eukaryotes.

Let's take a look at the proteins and enzymes that carry out replication, seeing how they work together to ensure accurate and complete replication of DNA.

The basic idea

DNA replication is semiconservative, meaning that each strand in the DNA double helix acts as a template for the synthesis of a new, complementary strand.

This process takes us from one starting molecule to two "daughter" molecules, with each newly formed double helix containing one new and one old strand.



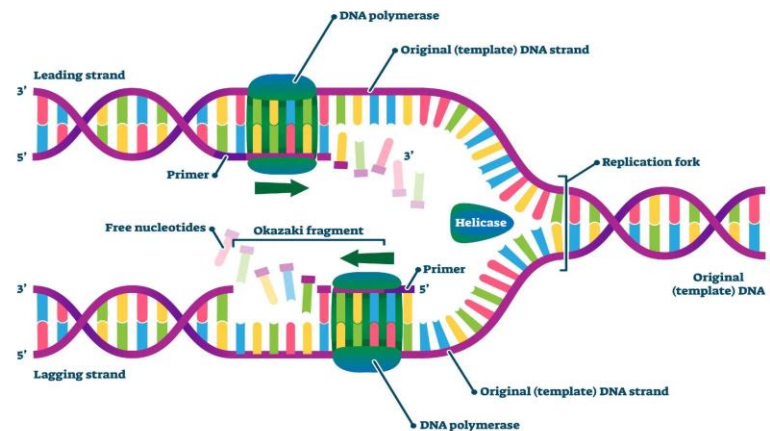
In a sense, that's all there is to DNA replication! But what's actually most interesting about this process is how it's carried out in a cell.

Cells need to copy their DNA very quickly, and with very few errors (or risk problems such as cancer). To do so, they use a variety of enzymes and proteins, which work together to make sure DNA replication is performed smoothly and accurately.

DNA polymerase

One of the key molecules in DNA replication is the enzyme DNA polymerase. DNA polymerases are responsible for synthesizing DNA: they add nucleotides one by one to the growing DNA chain, incorporating only those that are complementary to the template.

DNA POLYMERASE



ERASMUS+

DNA polymerase

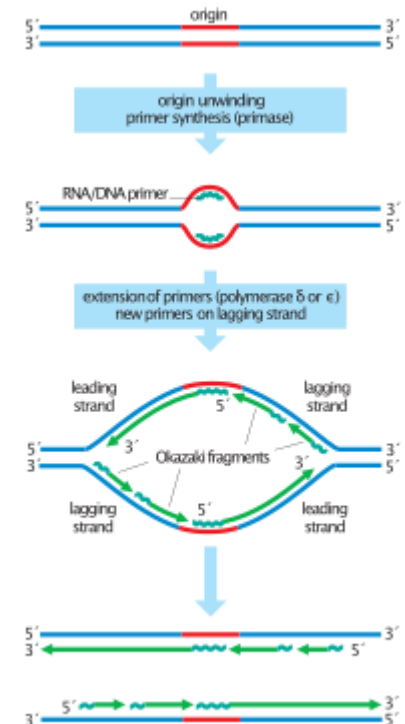
Here are some key features of DNA polymerases:

- They always need a template
- They can only add nucleotides to the 3' end of a DNA strand
- They can't start making a DNA chain from scratch, but require a pre-existing chain or short stretch of nucleotides called a primer
- They proofread, or check their work, removing the vast majority of "wrong" nucleotides that are accidentally added to the chain

Starting DNA replication

Replication always starts at specific locations on the DNA, which are called origins of replication and are recognized by their sequence.

Specialized proteins recognize the origin, bind to this site, and open up the DNA. As the DNA opens, two Y-shaped structures called replication forks are formed, together making up what's called a replication bubble. The replication forks will move in opposite directions as replication proceeds.



Primers and primase

DNA polymerases can only add nucleotides to the 3' end of an existing DNA strand. (They use the free -OH group found at the 3' end as a "hook," adding a nucleotide to this group in the polymerization reaction.) How, then, does DNA polymerase add the first nucleotide at a new replication fork?

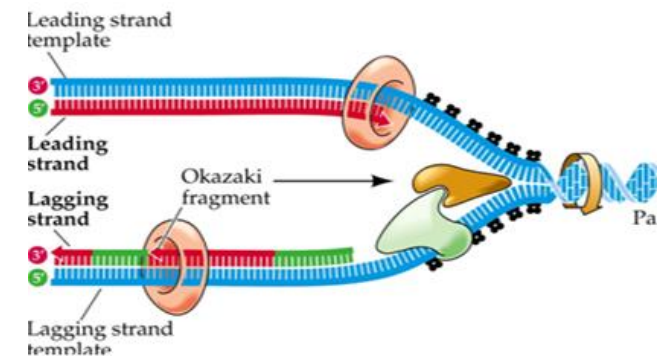
Alone, it can't! The problem is solved with the help of an enzyme called primase. Primase makes an RNA primer, or short stretch of nucleic acid complementary to the template, that provides a 3' end for DNA polymerase to work on. A typical primer is about five to ten nucleotides long. The primer primes DNA synthesis, i.e., gets it started.

Once the RNA primer is in place, DNA polymerase "extends" it, adding nucleotides one by one to make a new DNA strand that's complementary to the template strand.

Leading and lagging strands

DNA polymerases can only make DNA in the 5' to 3' direction, and this poses a problem during replication. A DNA double helix is always anti-parallel; in other words, one strand runs in the 5' to 3' direction, while the other runs in the 3' to 5' direction. This makes it necessary for the two new strands, which are also antiparallel to their templates, to be made in slightly different ways.

One new strand, which runs 5' to 3' towards the replication fork, is the easy one. This strand is made continuously, because the DNA polymerase is moving in the same direction as the replication fork. This continuously synthesized strand is called the leading strand.



Leading and lagging strands

The other new strand, which runs 5' to 3' away from the fork, is trickier. This strand is made in fragments because, as the fork moves forward, the DNA polymerase (which is moving away from the fork) must come off and reattach on the newly exposed DNA. This tricky strand, which is made in fragments, is called the lagging strand.

The small fragments are called Okazaki fragments, named for the Japanese scientist who discovered them. The leading strand can be extended from one primer alone, whereas the lagging strand needs a new primer for each of the short Okazaki fragments.

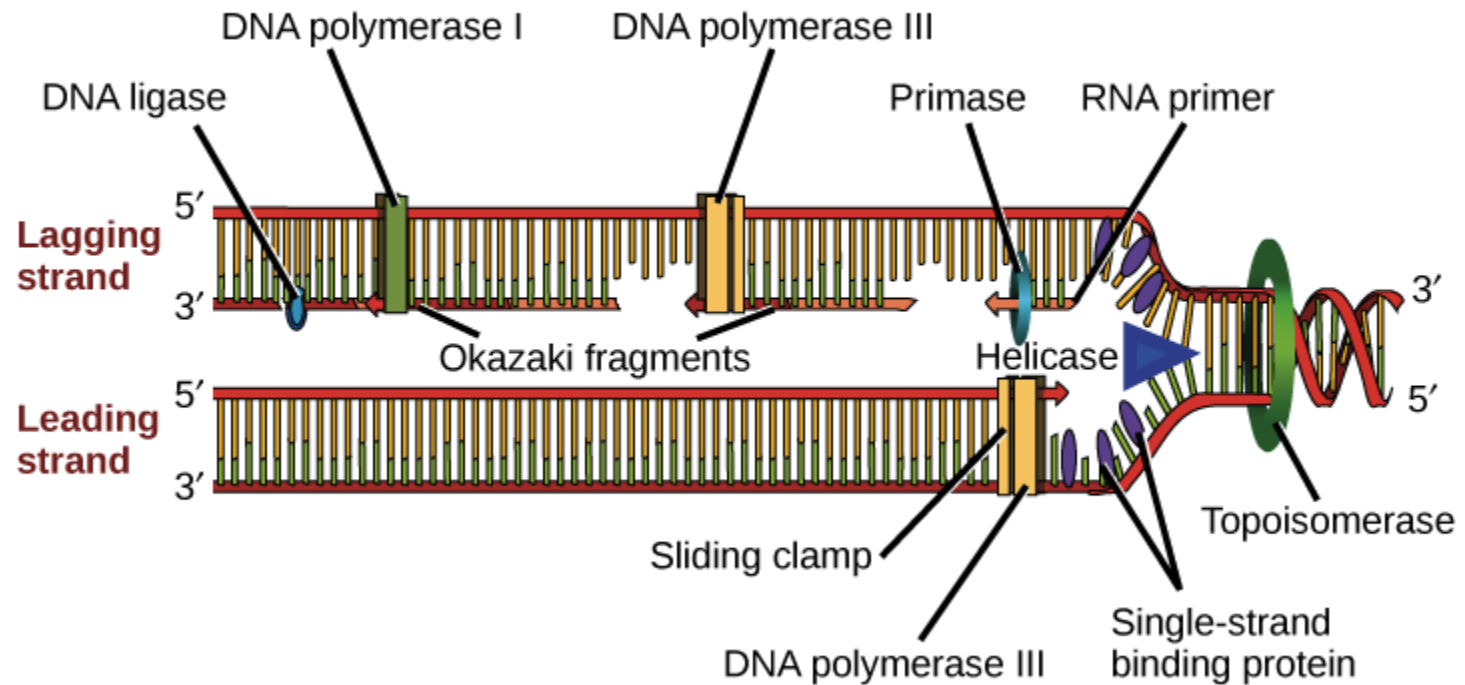
The maintenance and cleanup crew

Some other proteins and enzymes, in addition the main ones above, are needed to keep DNA replication running smoothly. One is a protein called the sliding clamp, which holds DNA polymerase III molecules in place as they synthesize DNA. The sliding clamp is a ring-shaped protein and keeps the DNA polymerase of the lagging strand from floating off when it re-starts at a new Okazaki fragment.

The maintenance and cleanup crew

Topoisomerase also plays an important maintenance role during DNA replication. This enzyme prevents the DNA double helix ahead of the replication fork from getting too tightly wound as the DNA is opened up. It acts by making temporary nicks in the helix to release the tension, then sealing the nicks to avoid permanent damage.

Finally, there is a little cleanup work to do if we want DNA that doesn't contain any RNA or gaps. The RNA primers are removed and replaced by DNA through the activity of DNA polymerase I, the other polymerase involved in replication. The nicks that remain after the primers are replaced get sealed by the enzyme DNA ligase.





- Helicase opens up the DNA at the replication fork.
- Single-strand binding proteins coat the DNA around the replication fork to prevent rewinding of the DNA.
- Topoisomerase works at the region ahead of the replication fork to prevent supercoiling.
- Primase synthesizes RNA primers complementary to the DNA strand.
- DNA polymerase III extends the primers, adding on to the 3' end, to make the bulk of the new DNA.
- RNA primers are removed and replaced with DNA by DNA polymerase I.
- The gaps between DNA fragments are sealed by DNA ligase.

DNA replication in eukaryotes

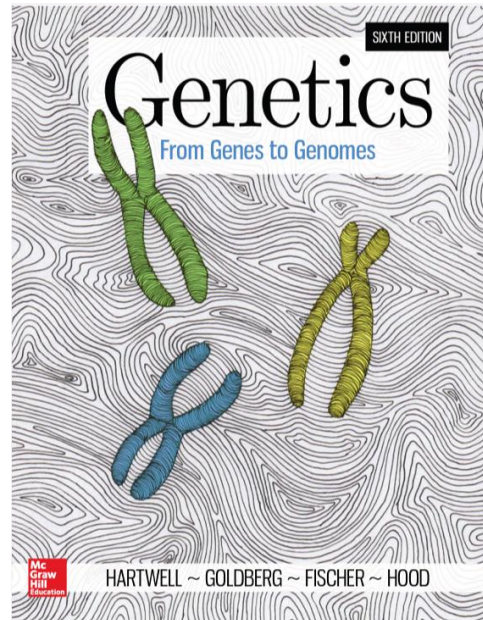
The basics of DNA replication are similar between bacteria and eukaryotes such as humans, but there are also some differences:

- Eukaryotes usually have multiple linear chromosomes, each with multiple origins of replication. Humans can have up to 100 000 origins of replication
- Most of the *E. coli* enzymes have counterparts in eukaryotic DNA replication, but a single enzyme in *E. coli* may be represented by multiple enzymes in eukaryotes. For instance, there are five human DNA polymerases with important roles in replication
- Most eukaryotic chromosomes are linear. Because of the way the lagging strand is made, some DNA is lost from the ends of linear chromosomes (the telomeres) in each round of replication



Review Questions

- Explain the main stages of DNA replication.
- List the key molecules (enzymes) in DNA replication process and explain their role.



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 4. Genes and Proteins



Contents

- Introduction
- Genes Encode Proteins
- Mutation
- Inherited mutations
- Natural selection



Introduction

If the code is thought of as a string of letters on a sheet of paper, then the genes are made up of distinct words that form sentences and paragraphs that give meaning to the pattern of letters. What is created from the complex and diverse DNA codes is protein, a class of macromolecules that carries out most of the activities in the cell.



Introduction

The information content of DNA resides in the sequence of its bases. The four bases in each chain are like the letters of an alphabet; they may follow each other in any order, and different sequences spell out different “words.” Each word has its own meaning, that is, its own effect on phenotype. AGTCAT, for example, means one thing, while CTAGGT means another. Although DNA has only four different letters, or building blocks, the potential for different combinations and thus different sets of information in a long chain of nucleotides is staggering. Some human chromosomes, for example, are composed of chains that are 250 million nucleotides long; because the different bases may follow each other in any order, such chains could contain any one of $4^{250,000,000}$ (which translates to 1 followed by 150,515,000 zeros) potential nucleotide sequences.



Genes Encode Proteins

Since the rediscovery of Mendel's work in 1900, the definition of the gene has progressed from an abstract unit of heredity to a tangible molecular entity capable of replication, transcription, translation, and mutation. Genes are composed of DNA and are linearly arranged on chromosomes. Some genes encode structural and regulatory RNAs. There is increasing evidence from research that profiles the transcriptome of cells (the complete set all RNA transcripts present in a cell) that these may be the largest classes of RNAs produced by eukaryotic cells, far outnumbering the protein-encoding messenger RNAs (mRNAs), but the 20,000 protein-encoding genes typically found in animal cells, and the 30,000 protein-encoding genes typically found in plant cells, nonetheless have huge impacts on cellular functioning.

Protein-encoding genes specify the sequences of amino acids, which are the building blocks of proteins. In turn, proteins are responsible for orchestrating nearly every function of the cell. Both protein-encoding genes and the proteins that are their gene products are absolutely essential to life as we know it.

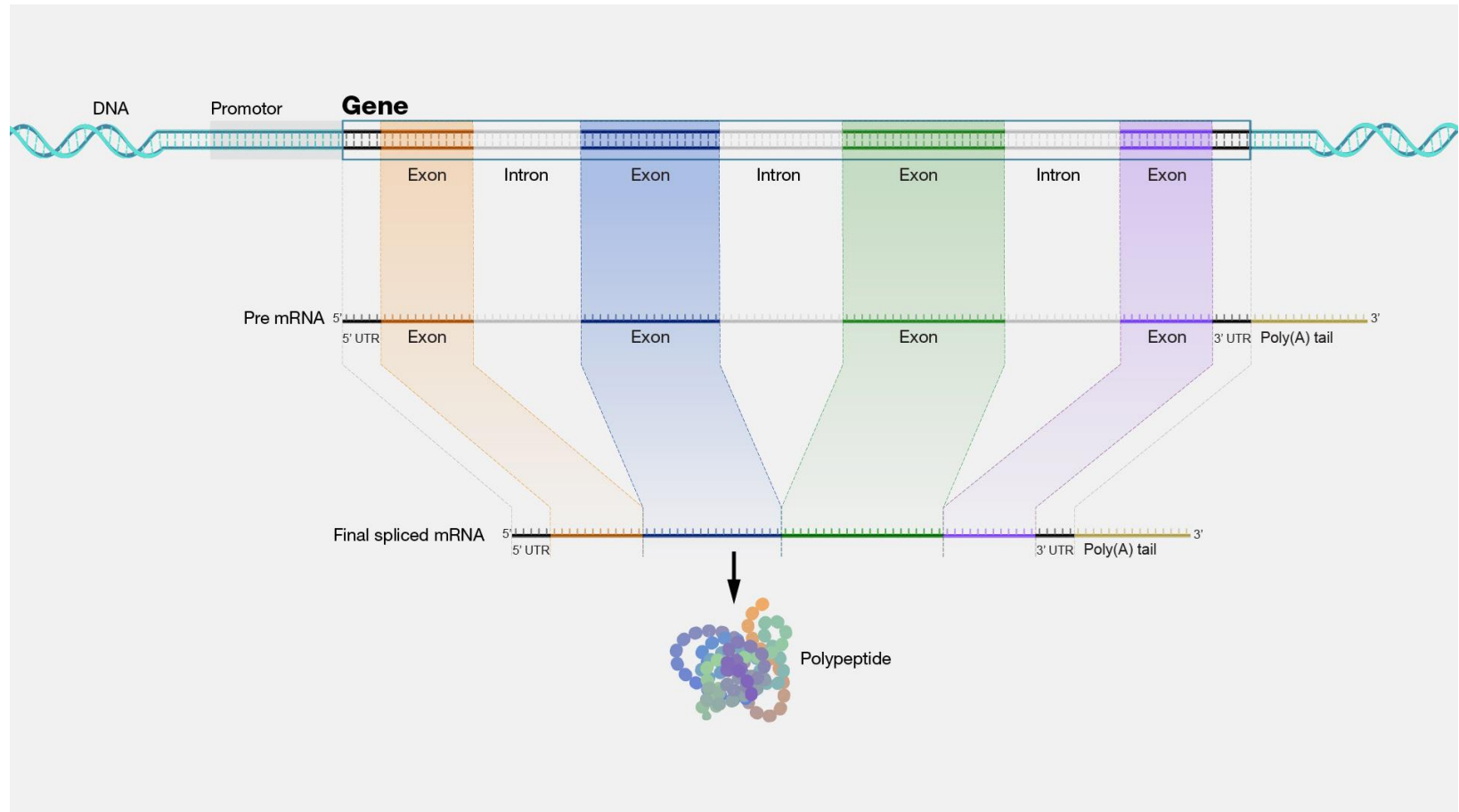


Genes Encode Proteins









Replication, Transcription, and Translation are the three main processes used by all cells to maintain their genetic information and to convert the genetic information encoded in DNA into gene products, which are either RNAs or proteins, depending on the gene. In eukaryotic cells, or those cells that have a nucleus, replication and transcription take place within the nucleus while translation takes place outside of the nucleus in cytoplasm. In prokaryotic cells, or those cells that do not have a nucleus, all three processes occur in the cytoplasm.

Replication is the basis for biological inheritance. It copies a cell's DNA. The enzyme DNA polymerase copies a single parental double-stranded DNA molecule into two daughter double-stranded DNA molecules. Transcription makes RNA from DNA. The enzyme RNA polymerase creates an RNA molecule that is complementary to a gene-encoding stretch of DNA. Translation makes protein from mRNA. The ribosome generates a polypeptide chain of amino acids using mRNA as a template. The polypeptide chain folds up to become a protein.

Genes Encode Proteins



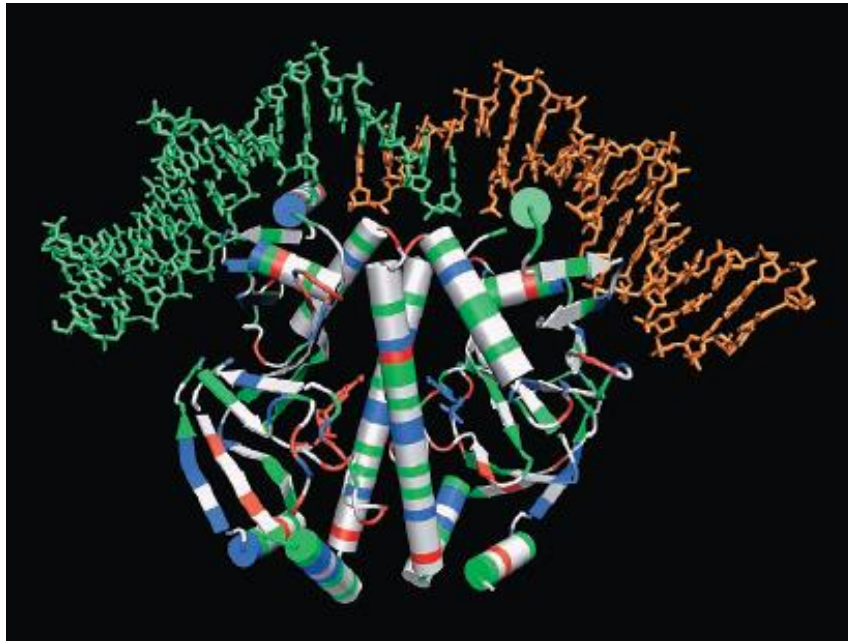
Most Genetic Information Is Read from Unwound DNA Chains

| Base | Icons |
|---|---|
| Purines | |
|  | =  |
| Adenine (A) | |
|  | =  |
| Guanine (G) | |
| Pyrimidines | |
|  | =  |
| Thymine (T) | |
|  | =  |
| Cytosine (C) | |



The unwinding of a DNA molecule exposes a single sequence of bases on each of two strands. Proteins read the information in a single DNA strand by synthesizing a stretch of RNA (a process called transcription) or DNA (a process called replication) complementary to a specific sequence.

Some Genetic Information Is Accessible Without Unwinding DNA



Some proteins can recognize and bind to specific base pair sequences in double-stranded DNA. This information emerges mainly from differences between the four bases that appear in the major and minor grooves. Within the grooves, certain atoms at the periphery of the bases are exposed, and particularly in the major groove, these atoms may assume spatial patterns that provide chemical information. Proteins can access this information to sense the base sequence in a stretch of DNA without disassembling the double helix. Sequence-specific DNA-binding proteins include transcription factors that turn genes on and off as well as bacterial restriction enzymes that cut DNA at particular sites.

In Some Viruses, RNA Is the Repository of Genetic Information

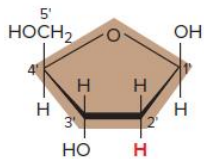
DNA carries the genetic information in all cellular forms of life and in many viruses. Prokaryotes such as *Escherichia coli* bacteria carry their DNA in a double-stranded, covalently closed circular chromosome. Eukaryotic cells package their DNA in double-stranded linear chromosomes. DNA viruses carry it in small molecules that are single- or double-stranded, circular, or linear.

By contrast, retroviruses, which include those that cause polio and AIDS, use ribonucleic acid, or RNA as their genetic material.

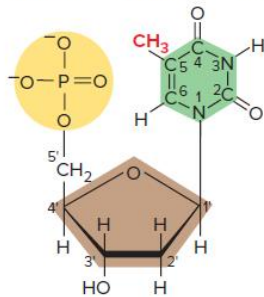
Differences in the chemical structure of DNA and RNA

DNA

- Deoxyribose



- Thymine deoxyribonucleotide

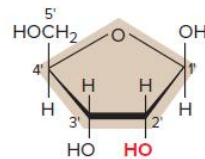


- Usually double-stranded

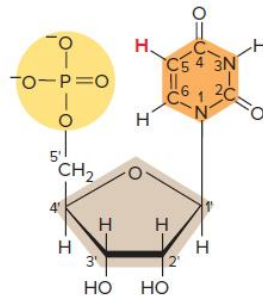


RNA

- Ribose



- Uracil ribonucleotide

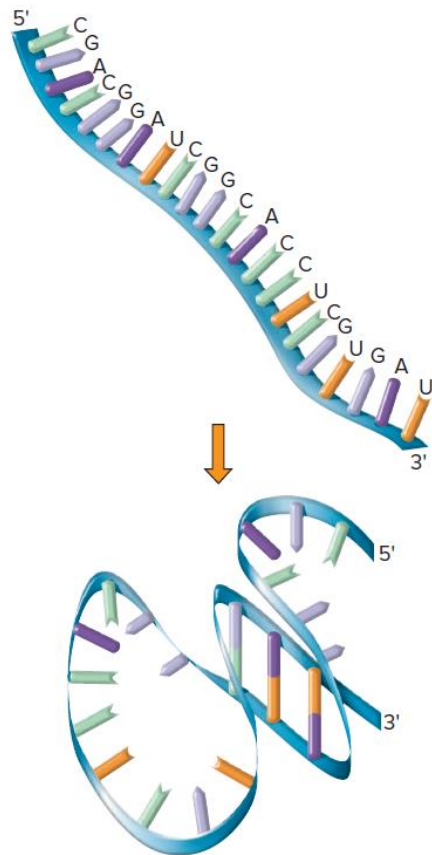


- Usually single-stranded



Three major chemical differences distinguish RNA from DNA. First, RNA takes its name from the sugar ribose, which it incorporates instead of the deoxyribose found in DNA. Second, RNA contains the base uracil (U) instead of the base thymine (T); U, like T, base pairs with A. Finally, most RNA molecules are single-stranded and contain far fewer nucleotides than the very long DNA molecules found in nuclear chromosomes.

Differences in the chemical structure of DNA and RNA



Within a single-stranded RNA molecule, folding can bring together two oppositely oriented regions that carry complementary nucleotide sequences to form a short, basepaired stretch within the molecule. This means that, compared to the relatively simple, double-helical shape of a DNA molecule, many RNAs have a complicated structure of short double-stranded segments interspersed with single-stranded loops. RNA has the same ability as DNA to carry information in the sequence of its bases, but it is much less stable than DNA. In addition to serving as the genetic material for an array of viruses, RNA fulfills several vital functions in all cells. For example, it participates in gene expression and protein synthesis, as presented in Chapter 8. RNA also plays a surprisingly significant role in DNA replication, which we now describe.

- Genes are contained in chromosomes, which are in the cell nucleus.
- A chromosome contains hundreds to thousands of genes.
- Every normal human cell contains 23 pairs of chromosomes, for a total of 46 chromosomes.
- A trait is any gene-determined characteristic and is often determined by more than one gene.
- Some traits are caused by mutated genes that are inherited or that are the result of a new gene mutation.

Proteins are probably the most important class of material in the body. Proteins are not just building blocks for muscles, connective tissues, skin, and other structures. They also are needed to make enzymes. Enzymes are complex proteins that control and carry out nearly all chemical processes and reactions within the body. The body produces thousands of different enzymes. Thus, the entire structure and function of the body is governed by the types and amounts of proteins the body synthesizes. Protein synthesis is controlled by genes, which are contained on chromosomes.

The **genotype** (or genome) is a person's unique combination of genes or genetic makeup. Thus, the genotype is a complete set of instructions on how that person's body synthesizes proteins and thus how that body is supposed to be built and function.

The **phenotype** is the actual structure and function of a person's body. The phenotype is how the genotype manifests in a person—not all the instructions in the genotype may be carried out (or expressed). Whether and how a gene is expressed is determined by a complex interaction of multiple factors including genotype, gene expression, environmental factors (including illnesses and diet), and other factors, some of which are unknown.

A **karyotype** is a picture of the full set of chromosomes in a person's cells.

Mutation

To prevent mistakes during replication, cells have a “proofreading” function to help ensure that bases are paired properly. There are also chemical mechanisms to repair DNA that was not copied properly. However, because of the billions of base pairs involved in, and the complexity of, the protein synthesis process, mistakes may happen. Such mistakes may occur for numerous reasons (including exposure to radiation, drugs, or viruses) or for no apparent reason. Minor variations in DNA are very common and occur in most people. Most variations do not affect subsequent copies of the gene. Mistakes that are duplicated in subsequent copies are called mutations.

Inherited mutations

Inherited mutations are those that may be passed on to offspring. Mutations can be inherited only when they affect the reproductive cells (sperm or egg). Mutations that do not affect reproductive cells affect the descendants of the mutated cell (for example, becoming a cancer) but are not passed on to offspring.

Mutations may be unique to an individual or family, and most harmful mutations are rare. Mutations that become so common that they affect more than 1% of a population are called polymorphisms (for example, the human blood types A, B, AB, and O). Most polymorphisms have little or no effect on the phenotype (the actual structure and function of a person's body).

Mutations may involve small or large segments of DNA. Depending on its size and location, the mutation may have no apparent effect or it may alter the amino acid sequence in a protein or decrease the amount of protein produced. If the protein has a different amino acid sequence, it may function differently or not at all. An absent or nonfunctioning protein is often harmful or fatal. For example, in phenylketonuria, a mutation results in the deficiency or absence of the enzyme phenylalanine hydroxylase. This deficiency allows the amino acid phenylalanine (absorbed from the diet) to accumulate in the body, ultimately causing severe intellectual disability.

In rare cases, a mutation introduces a change that is advantageous. For example, in the case of the sickle cell gene, when a person inherits two copies of the abnormal gene, the person will develop sickle cell disease. However, when a person inherits only one copy of the sickle cell gene (called a carrier), the person develops some protection against malaria (a blood infection). Although the protection against malaria can help a carrier survive, sickle cell disease (in a person who has two copies of the gene) causes symptoms and complications that may shorten life span.

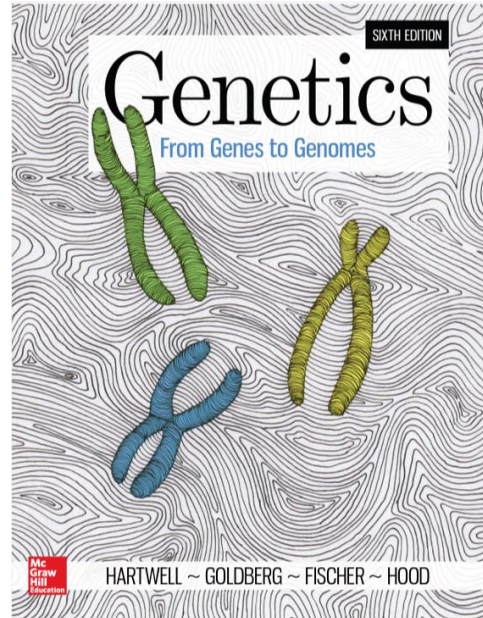
Natural selection

Natural selection refers to the concept that mutations that impair survival in a given environment are less likely to be passed on to offspring (and thus become less common in the population), whereas mutations that improve survival progressively become more common. Thus, beneficial mutations, although initially rare, eventually become common. The slow changes that occur over time caused by mutations and natural selection in an interbreeding population collectively are called evolution.



Review Questions

- Where is genetic information stored?
- Explain the differences in the chemical structure of DNA and RNA



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 5. Gene Expression



Contents

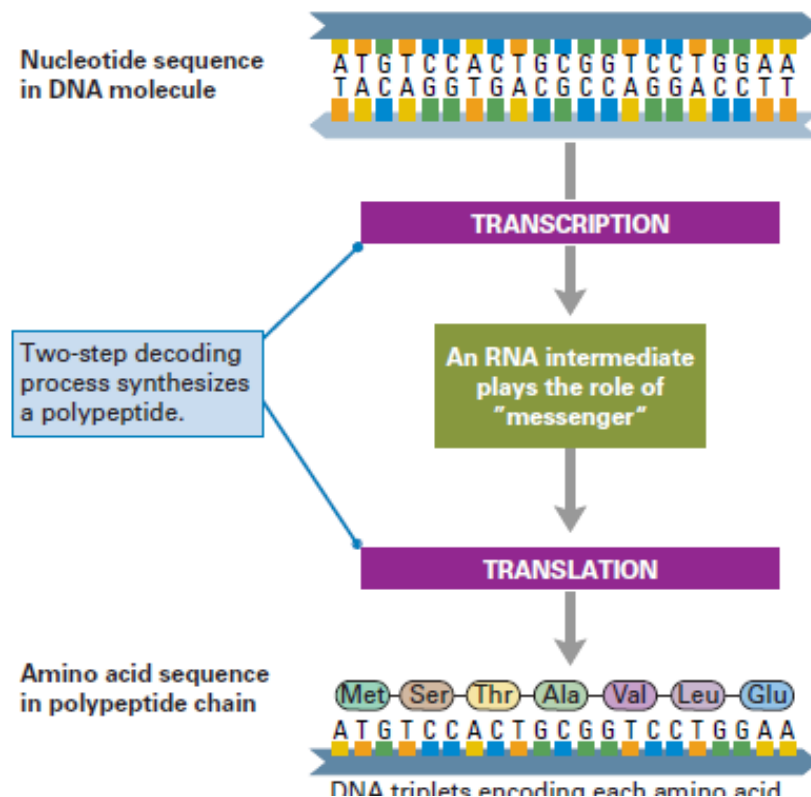
- Introduction
- The Central Dogma
- Transcription
- Translation
- The Genetic Code



Introduction

Watson and Crick were correct in proposing that the genetic information in DNA is contained in the sequence of bases in a manner analogous to letters printed on a strip of paper. In a region of DNA that directs the synthesis of a protein, the genetic code for the protein is contained in only one strand, and it is decoded in a linear order. A typical protein is made up of one or more polypeptide chains; each polypeptide chain consists of a linear sequence of amino acids connected end to end. For example, the enzyme PAH consists of four identical polypeptide chains, each 452 amino acids in length. In the decoding of DNA, each successive “code word” in the DNA specifies the next amino acid to be added to the polypeptide chain as it is being made. The amount of DNA required to code for the polypeptide chain of PAH is therefore $452 \times 3 = 1356$ nucleotide pairs. The entire gene is very much longer—about 90,000 nucleotide pairs. Only 1.5 percent of the gene is devoted to coding for the amino acids. The noncoding part includes some sequences that control the activity of the gene, but it is not known how much of the gene is involved in regulation.

The Central Dogma



There are 20 different amino acids. Only four bases code for these 20 amino acids, with each “word” in the genetic code consisting of three adjacent bases. For example, the base sequence ATG specifies the amino acid methionine (Met), TCC specifies serine (Ser), ACT specifies threonine (Thr), and GCG specifies alanine (Ala). There are 64 possible three-base combinations but only 20 amino acids because some combinations code for the same amino acid. For example, TCT, TCC, TCA, TCG, AGT, and AGC all code for serine (Ser), and CTT, CTC, CTA, CTG, TTA, and TTG all code for leucine (Leu). An example of the relationship between the base sequence in a DNA duplex and the amino acid sequence of the corresponding protein. This particular DNA duplex is the human sequence that codes for the first seven amino acids in the polypeptide chain of PAH.

DNA codes for protein not directly but indirectly through the processes of transcription and translation. The indirect route of information transfer,

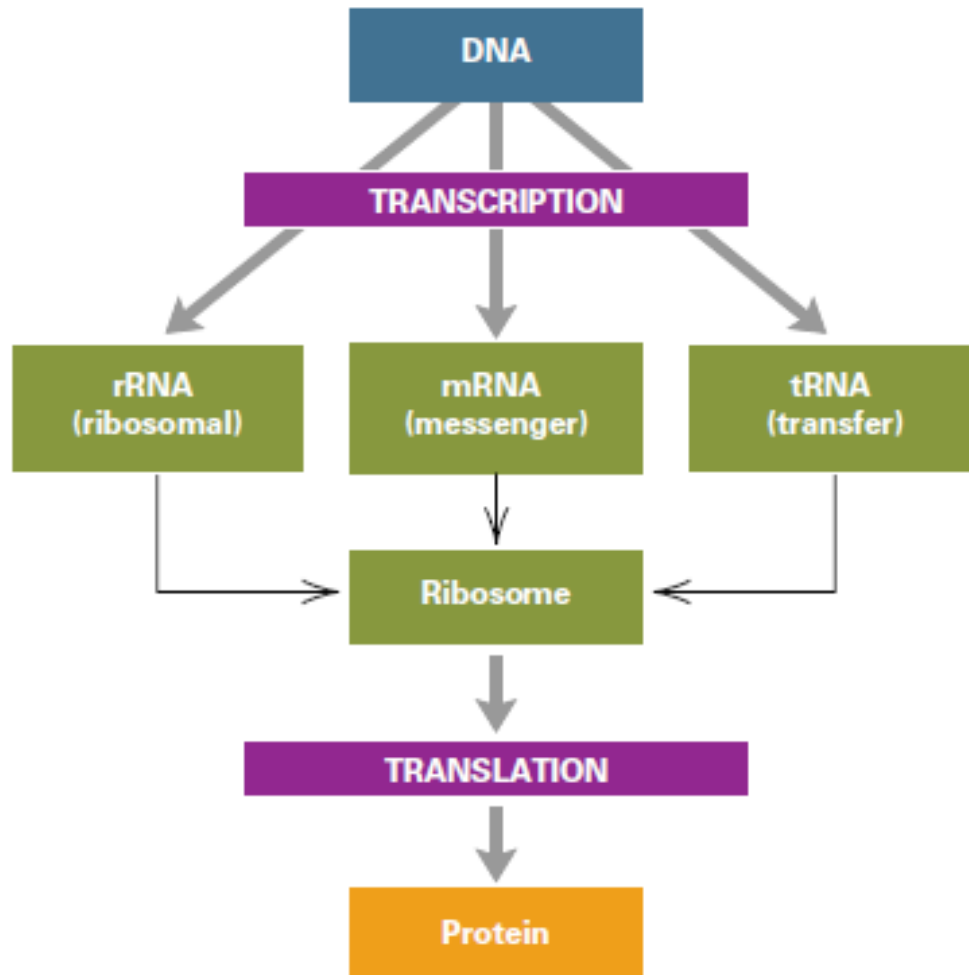
DNA → RNA → Protein

is known as the central dogma of molecular genetics. The term dogma means “set of beliefs”; it dates from the time the idea was put forward first as a theory. Since then the “dogma” has been confirmed experimentally but the term persists. The main concept in the central dogma is that DNA does not code for protein directly but rather acts through an intermediary molecule of ribonucleic acid (RNA). The structure of RNA is similar to, but not identical with, that of DNA. There is a difference in the sugar (RNA contains the sugar ribose instead of deoxyribose), RNA is usually single-stranded (not a duplex), and RNA contains the base uracil (U) instead of thymine (T), which is present in DNA. Actually, three types of RNA take part in the synthesis of proteins:

- A molecule of messenger RNA (mRNA), which carries the genetic information from DNA and is used as a template for polypeptide synthesis. In most mRNA molecules, there is a high proportion of nucleotides that actually code for amino acids. For example, the mRNA for PAH is 2400 nucleotides in length and codes for a polypeptide of 452 amino acids; in this case, more than 50 percent of the length of the mRNA codes for amino acids.
- Several types of ribosomal RNA (rRNA), which are major constituents of the cellular particles called ribosomes on which polypeptide synthesis takes place.
- A set of transfer RNA (tRNA) molecules, each of which carries a particular amino acid as well as a three-base recognition region that base-pairs with a group of three adjacent bases in the mRNA. As each tRNA participates in translation, its amino acid becomes the terminal subunit added to the length of the growing polypeptide chain. The tRNA that carries methionine is denoted tRNA^{Met}, that which carries serine is denoted tRNA^{Ser}, and so forth.

The central dogma is the fundamental principle of molecular genetics because it summarizes how the genetic information in DNA becomes expressed in the amino acid sequence in a polypeptide chain:

The sequence of nucleotides in a gene specifies the sequence of nucleotides in a molecule of messenger RNA; in turn, the sequence of nucleotides in the messenger RNA specifies the sequence of amino acids in the polypeptide chain.

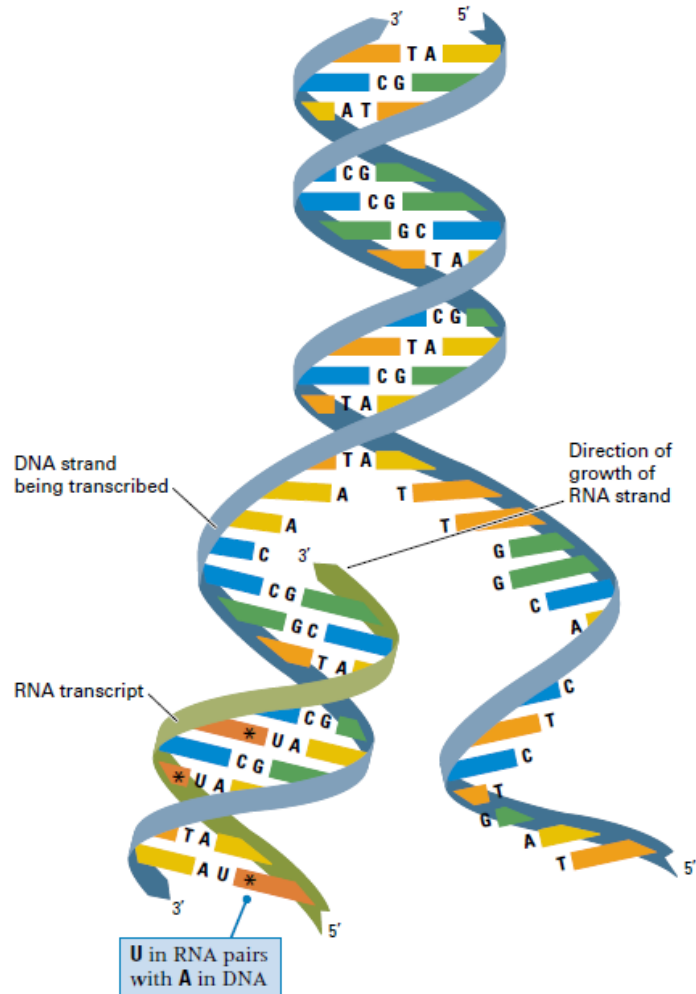


Given a process as conceptually simple as DNA coding for protein, what might account for the additional complexity of RNA intermediaries? One possible reason is that an RNA intermediate gives another level for control, for example, by degrading the mRNA for an unneeded protein. Another possible reason may be historical. RNA structure is unique in having both an informational content present in its sequence of bases and a complex, folded three-dimensional structure that endows some RNA molecules with catalytic activities. Many scientists believe that in the earliest forms of life, RNA served both for genetic information and catalysis. As evolution proceeded, the informational role was transferred to DNA and the catalytic role to protein. However, RNA became locked into its central location as a go-between in the processes of information transfer and protein synthesis.

This hypothesis implies that the participation of RNA in protein synthesis is a relic of the earliest stages of evolution— a “molecular fossil.” The hypothesis is supported by a variety of observations. For example,

- (1) DNA replication requires an RNA molecule in order to get started
- (2) an RNA molecule is essential in the synthesis of the tips of the chromosomes
- (3) some RNA molecules act to catalyze key reactions in protein synthesis

Transcription



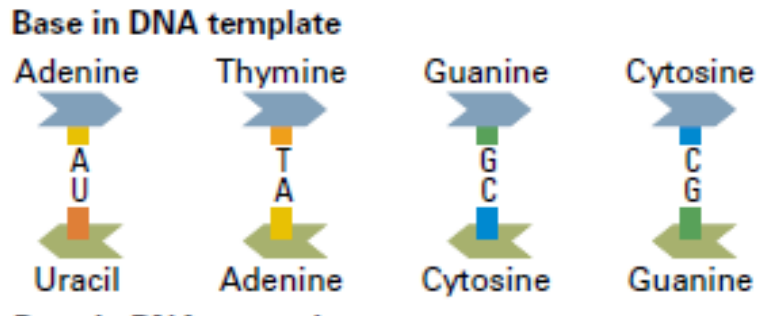
The manner in which genetic information is transferred from DNA to RNA. The DNA opens up, and one of the strands is used as a template for the synthesis of a complementary strand of RNA. The process of making an RNA strand from a DNA template is **transcription**, and the RNA molecule that is made is the transcript.

The base sequence in the RNA is complementary (in the Watson–Crick pairing sense) to that in the DNA template, except that U (which pairs with A) is present in the RNA in place of T. Each RNA strand has a polarity—a 5' end and a 3' end—and, as in the synthesis of DNA, nucleotides are added only to the 3' end of a growing RNA strand. Hence the 5' end of the RNA transcript is synthesized first, and transcription proceeds along the template DNA strand in the 3'-to-5' direction. Each gene includes nucleotide sequences that initiate and terminate transcription.



The RNA transcript made from any gene begins at the initiation site in the template strand, which is located “upstream” from the amino acid–coding region, and ends at the termination site, which is located “downstream” from the amino acid–coding region. For any gene, the length of the RNA transcript is very much smaller than the length of the DNA in the chromosome.

Translation

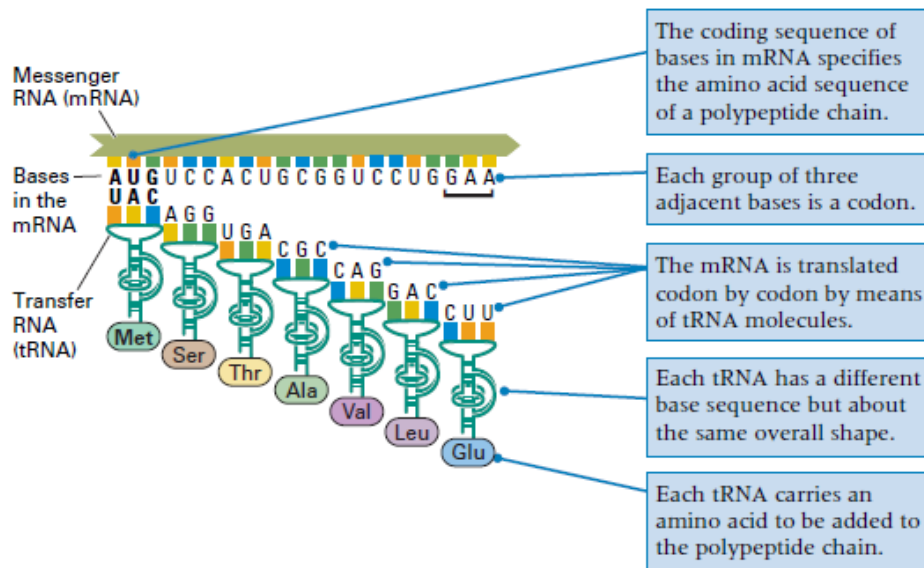


The correct amino acid is attached to the other end of the tRNA, and when the tRNA comes into line, the amino acid to which it is attached becomes the most recent addition to the growing end of the polypeptide chain.

The synthesis of a polypeptide under the direction of an mRNA molecule is known as translation. Although the sequence of bases in the mRNA codes for the sequence of amino acids in a polypeptide, the molecules that actually do the “translating” are the tRNA molecules. The mRNA molecule is translated in nonoverlapping groups of three bases called codons. For each codon in the mRNA that specifies an amino acid, there is one tRNA molecule containing a complementary group of three adjacent bases that can pair with those in that codon.

The role of tRNA in translation and can be described as follows:

The mRNA is read codon by codon. Each codon that specifies an amino acid matches with a complementary group of three adjacent bases in a single tRNA molecule. One end of the tRNA is attached to the correct amino acid, so the correct amino acid is brought into line.



The role of tRNA in translation and can be described as follows:

The mRNA is read codon by codon. Each codon that specifies an amino acid matches with a complementary group of three adjacent bases in a single tRNA molecule. One end of the tRNA is attached to the correct amino acid, so the correct amino acid is brought into line.

The Genetic Code

| Second nucleotide in codon | | | | | | | | | | | | | | | | | | |
|------------------------------------|---|-----|-----|---|---------------|-----|-----|---|-----------|-----|-------------|---|---------------|-------------|-----|------------|------------------------------------|----------|
| First nucleotide in codon (5' end) | U | | | | C | | | | A | | | | G | | | | Third nucleotide in codon (3' end) | |
| | U | UUU | Phe | F | Phenylalanine | UCU | Ser | S | Serine | UAU | Tyr | Y | Tyrosine | UGU | Cys | C | | Cysteine |
| | | UUC | Phe | F | Phenylalanine | UCC | Ser | S | Serine | UAC | Tyr | Y | Tyrosine | UGC | Cys | C | | Cysteine |
| | | UUA | Leu | L | Leucine | UCA | Ser | S | Serine | UAA | Termination | | UGA | Termination | | | | |
| | | UUG | Leu | L | Leucine | UCG | Ser | S | Serine | UAG | Termination | | UGG | Trp | W | Tryptophan | | |
| | C | CUU | Leu | L | Leucine | CCU | Pro | P | Proline | CAU | His | H | Histidine | CGU | Arg | R | | Arginine |
| | | CUC | Leu | L | Leucine | CCC | Pro | P | Proline | CAC | His | H | Histidine | CGC | Arg | R | | Arginine |
| | | CUA | Leu | L | Leucine | CCA | Pro | P | Proline | CAA | Gln | Q | Glutamine | CGA | Arg | R | | Arginine |
| | | CUG | Leu | L | Leucine | CCG | Pro | P | Proline | CAG | Gln | Q | Glutamine | CGG | Arg | R | | Arginine |
| | A | AUU | Ile | I | Isoleucine | ACU | Thr | T | Threonine | AAU | Asn | N | Asparagine | AGU | Ser | S | | Serine |
| | | AUC | Ile | I | Isoleucine | ACC | Thr | T | Threonine | AAC | Asn | N | Asparagine | AGC | Ser | S | | Serine |
| | | AUA | Ile | I | Isoleucine | ACA | Thr | T | Threonine | AAA | Lys | K | Lysine | AGA | Arg | R | | Arginine |
| | | AUG | Met | M | Methionine | ACG | Thr | T | Threonine | AAG | Lys | K | Lysine | AGG | Arg | R | | Arginine |
| | G | GUU | Val | V | Valine | GCU | Ala | A | Alanine | GAU | Asp | D | Aspartic acid | GGU | Gly | G | | Glycine |
| | | GUC | Val | V | Valine | GCC | Ala | A | Alanine | GAC | Asp | D | Aspartic acid | GGC | Gly | G | | Glycine |
| | | GUA | Val | V | Valine | GCA | Ala | A | Alanine | GAA | Glu | E | Glutamic acid | GGA | Gly | G | | Glycine |
| | | GUG | Val | V | Valine | GCG | Ala | A | Alanine | GAG | Glu | E | Glutamic acid | GGG | Gly | G | | Glycine |

Codon

Three-letter and single-letter abbreviations

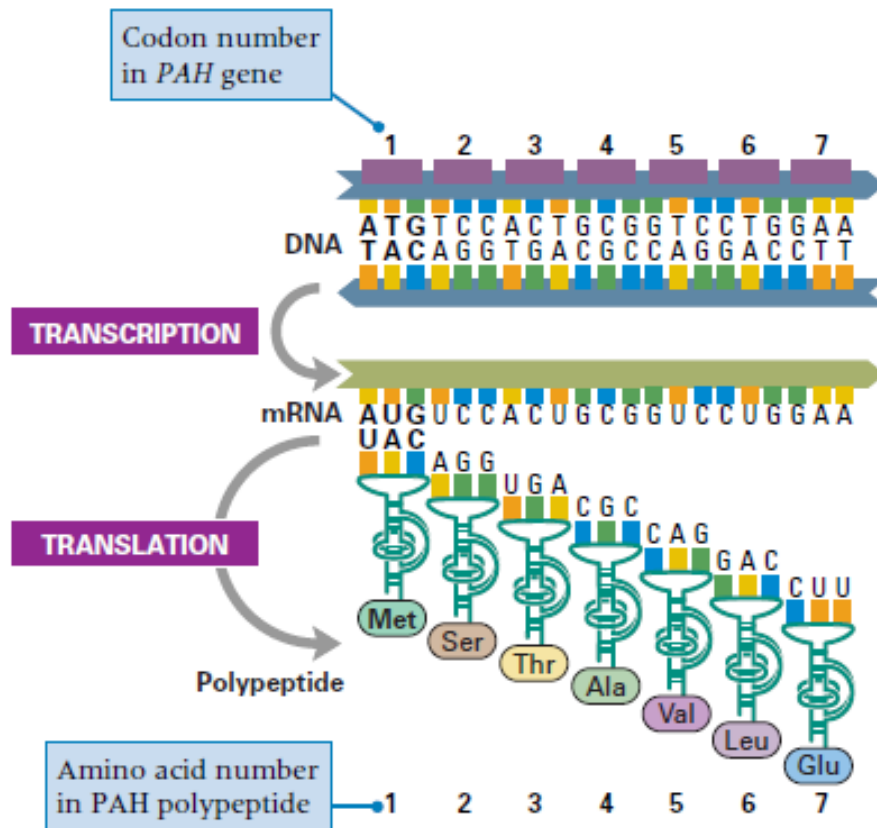
ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

In addition to the 61 codons that code only for amino acids, there are four codons that have specialized functions:

- The codon AUG, which specifies Met (methionine), is also the “start” codon for polypeptide synthesis. The positioning of a tRNA^{Met} bound to AUG is one of the first steps in the initiation of polypeptide synthesis, so all polypeptide chains begin with Met. (Many polypeptides have the initial Met cleaved off after translation is complete.) In most organisms, the tRNA^{Met} used for initiation of translation is the same tRNA^{Met} used to specify methionine at internal positions in a polypeptide chain.
- The codons UAA, UAG, and UGA are each a “stop” that specifies the termination of translation and results in release of the completed polypeptide chain from the ribosome. These codons do not have tRNA molecules that recognize them but are instead recognized by protein factors that terminate translation.

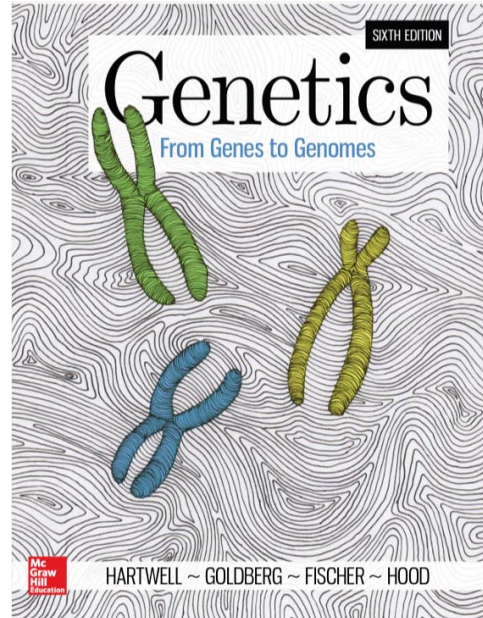


The central dogma in action. The DNA that encodes PAH serves as a template for the production of a messenger RNA, and the mRNA serves to specify the sequence of amino acids in the PAH polypeptide chain through interactions with the ribosome and tRNA molecules.



Review Questions

- Explain the process of gene expression.
- Explain the process of protein synthesis in a living cell.



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 1. Molecular Genetics

Lesson 6. Mutation, Protein Folding and Stability



Contents

- Introduction
- Types of DNA Mutations and Their Impact
- Types of Changes in DNA
- Protein Folding and Stability



Introduction

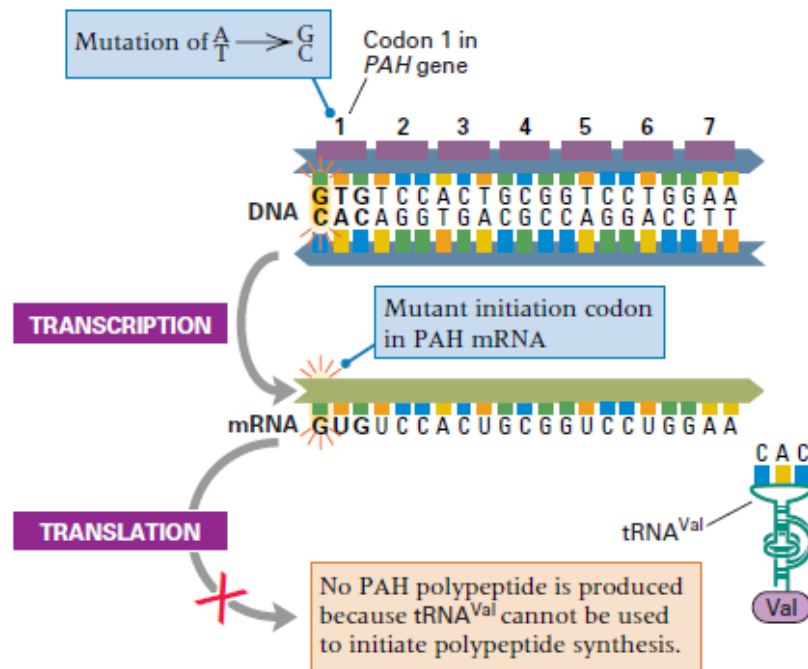
The term **mutation** refers to any heritable change in a gene (or, more generally, in the genetic material) or to the process by which such a change takes place. One type of mutation results in a change in the sequence of bases in DNA. The change may be simple, such as the substitution of one pair of bases in a duplex molecule for a different pair of bases. For example, a CG pair in a duplex molecule may mutate to TA, AT, or GC. The change in base sequence may also be more complex, such as the deletion or addition of base pairs. Geneticists also use the term **mutant**, which refers to the result of a mutation. A mutation yields a mutant gene, which in turn produces a mutant mRNA, a mutant protein, and finally a mutant organism that exhibits the effects of the mutation—for example, an inborn error of metabolism.

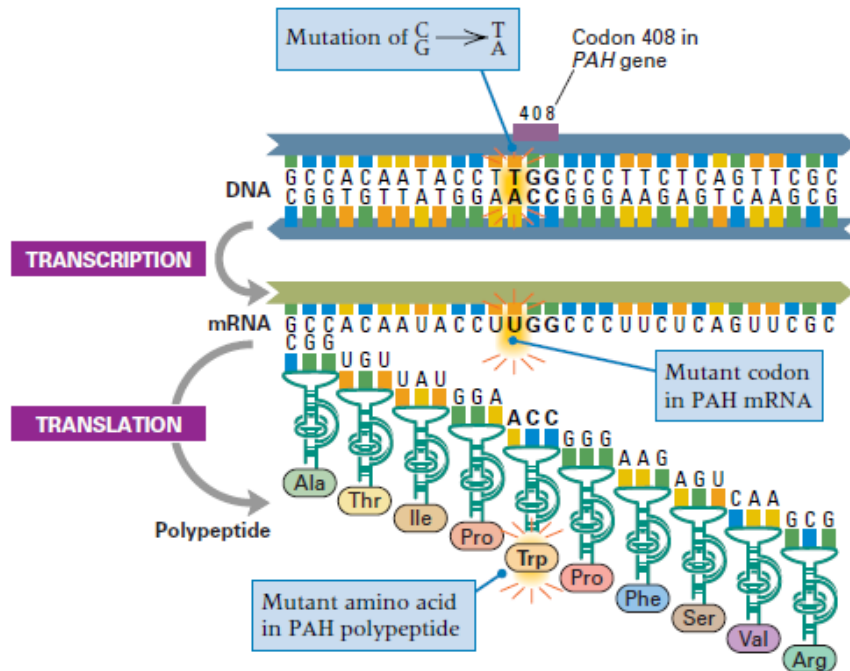
When do genetic mutations happen?

Genetic mutations occur during cell division when your cells divide and replicate. There are two types of cell division:

- **Mitosis:** The process of making new cells for your body. During mitosis, your genes instruct your cells to split into two by making a copy of your chromosomes.
- **Meiosis:** The process of making egg and sperm cells for the next generation. During meiosis, chromosomes copy themselves with half the amount of chromosomes as the original (from 46 to 23). That's how you're able to get your genetic material equally from each parent.

DNA from patients from all over the world who have phenylketonuria has been studied to determine what types of mutations are responsible for the inborn error. There are a large variety of mutant types. More than 400 different mutations have been described in the gene for PAH. In some cases part of the gene is missing, so the genetic information to make a complete PAH enzyme is absent. In other cases the genetic defect is more subtle, but the result is still either the failure to produce a PAH protein or the production of a PAH protein that is inactive.





One PAH mutant that is quite common is designated R408W, which means that codon 408 in the PAH polypeptide chain has been changed from one coding for arginine (R) to one coding for tryptophan (W). This mutation is one of the four most common among European Caucasians with PKU.

What are genetic disorders?

A genetic disorder is a condition caused by changes in your genome, or the genetic material present in a human. It includes your DNA, genes and chromosomes. Several factors cause genetic conditions, including:

- Mutation of one gene (monogenic).
- Mutation of multiple genes (multifactorial inheritance).
- Mutation of one or more chromosomes.
- Environmental factors (chemical exposure, UV rays) that change your genetic makeup.

You can inherit the genetic condition from your parents (if it's germ cell DNA in the sperm or egg) or the genetic condition can happen randomly, without having a history of the genetic condition in your family.

What are common genetic disorders?

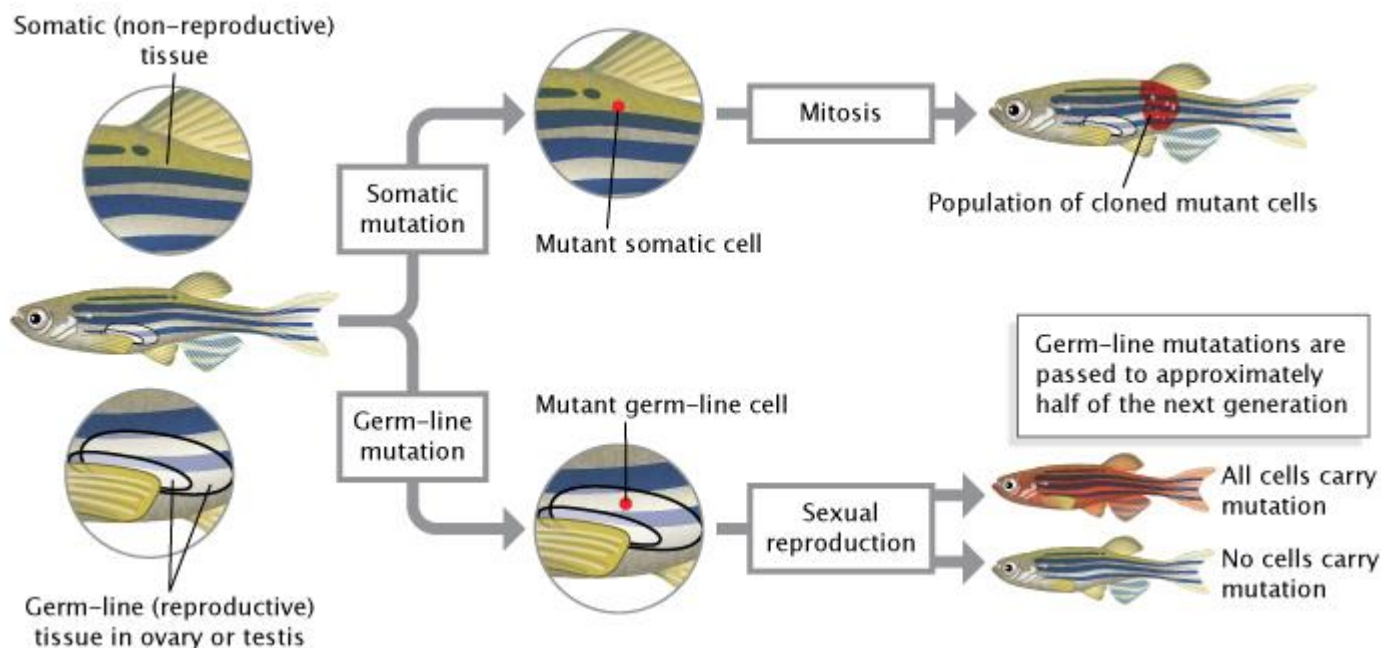
There are thousands of genetic conditions that exist. Some of the most common genetic conditions are:

- Alzheimer's disease.
- Some cancers.
- Cystic fibrosis.
- Down syndrome.
- Sickle cell disease.

The Relationship Between Mutations and Polymorphisms

While a mutation is defined as any alteration in the DNA sequence, biologists use the term "single nucleotide polymorphism" (SNP) to refer to a single base pair alteration that is common in the population. Specifically, a polymorphism is any genetic location at which at least two different sequences are found, with each sequence present in at least 1% of the population. Note that the term "polymorphism" is generally used to refer to a normal variation, or one that does not directly cause disease. Moreover, the cutoff of at least 1% prevalence for a variation to be classified as a polymorphism is somewhat arbitrary; if the frequency is lower than this, the allele is typically regarded as a mutation.

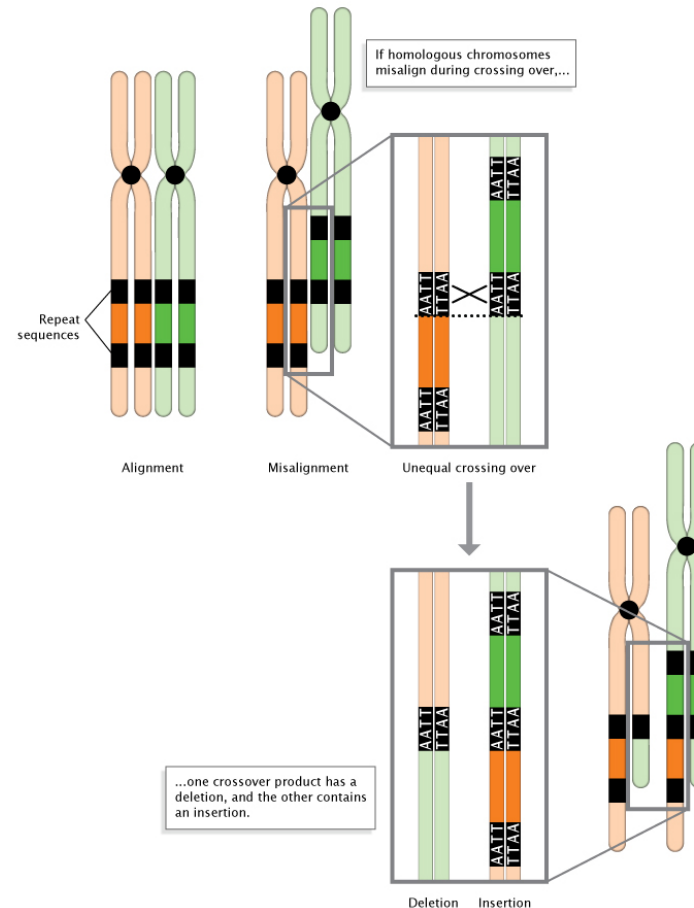
Types of Changes in DNA



Types of DNA Mutations and Their Impact

| Class of Mutation | Type of Mutation | Description | Human Disease(s) Linked to This Mutation |
|-----------------------|--------------------------------|---|---|
| Point mutation | Substitution | One base is incorrectly added during replication and replaces the pair in the corresponding position on the complementary strand | Sickle-cell anemia |
| | Insertion | One or more extra nucleotides are inserted into replicating DNA, often resulting in a frameshift | One form of beta-thalassemia |
| | Deletion | One or more nucleotides is "skipped" during replication or otherwise excised, often resulting in a frameshift | Cystic fibrosis |
| Chromosomal mutation | Inversion | One region of a chromosome is flipped and reinserted | Opitz-Kaveggia syndrome |
| | Deletion | A region of a chromosome is lost, resulting in the absence of all the genes in that area | Cri du chat syndrome |
| | Duplication | A region of a chromosome is repeated, resulting in an increase in dosage from the genes in that region | Some cancers |
| | Translocation | A region from one chromosome is aberrantly attached to another chromosome | One form of leukemia |
| Copy number variation | Gene amplification | The number of tandem copies of a locus is increased | Some breast cancers |
| | Expanding trinucleotide repeat | The normal number of repeated trinucleotide sequences is expanded | Fragile X syndrome, Huntington's disease |

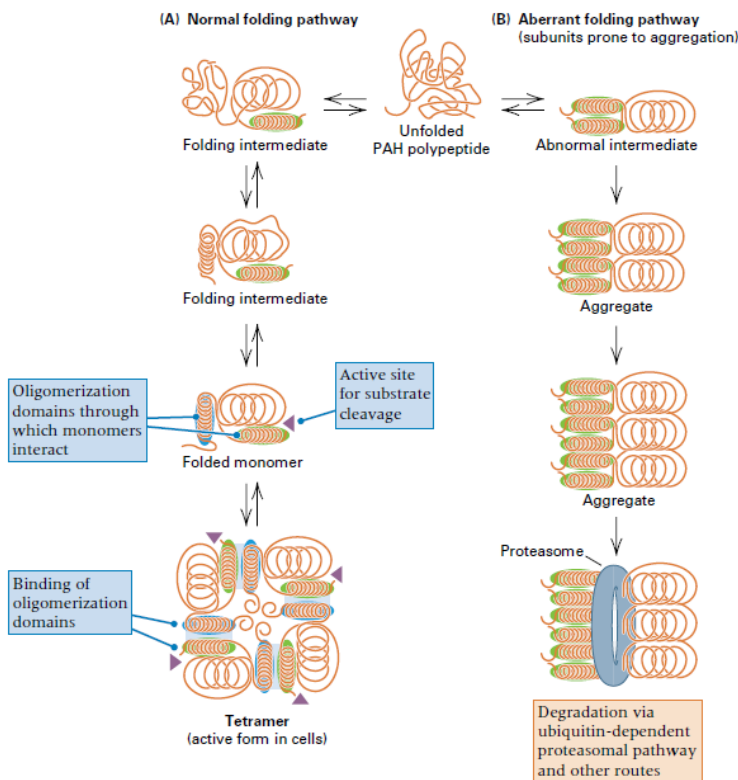
Unequal crossing-over during meiosis



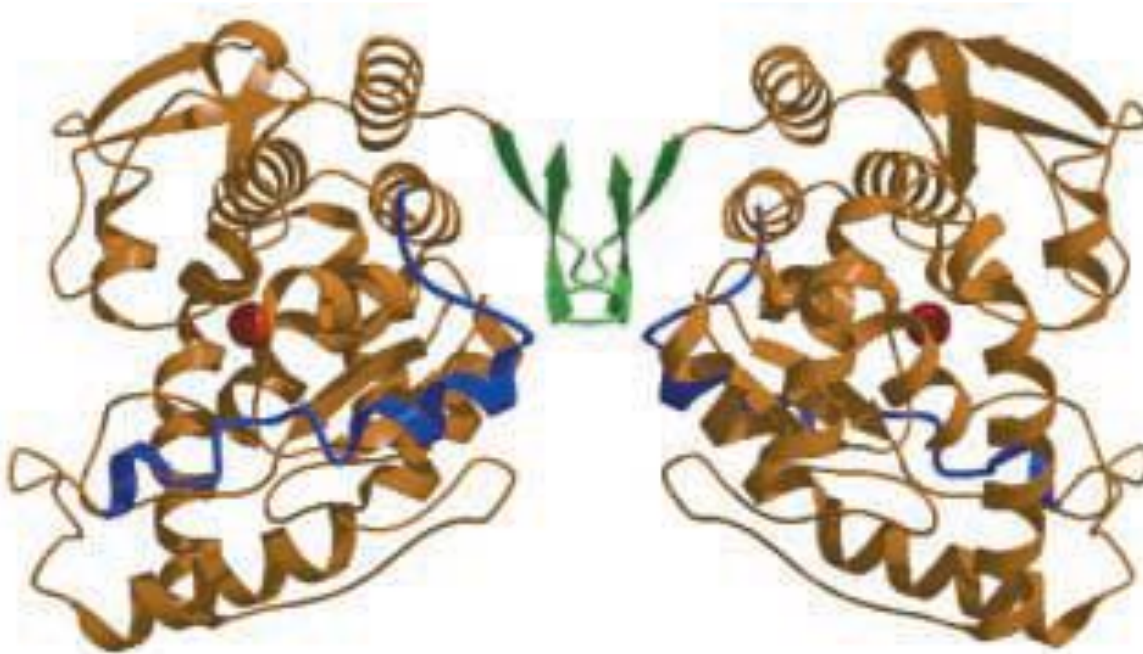
How Mutations Occur

- Mutations and the Environment
- Mutations Caused by Chemicals
- Spontaneous Mutations
- Errors During DNA Replication

Protein Folding and Stability



Some amino acid replacements perturb the ability of a protein to fold properly. (A) Normal folding in phenylalanine hydroxylase forms the active tetramer. (B) Abnormal folding of a mutant polypeptide chain results in the formation of polypeptide aggregates, which are progressively cleaved into the constituent amino acids through a ubiquitin-dependent proteosomal degradation pathway.



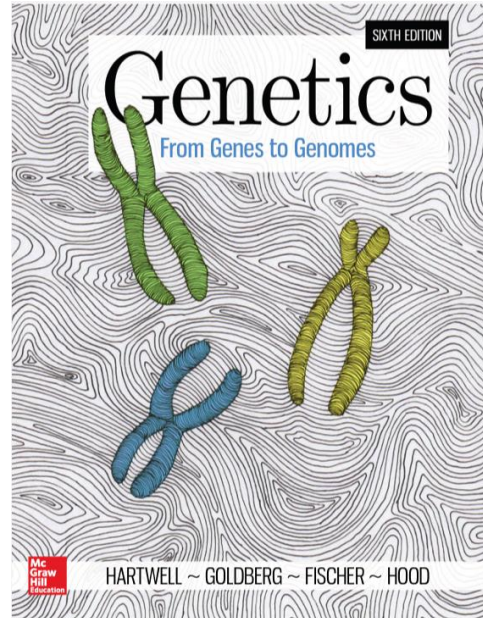
Three-dimensional structure of two of the four subunits in the active tetramer of phenylalanine hydroxylase. The chains of amino acids are represented as a sequence of curls, loops, and flat arrows, which represent different types of local structure. The oligomerization (in this case, tetramerization) domain is shown in green, and the catalytic domain containing the active site is shown in gold

Figure 1



Review Questions

- When do genetic mutations happen?
- What are genetic disorders?
- How Mutations Occur?



Leland Hartwell, Michael Goldberg, Janice Fischer, Lee Hood, Charles F. Aquadro, Bruce Bejcek. (2015) Genetics : from genes to genomes. McGraw-Hill Education, New York, USA



Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 1. Theoretical Modeling in Genomics



Contents

- Introduction
- Data-driven: Bioinformatics
- Flux balance analysis
- Ordinary differential equations
- Stochastic



Introduction

The aim of research in theoretical Modeling is to understand the inner workings of biological processes on a fundamental level, and from this to formulate accurate predictions leading to the design of experiments and applications. Hence, theoretical modeling tends to complement experimental research by offering new perspectives. Also known as modeling, this vivid topic may involve simulating biological systems *in silico*, analyzing large amounts of experimental data and building analytical theories to predict measurable phenomena.

Data-driven: Bioinformatics

The diversity of living systems is extreme. As of 2011, the number of different species of eukaryotes alone was estimated to be roughly (8.7 ± 1.3) million. These, in turn, may each contain thousands of genes in their DNA. As a result of this diversity, when studying living systems, one often ends up with vast amounts of data. Analyzing them and retrieving useful information has become a field in of itself: Bioinformatics.

Data-driven: Bioinformatics

Bioinformatics is applied in a wide range of areas. Some of the most important are the following:

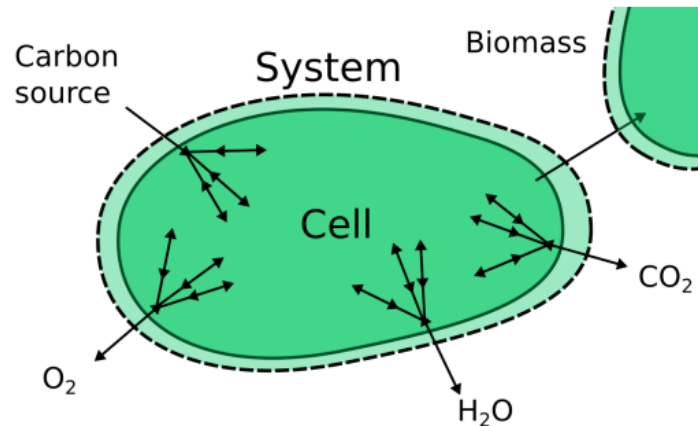
- Phylogenetics: the study of evolutionary relationships between genes and organisms.
- Omics: the characterization and quantification of biomolecules.
- Systems biology: the exploration of biological systems by studying the interactions between their components.
- Functional annotation: the characterization of genes and their function.
- Protein structure prediction and homology modeling: the study of the relationship between amino acid chain composition and the resulting three-dimensional conformation.
- Sequence alignment: the analysis of similarities between two or more DNA, RNA or protein sequences.

Flux balance analysis

Once there is enough data on a system, a model describing its properties can be built. However, the data available may at times not be sufficient yet for formulating this in a dynamic fashion. A common example for an alternative to such dynamic approaches, in the area of genome-scale metabolic models, is the so-called flux balance analysis method.

Flux balance analysis

Genome-scale models are meant to simulate a whole metabolic pathway or even an entire organism. As a result, they include many types of coupled chemical reactions and metabolites. These include so-called exchange reactions, which represent the import or export of metabolites, as well as reactions inside the system, which produce or consume the exchange metabolites and other intermediates.



Sketch of a genome-scale system of a biological cell. The arrows originating from or leading outside of the cell represent exchange reactions, while the arrows, which are entirely inside the cell, correspond to intracellular reactions.

Flux balance analysis

To perform dynamical simulations with systems of this scale and level of detail, one would encounter two substantial difficulties:

- First, computationally speaking, solving a partially coupled system of thousands of dynamic reaction equations is rather expensive.
- Second, all of the equations require empirical parameters, such as reaction rates, for many of which there is no available experimental data.

The flux balance analysis approach bypasses both of these problems by making two assumptions: steady-state conditions and optimization of the fluxes towards an objective. As a consequence, this formulation is meant to assume a static system and not to simulate dynamics.

Ordinary differential equations

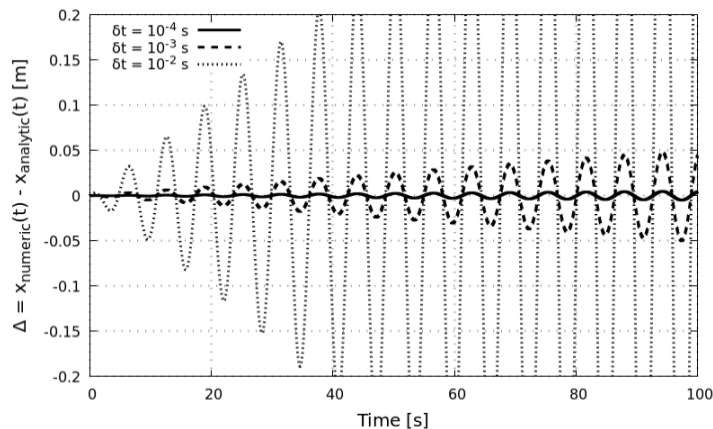
Dynamic deterministic algorithms belong to the category of those which simulate the time evolution of a system. They often work on a basis of one or more ordinary differential equations (ODEs). Solving such a model's equations provides, usually, a continuum of solutions, which distinguish themselves through their sole dependence on initial conditions. Once these initial conditions are specified, the dynamics is completely defined, and a unique solution can be found. In particular, this means that, excluding digitization and numerical errors, any simulation run with the same starting parameters results in exactly the same outcome.

Ordinary differential equations

If initial conditions are known, the time evolution can, in principle, be simulated with low computational effort. The trade-off lies in precision and stability. During each time step, the finite difference approximation leads to an error on the order δt^2 . As this accumulates, the numerical solution deviates further and further from the analytical one. This can be seen very well when comparing numerical solutions of the simple harmonic oscillator equation: $m d^2x/dt^2 = -kx$, with its analytical solution:

$$x(t) = A \cdot \cos\left(\sqrt{\frac{k}{m}} \cdot t + \phi\right),$$

where $A = 1$, $k = 1$ and $m = 1$ in this example.



Difference Δ between the analytic solution and numerical solution of the simple harmonic oscillator generated via the Euler algorithm. Each curve represents a different size of integration time step.

Stochastic

Some processes, for example the rolling of a die or the radioactive decay of an atom, cannot be described accurately by deterministic models, because they contain some inherent randomness. However, even though the exact dynamics of the system cannot be modeled, its average behavior is often of interest. It can also be that the computational cost of using an ODE-based approach is too high. These points support the use of stochastic models, and two common types of algorithms, Monte Carlo and Gillespie.

Monte Carlo algorithms

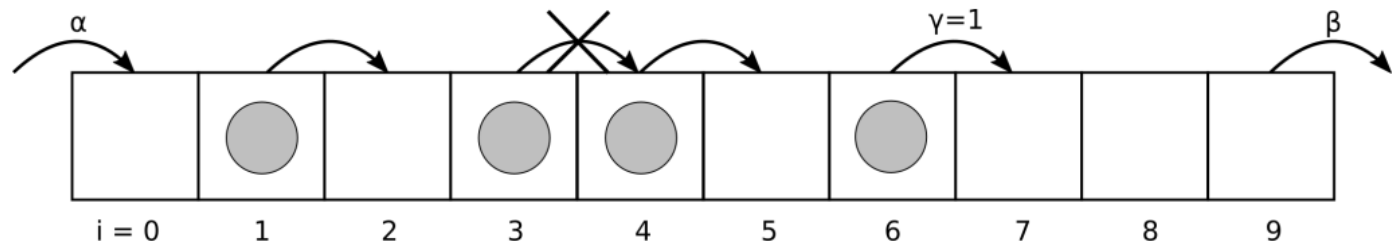
Monte Carlo algorithms are some of the first stochastic algorithms and possibly the most common.

The specific method varies, but the main point is stepwise sampling of random values from a distribution of system states. To put this in less abstract terms, consider the problem of finding the minimum of a function. A simple Monte Carlo algorithm would entail randomly picking and comparing function values from a given interval. If at each step the minimum is set as the smaller of the previous and current function value, a value close to the actual minimum is reached after a sufficient number of steps. A more refined Monte Carlo algorithm can also include biases towards certain system properties.

The Gillespie algorithm

The Gillespie algorithm can be viewed as an extension of Monte Carlo algorithms. Yet, it contains two key differences.

The first one is the fact that at each step the list of changes (also called reactions), which are currently possible within the rules and restrictions of the model, is updated. The new change to be made during the step is then randomly picked within this list. As a consequence, all changes selected can, for sure, be performed. This is different from the situation in regular Monte Carlo algorithms, where the randomly selected change is attempted but not always doable. A way to visualize the difference is to consider the so-called Totally Asymmetric Simple Exclusion Process (TASEP).





When simulating the system's behavior using a Monte Carlo algorithm, there are two choices. In the simplest algorithm, a lattice site is sampled from the list of all sites at each step. Then, it is checked if the site contains a particle, and finally, if the particle can move. As there may be many empty lattice sites, this can lead to a large number of rejected steps. When applying the second, more refined algorithm, the sampling is done only from the list of particles on the lattice. This reduces the number of un-doable steps, because empty lattice sites are no longer sampled. However, it is still possible that the sampled particle is unable to move, in which case the step is also not accepted.

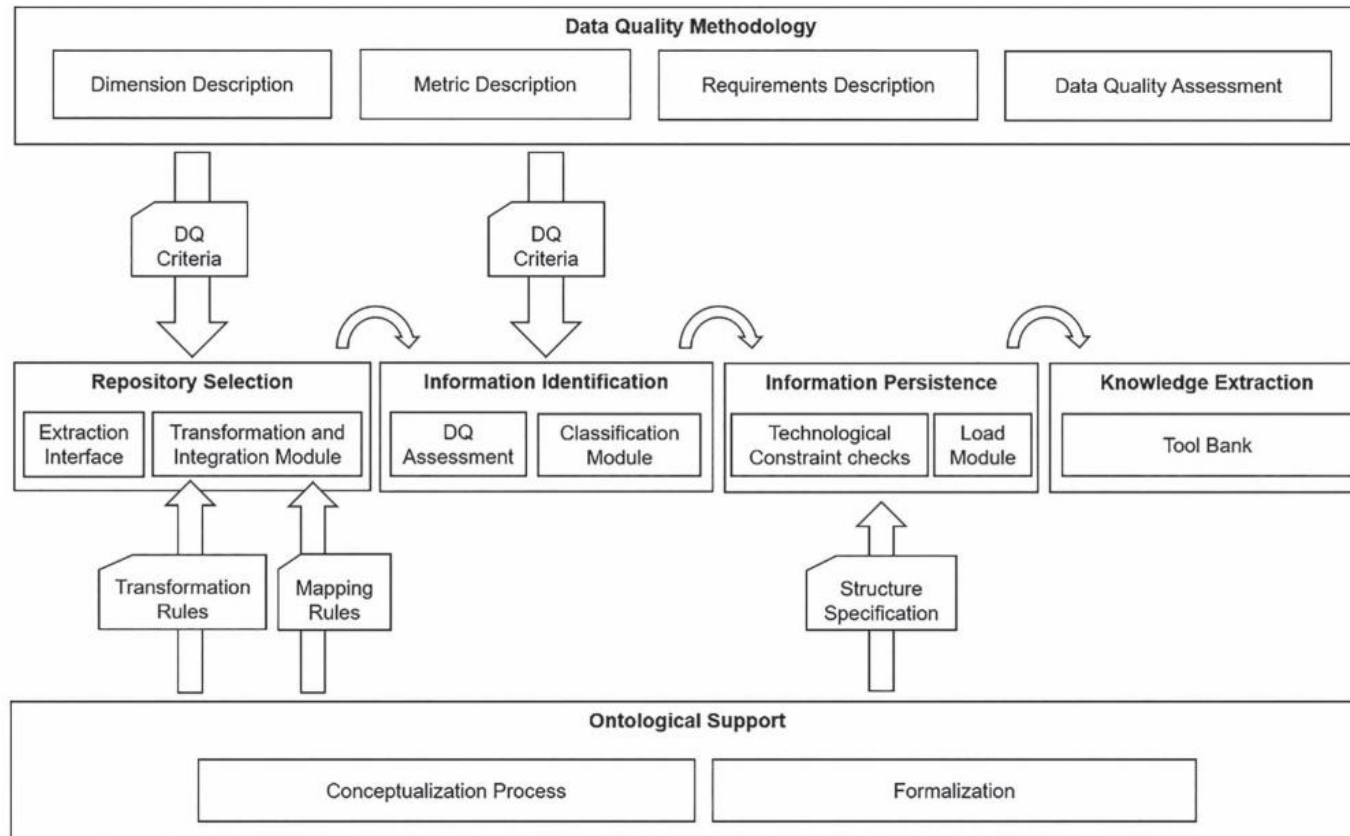
When using the Gillespie algorithm on the other hand, the sampling is done, not from the list of particles or lattice sites, but from the list of currently possible moves. In this way, any change made during a step is always possible and is therefore always accepted. This difference in sampling strategy can lead to drastic improvements in terms of computational effort. Using the example from Figure 10, the simple Monte Carlo algorithm would lead to sampling from a list of ten sites, seven of which would be rejected. The more refined algorithm would lead to sampling from a list of four particles, one of which would be rejected. Finally, in the Gillespie algorithm, the sampling would be done from a list of five moves, all of which would be accepted.

The second, and even more important difference, lies in the calculation of the "real" time. In contrast to regular Monte Carlo algorithms, in which each step lasts one unit of time, when using the Gillespie algorithm, at each step not only the change to the system, but also the time it takes, is calculated. This value is determined randomly, but the interval from which it is sampled is specified by using known parameters connected to the dynamics of the system. The consequence is that, on average, the "real" time inside the simulation corresponds to the time passing in the biological system. This is a considerable advantage, as it improves the comparability between the simulation and corresponding experimental data.

Conceptual modeling

With advances in genomic sequencing technology, a large amount of data is publicly available for the research community to extract meaningful and reliable associations among risk genes and the mechanisms of disease. However, this exponential growth of data is spread in over thousand heterogeneous repositories, represented in multiple formats and with different levels of quality what hinders the differentiation of clinically valid relationships from those that are less well-sustained and that could lead to wrong diagnosis. There is a conceptual models that play a key role to efficiently manage genomic data. These data must be accessible, informative and reliable enough to extract valuable knowledge in the context of the identification of evidence supporting the relationship between DNA variants and disease.

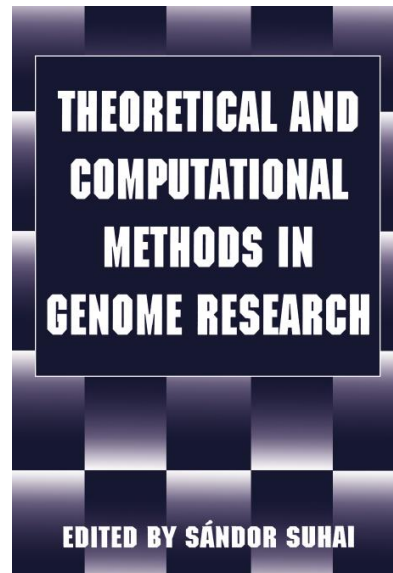
Framework for the management of genomic information





Review Questions

- Explain the methods of modelling in genomics.
- Explain the framework for the management of genomic information.



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York

Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 2. A Systematic Analysis of Gene Functions by the Metabolic Pathway Database



Contents

- Introduction
- Kyoto Encyclopedia of Genes and Genomes (KEGG)
- Reconstructing organism-specific pathways
- Comparing biological networks
- Metabolic pathway databases



Introduction

The genome sequencing projects of different organisms are fast producing catalogs of genes and gene products. The next obvious step is to understand functional implications, namely, to decipher both experimentally and computationally when, where, and how genes and molecules function in living organisms. In fact, our knowledge on the functioning of genes and molecules is also rapidly expanding owing to the advancement of experimental technologies in wide areas of molecular and cellular biology. In order to make full use of the information obtained by genome projects, it is essential that such functional data are properly computerized.

Introduction

| | LIGAND | KEGG | SwissProt | PIR | PDB |
|----------------------|--------|------|-----------|------|-----|
| Oxidoreductase (EC1) | 899 | 360 | 303 | 327 | 59 |
| Transferase (EC2) | 1002 | 306 | 332 | 319 | 47 |
| Hydrolase (EC3) | 992 | 172 | 428 | 370 | 107 |
| Lyase (EC4) | 332 | 160 | 129 | 126 | 21 |
| Isomerase (EC5) | 145 | 66 | 54 | 54 | 13 |
| Ligase (EC6) | 119 | 59 | 68 | 61 | 13 |
| Total | 3489 | 1123 | 1314 | 1257 | 260 |

LIGAND: enzymatic reactions

KEGG: metabolic pathways

SwissProt and PIR: amino acid sequences

PDB: 3D structures

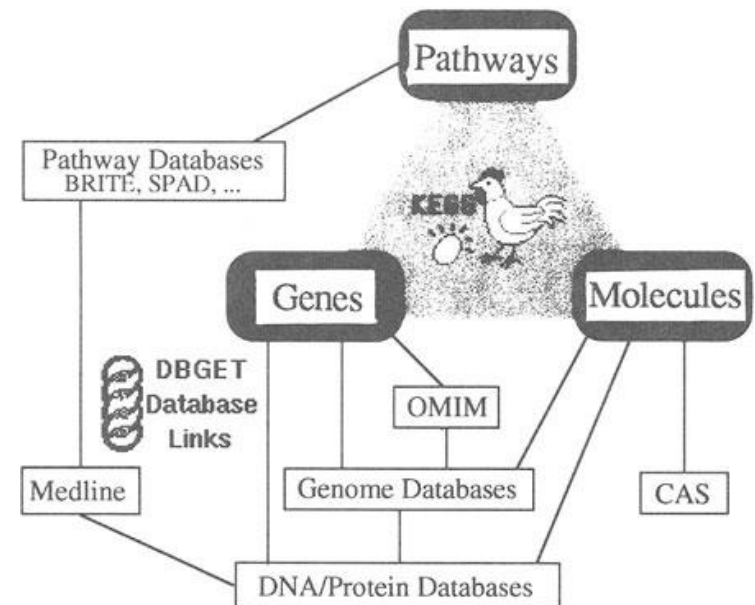
Kyoto Encyclopedia of Genes and Genomes (KEGG)

KEGG (Kyoto Encyclopedia of Genes and Genomes) is our attempt to computerize known pathways and to correlate them with gene catalogs. KEGG is constructed for use in the World Wide Web (WWW) and a preliminary version is made publicly available through the GenomeNet WWW server. KEGG consists of three types of data:

Genes - hierarchical text data

Molecules - hierarchical text data

Pathways - graphics data



Deductive Database for Metabolic Pathways

In order to compute possible pathways from a given list of enzymes, it is possible to organize a deductive database using CORAL. Suppose enzyme E catalyzes a chemical reaction with substrate X and product Y .

$$reaction(E, X, Y).$$

When the conversion of compound X to compound Y is a multistep process consisting of a number of enzymes, the enzymatic pathway is represented by:

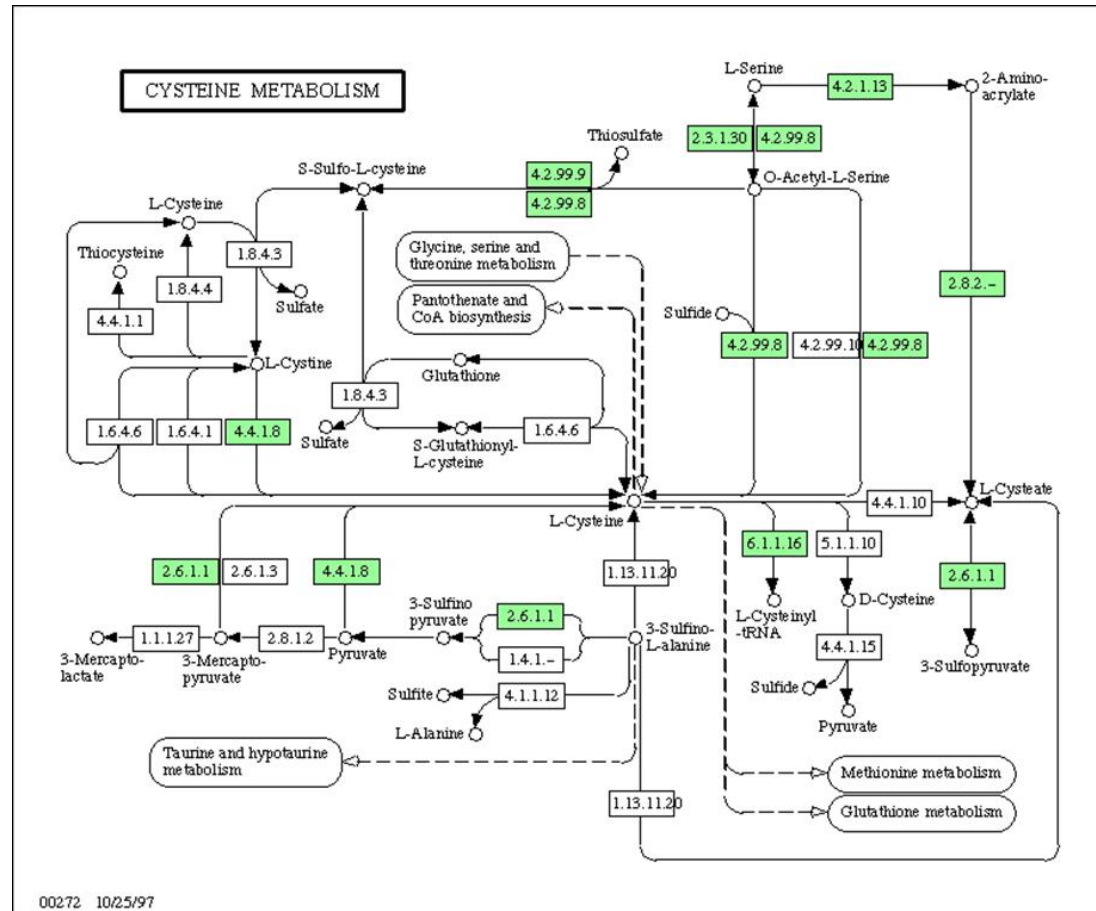
$$path(X, Y, [E]) \leftarrow reaction(E, X, Y).$$

$$path(X, Y, [E \mid EL]) \leftarrow reaction(E, X, Z), path(Z, Y, EL).$$

Given a catalog of enzyme genes, we can calculate possible pathways catalyzed by the gene products. Thus, for example, the correctness of gene identification in the genome sequencing project can be checked against the degree of completeness of the derived pathways. It is also possible to compare pathway diagrams and search for local similarities, in addition to analyzing sequence and structural similarities of enzymes on the pathway diagrams.

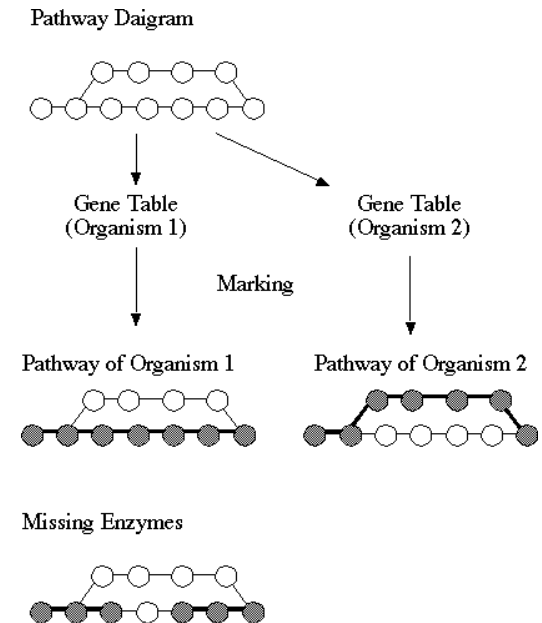


An example of the KEGG pathway diagram



Reconstructing organism-specific pathways

An organism-specific pathway is automatically generated by matching the reference pathway diagram and the gene catalog according to the EC number. When the gene for an enzyme exists in the gene catalog, the box representing the corresponding enzyme is marked by color on the pathway. The consecutive appearance of the colored boxes would then be considered an organism-specific pathway. In order for this procedure to be successful, the reference diagram should contain all known alternatives of reaction paths rather than just the consensus alone. In fact, we are learning from the complete genomes that different organisms have different sets of metabolic pathways that reflect the living environment and the strategy of adaptation. The automatic reconstruction of organism-specific pathways is a first step toward establishing a well verified set of reference diagrams.

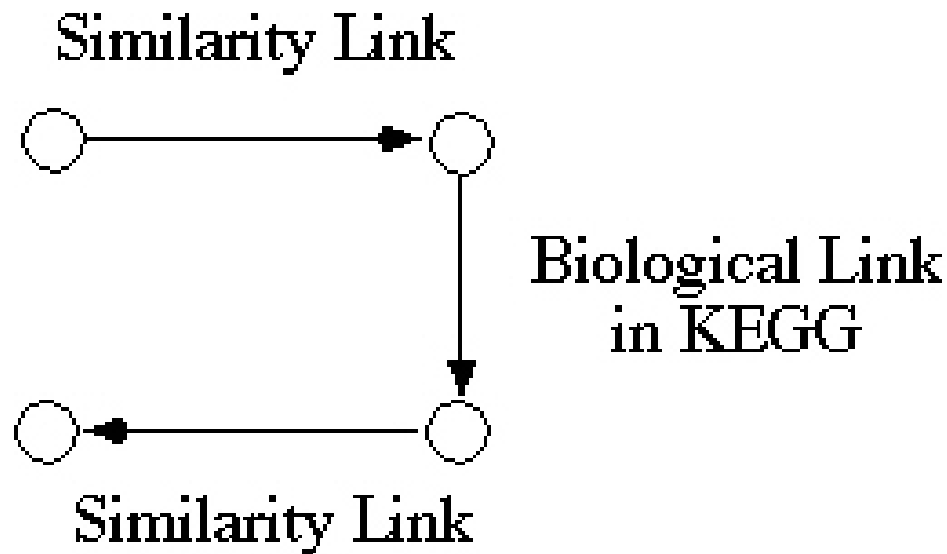


A schematic illustration of
generating organism-specific
pathways

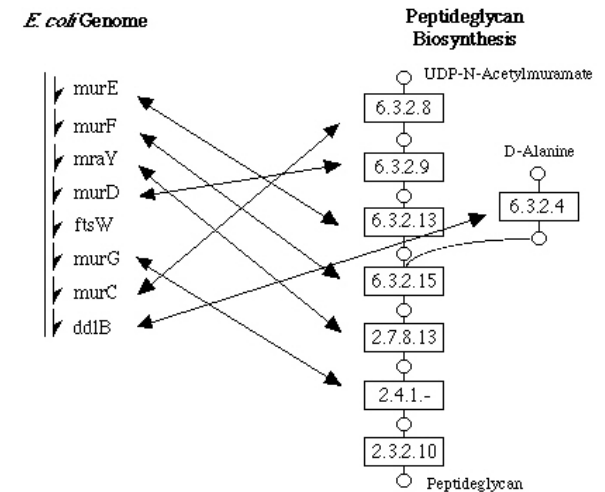
Computing paths of reasoning

| Type | Example |
|-----------------|--|
| factual link | cross-references in databases links between genes and functions |
| similarity link | sequence homology (orthology/paralogy) 3D similarity and complementarity |
| biological link | substrate-product relations in enzymatic reactions interacting molecules in a cell neighboring genes in the genome |

An example of the paths of reasoning that can be computed with a combination of various kinds of links.



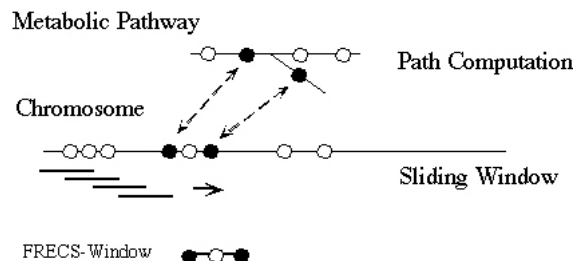
Chromosomal locations of genes is also informative for elucidating regulation or evolution of metabolism in yeast and bacteria. In KEGG the location and the order of enzyme genes in the genome can be examined graphically, as well as the location and the order of enzymes in the pathway. The example in Figure shows that a set of enzymes coded in close positions along the *E. coli* genome forms a block of reaction paths in the metabolic pathway, which is actually a general tendency as described below. Although the feature is not provided yet, the positional correlation in the genome, which is also represented by binary relations, will be included in the automatic biological reasoning process in KEGG.



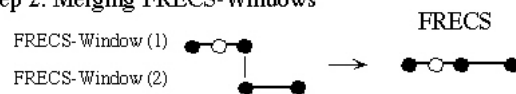
Comparing biological networks

(a)

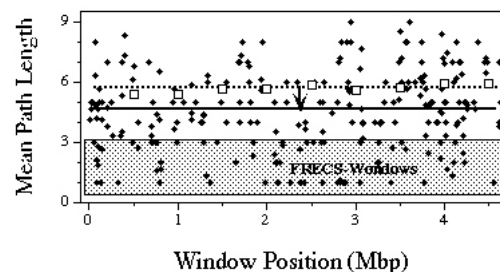
Step 1: Identification of FRECS-Windows



Step 2: Merging FRECS-Windows



(b)



(a) A functionally related enzymes coding segment (FRECS) is defined by the enzyme pairs that appear within a sliding window in the genome and that are separated by given path lengths in the metabolic pathway (see text). (b) The mean path length of enzyme pairs is plotted against the window position for two different window sizes: 1 Mbp (white boxes) and 10 kbp (black boxes). The dotted and solid lines show the average levels of the path length for 1 Mbp and 10 kbp windows, respectively, and an arrow is put to indicate the difference of the levels. FRECS corresponds, in this case, to the path length of 3 or less

Metabolic pathway databases

| Database/content | Reactions | Compounds | Pathways | Proteins | Version |
|------------------|-----------|-----------|----------|----------|---------|
| KEGG | 10307 | 17787 | 474838 | 6836 | 80.2 |
| MetaCyc | 15162 | 13585 | 2884 | 17090 | 20.5 |
| Reactome | 78927 | 88597 | 22072 | 120935 | 59 |
| PMN | 74013 | 2509 | 13117 | 279301 | 11.0 |

Although they are diverse in their scope, metabolic databases should be able to provide reliable information on four of the basic constituents of metabolic networks. These include biochemical reactions, enzymes catalyzing the reactions, pathways, and metabolites. Metabolites should be represented in their appropriate charged or neutral states in the databases. Similarly, the databases should incorporate reaction directionality information, elemental and charge balance and compartment to which the reaction and compounds belong to.

Reactome

Reactome is publically accessible, open-source, manually curated and peer-reviewed database of human pathways. The primary goal of Reactome is to provide molecular details of signal transduction, transport, metabolism, DNA replication and other cellular processes as an ordered network of molecular transformations of Homo sapiens. The latest version includes organisms from other domains of life. Users can exploit three important features of Reactome. They can browse, visualize and analyze reactions, metabolites, and enzymes in each pathway and download the information in various formats.

MetaCyc

MetaCyc is a highly curated database of experimentally validated metabolic pathways from all domains of life. It contains pathways derived from extensively large number of primary literature. MetaCyc aims to serve as a general reference database on metabolism. An important feature of MetaCyc is its pathway tool that can be used to computationally predict metabolic network models of any organism from a sequenced genome.

PMN

Plant metabolic network is a metabolic pathways database that hosts one reference database (PlantCyc) and 22 species-specific databases. All of them especially focus on plant metabolic pathways. At the center of PMN is PlantCyc, which is a metabolic pathway reference database containing more than 900 pathways, their catalytic enzymes and genes. Furthermore, PlantCyc contains compounds from over 350 plant species. The data source in PlantCyc covers pathways from experimentally validated literature and curated by PMN and its collaborators.

Comparison of the databases based on additional functions

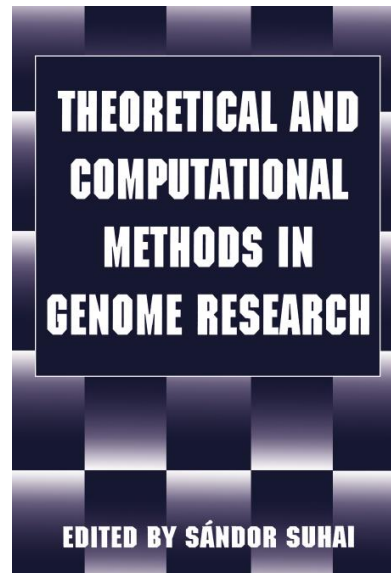
| Name | Tools | Function |
|----------|---------------------|---|
| Reactome | Pathway browser | <ul style="list-style-type: none"> • A tool for visualizing and interacting with Reactome biological pathways |
| | Analyze data | <ul style="list-style-type: none"> • Merges pathway identifier mapping, over-representation, and expression analysis tools into a single tabbed data analysis portal with integrated visualization and summary features, which can accept a gene, protein or small molecule list, or an expression dataset |
| | Species comparison | <ul style="list-style-type: none"> • Allows users to compare pathways between human and any of the other species inferred from Reactome by orthology |
| | Reactome FI network | <ul style="list-style-type: none"> • Cytoscape plugin designed to find network patterns related to cancer and other types of diseases |
| MetaCyc | Pathway tools | <ul style="list-style-type: none"> • Development of organism-specific databases • Metabolic reconstruction and modeling • Scientific visualization, web publishing • Visual analysis of gene expression and metabolomics datasets • Computational inferences • Comparative genome and pathway analyses • Analysis of biological networks |
| PMN | E2P2 | <ul style="list-style-type: none"> • Enzyme function annotation software. Predicts metabolic enzymes in a sequenced genome |
| | SAVI | <ul style="list-style-type: none"> • Pathway validation software. Processes predicted metabolic pathways using pathway metadata such as taxonomic distribution and key reactions and makes decisions about which pathways to keep, remove, or subject to manual validation |
| KEGG | PlantClusterFinder | <ul style="list-style-type: none"> • A pipeline to predict metabolic gene clusters from plant genomes |
| | KegHier | <ul style="list-style-type: none"> • Java application for browsing BRITE hierarchy files |
| | KegArray | <ul style="list-style-type: none"> • Java application for microarray data analysis |
| | KegDraw | <ul style="list-style-type: none"> • Java application for drawing compound and glycan structures |

Model repositories

Since the models come from diverse research groups or individuals, one of the current challenges related to metabolic network models is the inconsistencies in representing their components such as reaction, or compound names, and formulae, and symbols used to designate cellular compartments. Therefore, the online model repositories should go beyond simply serving as a repository to platforms for systematic analysis and standardization of these models, the efforts of BiGG and MetaNetX.org can be mentioned in this regard. Furthermore, the model repositories are expected to provide concise descriptions of the models such as simulation conditions, this may include in-silico growth media condition, energy requirements, objective function, and file compatibility information. Bearing this in mind, the following section will assess some of the most common metabolic network model databases.

Review Questions

- Explain how the organism-specific pathways could be reconstructed.
- Give some examples of metabolic pathway databases and what information could be obtained from them.



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York



Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 3. Polymer Dynamics of DNA, Chromatin, and Chromosomes



Contents

- Introduction
- The Genome as an Elastic Chain
- Brownian Dynamics and Monte-Carlo Simulations of Superhelical DNA
- The Chromatin Model



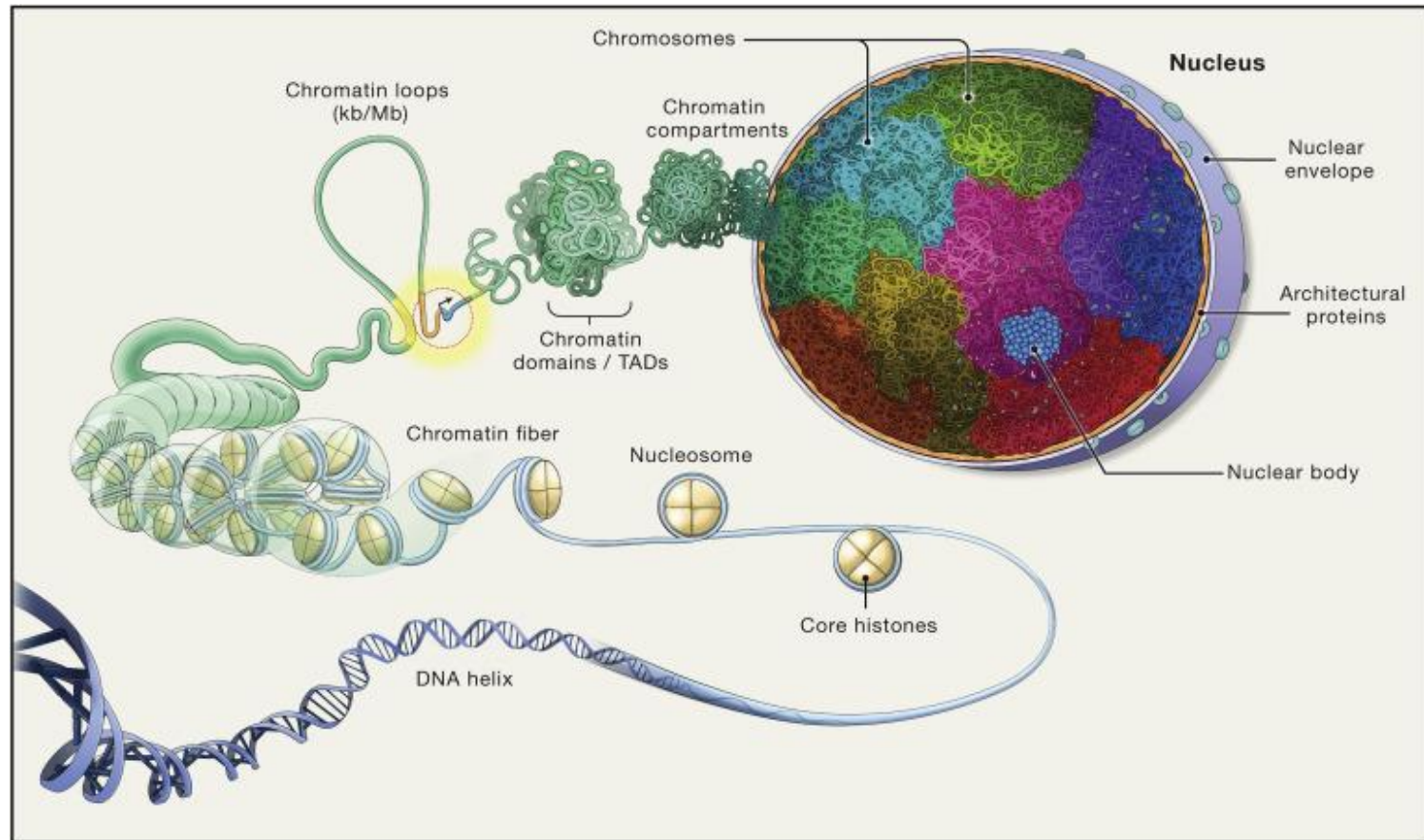
Introduction

The spatial organization of DNA in the eukaryotic nucleus plays a prominent role in determining gene activity. On the level of local DNA structure (some 100-1000 bp), the action of transcription factors is often mediated by DNA looping between the regulatory site and the promoter, and the structure of the intervening DNA has a decisive influence on the strength of the interaction between the transcription factor and the RNA polymerase initiation complex. On a higher level of organization, a great number of cases have been described where the expression of a gene is determined by the packaging of its DNA into a more or less compact chromatin structure.

The Genome as an Elastic Chain

The coarse-grained view of the genome as an elastic filament is sufficient as long as one is interested in questions like the kinetics of approach of distant pieces of DNA, the structural fluctuations of a supercoil or a chromatin fiber, or the three-dimensional organization of a whole chromosome. Such processes occur on length scales much greater than interatomic distances, typically $> 100\text{m}$, and on time scales much slower than molecular vibrations, typically $> 1\text{ ns}$; thus, atomic detail may be safely neglected.

The Genome as an Elastic Chain



The Genome as an Elastic Chain

For describing the large-scale motions of a long DNA molecule (several hundreds or thousands of base pairs), the elastic-filament approximation is also necessary: atomic level calculations on such a large system, including solvent water molecules, are unthinkable with present-day computing equipment.

The Genome as an Elastic Chain

For describing the large-scale motions of a long DNA molecule (several hundreds or thousands of base pairs), the elastic-filament approximation is also necessary: atomic level calculations on such a large system, including solvent water molecules, are unthinkable with present-day computing equipment.

- Elastic Parameters
- Topological Constraints
- Electrostatic Interactions
- Hydrodynamic Interactions

Brownian Dynamics and Monte-Carlo Simulations of Superhelical DNA

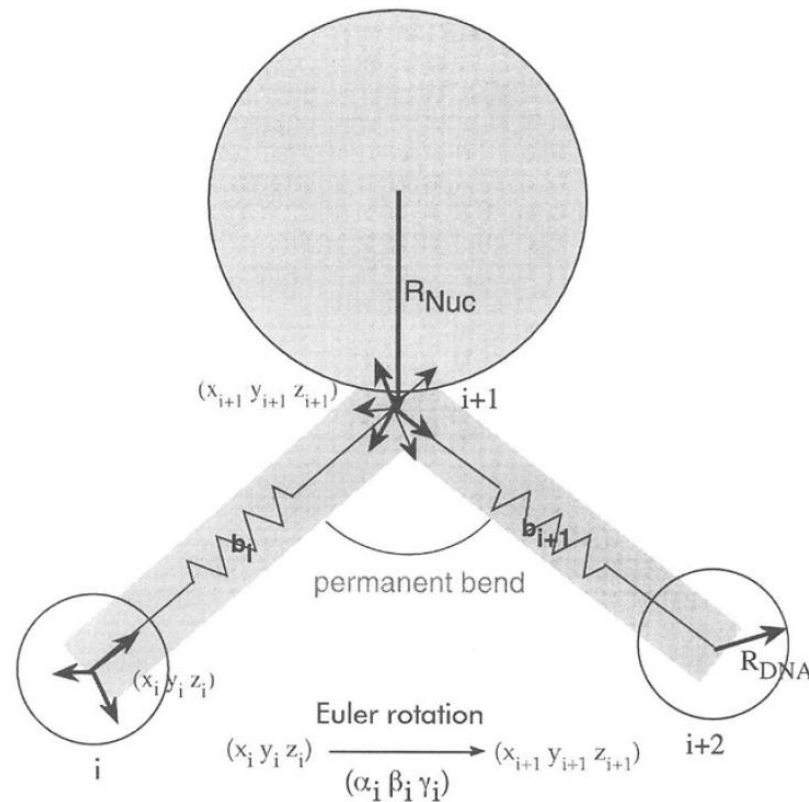
For calculating the dynamics of DNA, equations of motion for the segmented DNA chain have to be set up using the intramolecular interaction potentials described above and including the thermal motion through a random force. This is the Brownian Dynamics (BD) method which several groups applied in interpreting experimental data from fluorescence depolarization, dynamic light scattering, or triplet anisotropy decay. Circular DNA has recently also been modeled by a BD approach.

The Chromatin Model

Recent revisions of the existing structural data on chromatin fibers suggest a local 'zig-zag' arrangement of the nucleosomes which is folded up into a random-walk higher order structure. Randomness can occur either through non-uniform linker DNA length, causing a twist of one nucleosome with respect to the other, or by thermal fluctuations.

Based on these ideas, one can develop a BD model for the chromatin structure through some simple extensions of the elastic chain DNA model.

The Chromatin Model



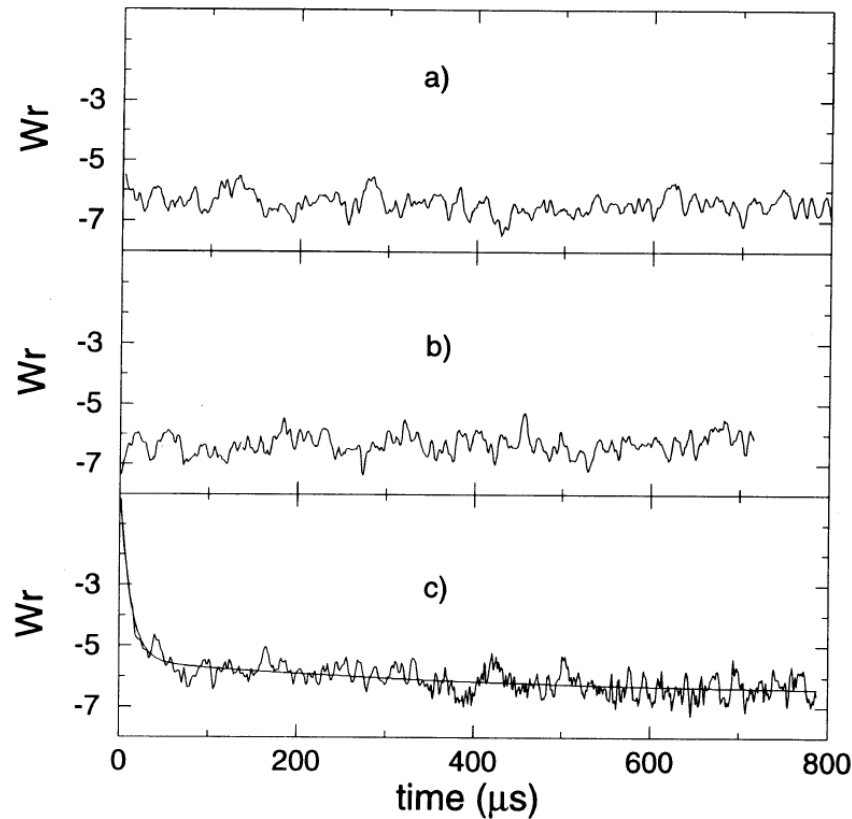


R-G Loop Model of Interphase Chromosomes

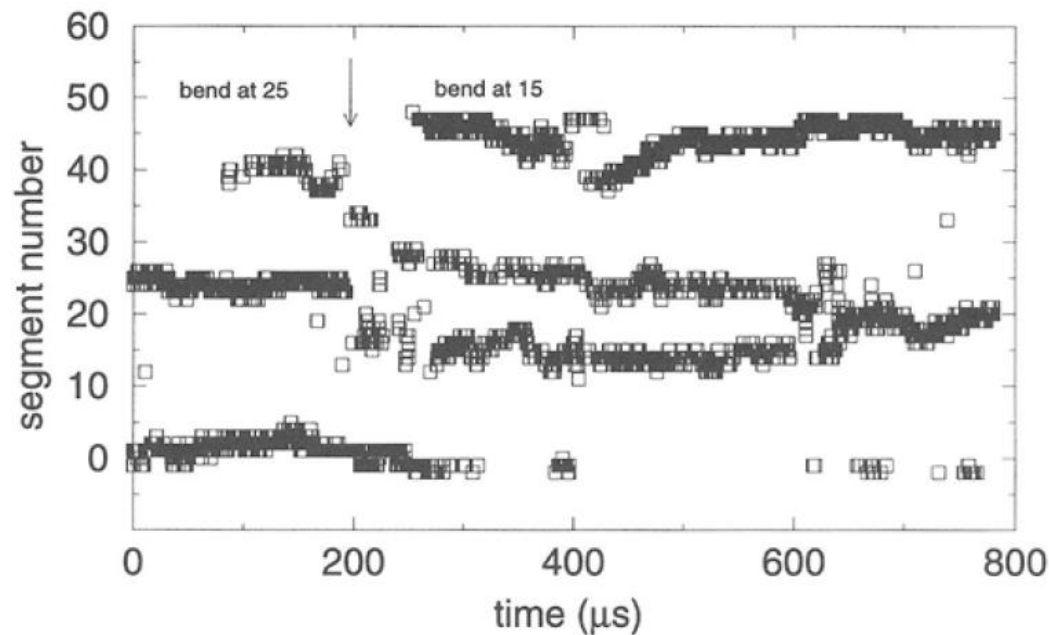
The basic structural element of the chromosome model is the chromatin fiber. Although the explicit structure of chromatin fibers in human interphase nuclei is still under discussion, their average geometrical properties on length scales above some thousand base pairs can be described by a polymer consisting of rigid segments of a certain length (Kuhn length) and an excluded volume interaction representing the diameter of the chromatin fiber. Motivated by the different properties of bands, e.g. density of genes or time of replication, it is assumed a different organization of chromosome bands in interphase as well. Each 'activated' band (e.g. a gene rich R-band) forms a single giant loop in the 3-10 Mbp range, while chromatin within 'inactivated' bands (e.g. G- or C-bands) is folded into several small loops about 100 kbp each.



Writhing kinetics for trajectories starting from a Monte Carlo-generated conformation flat circle (a), a straight interwound structure (b) and a flat circle (c). The straight line in c) is a double-exponential fit to the kinetic curve.

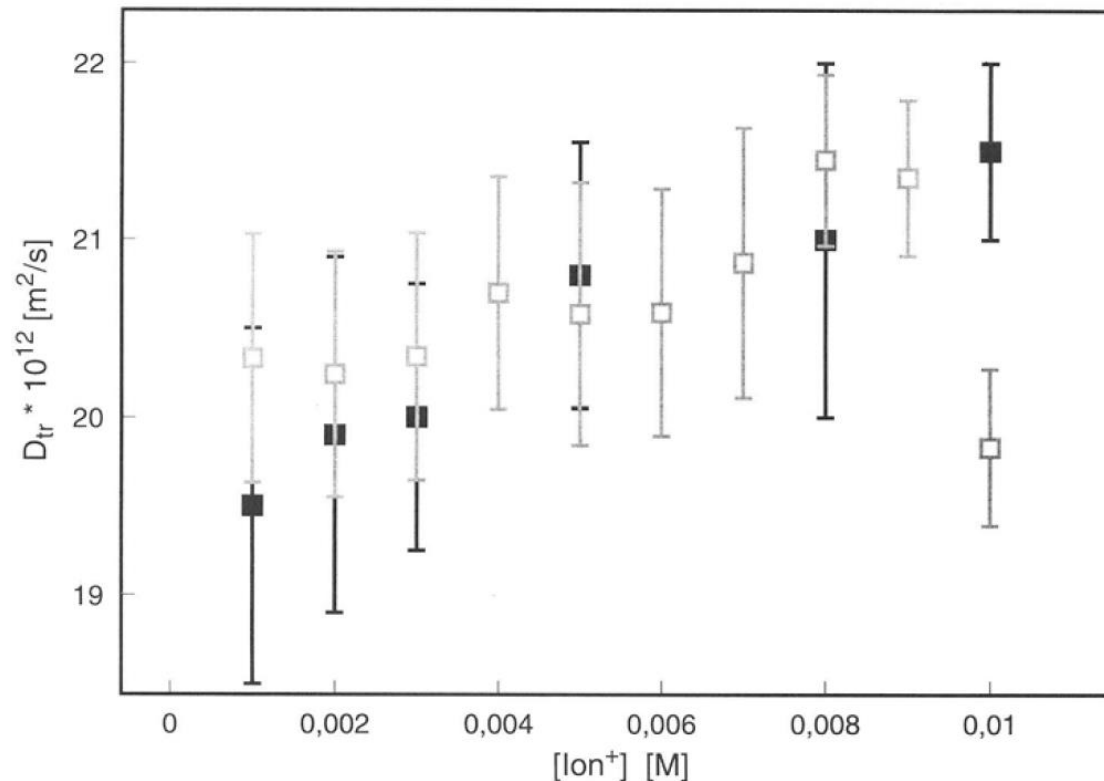


Chromatin Modelling



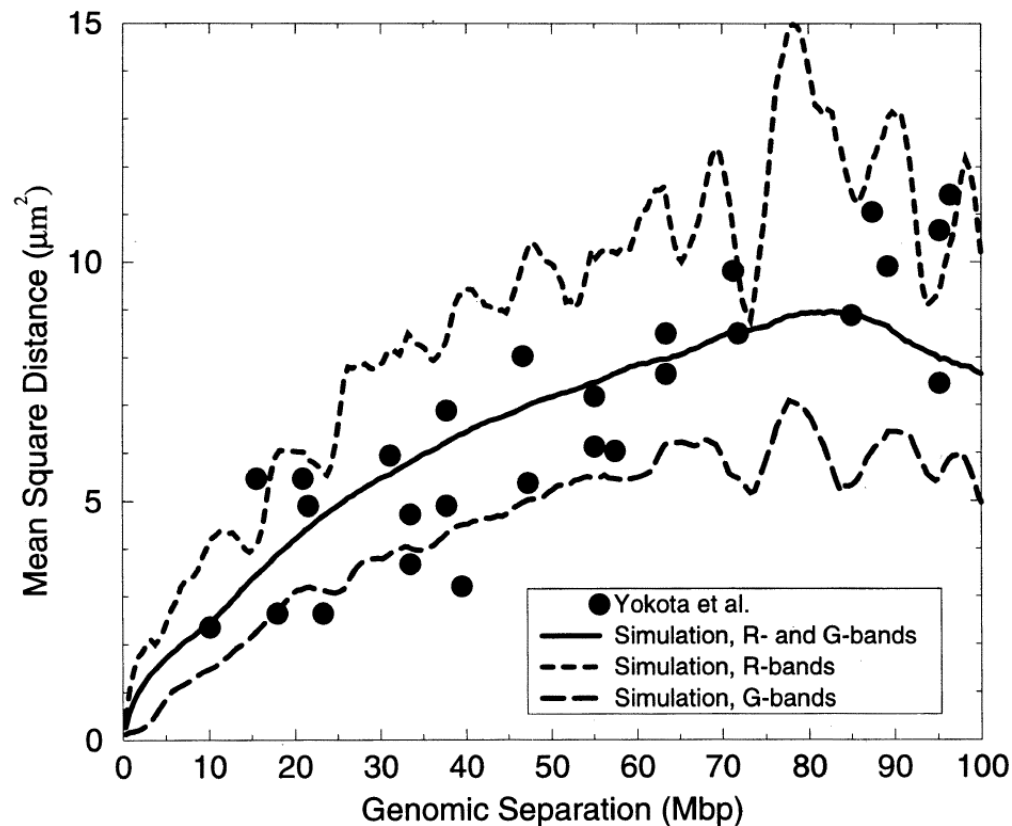
End loop positions vs. time for a trajectory starting from an interwound superhelix of an 1870 bp DNA with $\Delta Lk = -10$ and a permanent bend inserted at position 25. At $t=200 \mu s$, the bend was displaced to position 15

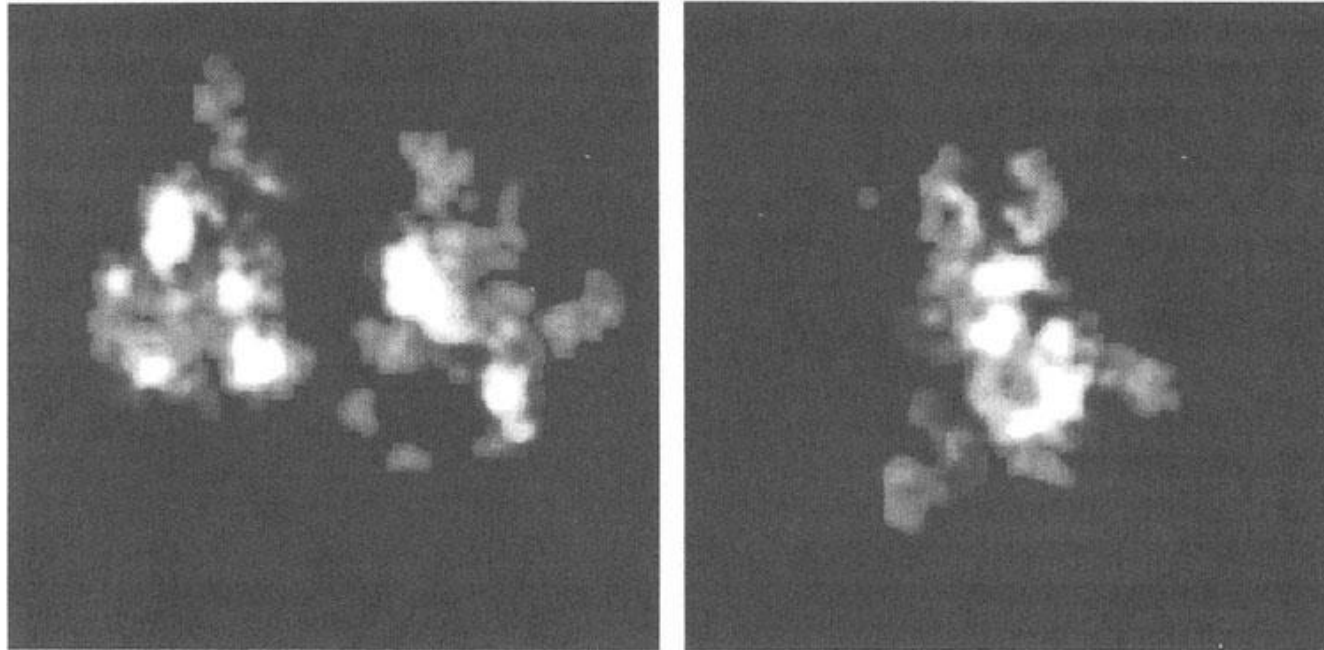
Chromosome Organization



Diffusion coefficients of dinucleosomes as measured by Yao et al. as a function of ionic strength (■). (□) D calculated from a simulated trajectory.

Spatial distances of genetic markers as a function of their separation in Mbp in 2-dimensional projections of interphase nuclei. (•) experimental data by Yokota et al., solid line: model data, averaged over R- and Gbands, dashed lines: R-bands only (upper), G-bands only (lower).



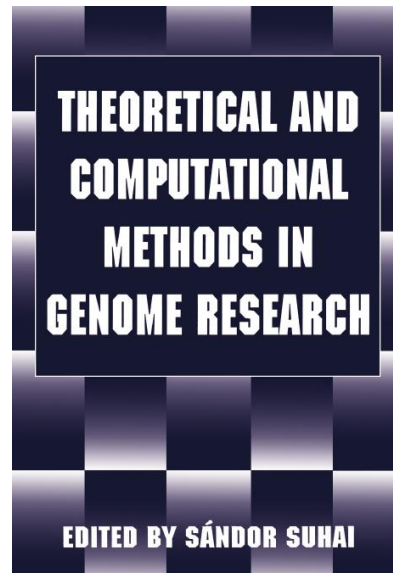


Simulated three-dimensional confocal section of the p- (left) and q- (right) arms of human chromosome 3. The p-arm shows up in two parts, since the central part is behind the image plane. The separation of the two arm regions predicted from the model is clearly visible.



Review Questions

- Explain the model of the genome as an elastic chain.
- Explain the chromatin model.



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York



Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 4. Sequence Patterns Diagnostic of Structure and Function



Contents

- Introduction
- Patterns Diagnostic of Structure
- Hidden Markov Model
- Combined Patterns
- Pattern Evaluation



Introduction

There currently exists a vast wealth of amino acid sequence data from a great many different genes and different organisms. It is anticipated that this wealth will continue to increase. These data represent much of our current raw knowledge of the biological systems on earth. Yet our ability to exploit these data is still limited: first, by our inability to predict a protein's biochemical functions or the structure encoding those functions directly from the knowledge of its amino acid sequence; and second, by our limited ability to carry out direct experimental determination of either the function or structure. Interestingly, it is currently easier to predict function rather than structure. This is true even though knowledge of the folded structure is essential to understanding the detailed steric constraints and side chain chemistry that determine the function(s). Only a small fraction of genes have had biochemical or cellular functions directly determined by experiment.



Introduction

Through the recognition of sequence similarities or shared patterns of sequence elements, we have been able to infer the function of tens of thousands of additional genes. This has proven a powerful approach. By suggesting a probable function for a newly determined sequence, only a few experimental tests need to be carried out for confirmation. In some cases the mere suggestion provided by a shared set of common elements generates obvious insight into past experimental observations making validation all but obvious. Thus recognition of biologically significant sequence similarity has become the Rosetta Stone of modern molecular biology. The tools used depend on methodologies drawn from mathematics, computer science, and linguistics as well as physics, chemistry and the rest of biology. Finally it is believed that analogous approaches will be developed for the accurate prediction of protein structure.

Patterns Diagnostic of Structure

The most obvious and simple structural patterns are those associated with secondary structures. For example, a hydrophobicity plot along a protein sequence showing a clear period of 3.6, 2.0 or zero is a pattern indicative of surface amphipathic helices, surface strands or a completely buried region (including transmembrane segments). Such patterns can be expressed as regular expressions (the lowest rung of the Chomsky formal language hierarchy) or environmental profiles. The latter is a sequence of amino acid preference vectors, one associated with each patterned structural position.

Patterns Diagnostic of Structure

Consensus patterns

Only the conserved
residues are represented:

```
CLVKWVYPFLWIDSK
CLVRWMYPKLPISK
CLIKWAYPYLWIESK
CLIKWAYPFLWIDSK
-----
CL--W-YP-L-I-SK
```

Regular Expressions

Using minimal amino acid
covering or probable
substitution classes:

```
CLVKWVYPFLWIDSK
CLVRWMYPKLPISK
CLIKWAYPYLWIESK
CLIKWAYPFLWIDSK
-----
CLanWfYPxLxIrSK
```

Frequency Profiles

Matrix containing
frequencies of occurrence
for each amino acid at
each residue:

```
CLVKWVYPFLWIDSK
CLVRWMYPKLPISK
CLIKWAYPYLWIESK
CLIKWAYPFLWIDSK
-----
```

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|--|---|--|--|--|--|--|--|--|
| A | 0 | 0 | 0 | 0 | 0 | 2 | | 0 | | | | | | | |
| C | 4 | 0 | 0 | | | | | 0 | | | | | | | |
| D | 0 | | | | | | | 0 | | | | | | | |
| E | 0 | | | | | | | 0 | | | | | | | |
| F | 0 | | | | | | | 1 | | | | | | | |
| . | | | | | | | | | | | | | | | |
| .. | | | | | | | | | | | | | | | |
| Y | 0 | | | | | | | 1 | | | | | | | |
| W | 0 | 0 | | | | | | 0 | | | | | | | |

Normally converted to a matrix of
log likelihood ratios.

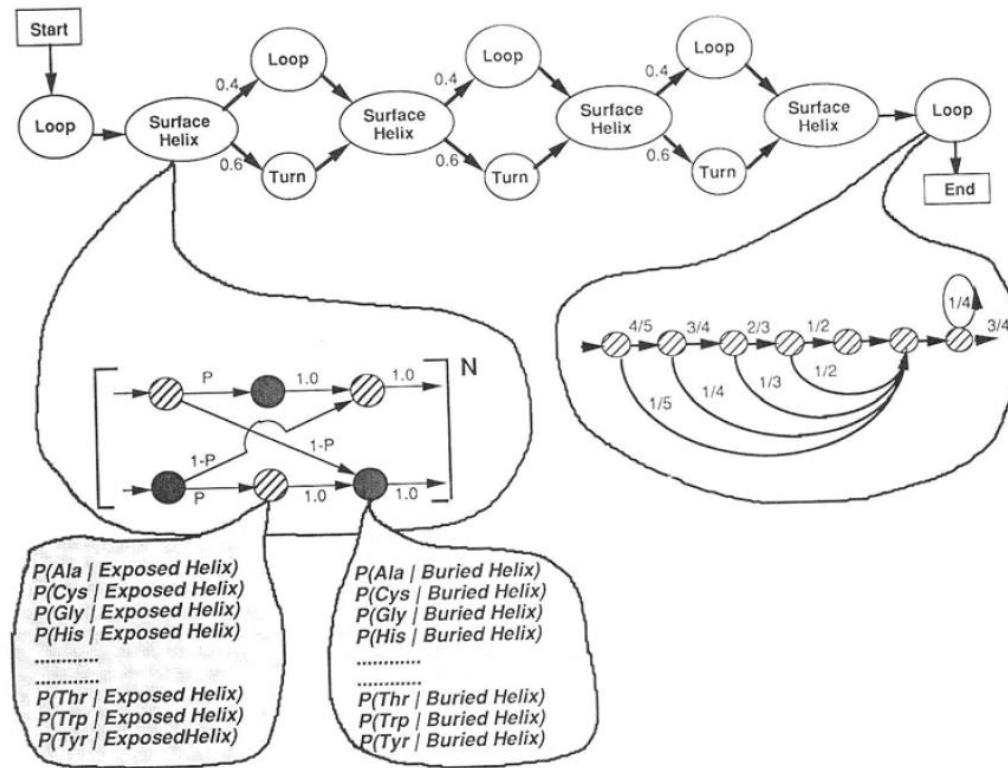
Converted to a UNIX *grep*-style
regular expression statement

```
CL[ILV][KR]W[IVLMFWYCA]YP.L.I[DENQKRH]SK
```

Patterns Diagnostic of Structure

Structural patterns can be considerably more complex, however. They can be hierarchical, consisting of a pattern of secondary structures (indicative of a particular 3-D folding class) each of which is, in turn, composed of a pattern of positional environments (with associated amino acid and/or hydrophobicity preferences). For example, the class of simple four helix parallel bundles would have a pattern consisting of four nearly equal length amphipathic helices separated by three or more loop- or turn-compatible amino acids. Each helix in turn would be represented by a pattern of buried and exposed positions, each of which would be associated with a vector of helix-compatible amino acid preferences. Here one generally needs to move up the Chomsky hierarchy to "context free grammars" to be able to enforce the "non local" constraint of similar lengths among the four helices. This can be done using Discrete State Models or Hidden Markov Chains.

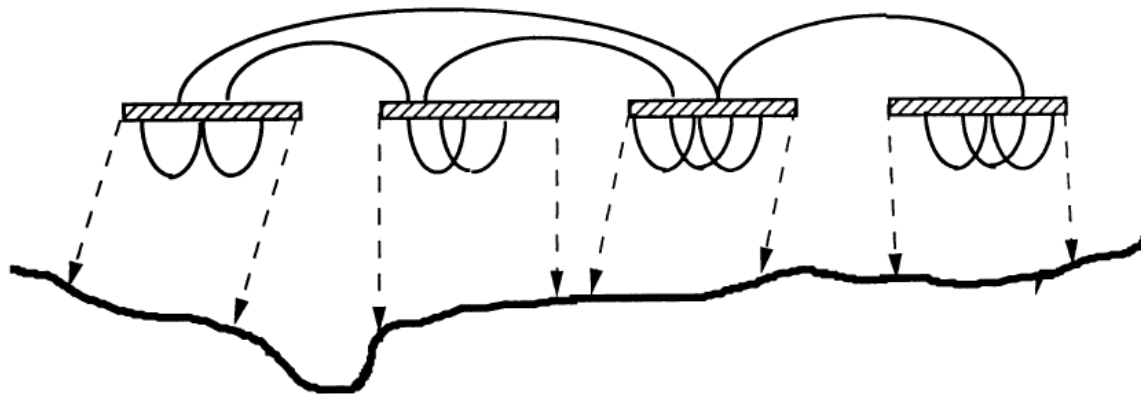
A hierarchical graphic representation of a Discrete State Model of a four helix parallel bundle



Patterns Diagnostic of Structure

Models used for the "Threading" approach to the inverse protein folding problem are often represented by even more complex patterns. There, in addition to containing patterns of sequence position environments and the associated amino acid preferences, pairwise constraints or contact preferences are added. These can be represented by a pattern of pairwise amino acid preferences associated with an adjacency matrix (or contact graph) among some subset of positions. Usually there is a score of pseudo energy associated with each pair of implied contacts for any particular threading or sequence alignment to structural model. Here one is forced to a context-free representation such as the Random Markov Fields.

A graphical representation of a possible set of pairwise contacts to be included in a threading model



Patterns Diagnostic of Function

As in the simplest case of a structural pattern, regular expressions" or profiles are used to represent the patterns of conserved amino acids common to a given functional protein family. It should be obvious that such patterns normally include many structural elements, since identical functions are nearly always encoded within similar structures. In addition, there are common pattern elements included, only because the allowed variation has not yet been observed. Both of these are just the expected result of the conservative/opportunistic nature of evolution. While it is in general difficult to separate these classes of elements, it is often possible to clearly identify experimentally those pattern elements that are essential to an "active site". As for the inclusion of chance common elements, one assumes that with sufficient sampling (taxonomic breadth) or biochemical modeling that these can be kept to a minimum.

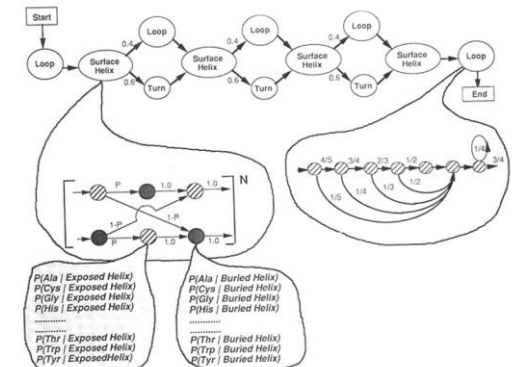
Patterns Diagnostic of Function

Pattern alignment is particularly useful since it not only suggests an overall function, but also information as to which parts of a new sequence are most likely to be critical to its function. However, there are limitations to most current implementations. None of the current pattern identification methods deal convincingly with families of proteins sharing heterologous sets of functional and/or structural domains. Current methods produce patterns of common elements while eliminating heterologous sequence elements. This is their strength when dealing with single-domain proteins, for it is the highly variable regions among homologous sequences that can give rise to random and misleading full database search matches. However, in the heterologous multiple domain protein families, entire and important domains may also be discarded or be reduced to near complete degeneracy in the generation of a single pattern common to all.

Hidden Markov Model

Considerable research has taken place in the area of speech pattern recognition using Hidden Markov or "discrete-state" models. While there have been limited applications to protein sequence analysis, the problems are conceptually very similar. Speech is a linear sequence of elements. Its pattern interpretations are highly complex and very context dependent, not different from the protein sequence's encoding of structure and function. Such models can be viewed as signal generators, or amino acid sequence generators.

Each of the N nodes represents a modeled state. The connecting arcs represent the allowed transitions between the states and its probability. Each model state includes a subset of "hidden" output states. Each output state defines a distribution over some alphabet of output symbols, or in our case the amino acids. Hidden Markov Model represented patterns are thus composed of an N by N state-to-state transition matrix and a set of output state distribution functions.



Combined Patterns

In those cases where a functional sequence pattern is thought to contain primarily only those sequence elements directly encoding the function rather than indirectly encoding the structure common to that functional family, the patterns are often too short or of insufficient specificity to be diagnostic. A classic example is the pattern of residues associated with the catalytic triad of the Serine proteases. Thus since all of the known Serine proteases are found to be all beta (crudely wound double) sheets, a pattern composed of the highly sensitive, but non specific catalytic triad "embedded" in a general all beta structural pattern might have greatly improved specificity without reduced sensitivity.

Pattern Evaluation

At a minimum, two quantities should be reported for any pattern: its sensitivity and specificity (reflecting type I and type II errors).

Sensitivity = $TP(\text{True Positives}) / \{TP + FN(\text{False Negatives})\}$

and

Specificity (selectivity) = $TN(\text{True Negatives}) / \{TN + FP(\text{False Positives})\}$

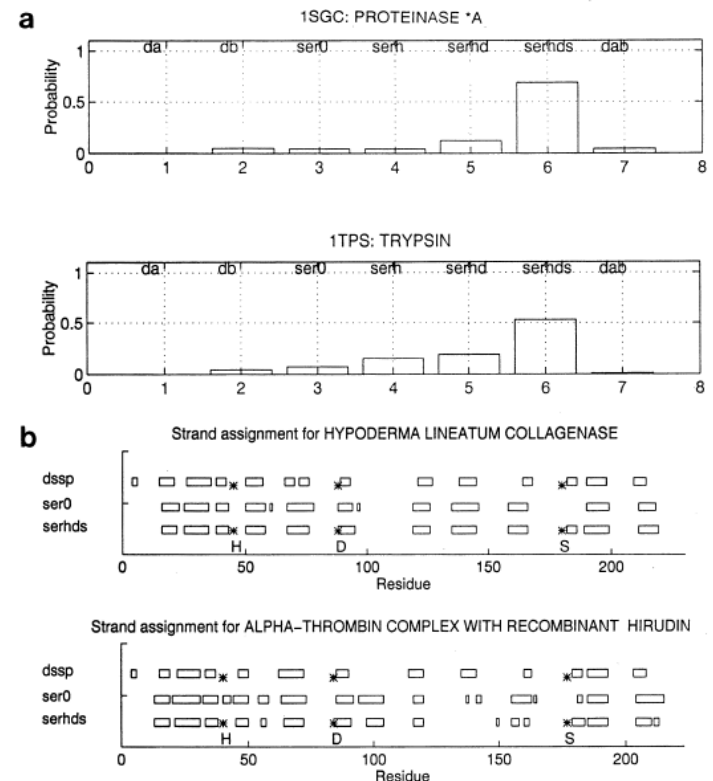
Here TP is the number of defining (or positive) set sequence matches and FP is the number of control (or negative) set matches. The sensitivity of a pattern (e.g., diagnostic of zinc fingers) reflects the likelihood that all occurrences (e.g., all "real" zinc fingers) have been and thus will be found. It reflects how sensitive the pattern is as a detector (e.g., of zinc fingers). The specificity is a measure of the probability of correctly identifying the negative instances. The diagnostic ability of a pattern can be conveniently summarized in a normal two-by-two truth or contingency table, given a positive and negative control set.

Pattern Evaluation

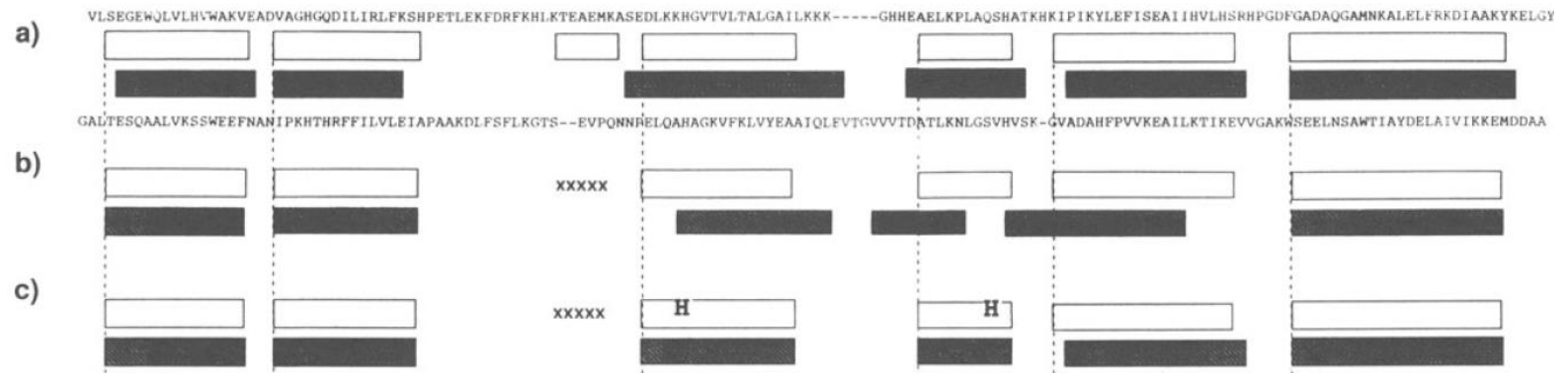
The computer science and statistical pattern recognition literature is replete with discussions of estimating class membership using the Bayesian approach. Thus it may be useful to identify the relationship between these discussions and sensitivity and specificity. Let P_i be a pattern and f_j the protein structure or functional class. One can then define the relationship between the conditional probabilities from the definition of joint probability. This is known as Bayes' relation:

$$P(f_j | p_i) = \frac{P(p_i | f_j) * P(f_j)}{P(p_i)}$$

(a) Relative probabilities of seven different Discrete State Models having generated two different Serine protease amino acid sequences. (b) Two examples of increased secondary structure predictive accuracy with increasing sequence specificity combined with an all beta Discrete State Model.



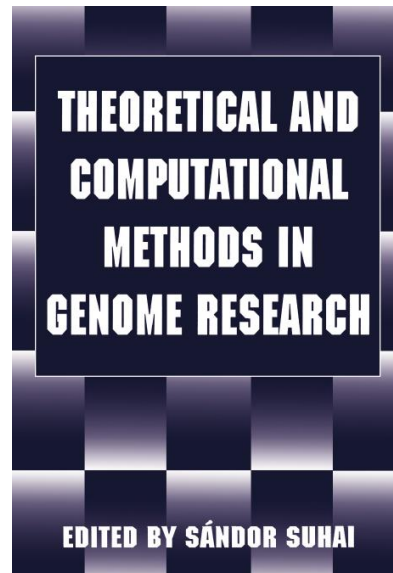
Leghemoglobin sequence threaded through the myoglobin structural model. (A) Reference alignment. (B) Short "D" helix removed, no active site constraints. (C) Short "D" helix removed, requiring model elements at positions E7 and F8 to be occupied by histidine, H, from the sequence.





Review Questions

- Explain the patterns diagnostic of structure.
- Explain the patterns diagnostic of function.
- How could the pattern be evaluated?



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York



Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 5. Recognizing Functional Domains in Biological Sequences



Contents

- Introduction
- Weight matrices
- Alignments requiring gaps
- Vector interpretations
- Discriminant matrices



Introduction

Weight matrices are general representations of sequence specificity exhibited by common sites. They were originally developed to represent regulatory sites on DNA (or RNA), such as ribosome binding sites, promoters, splice sites and others. They have also been extended to represent protein motifs, and are used in the BLOCKS database of conserved regions in protein families. Those approaches all used sites of constant size; that is the alignment of the sites did not require the addition of any gaps. Gribskov et al. showed how the same basic idea can be extended to alignments that include gaps, in what are called "profiles." This extension requires dynamic programming to find the optimal alignments between the matrix and the sequence. More recently Hidden Markov Models (HMMs) have been used to define the patterns in conserved sequences. Although classical weight matrices, profiles and HMMs differ in many details, they can also be viewed as variations of a common theme, as will be described briefly below.

Introduction to weight matrices

The idea derives from thinking that each position within a site contributes something to its activity. Presumably the consensus base at a position would contribute the most to the activity, but any single change away from the consensus is likely to only diminish the activity somewhat, and not eliminate it altogether. This will, of course, depend on the type of site; some protein binding sites might be highly specific, like restriction sites, so that any changes from the consensus base will effectively reduce the activity beyond detection. But for sites like promoters this doesn't seem to be the case.

| C | T | A | T | A | A | T | C |
|-----|-----|----|-----|-----|-----|---|---|
| -38 | 19 | 1 | 12 | 10 | -48 | | |
| -15 | -38 | -8 | -10 | -3 | -32 | | |
| -13 | -48 | -6 | -7 | -10 | -48 | | |
| 17 | -32 | 8 | -9 | -6 | 19 | | |

| C | T | A | T | A | A | T | C |
|---|-----|-----|----|-----|-----|-----|---|
| | -38 | 19 | 1 | 12 | 10 | -48 | |
| | -15 | -38 | -8 | -10 | -3 | -32 | |
| | -13 | -48 | -6 | -7 | -10 | -48 | |
| | 17 | -32 | 8 | -9 | -6 | 19 | |

| C | T | A | T | A | A | T | C |
|---|---|-----|-----|----|-----|-----|-----|
| | | -38 | 19 | 1 | 12 | 10 | -48 |
| | | -15 | -38 | -8 | -10 | -3 | -32 |
| | | -13 | -48 | -6 | -7 | -10 | -48 |
| | | 17 | -32 | 8 | -9 | -6 | 19 |

Using a matrix to score positions in a sequence. At each location of the matrix the score is the sum of the matrix elements corresponding to the sequence at that position (the circled elements).

Introduction to weight matrices

The weight matrix has values for all possible bases at each position within the potential site, whereas the sequence has particular bases at each position. The weight matrix values for those bases are summed to get the total score for the particular site. In the case of a protein that simply binds to DNA, the values of the weight matrix can be thought of as partial binding energies contributed by each base interacting with the protein, and the sum of those is the total binding energy for the protein to the site.

| | | | | | | |
|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 1 | 1 | 0 |
| C | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 1 | 0 | 1 | 0 | 0 | 1 |

| | | | | | | |
|---|----|----|----|----|----|----|
| A | 0 | 10 | 8 | 10 | 10 | 0 |
| C | 1 | 0 | 1 | 1 | 3 | 0 |
| G | 1 | 0 | 1 | 1 | 1 | 0 |
| T | 10 | 0 | 10 | 1 | 1 | 10 |

| | | | | | | |
|---|----|----|----|----|----|----|
| A | 2 | 95 | 26 | 59 | 51 | 1 |
| C | 9 | 2 | 14 | 13 | 20 | 3 |
| G | 10 | 1 | 16 | 15 | 13 | 0 |
| T | 79 | 3 | 44 | 13 | 17 | 96 |

| | | | | | | |
|---|-----|-----|----|-----|-----|-----|
| A | -38 | 19 | 1 | 12 | 10 | -48 |
| C | -15 | -38 | -8 | -10 | -3 | -32 |
| G | -13 | -48 | -6 | -7 | -10 | -48 |
| T | 17 | -32 | 8 | -9 | -6 | 19 |

Different matrices that represent the -10 region of E. coli promoters. The top matrix is a simple consensus sequence, whereas each following matrix adds additional information about the patterns observed at -10 regions.

| | |
|------|-------|
| AAAA | 2.63 |
| AAAC | 2.63 |
| AAAG | 2.63 |
| AAAT | 0.46 |
| AACA | 1.79 |
| AACC | 1.79 |
| AACG | 1.79 |
| AACT | -0.38 |
| . | |
| . | |
| AGCG | -0.25 |
| AGCT | -2.42 |
| AGGA | 0.59 |
| . | |
| . | |
| CTTT | -0.37 |
| GAAA | 4.01 |
| . | |
| . | |
| TTTG | 2.63 |
| TTTT | 0.46 |

| | | | | |
|---|-------|-------|-------|-------|
| A | -0.55 | +1.38 | +0.42 | +1.38 |
| C | 0 | +0.55 | -0.42 | +1.38 |
| G | +0.83 | -0.66 | +0.42 | +1.38 |
| T | +0.83 | +0.42 | 0 | -0.79 |

The left column represents the binding energy to all 256 4-long sequences (only some of them are shown). The matrix on the right provides the same information because every value in the list on the left can be calculated using this matrix.

Alignments requiring gaps

Weight matrices can be extended to include alignments with gaps, as in profiles. In this case the matrix contains extra rows to indicate the score associated with a gap in the alignment. Typically there will be scores for both the introduction of a gap and a different score for the extension of a gap, as it is well accepted that those two events have different probabilities. In HMM methods, there will be different scores for the introduction of insertions than for deletions, but it has been shown how both HMMs and profiles can be described in the same syntactical framework. The training method is related to Expectation-Maximization (EM) methods used previously for ungapped alignments and alignments with restricted gaps. Other methods are also capable of finding gapped alignments and determining the matrix representations of those alignments simultaneously.

A.

| | | | | | | |
|---|-----|-----|-----|-----|-----|-----|
| A | 9 | 214 | 63 | 142 | 118 | 8 |
| C | 22 | 7 | 26 | 31 | 52 | 13 |
| G | 18 | 2 | 29 | 38 | 29 | 5 |
| T | 193 | 19 | 124 | 31 | 43 | 216 |

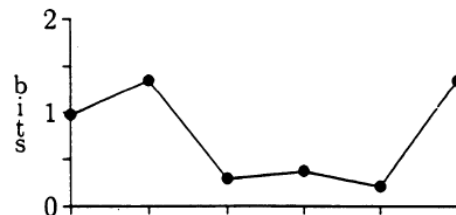
B.

| | | | | | | |
|---|------|------|------|------|------|------|
| A | 0.04 | 0.88 | 0.26 | 0.59 | 0.49 | 0.03 |
| C | 0.09 | 0.03 | 0.11 | 0.13 | 0.22 | 0.05 |
| G | 0.07 | 0.01 | 0.12 | 0.16 | 0.12 | 0.02 |
| T | 0.80 | 0.08 | 0.51 | 0.13 | 0.18 | 0.89 |

C.

| | | | | | | |
|---|-------|-------|-------|-------|-------|-------|
| A | -2.76 | 1.82 | 0.06 | 1.23 | 0.96 | -2.92 |
| C | -1.46 | -3.11 | -1.22 | -1.00 | -0.22 | -2.21 |
| G | -1.76 | -5.00 | -1.06 | -0.67 | -1.06 | -3.58 |
| T | 1.67 | -1.66 | 1.04 | -1.00 | -0.49 | 1.84 |

D.



Determining the specificity matrix and information content of the -10 regions of promoters. The top matrix shows the occurrence of each base at each position in a collection of 242 promoters. The next matrix converts those numbers into the fraction of each base at each position. The third matrix is the log-likelihood ratio of the observed frequencies divided by the expected, using 0.25 as the expected occurrence for any base. (The logarithm is taken to base 2.)

Vector interpretations

It is often useful to think of the weight matrices and the sequences in terms of a "sequence space." The weight matrix itself is a point, or vector to that point, in the space. And, of course, each sequence is also a vector in that space. However, we need to think of the space as defined by the weight matrix, have dimensions not only for all possible bases at each position, but also the possibilities of gaps, gap extensions and non-independent positions. In order to determine the vector for a particular sequence (or site) in this space, it is first necessary to align it with the matrix. The score for any sequence, given a matrix, is then just the dot-product of the vector for the matrix with the vector for the sequence. The vector for the sequence just has a 1 for each base that occurs at each position (corresponding the circles below the sequence) and a 0 for all of the bases that do not occur. This is commonly used in statistical methods and is known as "dummy encoding," where the variables that exist for a particular example are given 1s, and the variables that do not exist are given 0s. Aligned sequences with gaps are still encoded the same way, but now the 1s may refer to positions of gaps, or gap extensions, instead of just to residues at positions. Then the score of any aligned sequence (called $Sc(S_i)$ for the sequence S_i) with the matrix is just the sum of the variables that occur, and is simply the dot-product of the sequence with the weight matrix:

$$Sc(S_i) = \vec{W} \cdot \vec{S}_i = |\vec{W}| |\vec{S}_i| \cos \alpha_i$$

Discriminant matrices

One might want to use a weight matrix in a somewhat different manner. Rather than just take a collection of known sites and try to represent them effectively, or even try to make a matrix with good quantitative predictions of site activity, one might have the goal of finding a matrix that is optimally effective at discriminating true sites from other sequences, non-sites. For example, if one has a collection of sites that are known to be functional and another set of sequences that are known to not be functional sites, one could try to find a matrix that distinguished the two sets by giving every site a score higher than any non-site. In fact, this was the first application of weight matrices, which used a simple neural network called a "perceptron" to determine the matrix elements.

Discriminant matrices

A given matrix will determine a score for any particular sequence. If the sites are binding sites for some protein, this approach is directly related to finding the sites with maximum specific binding energy for the protein. The method works by calculating the "partition function," or the total binding probability, over all of the possible sites in the collection (the entire genome for a DNA binding protein). Then the objective is to find the weight matrix that maximizes

$$U = \frac{1}{N} \sum_{i=1}^N \vec{W} \cdot \vec{S}_i - \ln Y$$

where the sum is over all N sites and Y is the average likelihood (i.e., $e^{\text{dot-product}}$) over the entire collection of sequences, both sites and non-sites.

Discriminant matrices

Current approaches use a variety of methods, from comparing a new sequence to individual sequences in the database, such as with BLAST or FASTA, to comparing the new sequence to family representations such as profiles, HMMs, BLOCKS or Prosite patterns. As an initial test it was compared the ability of the optimized matrices to distinguish family members to that of the BLOCKS database. This database derives from the protein families described in Prosite, but represents the most conserved, ungapped regions as weight matrices that can be used to score new sequences and determine whether or not they are members of the family. Previously the weight matrices have been determined solely from the members of the family, not taking into account the sequences of non-family member proteins. Much emphasis has been placed on how to weight the sequences of the family members so as to compensate for the bias in the representation in the current database and to compensate for limited sample sizes.

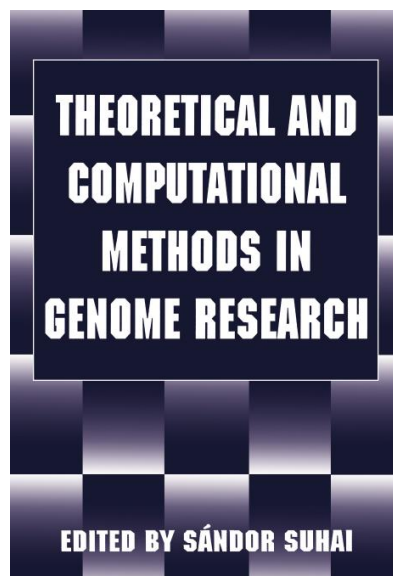


Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness



Review Questions

- Explain the role of matrices for functional domain recognition.
- What is the "partition function," or the total binding probability and how could be calculated?



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York



Module 2. Genetics

Topic 2. Theoretical Modeling in Genomics

Lesson 6. Computer Simulations of Protein- DNA Interactions



Contents

- Introduction
- Detection programs or online servers that can detect active sites
- Strategies in the ligand placement
- Rigid body docking
- Flexible docking



Introduction

Molecular recognition is an essential component in almost all biomolecular processes, specifically in processes relating to transcription and translation of the genetic material. Much progress has been made in recent years towards characterizing several such systems in structural terms, providing insight into the fundamental issue of the structural basis for sequence dependent interactions and binding; in particular one can identify some principles of recognition and structural organization within the transcription factor families.



Molecular dynamics (MD) simulation provides a very detailed, structural and dynamic, description of biomolecular systems; this level of detail, which is very difficult to obtain by other means, is very valuable for a thorough understanding of the subtle balance between competing interactions involved in molecular recognition processes. From a comparison of calculated interaction energies (enthalpies) in substrate:protein complexes, with calculated free energy values as well as with experimental data, it is quite clear that straightforward, intuitive guesses of the outcome of mutation experiments in complicated systems are unreliable. The influences of slight structural changes, interplay with solvent and ions, and entropic effects are very difficult to guess; more precise methods, like free energy perturbation or potential of mean force calculations, therefore are necessary. Although some aspects of these system may also require combined molecular mechanics/quantum mechanics energy calculations, the non-covalent binding processes that are the focus of this report have been studied using classical mechanics and empirical energy functions.

Simulations of Protein-DNA Systems

A few dynamics simulations of protein-DNA systems have been reported in the literature. The lac repressor headpiece in complex with its operator was simulated using distance restraints from NMR studies, with a 50ps piece of unrestrained trajectory. From this simulation the specific contacts between the protein and the DNA were found to be formed mainly by nonpolar contacts, and to a lesser extent through water-mediated hydrogen bonds. The DNA binding domain of the glucocorticoid receptor (GRDBD) in complex with DNA has been the subject of studies by three groups using slightly different approaches. Harris et al have run a 1ns simulation of a model of the GRDBD dimer bound to DNA with a 10 Å shell of water surrounding the system (yielding a total of 9625 atoms), and Bishop & Schulten have reported two 90ps simulations of the GRDBD dimer bound to two DNA sequences with different spacings between the recognition half sites, using an ellipsoidal geometry for the surrounding water (ca 13600 atoms in total). Luisi et al's simulations of the GRDBD system have used the crystallographic coordinates for the complex in a 42 Å radius sphere of water (ca 31000 atoms in total), which was simulated for 200ps. Complexes of GRDBD monomers bound to various halfsite DNA sequences were simulated for 200ps, and also used in free energy perturbation calculations of the relative affinities of the GRDBD for these DNA sequences.



Simulations of Other Nucleic Acid Complexes

The thermodynamics of nucleic acid interactions with proteins and small ligands have also been simulated with results in good agreement with experiment.

DNA-Netropsin Complex

The small dicationic peptide drug netropsin binds preferentially to the minor groove of A-T regions of DNA, which are favored by 4.4 kcal/mol over G-C regions, a difference that has been attributed to the repulsion of the amino group protruding into the minor groove of a G-C pair, but which is absent in an A-T pair. We have applied the thermodynamic cycle-perturbation method to calculate this change in free energy of binding with very encouraging results. We obtained a value for $\Delta\Delta G$ in the range from 3.4 to 4.0 kcal/mol with an estimated error of 1-2 kcal/mol. The results do not seem to depend on the size of the system or the length of the simulations within the limits we have tried: 1Q-20ps runs of either a DNA hexamer duplex in water (1200 atoms) or a DNA octamer duplex in water (2600 atoms).

Ribonuclease T₁

Ribonuclease T₁ is a small enzyme which selectively cleaves single-stranded RNA on the 3' side of guanines, and for which the 3D structure is known with and without bound substrate. This makes the enzyme ideal for the study of protein-nucleic acid interactions. Our work on this system has so far focussed on changes in the structure of the wild-type enzyme, particularly in its active site, upon binding of the inhibitor 2'GMP. Initial experimental studies using time-resolved fluorescence showed the presence of a picosecond motion of Trp59 (the only tryptophan in the enzyme) which was altered by the binding of 2'GMP. This was also observed in a series of molecular dynamics simulations of the same systems, in water as well as in vacuo.

Ribonuclease T₁

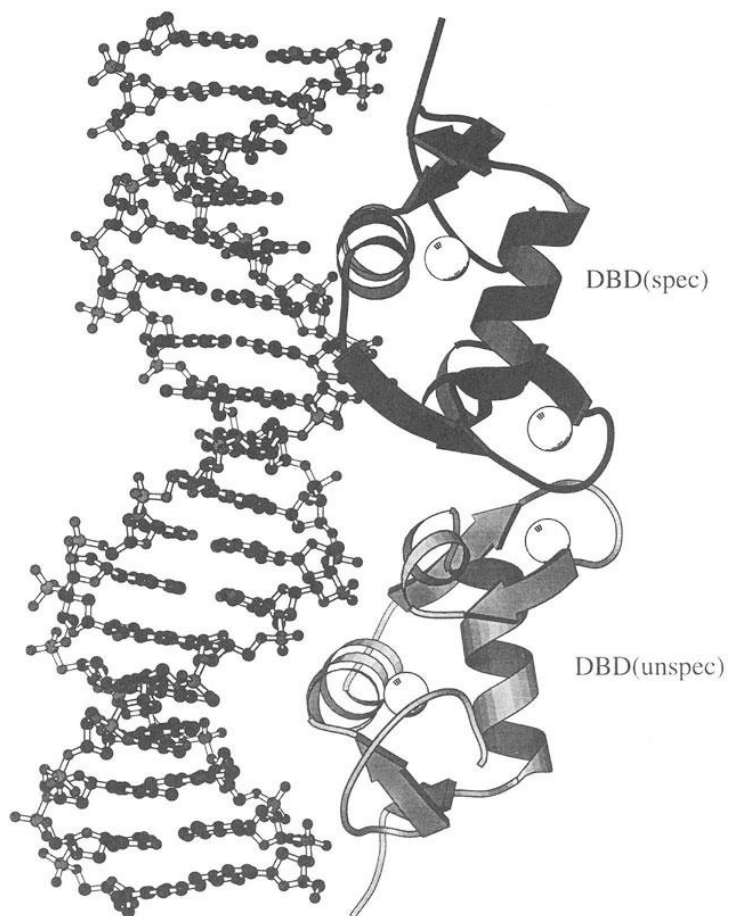
From a series of free energy simulations comparing the binding of 2-aminopurine versus 3'GMP to the enzyme was found that 2-aminopurine has a less favorable interaction (by -5 kcal/mol) which is reasonable; the effect of the compensatory mutation Glu46-Gln46 seems mainly to be a reduction of the affinity for GMP, with almost no direct effect on 2-aminopurine binding.

Nuclear Hormone Receptors

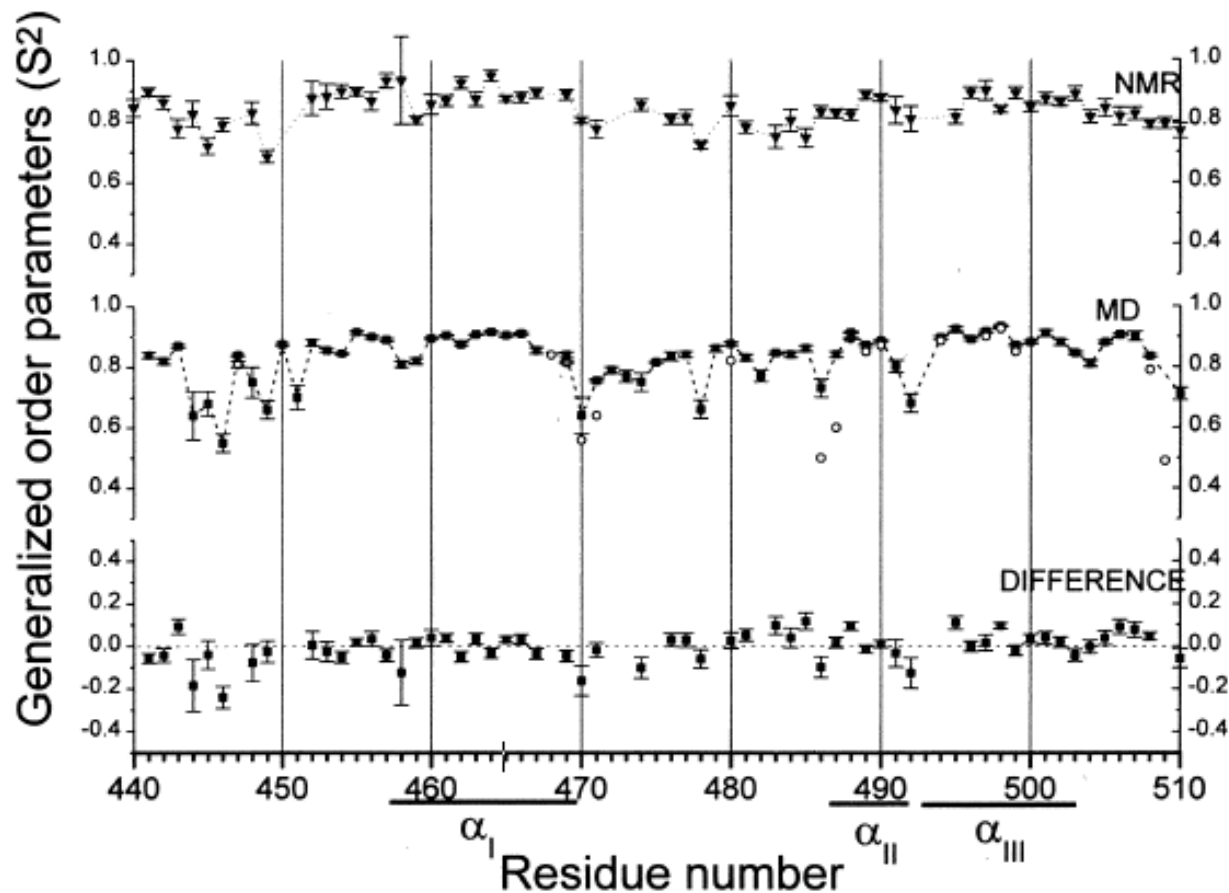
The transcription factors which are activated by hormones such as glucocorticoids, estrogens, and retinoids, constitute the large family of nuclear hormone receptors. These receptors seem to have a common evolutionary origin and to have evolved into three distinct subfamilies, one of which is the steroid hormone receptor group. The receptors are multidomain proteins, with the different functions needed within the signal transduction system performed by different domains, and they bind to the response elements of the DNA in various dimeric combinations. Some of the main domains contain the ligand binding, DNA binding, and transactivation functions.

Nuclear Hormone Receptors

There are two families of response elements for steroid receptors, the glucocorticoid response elements (GRE) with the halfsite sequence TGTTCT and the estrogen response elements (ERE) with the halfsite sequence TGACCT. Crystallographic structures have been determined for systems of both kinds.

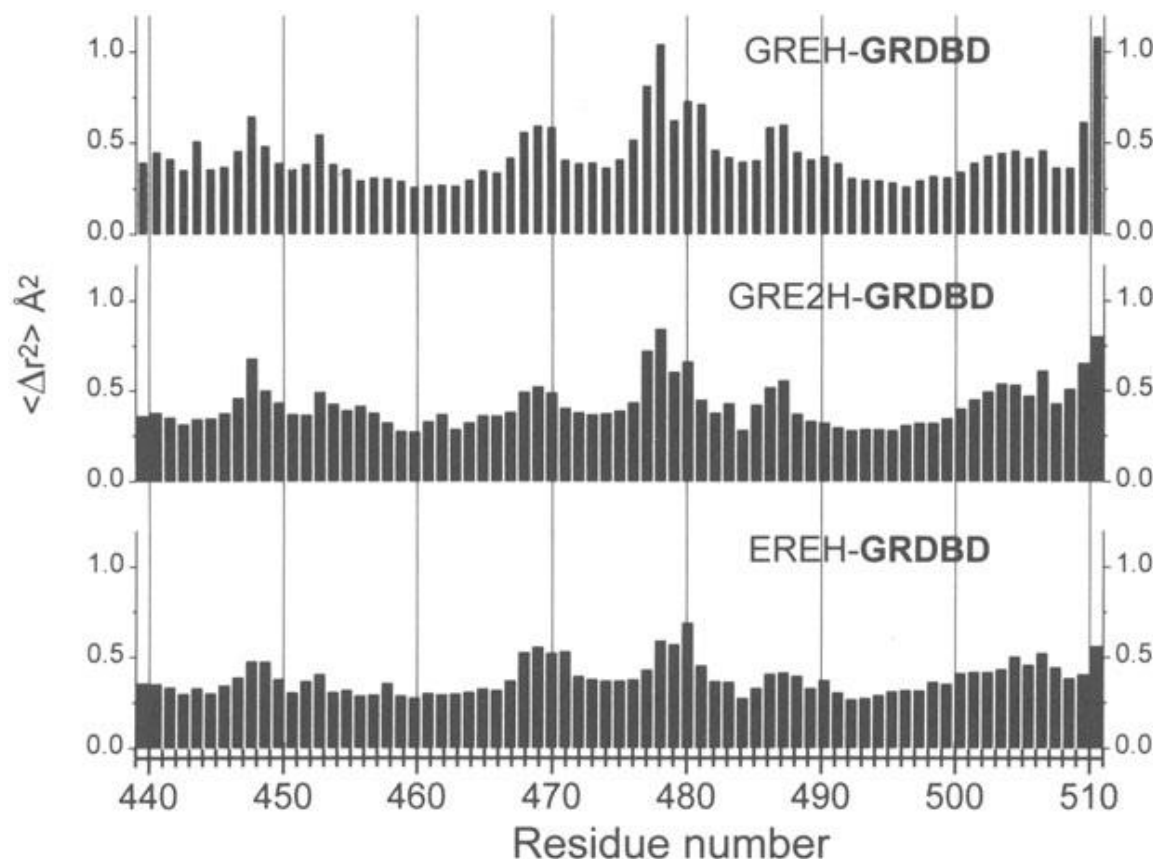


Schematic drawing of the GRE:GRDBD
complex



Order parameters vs residue in the GRDBD monomer

Mean square atomic fluctuations in different complexes of GRDBD and DNA response elements



Protein-DNA interactions

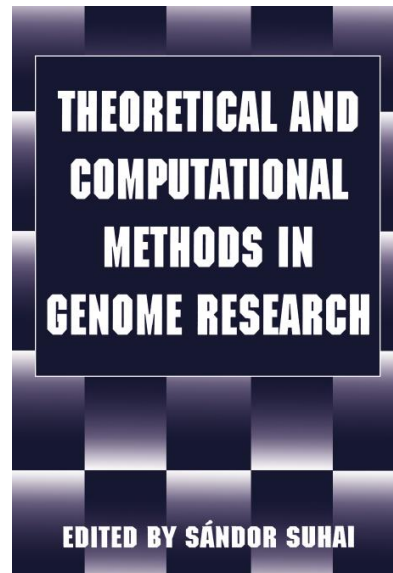
Overall the structure of the complex is maintained, a small bend in the DNA becomes slightly more pronounced due to somewhat closer contacts between protein and DNA, and some local distortions in the DBD are also seen. There is one region of DBD where the NMR and X-ray structures differ, and in simulations of the DBD monomer in solution, starting from either of these conformers, the simulated structure remains close to its initial conformation indicating that both conformers are in local minima. Results on backbone dynamics correlate well with NMR relaxation data from ^{15}N -labelled proteins, and in the cases with significant discrepancies an analysis of the actual motions as found in the simulated trajectories revealed that they were generally the result of infrequent processes, e.g. a backbone N-H hydrogen bond would jump back and forth between different H-bond accepting groups, causing insufficient sampling.

Simulations of protein:DNA systems are feasible on a nanosecond timescale, and yield structural, dynamic and thermodynamic results which agree well with available experimental data. In addition to this we find details concerning the dynamics at the contact interfaces in terms of direct as well as water mediated hydrogen bonds that are not easily seen in experiments. We could also make some suggestions as to the underlying reasons for differences in the stabilities and cooperativity of the complex formation in the GRDBD:GREH and GRDBD:EREH cases, such as the stiffening of the GRDBD when bound to EREH.



Review Questions

- Give some examples of simulations of protein-DNA systems.
- How can the simulations of protein-DNA systems be done?



Sandor Suhai (1997) Theoretical and Computational
Genome Research, Springer Science+Business Media
New York



Module 2. Genetics

Topic 3. Computer Genomics

Lesson 1. DNA Decoding



Contents

- Introduction
- Sequencing Platforms
- Whole-Genome Sequencing and Whole-Transcriptome Sequencing



Introduction

The use of computational methodologies for analysis of biological data is not recent; however, with the reduction of the cost of DNA sequencing associated with the increase in the volume of genomic data produced by the sequencing platforms, it has become essential to use computational approaches to handle and extract more information from the data of complete genomes and/or transcriptomes using bioinformatics tools.

Sequencing Platforms

Decoding DNA in biological samples has become an essential step in a variety of research applications. With the advancement of technologies, DNA was sequenced more quickly and identified with greater precision in terms of genetic composition and organization, which is fundamental information for understanding biological processes, in addition to directing post-genomic studies such as transcriptomics.



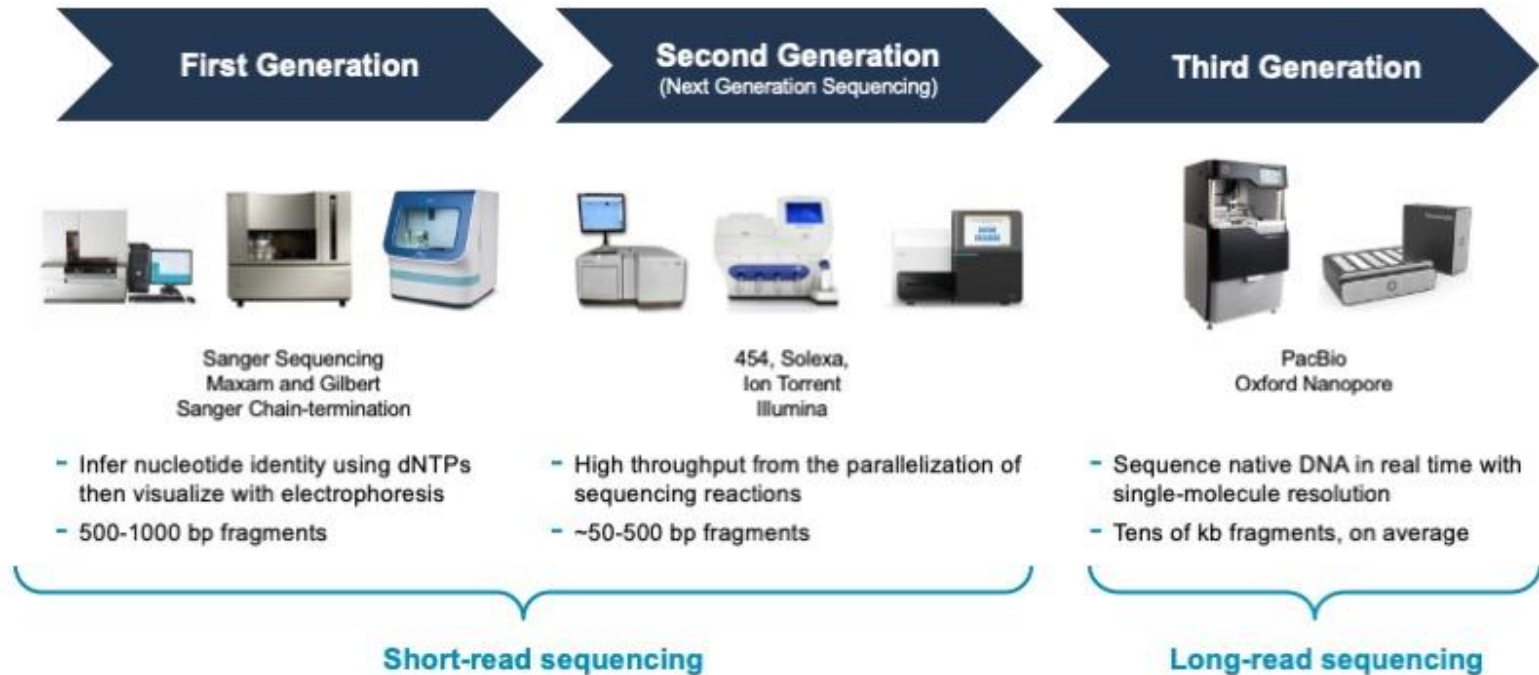
Next generation sequencing (NGS)

Next generation sequencing (NGS), also known as high-throughput sequencing, is a powerful technology that has revolutionized the field of genomics, transcriptomics, and even proteomics. It allows the rapid and cost-effective determination of DNA and RNA sequences and detection of base modifications, enabling the analysis of entire genomes, transcriptomes, and metagenomes at unprecedented resolution and depth. The ability to generate vast amounts of sequence data in a short amount of time has revolutionized many areas of research and has led to numerous breakthroughs: detailed annotation of genomes, as well as the identification of genetic variants associated with diseases, deeper understanding of gene regulation, cellular differentiation, and functional pathways, and, in agriculture, improved crop breeding and pest management.



Next generation sequencing (NGS)

Nowadays, there are several sequencing platforms available, each with their unique features that provide alternative options for sequencing mechanisms, read lengths, run times, ease-of-use, and scalability. First-generation platforms are based on classical chain-termination method, colloquially refers to as Sanger sequencing. Second-generation NGS platforms are based on sequencing by synthesis or a pH change detection and generate short reads. These platforms include Ion Torrent devices, sequencers from Illumina, as well as from BGI Group. Third generation NGS platforms can generate long reads, via single molecule real-time or nanopore sequencing. Next, we will review the most used sequencing technologies and their different principles. Understanding the characteristics of each of them is essential to know which one is best suited when designing your studies.



The evolution of DNA sequencing tools

Sequencing procedures

- The chain-termination method: Sanger sequencing
- Semiconductor sequencing
- Sequencing by synthesis: bridge amplification
- Combinatorial probe-anchor synthesis: nanoball sequencing
- Single molecule real-time sequencing (SMRT)
- Nanopore sequencing



The chain-termination method: Sanger sequencing

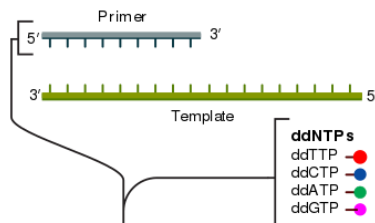
The DNA structure, described by James Watson and Francis Crick in 1953, made it possible for other researchers to apply sequencing methodologies to determine the nucleotide sequence of nucleic acids, called first generation sequencing technologies. In this first generation, the initial efforts were focused on RNA sequencing, generating, in 1965 by Robert Holley et al., the first complete nucleic acid sequence, the tRNA alanine from *Saccharomyces cerevisiae*. At that time, several researchers started adapting their methods to sequence DNA. From the mid-1970s, DNA sequencing was leveraged through the Maxam–Gilbert Method and the “Plus and minus” Method. However, the breakthrough that changed the process of how DNA was sequenced only came in 1977, when Sanger’s dideoxy “chain termination” technique was published.

The Sanger method has undergone numerous changes, such as the development of semi-automatic sequencers with electrophoresis in capillaries filled with gel, and a detection system using confocal fluorescence excited by laser, which brought advantages such as: reducing the handling of toxic chemicals, application of samples by electroinjection and simultaneous electrophoresis with up to 384 independent capillaries, for the generation of fragments of approximately 750 base-pairs. In 1986, Leroy Hood's laboratory in Caltech (California—USA), together with Applied Biosystems, launched the first semi-automatic sequencer, based on the Sanger method.

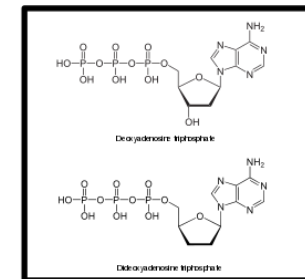
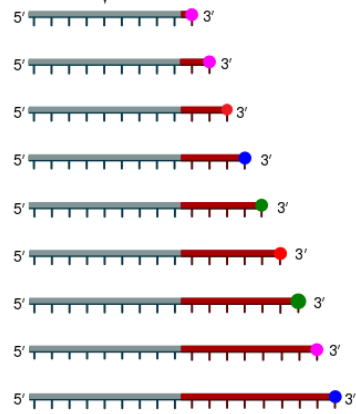
Sanger sequencing

① Reaction mixture

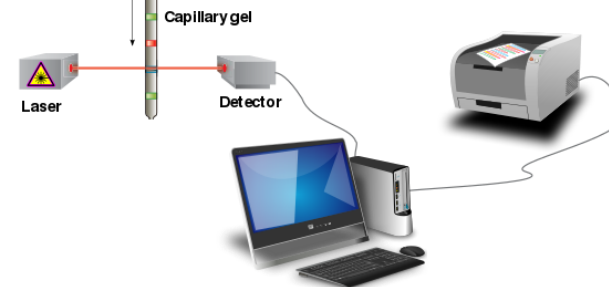
- ▶ Primer and DNA template ▶ DNA polymerase
- ▶ ddNTPs with flouochromes ▶ dNTPs (dATP, dCTP, dGTP, and dTTP)



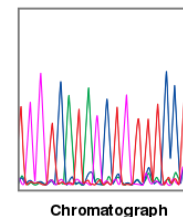
② Primer elongation and chain termination



③ Capillary gel electrophoresis separation of DNA fragments



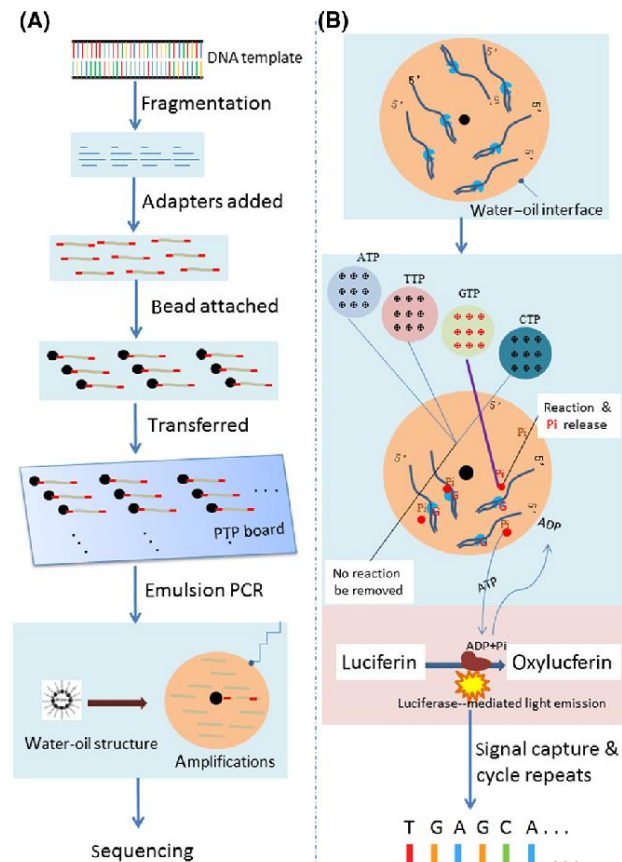
④ Laser detection of flouochromes and computational sequence analysis



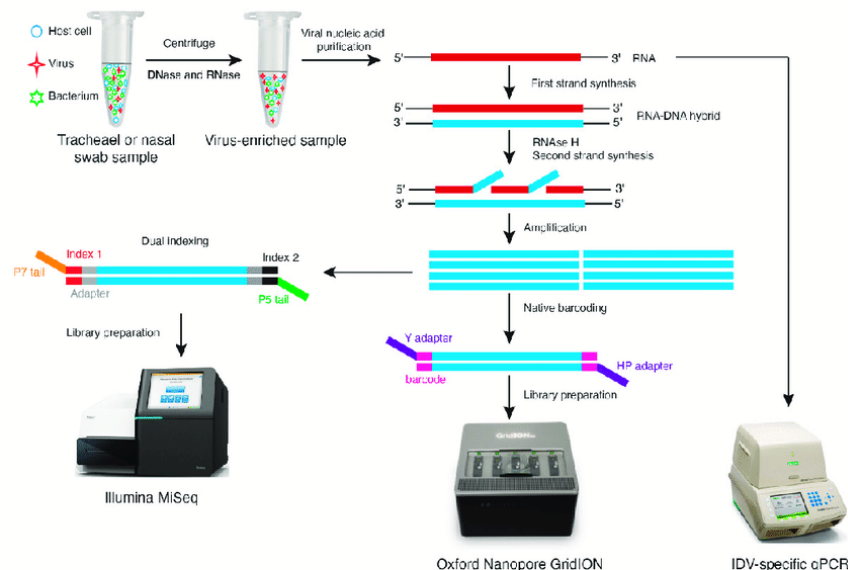


Simultaneously, with Sanger's sequencing efforts, a luminescent method was developed to measure pyrophosphate synthesis. Pyrosequencing, as this technique was called, was subsequently licensed to 454 Life Sciences, a biotechnology company founded by Jonathan Rothberg. This company evolved into the first major commercial "next-generation sequencing" (NGS) technology. In 2004, the first high-throughput (HTS) sequencing machine that was massively available to the public was the 454 GS/20 Roche machine, which offered an increase in the number of reads (up to 100 bp), as well as improved data quality. The greater number of reactions, generated in parallel sequencing on a micrometer scale, often made it possible as a result of improvements in microfabrication and high-resolution images, was the point that defined the second generation of DNA sequencing.

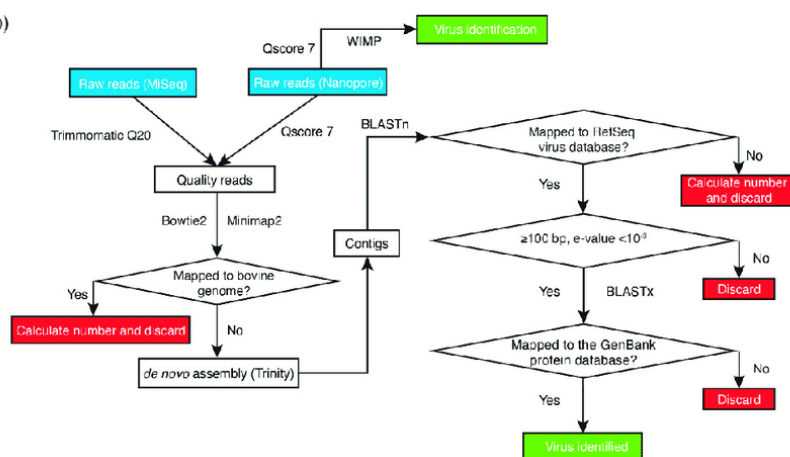
Schematic representation of Roche 454 pyrosequencing.



(a)



(b)

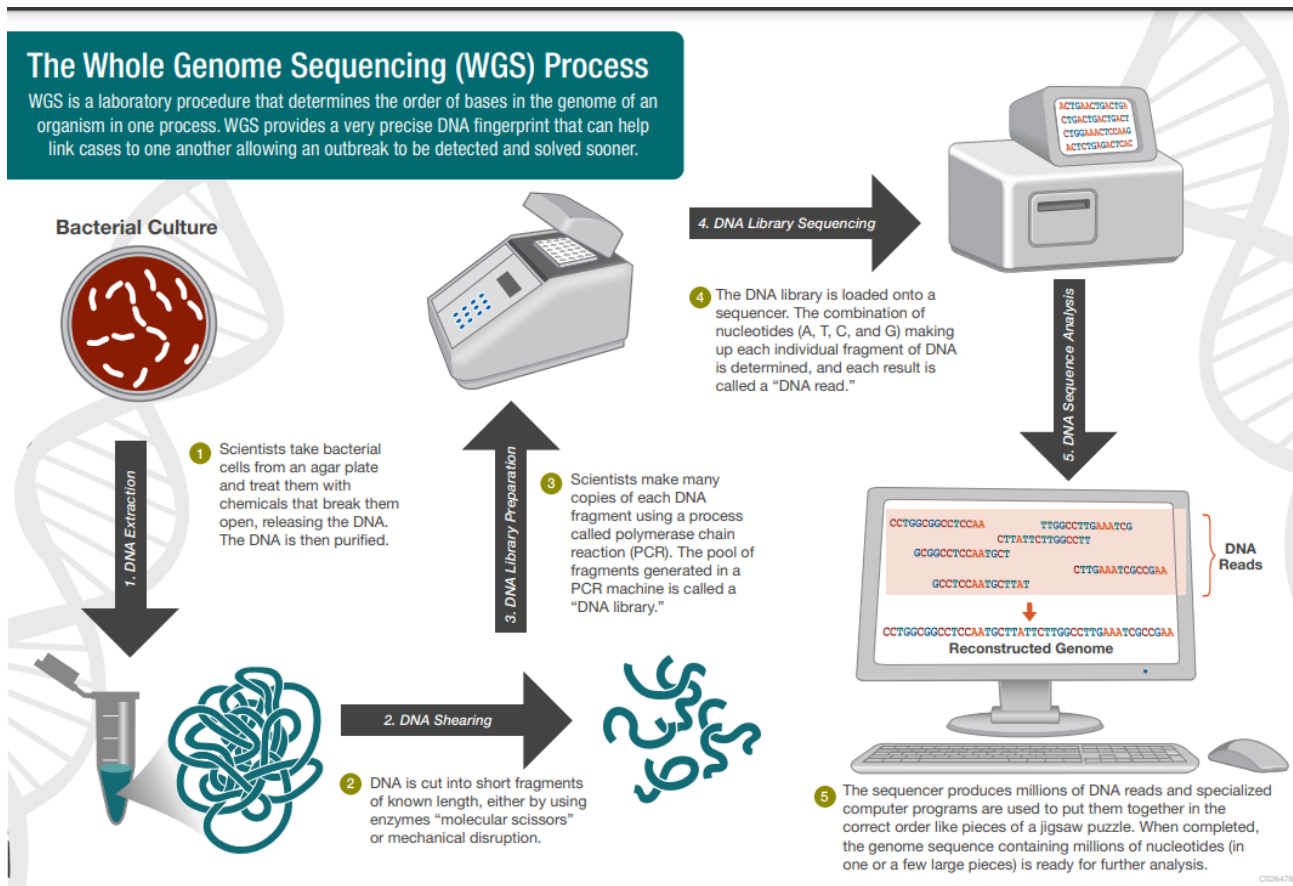


In early 2012, the first nanopores sequencing platform was announced by Oxford Nanopore, introducing two main versions of sequencers: GridION and MinION, capable of generating large amounts of data, with a simple sample preparation resulting in long reads to a low cost. MinION is a small, portable device, capable of sequencing 30 Gb of DNA, while GridION can generate up to 150 Gb of data transmitted in real time for immediate analysis. Another technology launched by the same company was PromethION, which can generate up to 8 Tb of data.

Whole-Genome Sequencing and Whole-Transcriptome Sequencing

The improvement of NGS technologies made it possible to carry out genome sequencing and complete transcriptome projects on a large scale. The analysis of genomes and complete transcriptome enabled the identification of gene function within the biological context. For example, with the sequencing of a genome, it was possible the identification of genes that generally do not function independently, and their functions are not controlled directly by the promoter, but by many other regulatory elements, such as intensifiers, response elements, and silencers.

Whole-Genome Sequencing



CS204789-A

Overview of the most common sequencing platforms

| Platform | Advantages | Drawbacks | Recommended applications |
|-------------------------------------|---|---|--|
| Sanger | <ul style="list-style-type: none"> Costs (low target number) Established workflow Simple data analysis | <ul style="list-style-type: none"> Sensitivity Scalability Sample input requirements | TRS, validation of NGS data |
| Ion Torrent | <ul style="list-style-type: none"> Costs Speed | <ul style="list-style-type: none"> Short length reads Accuracy | TRS, metagenomics |
| Illumina | <ul style="list-style-type: none"> Sensitivity Amount of generated data with same DNA High throughput | <ul style="list-style-type: none"> Costs for low target numbers Short length reads | WGS, WES, TRS, RNAseq, epigenomics, metagenomics |
| BGI Group | <ul style="list-style-type: none"> Accuracy No optical duplicates | <ul style="list-style-type: none"> Short length reads | WGS, WES, TRS |
| Pacific Biosciences | <ul style="list-style-type: none"> Long reads High accuracy with CCS mode Direct detection of epigenetic modifications | <ul style="list-style-type: none"> Costs Large amounts of starting material Error rate with CLR mode | WGS, TRS, RNAseq |
| Oxford Nanopore Technologies | <ul style="list-style-type: none"> Very long reads Direct sequencing of RNA Detection of RNA modifications | <ul style="list-style-type: none"> Costs Error rate Large amounts of starting material | WGS*, TRS, RNAseq, epigenomics, metagenomics |

*Small whole genome sequences.



Review Questions

- Explain the Next generation sequencing.
- What sequencing procedures exist?



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 2. Genetics

Topic 3. Computer Genomics

Lesson 2. Sequence Alignment



Contents

- Introduction
- Biological Sequence Alignment
- Pairwise Sequence Alignment and Dynamic Programming
- Multiple Sequence Alignment (MSA)



Introduction

The comparison of biological sequences allows us to confront the differences between organisms and species at the gene level. Comparative genomics, a branch of science that exhaustively uses bioinformatics techniques to track genes in various species and study their similarities and differences, uses these studies to infer the functional and structural characteristics of newly discovered or existing proteins. The analysis of biological sequences does not differ much from the techniques used to compare strings and texts and, therefore, the concept of alignment becomes very important. Sequences that evolve in species and clades through mutations include insertions, deletions (indels), and incompatibilities. When comparing two biological sequences, an alignment is generated to visualize the differences between the sequences at each position.



Introduction

Sequence alignment is one of the main tasks of bioinformatics. It consists of aligning a query sequence with a reference sequence, which is usually in a public database of sequences, with the aim of determining whether they have correspondences with each other that are statistically significant. It differs from the classic computational problem of exact string matching, where there is an interest in finding exact matches. String alignment is an approximate string match or string match problem that allows for errors. The problem, in its most general form, is to find into a text (or sequence of characters) the position where a certain pattern occurs, allowing a limited number of errors in the correspondences. The distance between the two sequences is defined as the minimum sequence of operations necessary to transform one into the other. With respect to probability, a cost is assigned to operations, so that the most likely operations cost less. The objective is to minimize the total cost. Ultimately, the final goal of sequence alignment is to determine the similarity between parts of the genomic code. Among the known applications of this type of task, we can mention the discovery of genes, prediction of function, and assembly of the genome sequence.

Biological Sequence Alignment

An alignment between two strings is simply the matching of pairs between the letters in each string. The alignment of nucleotide or amino acid sequences is able to reflect the evolutionary relationship between two or more homologous sequences that share a common ancestor. If the same letter is present in both sequences, the position was preserved in the evolution. If the letters are different, then it is possible to infer that the two strings are derived from an ancestral letter (which may be one of the two or none). However, sequences that are homologous can have different lengths, which can be partly explained by insertions or deletions in the sequences. In this way, a letter or a section of letters can be paired with dashes in the other sequence to signify this insertion or exclusion.

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | T | T | G | C | A | T | C | A | A | G | C | T | A | T | A | A | A | T | T | G | C | A | A |
| A | T | G | - | A | C | - | C | A | A | - | G | - | A | T | A | A | A | - | - | C | C | A | A |

Pairwise Sequence Alignment and Dynamic Programming

Pairwise alignment consists of comparing two sequences with one another to find the best possible alignment between them. The process involves a scoring system for each position where there is a match, mismatch, and indels. Since matches are preferred over deletions, they normally receive the highest scores and the lowest scores are assigned to insertions. The similarity between two sequences is inversely proportional to the number of mismatches and indels in the alignment. Different scoring models were developed based on the statistically relevant frequency of one amino acid becoming another.

There are two types of alignments for sequence analysis in pairs based on the dynamic programming method: Global and Local Alignment.

Global Alignment

Also called end-to-end alignment. The idea behind the method is to try to align all the residues in each sequence. This approach is useful when the sequences being compared are similar and of approximate size. Needleman and Wunsch were the first to present an algorithm capable of finding the global alignment between two amino acid sequences. The algorithm is based on dynamic programming and achieves the global alignment of two sequences. The algorithm covers three main steps: initialization, calculation, and trace back. A matrix of dimensions i, j is initialized, where i and j are the length of the two strings in comparison. Next, the highest score $F(i,j)$ for each comparison in each position is calculated,

$$F(i,j) = \max \{F(i-1, j-1) + s(X_i, Y_i), F(i-1, j) - d, F(i, j-1) - d\},$$

where $s(X_i, Y_i)$ is the match/mismatch score and d is the penalty for deletion.

After calculating the maximum score for each position in the matrix, the trace back starts from the last cell (bottom right) in the matrix. At each step, it moves from the current cell to the one from which the current cell value was derived. A match or mismatch is assigned if the maximum score was derived from a diagonal cell. An insertion/deletion is assigned if the score was derived from the top or left cell. After the trace back is complete, there are two sequences aligned end to end with an optimal alignment score.

Local Alignment

This type of arrangement is most useful for different sequences that probably contain regions of similarity in the larger context of the sequence. Smith and Waterman (Smith et al. 1981) introduced a different algorithm for scoring similarities in order to find optimal local alignment subsequences, even at the cost of the global score. The algorithm achieves local alignment of strings and is quite similar to the Needleman–Wunsch’s method. Local alignment can be used in situations where you want to align smaller substrings from two sequences. In the biological context, such a situation can arise during the search for a domain or motif within larger sequences. The algorithm comprises the same steps as Needleman–Wunsch, however, with two main differences. The calculation of the maximum score also includes an option of 0:

$$F(i,j) = \max \{0, F(i-1, j-1) + s(X_i, Y_i), F(i-1, j) - d, F(i, j-1) - d\}.$$

| | | M | V | S | S | D |
|---|----|----|----|----|----|-----|
| | 0 | -2 | -4 | -6 | -8 | -10 |
| M | -2 | 2 | 0 | -2 | -4 | -6 |
| V | -4 | 0 | 4 | 2 | 0 | -2 |
| S | -6 | -2 | 2 | 6 | 4 | 2 |
| D | -8 | -4 | 0 | 4 | 5 | 6 |

Alignment 1

| M | V | S | S | D |
|---|---|---|---|---|
| M | V | S | - | D |

Alignment 2

| M | V | S | S | D |
|---|---|---|---|---|
| M | V | - | S | D |

Needleman–Wunsch matrix. The calculation uses scores: +2 for match, -1 for mismatch, and -2 for gap. The arrows show the matrix cell from where the value is generated. Cells with values in red show the trace back that creates the alignment

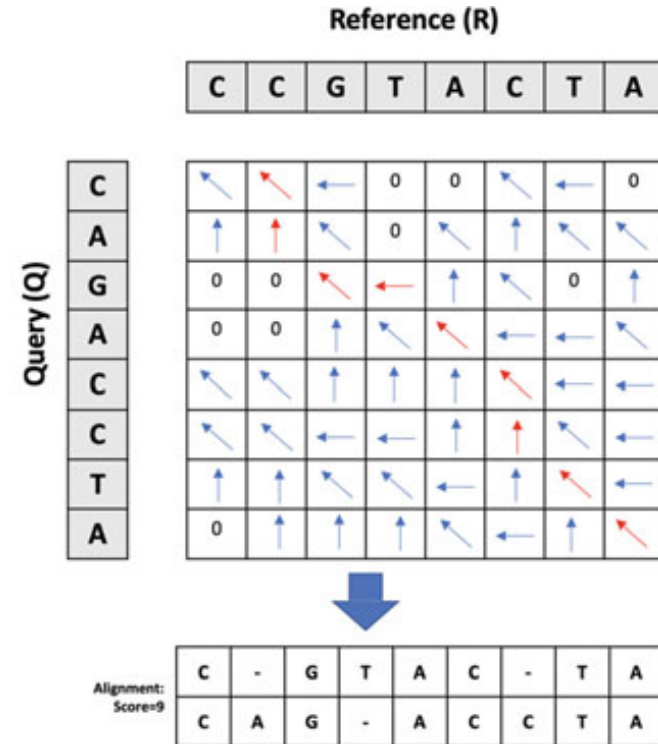
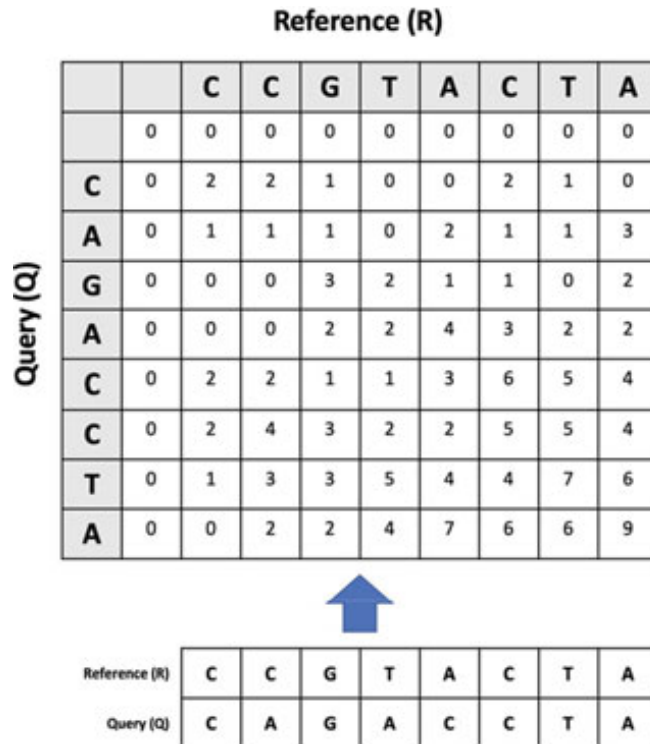
The assignment of 0 as the maximum score corresponds to the beginning of a new alignment. This allows the alignments to end anywhere in the matrix. The trace back, therefore, starts from the highest value of $F(i, j)$ in the matrix and ends where it finds 0.

Multiple Sequence Alignment (MSA)

When it comes to biological sequence analysis, one of the biggest challenges is to decode the large number and length of the sequences. Biological databases store a vast amount of proteins and DNA sequences and gather more than 100 million sequences, of the most distinct species of nature. Although alignment methods based on dynamic programming are quite accurate and can achieve good alignments based on scores, they are slow and impractical for these databases with millions of sequences. The time complexity of dynamic programming algorithms is $O(mn)$, that is, the product of the sequence lengths. As an initial way of trying to improve the speed of comparison between sequences, heuristic algorithms such as BLAST, BLAT, and FASTA were created. In the same direction, algorithms such as LSCluster, Usearch, Vsearch, Diamond, and Ghostx have been proposed to try to improve the search efficiency by similarity.

In general, these algorithms look for exact matches and extend the alignment of those matches, trying to estimate the ideal score alignment. Thus, heuristic algorithms with approximate correspondence approaches try to solve the multiple sequence alignment by finding similarities between them, as is the case of the CLUSTAL software family, which uses the progressive algorithm of Feng and Doolittle (1987).

Smith–Waterman matrix with linear gap penalty. The calculation uses scores: +2 for, -1 for mismatch, and 1 for gap. The left matrix represents input sequences and the right matrix represents sequences are alignments. The left matrix is the corresponding $(n + 1)$ by $(m + 1)$ score matrix. The right matrix is the trace back matrix, with red arrows indicating the optimal alignment path. The null pointer is represented as 0



BLAST (Basic Local Alignment Search Tool) is a software based on the idea that the best scoring sequence alignment should contain the largest number of identical matches or high-scoring sub-alignments. The algorithm works by performing the following steps:

1. Reducing the query sequence into small subsequences called seeds;
2. Searching for these seeds across the entire database looking for exact matches;
3. Extending the size of the exact matches into an ungapped alignment until a maximum scoring extension is reached.



The use of seeds to first search for exact matches greatly increases the entire search process and alignment without gaps loses only a small set of significant matches. BLAST's accuracy and sensitivity have made it one of the most widely used search algorithms in the biological world. A variant of BLAST called Position-Specific-Iterative BLAST (PSI-BLAST) extends the basic BLAST algorithm. PSI-BLAST performs several BLAST iterations and uses the hits found in one iteration as a query for the next iteration. Although PSI-BLAST responds slower to the large amount of calculations required, it is considered a reliable tool to finding distant homology relationships.



Although BLAST and PSI-BLAST are still widely used, some lately developed methods offer results with greater precision and sensitivity. Hidden Markov Models (HMM) have been used efficiently in numerous applications to understand and explore biological data. An example is HMM-HMM (HHblits) fast sequence search (Remmert et al. 2012). The tool can be used as an alternative to BLAST and PSI-BLAST and is 50–100 times more sensitive. This high sensitivity of the tool can be attributed to the algorithm that is based on the comparison of the HMM representations of the sequences. Although profile–profile or HMM–HMM alignments are very slow due to calculations, the HHblits prefilter reduces the required alignment scaling from millions to thousands, increasing its speed considerably. HHblits represents each sequence in the database as an HMM profile. This pre-processing reduces the number of HMM comparisons to search for similarity, selecting only those target sequences where the highest alignment without gap exists. At the end, a Smith–Waterman alignment shows a significant E-value.

There is another set of methods used to perform Multiple Sequence Alignment (MSA), while reducing errors inherent to progressive methods, they are called iteratives. These categories work in a similar way to progressive methods, but they realign the initial sequences repeatedly as well as they add new sequences to the growing MSA. A very used iteration-based algorithm is called MUSCLE (Multiple Sequence Alignment by Log-Expectation) and improves the performance of progressive methods through a more accurate distance measurement to assess the relationship between two sequences.

These methods exploit the search space in an account to find an almost optimal result. Usually, Metaheuristics are categorized in:

1. **Single-Solution Based Methods:** this category includes a local search algorithm that is primarily concerned with modifying and improving individual possible solutions. The unique solution-based method includes Tabu Search (TS),
2. **Simulated Annealing (SA), Variable Neighborhood Search (VNS), Iterated local Search (ILS),** but not restricted to these methods only. **2. Population-Based Methods:** population-based metaheuristics begin its process with the starting population and keep iterating until any stop criteria are met. These methods are imitations or inspired by natural phenomena. They use bio-inspired operators such as selection, crossing, mutation, to generate the pool of descendants from the previous population. This is the main difference between Metaheuristic single-solution based and population-based methods. Population-based Metaheuristic methods include Evolutionary Algorithms (EAs), Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Group Search Optimizer (GSO), Artificial Immune System (AIS), etc.



Review Questions

- Explain the two types of alignments for sequence analysis in pairs based on the dynamic programming method.
- What is characteristic for multiple sequence alignment?



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 2. Genetics

Topic 3. Computer Genomics

Lesson 3. Genome Assembly and Annotation



Contents

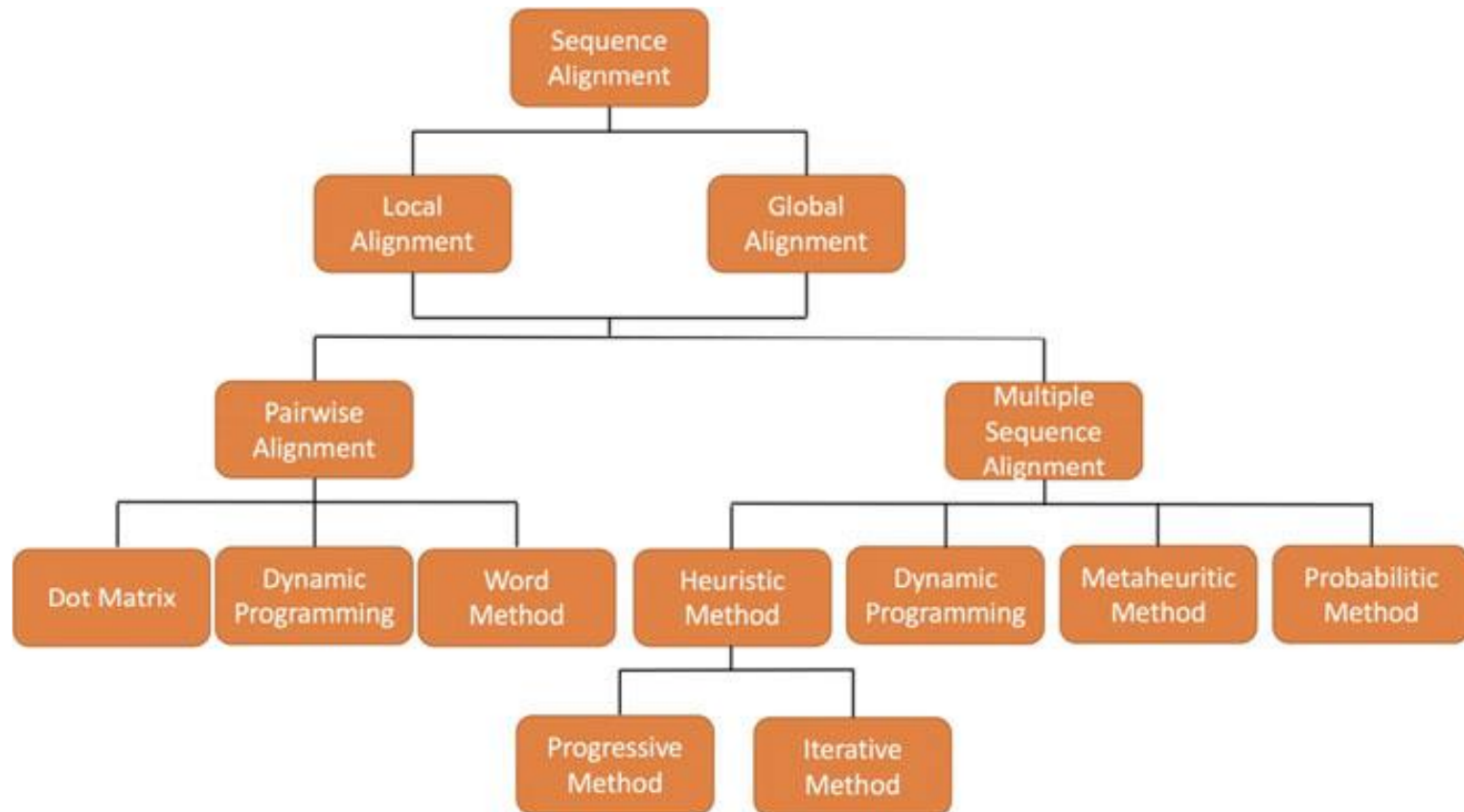
- Introduction
- Reference-Based Assembly
- De Novo Assembly
- Hybrid Assembly
- Gene Prediction and Annotation



Introduction

From the data generated by NGS technologies, several new applications have emerged, such as the study of microbial communities, the discovery of structural variants in genomes, and the analysis of gene structure and expression. Due to the small length of sequences generated by the most common NGS platforms, many of these analyses begin with the computational process of sequence assembly, which consists of grouping the generated fragments based on their base identity. There are two general approaches to assembling NGS fragments: reference-based and de novo approaches. In the reference-based assembly, a reference genome of the same organism or related species is used as a guide to align the reads; this is, in many cases, the analysis of resequenced data.

Different methods for sequence alignment



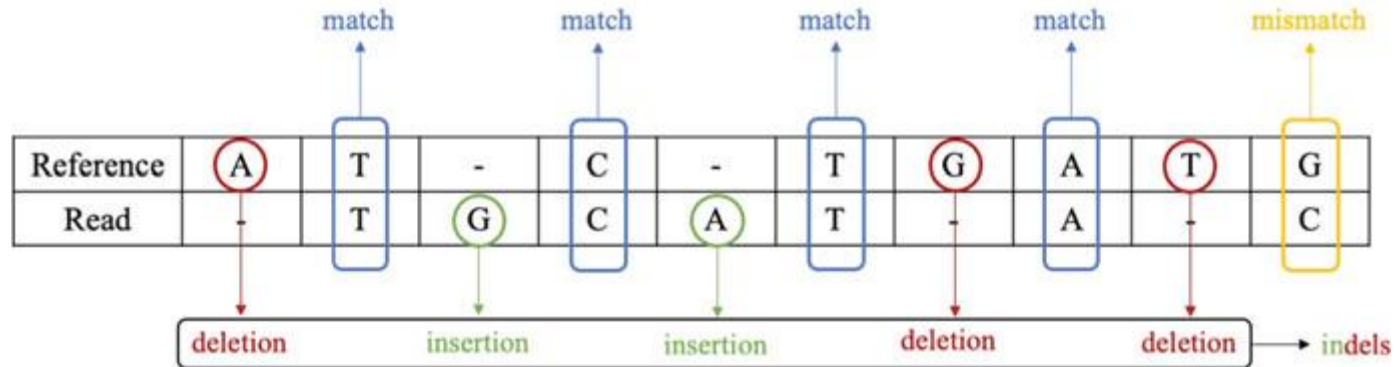
Reference-Based Assembly

The reference-based assembly requires less computational cost when compared to the de novo approach. It is a technique to identify the differences between the reads obtained in a sequencing compared to a previously available reference genome, traditionally used in resequencing, but not limited to this.

In the reference-based assembly each read is compared to the reference sequence, base by base, trying to map those bases. In this scenario, there are four possibilities: matches, mismatches, and insertions and deletions (indels). A match happens when one base is mapped against the reference sequence; a mismatch happens when one base is not mapped against the reference genome. An insertion happens when one base is present in the read sequence but not in the reference sequence, while a deletion happens when the base is present in the reference sequence but not in the read sequence. The combination of insertions and deletions is called indels. In order to evaluate how well the alignment is, the number of matches is counted and then divided by the size of the sequence. This division represents the percentage of identity.

Reference sequence: ATCTGATG (n = 8)

Read sequence: TGCATAC (n = 7)



matches = 4 mismatch = 1 insertions = 2 deletions = 3 percentage of identity = $4/7 \sim 51\%$

Due to the high coverage of sequencing provided by Next-Generation Sequencing platforms, analysis of variants in the genome such as SNPs and SNVs is based on mapping reads against a reference genome, where the alignment achieves a minimum score, usually represented by the amount of matches, mismatches, indels, and/or percentage of identity.

Mapping Algorithms and Tools

The alignment in the figure shows two very simple and small sequences to demonstrate simple concepts. In reality, considering the size of reference genome sequences and the amount of data generated by NGS technologies, some robust methodology is required. Mapping efficiency of reads against the reference sequence, highly accurate, is determinant for the quality of downstream analysis.

Over 50 different mapping algorithms exist. Most of them require special data structures, indices, constructed for reads of sequences and the reference sequence. Based on how these algorithms use their indices, it is possible to group them in two categories: hash tables-based algorithms and Burrows–Wheeler transform (BWT)-based algorithms.

Mapping Algorithms and Tools

Hash tables-based algorithms are grouped in types: those that hash the genome and those that hash the reads. The main concept for both types is to construct a hash table for subsequences of reads and genomes. The hash key for each entry in the hash table is a subsequence, and the value for that key is a list of the coordinates where this subsequence can be located. Examples of hash tables-based algorithms: GSNAP, FANGS, and MAQ.

BWT-based algorithms are very efficient in indexing data and maintaining small memory usage when a search is performed. Current BWT-based tools use a modified version of BWT algorithm that uses a different type of data structure, called FM-index, Created by Ferragina and Manzini (2000). The transformation of the genomes into a FM-index improves the search performance, improving the algorithm as a whole. Because of its efficiency, BWT-based algorithms became the most used in mapping applications. In this context, two software stand out: BWA and Bowtie.

Mapping Algorithms and Tools

Bowtie begins by constructing the FM-index for the reference sequence, then it uses a modified version of the Ferragina and Manzini mapping algorithm to locate the position of the alignment. Currently, two versions of bowtie can be found, Bowtie and Bowtie 2. Bowtie 2 was developed mainly to handle reads longer than 50 base-pairs, while the first version of bowtie handles only sequences up to 35 base-pairs.

BWA is very similar to Bowtie; it also uses a modified version of the Ferragina and Manzini (2000) mapping algorithm to find exact matches. To handle inexact matches, BWA searches for matches among subsequences of the reference sequence minding a certain distance defined. In general, Bowtie is best suited for most analyses, while BWA performs better for longer reads.

De Novo Assembly

The de novo assembly approach is based on the overlapping of the reads or part of them with another. This strategy is useful to unknown genomes: new strains or species, and is able to represent regions which cannot be identified by reference assembly due to its absence in the reference genome.

To improve the accuracy of the de novo assembly is highly recommended to remove the low quality bases (Phred metric) of the ends of the reads, and that reads with low quality scores to avoid missassemblies. After the launch of next generation sequencing platforms, the challenge became to group short readings (<30 bp) based on its identity to produce long sequences (contigs), and in turn contigs can be ordered and oriented to generate scaffolds.

De Novo Assembly

One of main difficulties is assembly of repeated regions, greater than reads length, of the genome and some bases or regions cannot be represented in the assembled genome, these regions are called gaps and are usually represented by N.

Actually, the reads length increases and the use of paired libraries is useful to address some repeated regions for the prokaryote, but it remains a real problem during the assembly of eukaryotes due to the larger repeated regions.

The main strategies used to assembly genomes are Eulerian De Bruijn Graph, Hamiltonian De Bruijn Graph, String graph, and Overlap-Layout-Consensus (OLC) which are implemented with some differences by many assemblers, such as ALLPATHS, Velvet, ABySS, SOAPdenovo, and SPAdes.

De Novo Assembly

Before executing any approach the user can adopt some strategies to correct errors on the reads, most of them based on the frequency of reads or k-mers to define the confident and erroneous sequences.

The quality of the results obtained from the genome assembly process can be evaluated based on the contig length, amount of bases and contig generated, and how large the sequences produced are, to explore the results the common metrics to be evaluated are: N50—The N50 value means that 50% of the bases generated by the assembly process are part of contigs with length greater than or equal to N50 value; NG50—the same for N50 but the percent of bases in the reference genome selected; L50—the number of contigs used to reach the N50 value. Other metrics should be evaluated too, such as number of base produced, number of contigs and missassemblies when a reference genome is available to check with software such as Quast.

Hybrid Assembly

New hybrid strategies have been developed to take advantage of each type of assembly. Among them, it is possible to highlight techniques that combine reads and assemblies from different sequencing technologies and different assembly algorithms that can be applied in several tasks, such as de novo assemblies, sequencing error correction, and sequence quality improvement. This type of hybrid assembly makes use of reads from different sequencers to reconstruct the genome, mostly using overlap-layout-consensus based methods. Another hybrid assembly approach occurs when different assemblers are used. Rather than performing assembly from reads, this kind of hybrid strategy, also known as meta-assembly, uses assemblies generated by different assemblers, combining the results (contigs and/or scaffolds) produced by those tools to produce a new sequence. However, the concepts of hybrid assembler and hybrid assembly should not be confused. When it comes to assemblers, “hybrid” refers to the ability of an assembler to work with short and long reads, while in relation to the assembly process, “hybrid” refers to the use of more than one type of assembly strategy (DBG/OLC), sequencer (regardless of read type), or input dataset (read/mount).

Gene Prediction and Annotation

After finishing the genome assembly process, having a whole genome or a draft, the identification of the Open Read Frames (ORFs), sequence between the start and stop codon, is the next step, followed by gene annotation: the process to get metadata regards the genic product for each ORF identified. The gene annotation often is based on biological database that shows the function, products, and processes that gene can be involved beyond other information. Some methods adopted to do the gene prediction are based on a training dataset, so when this set of genes chosen for training is not good, it can lead to bad results. For the training task, most programs today use Markov models (HMM—Hidden Markov Models or IMM—Interpolated Markov Models) (for example, SNAP; GlimmerHMM; GeneMark; GlimmerIMM) for this training, where the genes are modeled with the Markov models that use a series of states to represent a generic structure of the genes. Data training for gene prediction and annotation programs is often chosen at random from a subset of high-quality genes that ideally represent the variation found in a genome. When programs for gene prediction and annotation are trained on a grass genes subset with random GC content, they are effectively being trained on two classes of genes at the same time, and this may result in poor output when genes are predicted in new sequences of genome.

Gene Prediction and Annotation

Actually, the sequence of the human genome can be done for less than a thousand dollars. Due to this reduction in the sequencing price, there was an advance in the assembly and alignment algorithms. As a result, obtaining a high-quality assembly draft became an achievable goal for most genome projects. This caused the bottleneck in genomic studies to change focus, genome annotation has become a challenging task due to the difficulty of collecting or predicting proteins, mainly for large genomes, requiring other data sources, such as RNA-Seq and databases to train, optimize, and configure gene annotation tools.

The manual curation can be used to improve the quality of gene annotation to describe the Gene Ontology (GO) Terms or the gene products based on biological annotation database, such as Blast2GO and GoFeat, which use annotated genes and its structural similarity to take new information and insights, ever based on computational approaches representing most of the annotations found on the biological databases.

Gene Prediction and Annotation

The accuracy of gene annotation is essential to next analysis to evaluate the genes found and their relationship in the organism, which will drive to discoveries about functions and phenotypes which can be associated to the organism to many applications, such as pathogen–host interactions and antibiotic resistance.

One of the main limitations in the genomic annotation task comes from need of database with annotations already made. There are some areas of biology that are more studied and therefore have more data (complete data, better described, and sometimes curated) for known processes, beyond the amount of databases for specific analysis that are not integrated on the big databases such as Genbank, DDBJ, and EBI. Nowadays with the evolution of annotation programs, most of them are now automated, for example, RAST, PATRIC. These pipelines basically have two tasks: searching for patterns that identify the species gene (e.g., ESTs—Expressed Sequence Tag, proteins, RNA-Seq) and characterizing these patterns into a database (e.g., Interpro, Uniprot, Pfam using Blast or Diamond).



Review Questions

- What types of genome assembly exist?
- Explain each of them.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 2. Genetics

Topic 3. Computer Genomics

Lesson 4. Biological Interaction Networks



Contents

- Introduction
- Biological Network Properties
- Types of Biological Networks
 - ✓ Metabolic Networks
 - ✓ Signaling Network
 - ✓ Gene Regulation Network
 - ✓ Protein-Protein Interaction Network
 - ✓ Biological Co-Expression Network



Introduction

Biological networks are used in different biological sciences, such as the study of the interactome, cancer study, drug prediction, metagenome analysis, proteomic analysis, molecular interactions, and cell interactions, among other areas.

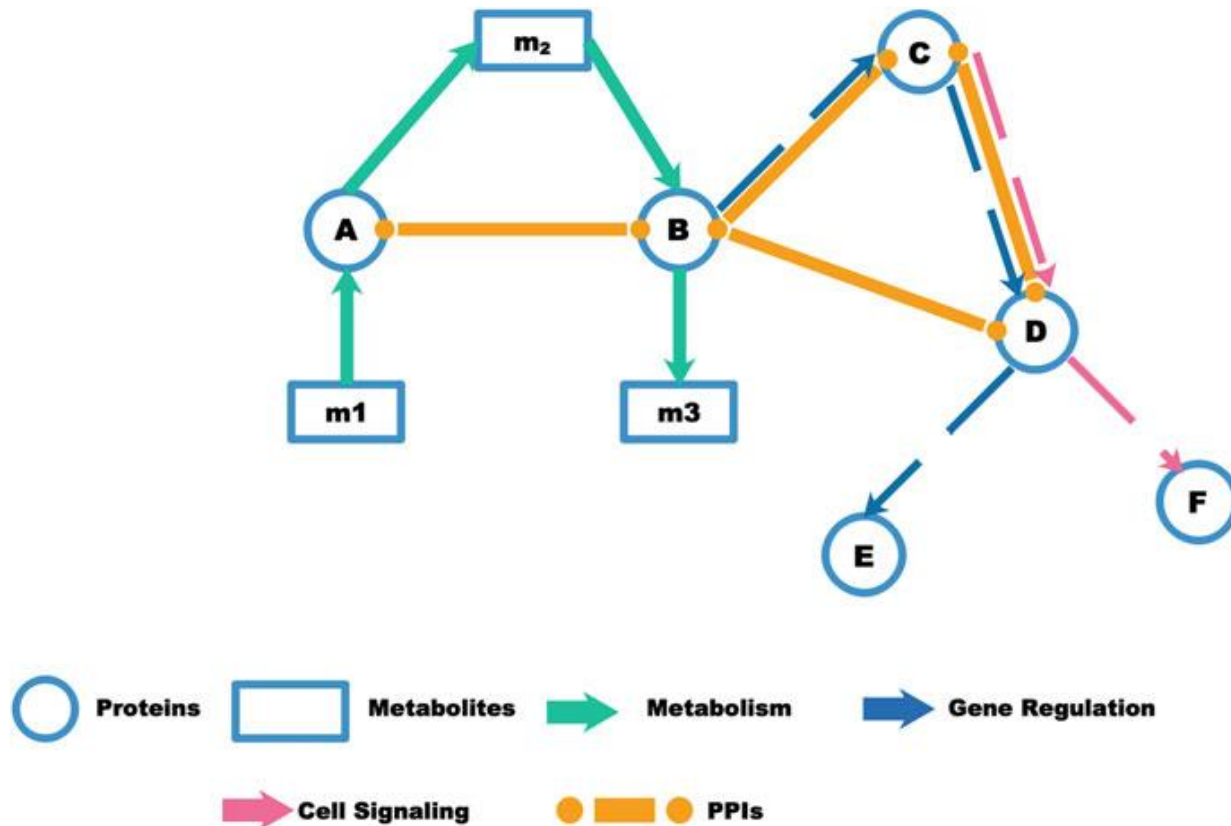
A biological network can be defined as a collection of units (biomolecules), potentially interacting as a system. In other words, a biological interaction network can be represented an abstraction of the interactions obtained through mathematical or computational models, where a uniform set of nodes connected by a uniform set of edges that can be directed or undirected are represented. In this type of network, the nodes can represent biomolecules (genes, protein, neuron, organisms, cells, among others), and the borders usually represent relationships and interactions (biochemical, transcriptional, energy flow, regulation, co-expression, metabolic, among other).



The study and analysis of networks is part of network biology. This paradigm allows us to understand the complex interactions of biomolecules within cells by representing and analyzing biological systems through tools and methods derived from graph theory, mathematics, physics, statistics, machine learning, and other, applies to and omics and biological data.

The inferences of biological networks using NGS data allow obtaining relevant information about expression and regulation processes inside the organisms. Biological interaction networks can be built using different methods of reverse engineering that use high and low throughput data, as well as statistical, mathematical, and computational techniques that allow reconstructing how the elements of biological networks integrate as a system.

Types of biological interactions that can be represented by networks





Biological networks have been used to study transcription-regulation processes in Escherichia Coli, where it has been demonstrated that essential molecular elements contribute to the specialization of the dynamics of global responses, which allows the bacteria to have a more robust and quicker response to processes and environmental signs. They are also used to study networks of metabolic interactions; study host-pathogen interactions, discover new measurements, identify biomarkers; identification of genes involved in specific cell cycle processes; identification of disease-related genes critical biological processes. Networks have been used to generate models of the relationship between elements of biological data sets, as well as the analysis of chromatin formation within cells; the identification of metabolic pathways related to genetic regulation; as well as to model protein-protein interactions that take place within organisms.

The networks of biological inferences can be classified into four types: protein-protein, Gene regulation networks, metabolic networks, signaling networks, and co-expression networks.

Biological Network Properties

The biological interaction networks have specific architectures and properties that enable the analysis and interpretation of the complexity of the interactions present within the different domains and elements present in them. Some of the elements and property that are part of the networks are as follows:

- Node: it is an individual element within the network;
- Edge: represent the interactions and interconnections between nodes;
- Components: are a group or groups of nodes that are mutually connected;
- Degree/connectivity: the number of edges connected between a focal node and the other nodes;
- Network density: describes the portion of potential connections on a network that are real connections. A potential connection is a connection that can exist between two nodes, regardless of whether they exist or not;
- Betweenness: is the metric that measures how a node is in the path between the other nodes. Nodes with a high centrality may have a strong influence due to their control over the passage of information within the network;

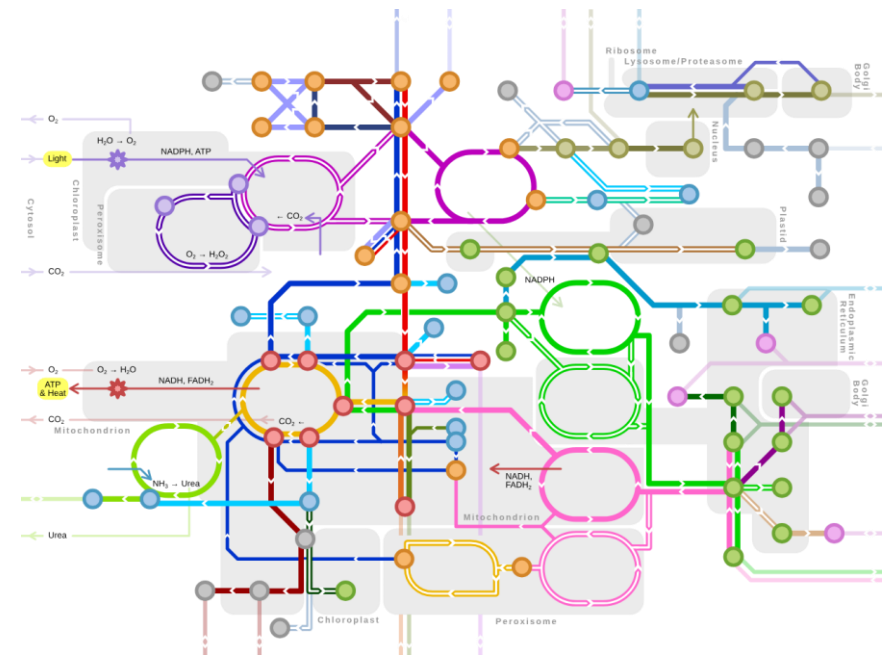
Biological Network Properties

- Closeness: is the measure of the shortest path between one node and the other nodes within the network;
- Clustering coefficient: it is a measure of the proportion of neighbors reached through a node to the other neighbors. This metric shows the degree to which the nodes in a network tend to group. This metric allows measuring the cohesion of the network;
- Degree of distribution: it is the distribution of the frequencies of the degrees of the nodes individually for an entire network;
- Modules or clustering: a set of densely interconnected nodes within the network;
- Motif: they are small subnets or patterns that are statistically overrepresented within the network;
- Clique: consists of a fully connected subnet within a given network;
- Directed graph: nodes in a directed graph are connected by an asymmetric relationship, such as predation;
- Undirected graph: nodes in an undirected graph are connected by a symmetric relationship, such as physical interactions.

Types of Biological Networks

Metabolic Networks

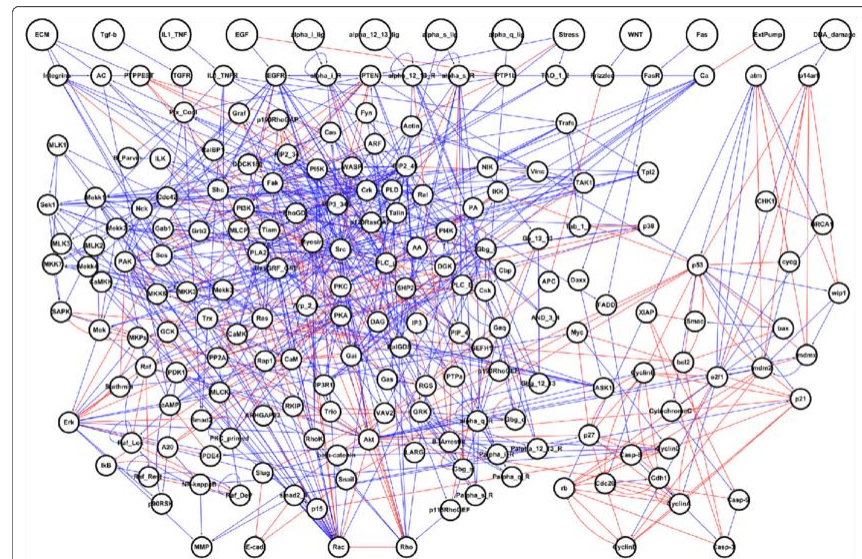
This network type allows the annotation of genes and metabolic ones by determining elements of relationships, structure, and dynamics of metabolic networks. This network infers the enzymatic function of a specific protein or reconstructs the metabolic pathways in which it participates.



Types of Biological Networks

Signaling Network

It allows representing abstractions of molecular interactions and chemical modifications that act in a chain to transport stimuli (hormones, pathogens, nutrients) detected by the cell membrane receptors to the cell nucleus, to coordinate the beginning of the appropriate metabolic and genetic responses. For the reconstruction of this type of network, techniques such as genetic knockout have been used, which allow studying the different responses of organisms to this stimulus.

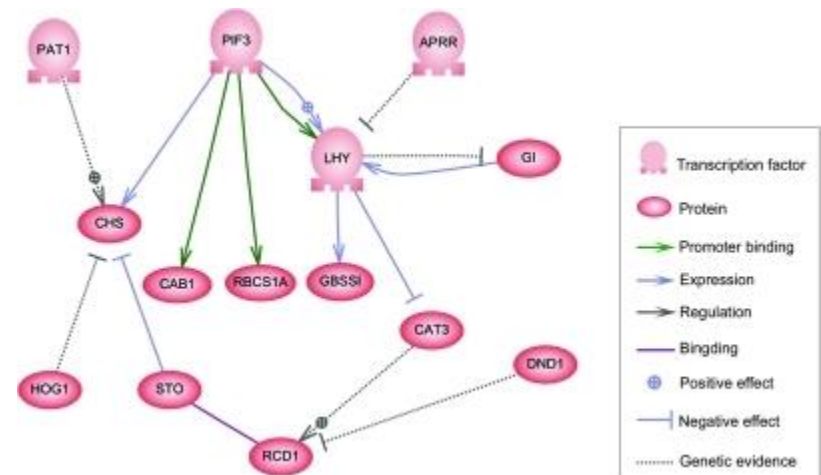


Types of Biological Networks

Gene Regulation Network

Also known as the transcription regulation network, networks represent the casual interactions between transcription factors and genes. They are usually represented as a directed graph, whose direction is defined by the genes' expression. The interconnections between genes can represent biochemical processes such as reaction, transformation, interaction, activation, or inhibition.

The gene regulation networks present the set of activation and inhibition gene interactions within cells. Several transcription factors and gene products participate in the transcription process, regulating, directly or indirectly, other biomolecules within the genome, through regulatory chains. On the other hand, feedback loops can also be generated within this process, regulating negatively (downregulated) or positively (upregulated) gene product production.



Mapping Algorithms and Tools

Protein-Protein Interaction Network

Protein-protein interaction (PPIs) networks consist of proteins and their interactions. In this network, the nodes represent the proteins, and the edges correspond to the interactions between the proteins. Proteins are organized into different putative complexes, each performing a specific task or process within cells. A protein-protein interaction occurs when two or more proteins come together temporarily to modify each other, trigger signal transduction, or perform specific biological functions for a prolonged time.

The construction and analysis of PPI networks enable identifying protein complexes, which permits the study and understanding of the mechanisms that regulate life, explaining the evolutionary orthology signal, the prediction of biological functions of uncharacterized proteins, and drug target detections for specific diseases. One of the most used techniques for detecting protein complexes is clustering techniques, which allow groups of proteins that share similarity or common domains to be identified.

Mapping Algorithms and Tools

Protein-Protein Interaction Network

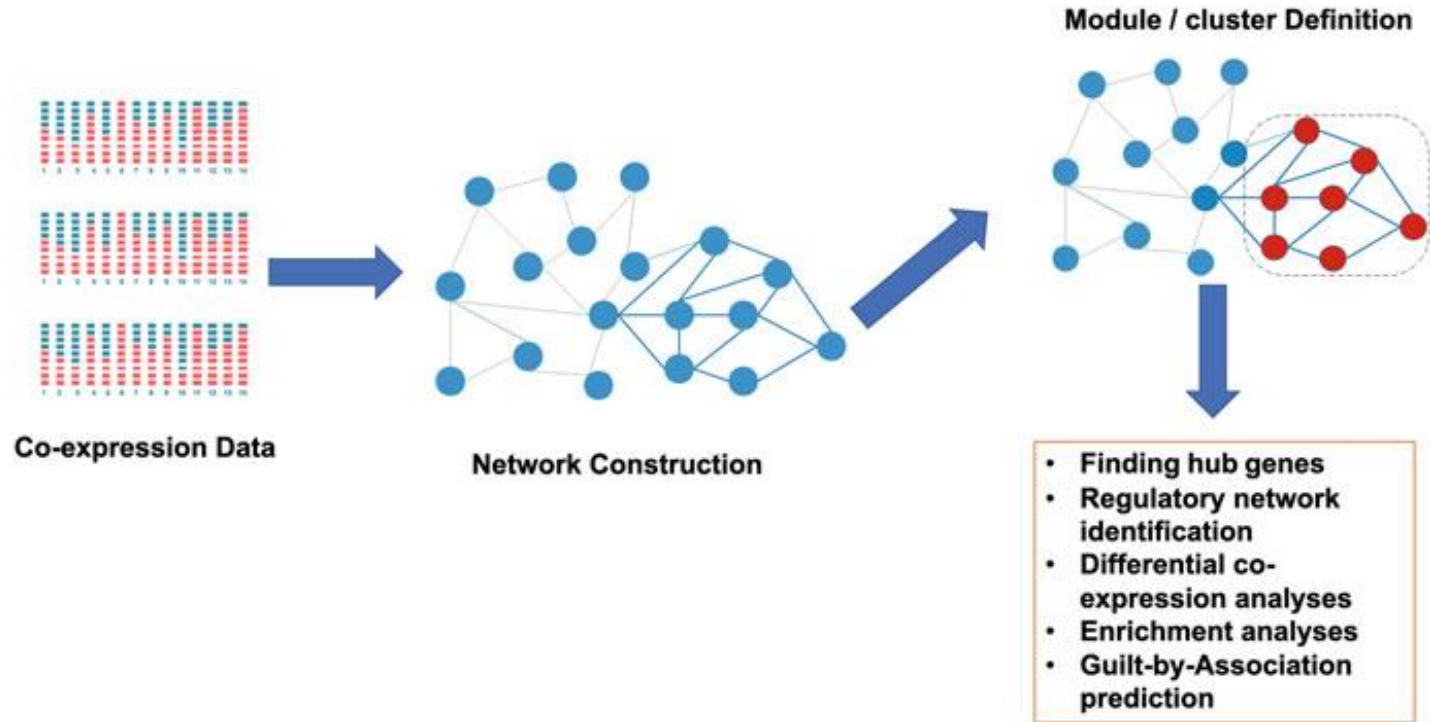
The PPI networks allow to represent the relationships that occur between proteins within cells, that is, the interactome of the organisms that are under study. The use of protein-protein interaction detection techniques, such as high-throughput affinity purification combined with mass spectrometry and the yeast two-hybrid assay and PPI prediction algorithms, have made it possible to construct and study more complex and complete interactomes. Despite the advances, it is essential to note that current knowledge about the interactome is incomplete and noisy. The techniques used have limitations in terms of the number of genuinely physiological interactions and present some false positives and false negatives.

Mapping Algorithms and Tools

Biological Co-Expression Network

These networks allow exploring the existing transcript-transcript associations and interactions, where genes are generally interconnected since there is an association of co-expression between them. This network is built from gene expression data and is represented as an undirected network.

Co-expression networks (CEN) can be used for various purposes, such as identifying genes with the most significant influences within networks, prioritizing disease candidate genes, functional annotation of genes, and identifying regulatory genes within networks.



Workflow for generating and analyzing a co-expression network. The figure shows the different kinds of analyzes that can be performed with this type of network.



Co-expression networks are built by analyzing gene expression profiles' similarity, using techniques such as the correlation coefficient, distance metrics, and developed algorithms based on statistical metrics. The interconnections between the genes are determined using a cut-off that allows structuring the interconnections within the networks. The use of these types of control through cut-off allows the network to represent complex processes and patterns within the organisms.

The CENs have properties such as transitivity, allow the identification of dense communities of genes within the networks, which indicate that the member genes of these communities are functionally related; inside the cluster, there are nodes (genes) that present a high degree of interconnection, these are called central genes (hubs), this type of genes are generally more important to define the functionality of the network, in addition to being able to explain the functioning of each module better. Another property of CENs is the free-scale property. This kind of network has a degree of distribution that follows the power-law distribution, in which most nodes have low degrees, in contrast to the existence of a few nodes with high degrees, which indicates the probability that one node can connect to another is directly proportional to its degree. Due to this property, it is possible to identify a small set of central genes (hubs) and a broad set of genes with few interactions. This characteristic allows networks to be more robust.



Co-expression networks have been used to understand the relationships between genes' expression and the study of different phenomena and interactions between genes. CENs were used to identify genes related to the synthesis and metabolization of fenbendazole and flunixin meglumine in pig livers; this research found eight gene modules that showed a high relation to the level of transcripts relating the metabolism of these medications.

Exciting research that used co-expression networks were carried out by Shaik and Ramakrishna. This study presents the common genes for responses to water and bacterial stress present in rice and Arabidopsis. The team was able to identify several common gene modules that showed high co-expression and specific hubs related to these stresses.

CENs were used to study the genes involved in developing the skeletons and muscle mass of mice for myostatin. In the study developed by Yang, the researchers built co-expression networks using microarray data, which allowed them to study biological processes and metabolic pathways related to the development of muscles and skeletons in wild mice. This study allowed confirmed and identified new transcriptional regulators.



CEN can be used for the detection of biomarkers. The research developed by the team of Zhao and Li (2019), studied gestational diabetes mellitus and managed to identify ten potential biomarkers that help in diagnosing and therapy of this disease through co-expression networks.

In the research of Yuan et al. (2018), biomarkers for the diagnosis of adrenocortical carcinoma were analyzed. Within this study, they analyzed 12 central genes (hubs) within the networks that showed a correlation associated with the prognosis and progress of the disease; another team used co-expression networks to detect candidate regulatory genes that present differential expression and that contribute to the spread of *Salmonella enterica* in pigs.



Review Questions

- Give some examples of biological interactions that can be represented by networks.
- Describe the mapping algorithms and tools.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 1. Introduction into Bioinformatics

Lesson 1. History of Bioinformatics



Contents

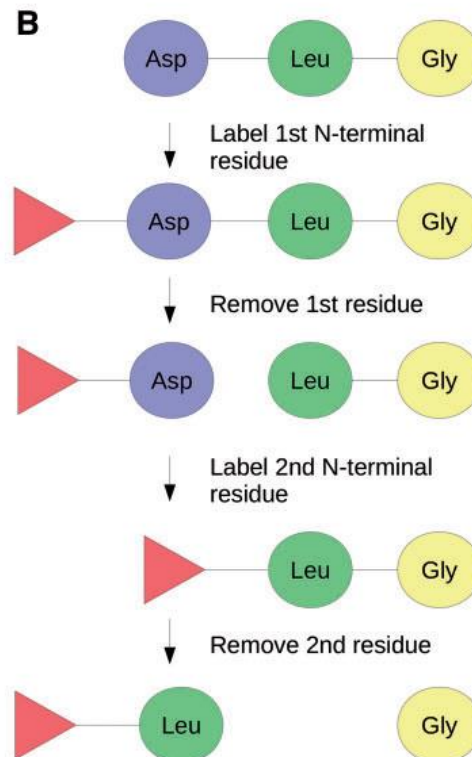
- Introduction
- The origins
- The computer-assisted genealogy of life
- Paradigm shift from protein to DNA analysis
- Parallel advances in biology and computer science
- Genomics, structural bioinformatics and the information superhighway



Introduction

Computers and specialized software have become an essential part of the biologist's toolkit. Either for routine DNA or protein sequence analysis or to parse meaningful information in massive gigabyte-sized biological data sets, virtually all modern research projects in biology require, to some extent, the use of computers. This is especially true since the advent of next generation sequencing (NGS) that fundamentally changed the ways of population genetics, quantitative genetics, molecular systematics, microbial ecology and many more research fields.

1950–1970: The origins



Automated Edman peptide sequencing. (A) One of the first automated peptide sequencers, designed by William J. Dreyer. (B) Edman sequencing: the first N-terminal amino acid of a peptide chain is labeled with phenylisothiocyanate (PITC, red triangle), and then cleaved by lowering the pH. By repeating this process, one can determine a peptide sequence, one N-terminal amino acid at a time.

Protein analysis was the starting point

In the late 1950s, in addition to major advances in determination of protein structures through crystallography [9], the first sequence (i.e. amino acid chain arrangement) of a protein, insulin, was published. This major leap settled the debate about the polypeptide chain arrangement of proteins. Furthermore, it encouraged the development of more efficient methods for obtaining protein sequences. The Edman degradation method emerged as a simple method that allowed protein sequencing, one amino acid at a time starting from the N-terminus. Coupled with automation, more than 15 different protein families were sequenced over the following 10 years.

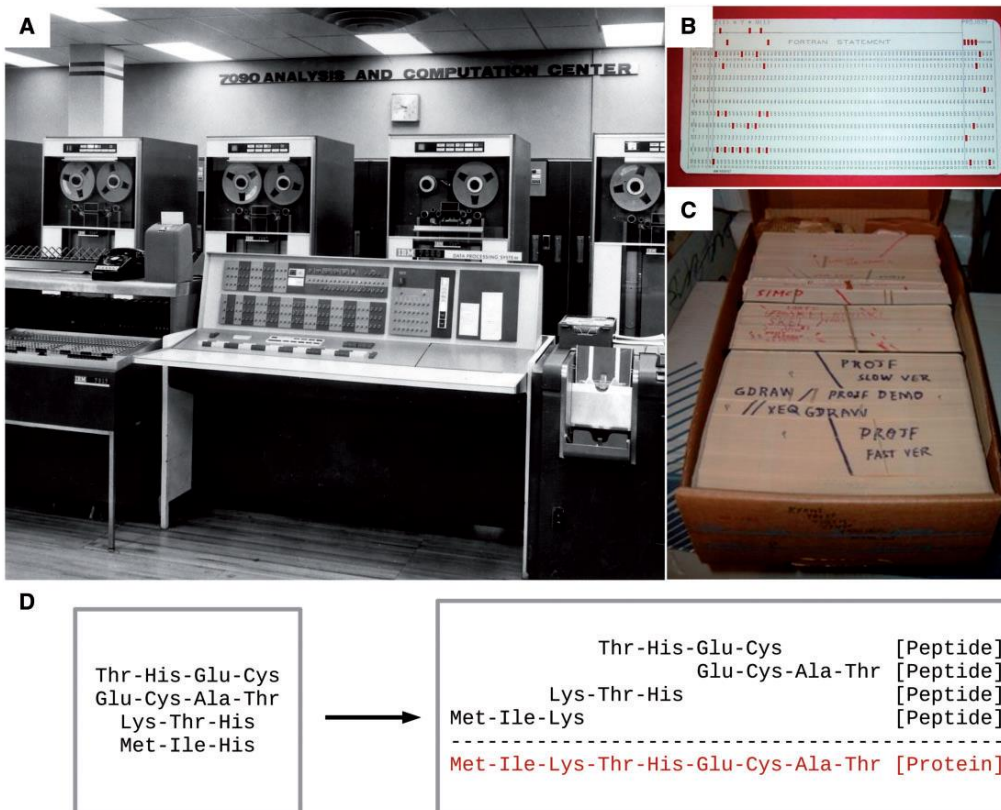
A major issue with Edman sequencing was obtaining large protein sequences and not sequencing a protein in itself but rather assembling the whole protein sequence from hundreds of small Edman peptide sequences.

Dayhoff: the first bioinformatician

Margaret Dayhoff (1925–1983) was an American physical chemist who pioneered the application of computational methods to the field of biochemistry. Dayhoff's contribution to this field is so important that David J. Lipman, former director of the National Center for Biotechnology Information (NCBI), called her 'the mother and father of bioinformatics'.

she began to work with Robert S. Ledley, a physicist who also sought to bring computational resources to biomedical problems. From 1958 to 1962, both combined their expertise and developed COMPROTEIN, 'a complete computer program for the IBM 7090' designed to determine protein primary structure using Edman peptide sequencing data. This software, entirely coded in FORTRAN on punch-cards, is the first occurrence of what we would call today a de novo sequence assembler.

Dayhoff: the first bioinformatician

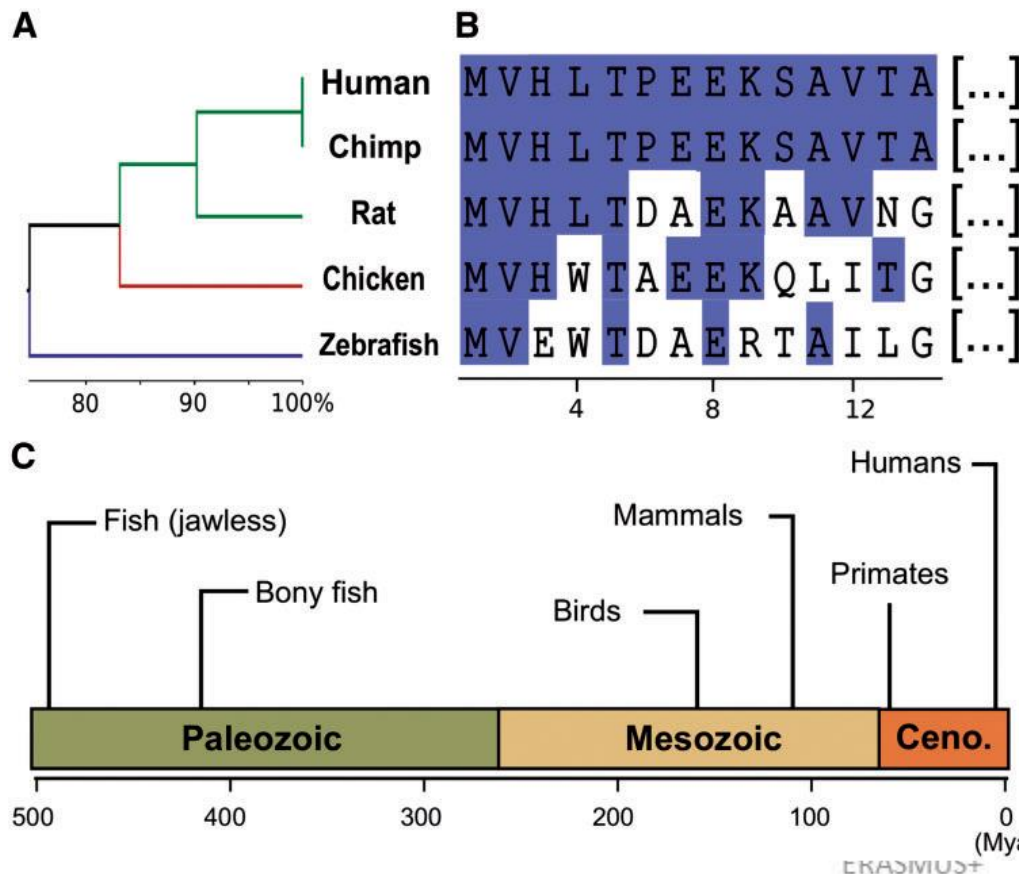


COMPROTEIN, the first bioinformatics software. (A) An IBM 7090 mainframe, for which COMPROTEIN was made to run. (B) A punch card containing one line of FORTRAN code (the language COMPROTEIN was written with). (C) An entire program's source code in punch cards. (D) A simplified overview of COMPROTEIN's input (i.e. Edman peptide sequences) and output (a consensus protein sequence).

The computer-assisted genealogy of life

Emile Zuckerkandl and Linus Pauling departed from this paradigm by investigating biomolecular sequences as ‘carriers of information’. Just as words are strings of letters whose specific arrangement convey meaning, the molecular function (i.e. meaning) of a protein results from how its amino acids are arranged to form a ‘word’. Both observed that orthologous proteins from vertebrate organisms, such as hemoglobin, showed a degree of similarity too high over long evolutionary time to be the result of either chance or convergent evolution (Ibid). The concept of orthology itself was defined in 1970 by Walter M. Fitch to describe homology that resulted from a speciation event

The computer-assisted genealogy of life



Sequence dissimilarity between orthologous proteins from model organisms correlates with their evolutionary history as evidenced by the fossil record. (A) Average distance tree of hemoglobin subunit beta-1 (HBB-1) from human (*Homo sapiens*), chimpanzee (*Pan troglodytes*), rat (*Rattus norvegicus*), chicken (*Gallus gallus*) and zebrafish (*Danio rerio*). (B) Alignment view of the first 14 amino acid residues of HBB-1 compared in (A) (residues highlighted in blue are identical to the human HBB-1 sequence). (C) Timeline of earliest fossils found for different aquatic and terrestrial animals

A mathematical framework for amino acid substitutions

In 1978, Dayhoff, Schwartz and Orcutt contributed to another bioinformatics milestone by developing the first probabilistic model of amino acid substitutions. This model, completed 8 years after its inception, was based on the observation of 1572 point accepted mutations (PAMs) in the phylogenetic trees of 71 families of proteins sharing above 85% identity. The result was a 20 x 20 asymmetric substitution matrix that contained probability values based on the observed mutations of each amino acid (i.e. the probability that each amino acid will change in a given small evolutionary interval). Whereas the principle of (i.e. least number of changes) was used before to quantify evolutionary distance in phylogenetic reconstructions, the PAM matrix introduced the of substitutions as the measurement of evolutionary change.

A mathematical framework for amino acid substitutions

| $10^4 P^a$ | | Ala A | Arg R | Asn N | Asp D | Cys C | Gln Q | ... | Val V |
|------------|-----|----------|----------|----------|----------|----------|----------|-----|----------|
| Ala | A | 9867 | 2 | 9 | 10 | 3 | 8 | ... | 18 |
| Arg | R | 1 | 9913 | 1 | 0 | 1 | 10 | ... | 1 |
| Asn | N | 4 | 1 | 9822 | 36 | 0 | 4 | ... | 1 |
| Asp | D | 6 | 0 | 42 | 9859 | 0 | 6 | ... | 1 |
| Cys | C | 1 | 1 | 0 | 0 | 9973 | 0 | ... | 2 |
| Gln | Q | 3 | 9 | 4 | 5 | 0 | 9876 | ... | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Val | V | 13 | 2 | 1 | 1 | 3 | 2 | ... | 9901 |

^aEach numeric value represents the probability that an amino acid from the *i*-th column be substituted by an amino acid in the *j*-th row (multiplied by 10 000).

1970–1980: Paradigm shift from protein to DNA analysis

Deciphering of the DNA language: the genetic code

The specifications for any living being (more precisely, its ‘proteins’) are encoded in the specific nucleotide arrangements of the DNA molecule. This view was formalized in Francis Crick’s sequence hypothesis (also called nowadays the ‘Central Dogma’), in which he postulated that RNA sequences, transcribed from DNA, determine the amino acid sequence of the proteins they encode. In turn, the amino acid sequence determines the three-dimensional structure of the protein.

Therefore, if one could figure out how the cell translates the ‘DNA language’ into polypeptide sequences, one could predict the primary structure of any protein produced by an organism by ‘reading its DNA’. By 1968, all of the 64 codons of the genetic code were deciphered [35]; DNA was now ‘readable’, and this groundbreaking achievement called for simple and affordable ways to obtain DNA sequences.

Cost-efficient reading of DNA

Being able to obtain DNA sequences from an organism holds many advantages in terms of information throughput. Whereas proteins must be individually purified to be sequenced, the whole genome of an organism can be theoretically derived from a single genomic DNA extract. From this whole-genome DNA sequence, one can predict the primary structure of all proteins expressed by an organism through translation of genes present in the sequence. Though the principle may seem simple, extracting information manually from DNA sequences involves the following:

1. comparisons (e.g. finding homology between sequences from different organisms);
2. calculations (e.g. building a phylogenetic tree of multiple protein orthologs using the PAM1 matrix);
3. and pattern matching (e.g. finding open reading frames in a DNA sequence).

Using DNA sequences in phylogenetic inference

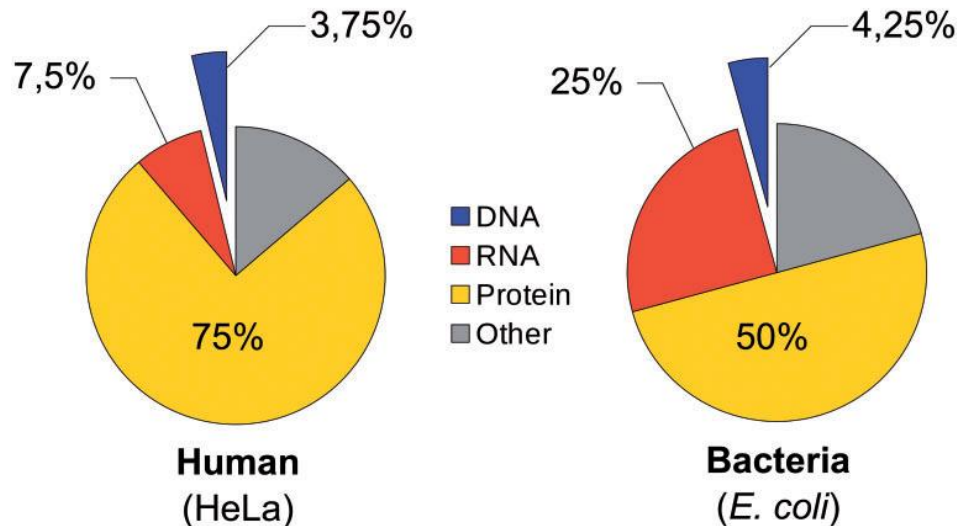
Felsenstein was the first to develop a maximum likelihood (ML) method to infer phylogenetic trees from DNA sequences. Unlike parsimony methods, which reconstruct an evolutionary tree using the least number of changes, ML estimation 'involves finding that evolutionary tree which yields the highest probability of evolving the observed data'. Following Felsenstein's work in molecular phylogeny, several bioinformatics tools using ML were developed and are still widely developed, as are new statistical methods to evaluate the robustness of the nodes. This even inspired in the 1990s, the use of Bayesian statistics in molecular phylogeny, which are still commonly used in biology.

1980–1990: Parallel advances in biology and computer science

Molecular methods to target and amplify specific genes

Genes, unlike proteins and RNAs, cannot be biochemically fractionated and then individually sequenced, because they all lay contiguously on a handful of DNA molecules per cell. Moreover, genes are usually present in one or few copies per cell. Genes are therefore orders of magnitude less abundant than the products they encode.

1980–1990: Parallel advances in biology and computer science



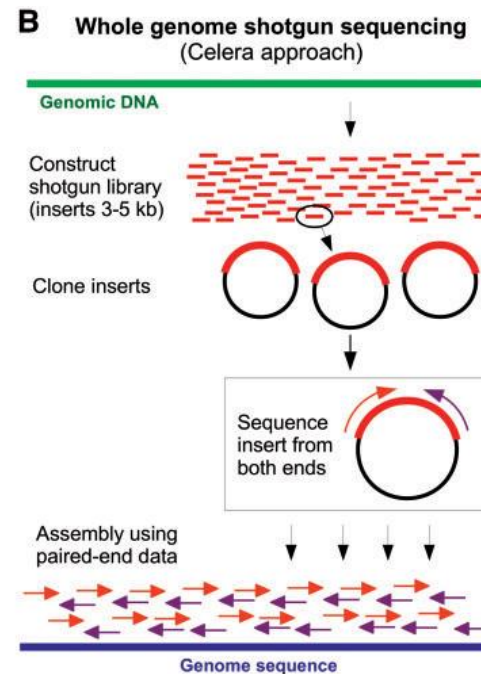
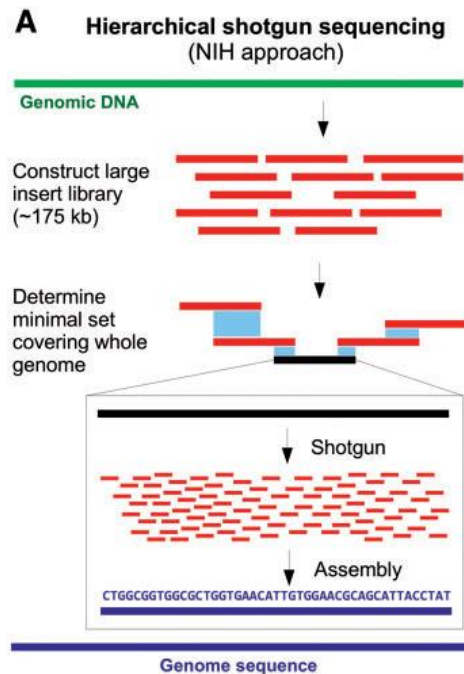
DNA is the least abundant macromolecular cell component that can be sequenced. Percentages (%) represent the abundance of each component relative to total cell dry weight. Data from the Ambion Technical Resource Library.

1990–2000: Genomics, structural bioinformatics and the information superhighway

| | Fortran ^a | C | R | Java |
|--|--|------------------------------|--|---|
| First appeared | 1957 | 1972 | 1993 | 1995 |
| Typical use | Algorithmics, calculations, programming modules for other applications | Optimized command-line tools | Statistical analysis, data visualization | Graphical user interfaces, data visualization, network analysis |
| Notable fields of application | Biochemistry, Structural Bioinformatics | Various | Metagenomics, Transcriptomics, Systems Biology | Genomics, Proteomics, Systems Biology |
| Specialized bioinformatics repository? | None | None | Bioconductor, [73], since 2002 | BioJava [74], since 2002 |
| Example software or packages | Clustal [32, 33], WHAT IF [75] | MUSCLE [76], PhyloBayes [77] | edgeR [78], phyloseq [79] | Jalview [80], Jemboss [81], Cytoscape [82] |

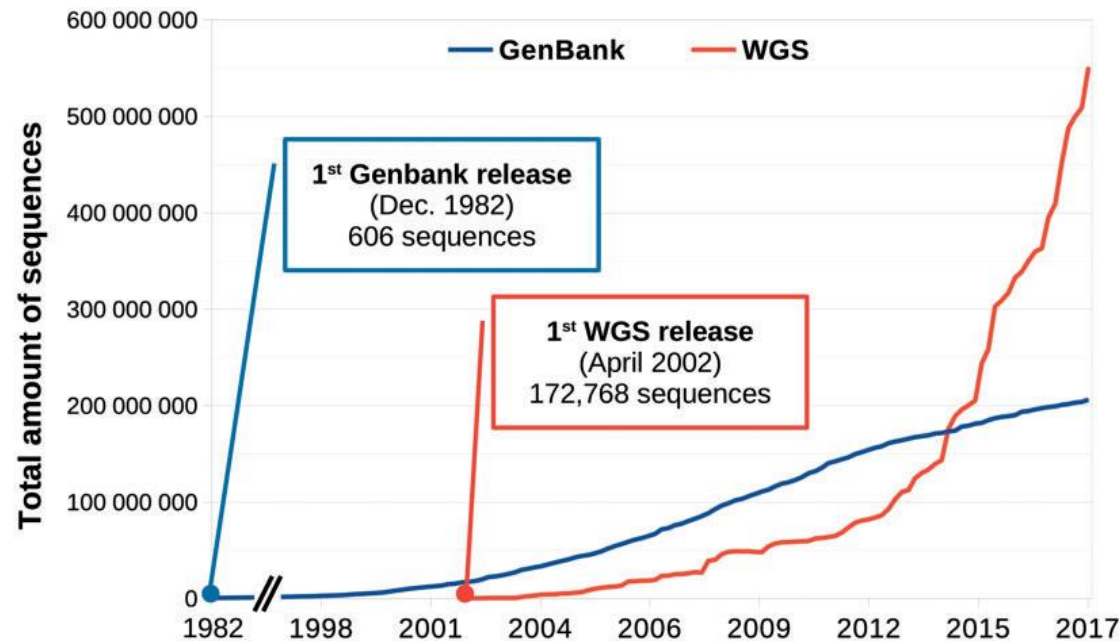
^aEven though the earliest bioinformatics software were written in Fortran, it is seldom used to code standalone programs nowadays. It is rather used to code modules for other programs and programming languages (such as C and R mentioned here).

Notable nonscripting and/or statistical programming languages used in bioinformatics



Hierarchical shotgun sequencing versus whole genome shotgun sequencing. Both approaches respectively exemplified the methodological rivalry between the public (NIH, A) and private (Celera, B) efforts to sequence the human genome. Whereas the NIH team believed that whole-genome shotgun sequencing (WGS) was technically unfeasible for gigabase-sized genomes, Venter's Celera team believed that not only this approach was feasible, but that it could also overcome the logistical burden of hierarchical shotgun sequencing, provided that efficient assembly algorithms and sufficient computational power are available. Because of the partial use of NIH data in Celera assemblies, the true feasibility of WGS sequencing for the human genome has been heavily debated by both sides

2000–2010: High-throughput bioinformatics

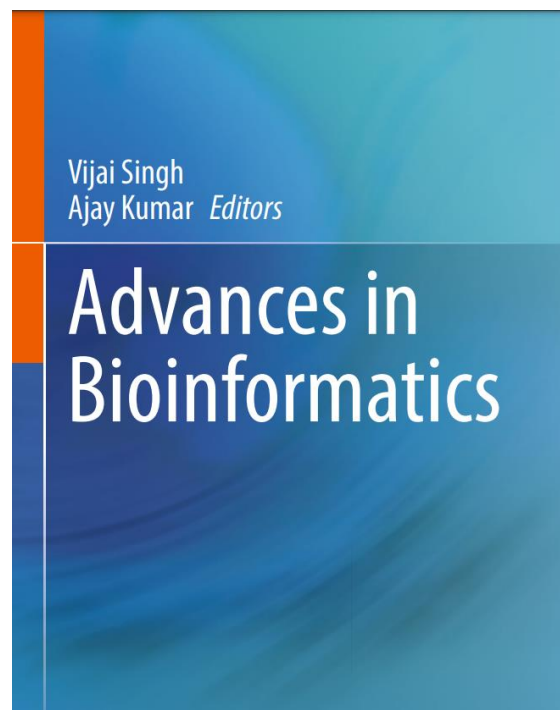


Total amount of sequences on the NCBI GenBank and WGS (Whole Genome Shotgun) databases over time. The number of draft/incomplete genomes has surpassed the amount of complete genome sequences in June 2014, and still continues to grow exponentially. Source: <https://www.ncbi.nlm.nih.gov/genbank/statistics/>



Review Questions

- Point out the main periods of development of bioinformatics.
- What is characteristic for each of them?



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 1. Introduction into Bioinformatics

Lesson 2. Objectives of Bioinformatics



Contents

- Introduction
- Elements of biology
- What is bioinformatics?
- Extension of Bioinformatics Concept
- About some problems in bioinformatics

Introduction

“The two technologies that will shape the next century are biotechnology and information technology”

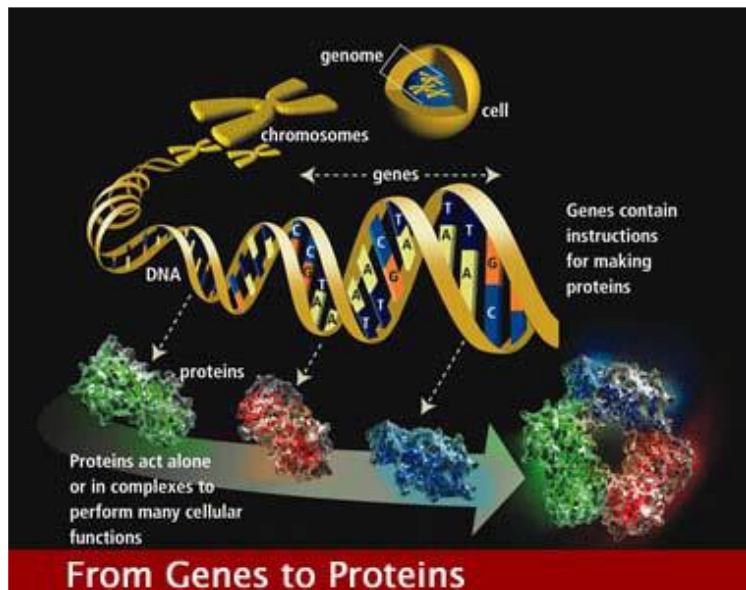
Bill Gates

“The two technologies that will have the greatest impact on each other in the new millennium are biotechnology and information technology”

Martina McGloughlin

Basic molecular biology

Most of 100 billion cells in the human body contains a copy of the entire human genome (all the genetic information necessary to build a human being).



The cell nucleus contains six feet of DNA packed into 23 pairs of chromosomes. We each inherit one set of 23 chromosomes from our mother and another set from our father. DNA contains the code for the body (genes) governing all aspects of cell growth and inheritance.

Protein, made up amino acids, are essential components of all organs and chemical activities.

Molecules of life

1. Small molecules

2. Proteins

3. DNA

4. RNA

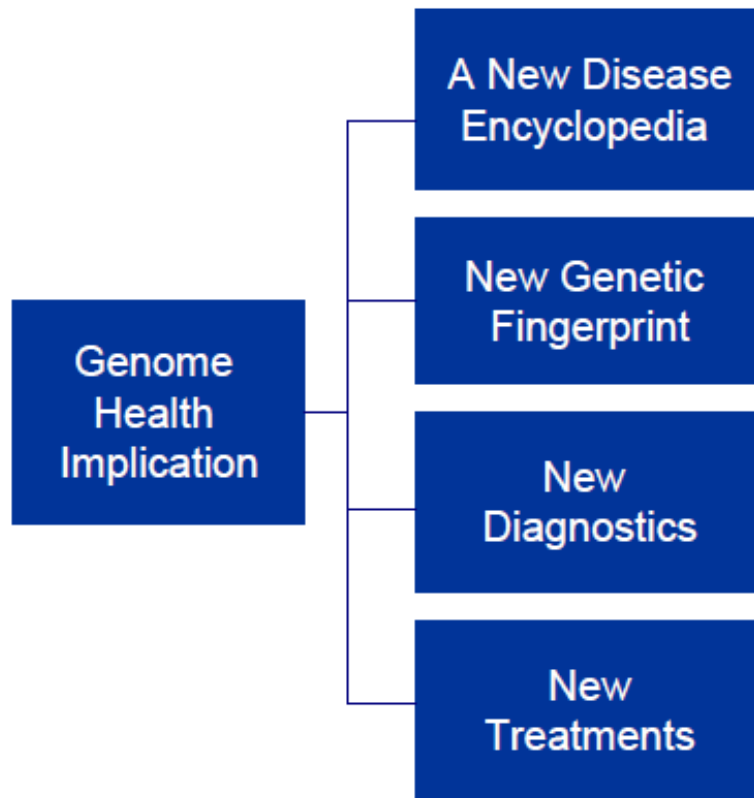
Biological macromolecules

Gene prediction problem

Gene prediction: It is an interesting question: given the genomic DNA sequence, can we tell where the genes are?

| Organism | The number of predicted genes | Part of the genome that encodes proteins (exons) |
|-------------------|-------------------------------|--|
| E.Coli (bacteria) | 5000 | 90% |
| Yeast | 6000 | 70% |
| <u>Worm</u> | 18,000 | 27% |
| <u>Fly</u> | 14,000 | 20% |
| <u>Weed</u> | 25,500 | 20% |
| <u>Human</u> | 30,000 | < 5% |

Human Genome Project



Goal

- identify all the approximately 30,000 genes in human DNA,
- determine the sequences of the 3 billion chemical base pairs that make up human DNA,
- store this information in databases,
- improve tools for data analysis,
- transfer related technologies to the private sector, and
- address the ethical, legal, and social issues (ELSI) that may arise from the project.

What is bioinformatics?

Bio: Molecular Biology

Informatics: Computer Science

Bioinformatics: Solving problems arising from biology using methodology from computer science.

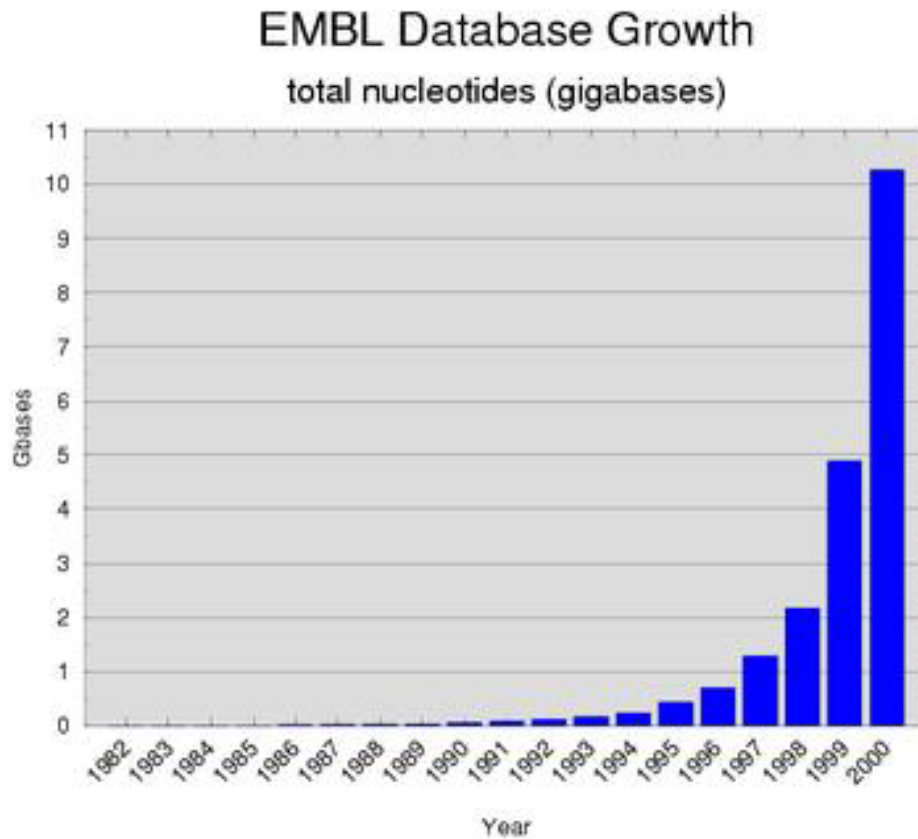
Synonyms: Computational biology,
Computational molecular biology,
Biocomputing

Paradigm Shift in Biology

The new paradigm, now emerging, is that all the 'genes' will be known (in the sense of being resident in databases available electronically), and that the starting point of a biological investigation will be theoretical. An individual scientist will begin with a **theoretical conjecture**, only **then turning to experiment** to follow or test that hypothesis.

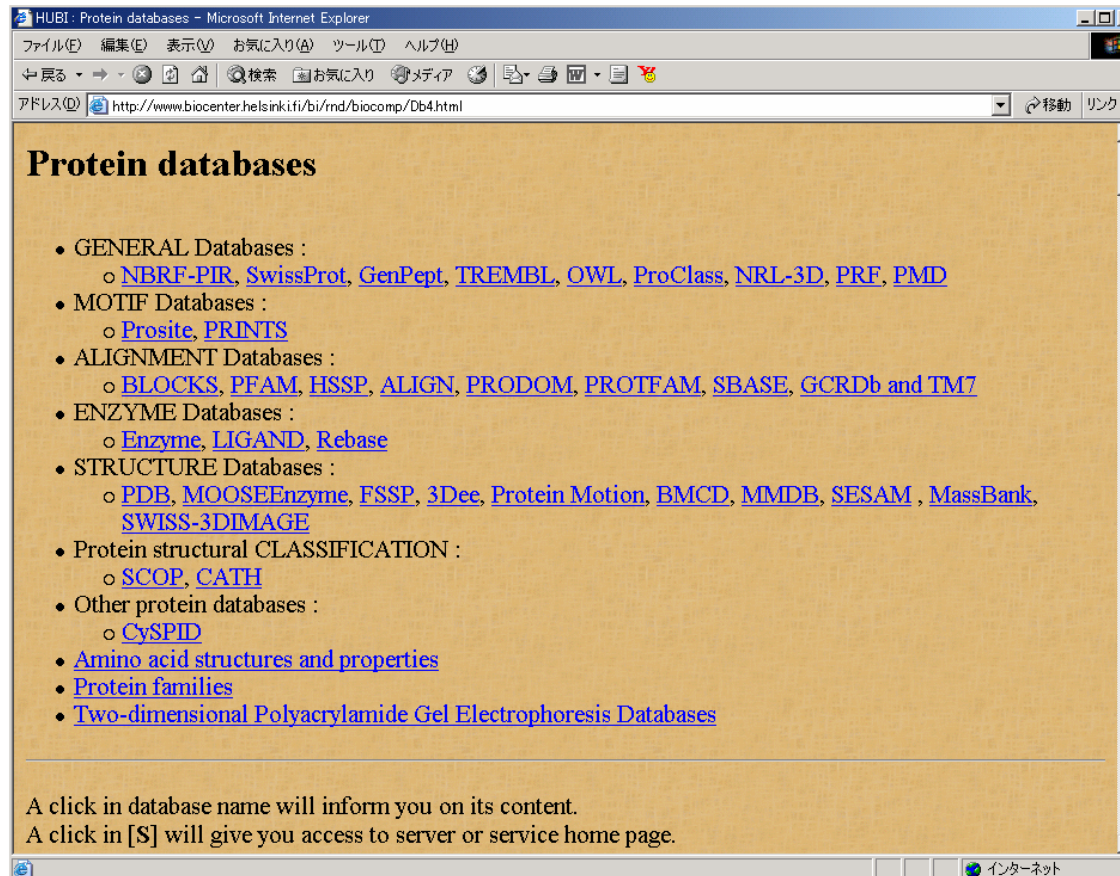
To use [the] flood of knowledge, which will pour across the computer networks of the world, biologists not only must become **computer literate**, but also **change their approach** to the problem of understanding life.

Base Pairs in GenBank



10,267,507,282
bases in
9,092,760
records.

Public databases



Extension of Bioinformatics Concept

■ Genomics

- Functional genomics
- Structural genomics

The identification and functional characterization of genes.

■ Proteomics: large scale analysis of the proteins of an organism

The study of gene expression at the protein level, by the identification and characterization of proteins present in a biological sample.

■ Pharmacogenomics: developing new drugs that will target a particular disease

The use of genetic information to predict the safety, toxicity and/or efficacy of drugs in individual patients or groups of patients.

■ Microarray (genome chip): DNA chip, protein chip

a new technology aims to monitor the whole genome on a single chip so that researchers can have a better picture of the interactions among thousands of genes simultaneously

37

Problems in Bioinformatics

Structure analysis

- Protein structure comparison
- Protein structure prediction
- RNA structure modeling

Pathway analysis

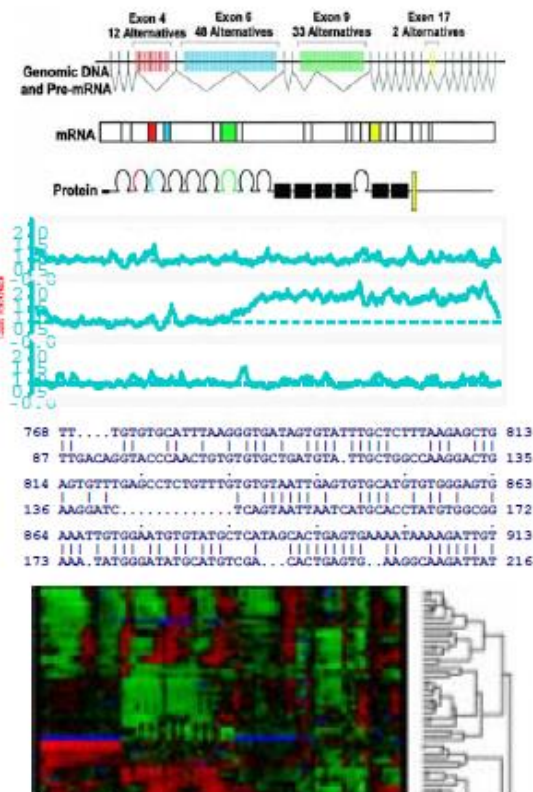
- Metabolic pathway
- Regulatory networks

Sequence analysis

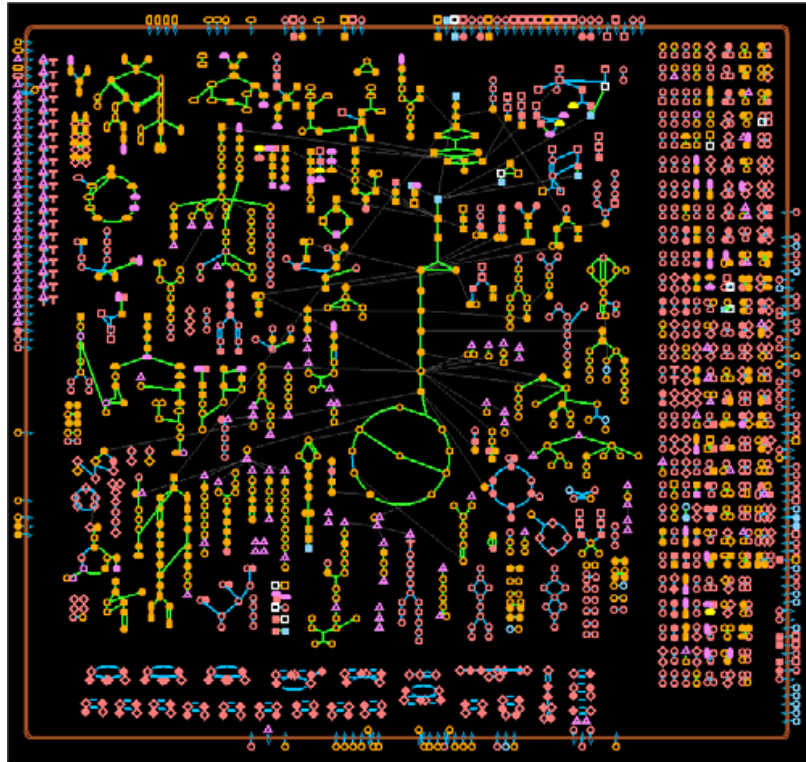
- Sequence alignment
- Structure and function prediction
- Gene finding

Expression analysis

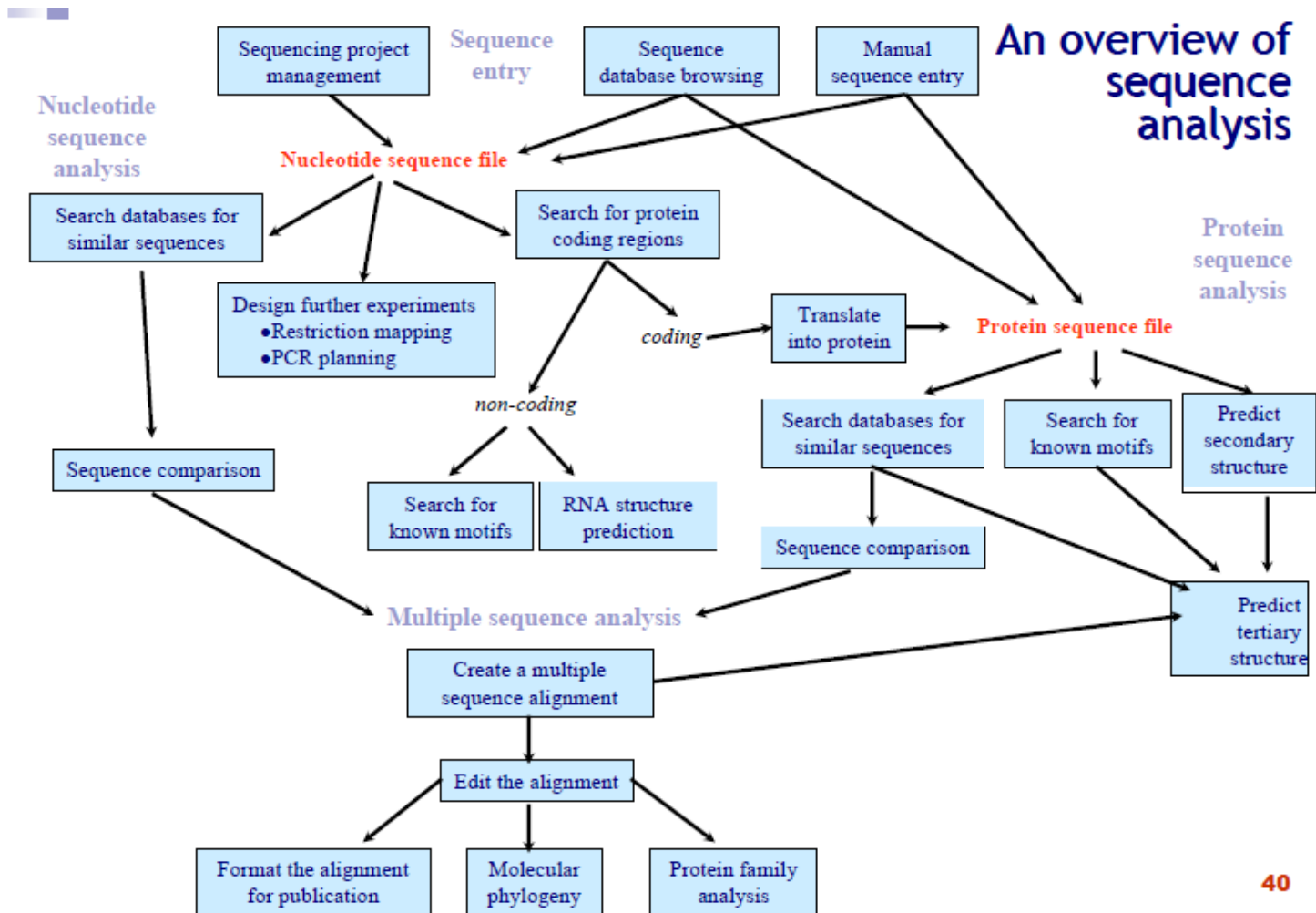
- Gene expression analysis
- Gene clustering



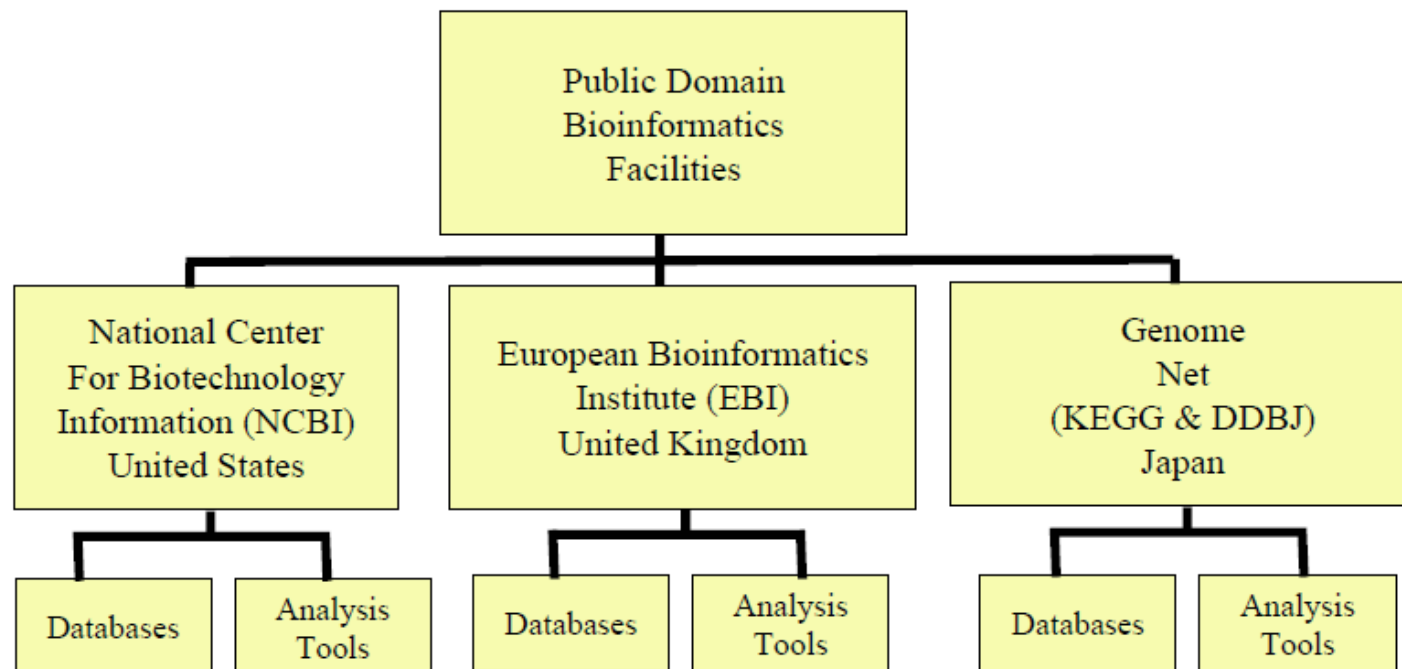
Pathway analysis



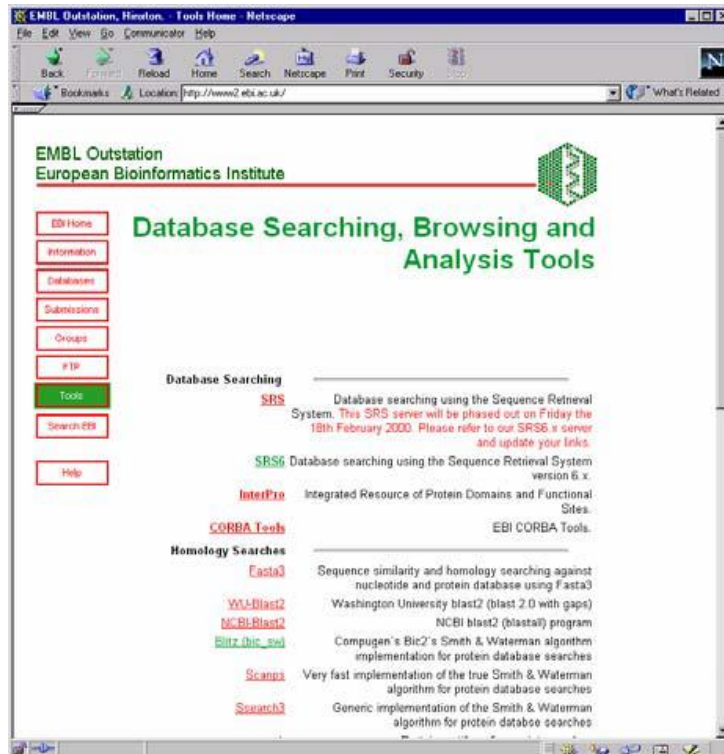
- A chemical reaction interconverts chemical compounds
- An enzyme is a protein that accelerates chemical reactions
- A pathway is a linked set of reactions



Primary public domain bioinformatics servers



Analysis Tools



The EBI maintains versions of major public domain sequence database searching and analysis tools, e.g. FASTA, CLUSTALW, BLAST, and Smith & Waterman implementations.

Challenges in Bioinformatics

Bioinformatics requires:

- Access to multiple distributed resources
- Needs information to be up-to-date
- Minimal data redundancy
- Robust applications
- Extendable applications
 - Monolithic App. vs. Components
- Portable software

Challenges in Bioinformatics

Explosion of information

- Need for faster, automated analysis to process large amounts of data
- Need for integration between different types of information (sequences, literature, annotations, protein levels, RNA levels etc...)
- Need for “smarter” software to identify interesting relationships in very large data sets

Lack of “bioinformaticians”

- Software needs to be easier to access, use and understand
- Biologists need to learn about the software, its limitations, and how to interpret its results



Review Questions

- What are the main objectives of bioinformatics?
- Give some examples of the challenges in bioinformatics.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 1. Introduction into Bioinformatics

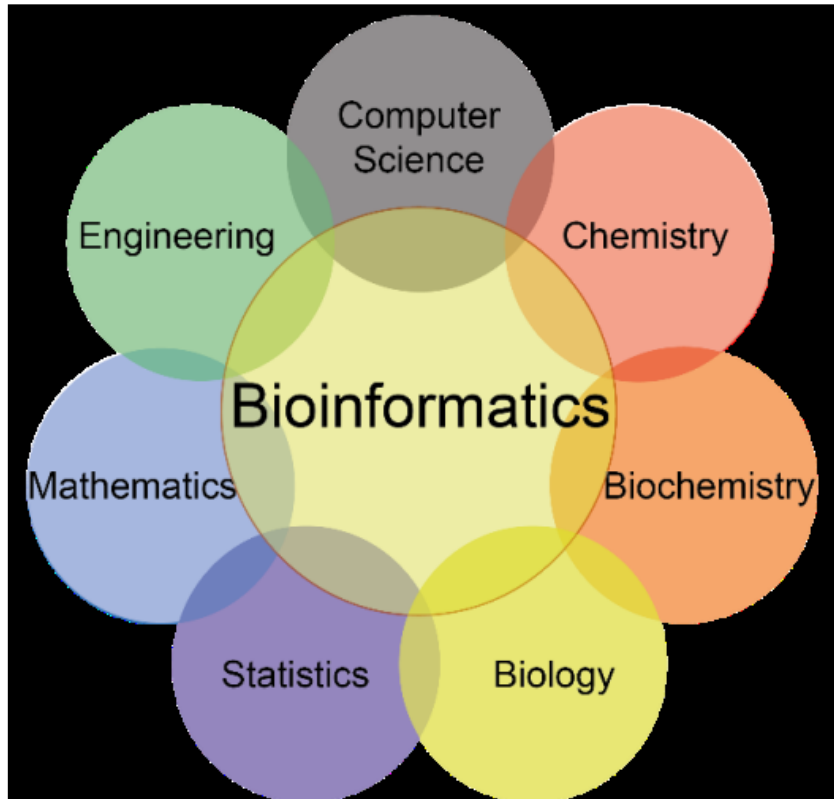
Lesson 3. Components of Bioinformatics



Contents

- Introduction
- Data
- Databases
- Various categories of Analyses

Introduction



Bioinformatics is a branch of science that integrates computer science, mathematics and statistics, chemistry and engineering for analysis, exploration, integration and exploitation of biological sciences data, in Research and Development

Bioinformatics deals with storage, retrieval, analysis and interpretation of biological data using computer based software and tools

Data

Database

Database Mining Tools

Data

➤ Nucleic Acid Sequences

- Raw DNA Sequences
- Genomic sequence tags (GSTs)
- cDNA sequences
- Expressed sequence tags (ESTs)
- Organellar DNA sequences
- RNA Sequences

➤ Protein sequences

➤ Protein structures

➤ Metabolic pathways

➤ Gel pictures

➤ Literature

Databases

A database is a vast collection of data pertaining to a specific topic e g nucleotide sequence, protein sequence etc in an electronic environment

- They are heart of bioinformatics
- Computerized storehouse of data (records).
- Allows extraction of specified records
- Allows adding, changing, removing, and merging of records
- Uses standardized formats

Databases: Types

- **Sequence Databases**
- **Structural Databases**
- **Enzyme Databases**
- **Micro array Databases**
- **Clinical Database**
- **Pathway Databases**
- **Chemical Databases**
- **Integrated Databases**
- **Bibliographic Databases**

Nucleotide Sequence Databases

NCBI GenBank: www.ncbi.nlm.nih.gov/GenBank)

EMBL: www.ebi.ac.uk/embl

DDBJ: www.ddbj.nig.ac.jp)

The 3 databases are updated and exchanged on a daily basis and the accession numbers are consistent.

There are no legal restriction in the usage of these databases. However, there are some patented sequences in the database.

The International Nucleotide Sequence Database Collaboration (INSDB)

Protein Databases

Protein sequence database

- Functions as repository of raw data: two types
- Primary
- Secondary

Protein structure database

Primary databases

SWISS PROT Groups at Swiss Institute of Bioinformatics (

It annotate the sequences

Describe protein functions

Its domain structures

Its post translations modifications

Provides high level of annotation

Minimum level of redundancy

High level of integration with other databases

TrEMBL

Computer annotated supplements of SWISS PROT that contains all the translations of EMBL nucleotide entries not yet integrated in SWISS PROT.

PIR Protein Information Resource, a division of NBRF in US

Collaborated with Munich Information Centre for Protein Sequences (and Japanese International Protein Sequence Database (

One an search for entries

Do sequence similarity

PIR also produces MRL 3 D db of sequences extracted from 3 D structures in PDB)

Secondary databases

Secondary db compile and filter sequence data from different primary db These db contain information derived from protein sequences and help the user determine whether a new sequence belong to a known protein family

PROSITE

db of short protein sequence patterns and profiles that characterise biologically significant sites in proteins

It is based on regular expressions describing characteristic sequences of specific protein families and domains

It is part of SWISS PROT, and maintained in the same way

PRINTS

PRINTS provides a compendium of protein fingerprints (groups of conserved motifs that characterise a protein family)

Now has a relational version, version, "PRINTS S"

BLOCKS

BLOCK patterns without gaps in aligned protein families defined by PROSITE, found by pattern searching and statistical sampling algorithms Automatically determined un gapped conserved segments

Pfam

Db of protein families defined as domains

For each domain, it contains a multiple alignment of a set of defining sequences and the other sequences in SWISS PROT and TrEMBL that can be matched to the alignment

Protein Structural Database

PDB (Protein Data Bank)

Main db of 3 D structures of biological macromolecules (determined by X ray crystallography and NMR)

PDB entries contain the atomic coordinates, and some structural parameters connected with the atoms or computed from the structures (secondary structure)

PDB provide primary archive of all 3 D structures for macromolecules such as proteins, DNA, RNA and various complexes

SCOP (Structural Classification of Proteins):

Db was started to with objective to classify protein 3 D structures in a hierarchical scheme of structural classes

It is based on data in a primary db, but adds information through analysis and organization (such as classification of 3 D structures into hierarchical scheme of folds, super families and families)

CATH (architecture, topology, homologous super family)

CATH perform hierarchical classification of protein domain structures Clusters proteins at four major structural levels

Database Mining Tools (Analysis Tools)

| Analysis Tool | Function |
|------------------------------|---|
| BLAST (NCBI, USA) | Used to analyse sequence information and detect homologous sequences |
| ENTREZ (NCBI, USA) | Used to access literature (abstracts), sequence and structure db |
| DNAPLOT (EBI, UK) | Sequence alignment tool |
| LOCUS LINK (NCBI, USA) | Assessing information on homologous genes |
| LIGAND (GenomNet, Japan) | A chemical db, allows search for a combination of enzymes and links to all publically accessible db. |
| BRITE (GenomNet, Japan) | Biomolecular relations information transmission and expression db; links to all publically accessible db. |
| TAXONOMY BROWSER (NCBI, USA) | Taxonomic classification of various species as well as genetic information |
| STRUCTURE | It support Molecular Modelling Database (MMDB) and software tools for structure analysis |

Karim Yedav, Department of Biochemistry

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Various categories of Analyses

Analysis of a single gene (protein) sequence

Similarity with other known genes

Phylogenetic trees; evolutionary relationships

Identification of well defined domains in the sequence

Sequence features (physical properties, binding sites, modification sites)

Prediction of sub cellular localization

Prediction of protein secondary and tertiary structures

Various categories of Analyses

Analysis of whole genomes

Location of various genes on the chromosomes,
correlation with function or evolution

Expansion/duplication of gene families

Which gene families are present, which missing?

Presence or absence of biochemical pathways

Identification of "missing" enzymes

Large scale events in the evolution of organisms

Various categories of Analyses

Analysis of genes and genomes with respect to function (Functional Annotation)

Transcriptomics Expression analysis micro array data (mRNA/transcript analyses)

Proteomics protein qualitative and quantitative analyses, covalent modifications

Comparison and analysis of biochemical pathways

Deletion or mutant genotypes vs phenotypes

Identification of essential genes, or genes involved in specific processes



Various categories of Analyses

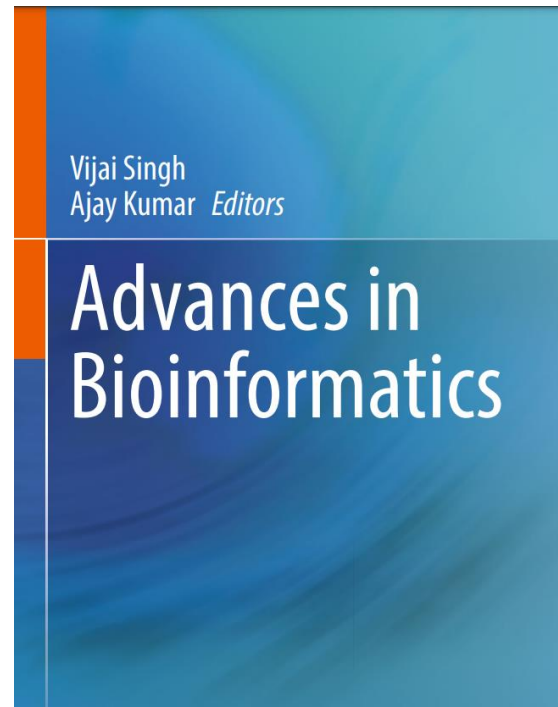
Comparative genomics

Identifying pathogen specific unique targets for
designing novel drugs



Review Questions

- What are the components of bioinformatics?
- Give some examples of data, database, database mining tools.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 1. Where to Find Protein and Ligand Structures



Contents

- Introduction
- PDB database
- Homology modelling
- The structures of the ligands
- Small molecular structures databases

Introduction

Molecular recognitions including enzyme-substrate, drug-protein, drug-nucleic acid, protein-nucleic acid, and protein-protein interactions play important roles in many biological processes such as signal transduction, cell regulation, and other macromolecular assemblies. Therefore, determination of the binding mode and affinity between the constituent molecules in molecular recognition is crucial to understanding the interaction mechanisms and to designing therapeutic interventions. Due to the difficulties and economic cost of the experimental methods for determining the structures of complexes, computational methods such as molecular docking are desired for predicting putative binding modes and affinities.



PDB database

rcsb.org

Working Sites | Mails | Journals | Symposia | Citations | Music | Articles for download | RGD | Google Преводач | cPanel X | Биоинформатика | Easyclass | The Next...

RCSB PDB | Deposit | Search | Visualize | Analyze | Download | Learn | About | Documentation | Careers | COVID-19 | MyPDB | Contact us

RCSB PDB PROTEIN DATA BANK

207,540 Structures from the PDB
1,068,577 Computed Structure Models (CSM)

3D Structures | Enter search term(s), Entry ID(s), or sequence | Include CSM | Search

Advanced Search | Browse Annotations | Help

PDB-101 | PDB | EMDatResource | NAKB | wwPDB Foundation | PDB-Dev

Facebook | Twitter | YouTube | Instagram

New: More Computed Structure Models (CSM) available | Learn more

Welcome

Deposit

Search

Visualize

Analyze

Download

Learn

RCSB Protein Data Bank (RCSB PDB) enables breakthroughs in science and education by providing access and tools for exploration, visualization, and analysis of:

- Experimentally-determined 3D structures from the **Protein Data Bank (PDB)** archive
- Computed Structure Models (CSM)** from AlphaFold DB and ModelArchive

These data can be explored in context of external annotations providing a structural view of biology.

Explore NEW Features

PDB-101 Training Resources

July Molecule of the Month

c-Abl Protein Kinase and Imatinib

PDB database

Determine whether the corresponding target has been deposited or not.

- **One way of doing this is to determine whether there is a protein in the PDB with a sequence that is 100% identical to that of the protein target.**
- **If the target protein has not yet been deposited in the PDB but the database contains a protein with a similar sequence, then homology modelling can be used to predict the 3D structure of the former protein.**

Homology modelling

Programs or webserver allow users to build these models by strictly controlling the values of several parameters, some databases of protein structures modelled by homology have been automatically created by these programs or webserver by using default parameter values when generating the homology model.

Examples of such databases are ModBase (http://modbase.compbio.ucsf.edu/modbase-cgi/search_form.cgi) and the SWISS-MODEL



ModBase: Database of Comparative Protein Structure Models



[Sali Lab Home](#) [ModWeb](#) [ModLoop](#) [ModBase](#) [ModEval](#) [PCSS](#) [FoXS](#) [IMP](#) [ModPipe](#) [Twitter](#)

[ModBase Home](#) [ModBase Datasets for User:Anonymous](#) [User Login](#) [Help](#) [News](#) [Contact](#) [Current Datasets](#)

- [General Information](#)
- [Statistics and Genome Datasets](#)
- [News](#)
- [Project Pages](#)
- [Authors and Acknowledgements](#)
- [Publications](#)
- [Downloads](#)
- [Related Resources](#)

Please address inquiries to:
modbase@salilab.org

MODBASE is *no longer under active development*, and contains theoretically calculated models, not experimentally determined structures. The models may contain *significant* errors.

ModBase Search

ModBase is a database of comparative protein structure models, calculated by our modeling pipeline [ModPipe](#).

Search type  Display type 

All available datasets are selected

To include the academic (comprehensive) dataset, go to [Current Datasets!](#)

Search by properties

Property 
Organism  or

[Advanced search](#)

Model Details Example

Database ID: [Q12321](#)

This page makes all models and model details for one sequence available.



Overview Example

[Dengue Virus](#)

Models for several sequences are displayed on the [Sequence Overview](#) page.

| | |
|---|---|
|  | <input type="checkbox"/> 159024817 nonstructural protein ns4a [dengue virus 2] |
|  | <input type="checkbox"/> 159024814 nonstructural protein ns2a [dengue virus 2] |
|  | <input type="checkbox"/> 159024816 nonstructural protein ns3 [dengue virus 2] |
|  | <input type="checkbox"/> ns4b |
|  | <input type="checkbox"/> 159024810 membrane glycoprotein precursor [dengue virus 2] |
|  | <input type="checkbox"/> 159024815 nonstructural protein ns2b [dengue virus 2] |



Tackling COVID-19 – Challenges Aiming at Bettering the Higher
Education Quality (TACOHEQ)
Project Number: 2020-1-BG01-KA226-HE-095131

Co-funded by the
Erasmus+ Programme
of the European Union



BIOZENTRUM
University of Basel
The Center for Molecular Life Sciences

SWISS-MODEL

Modelling Repository Tools Documentation Log in Create Account

SWISS-MODEL

is a fully automated protein structure homology-modelling server, accessible via the [ExPASy web server](#), or from the program DeepView (Swiss Pdb-Viewer).

The purpose of this server is to make protein modelling accessible to all life science researchers worldwide.

Start Modelling

Repository

Every week we model all the sequences for thirteen core species based on the latest UniProtKB proteome. Is your protein already modelled and up to date in **SWISS-MODEL Repository**?



Search SWISS-MODEL Repository



What's new

This website requires cookies, and the limited processing of your personal data in order to function. By using the site you are agreeing to this as outlined in our [privacy policy](#).

OK

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Protein Databases

Repository

(<http://swissmodel.expasy.org/repository/>) whose models can be easily accessed from links within appropriate UniProt Knowledgebase entries. At this point, it is worth pointing out that some years ago, docking into protein active sites built by sequence homology was considered to be a source of unreliable results. Nevertheless, the improvement in homology modelling methods has radically changed this point of view and they have now been successfully used in docking experiments.

The structures of the ligands

There are two main sources of ligand structures.

- The first source is computer programs that use graphic tools or languages such as SMILES (i.e. Simplified Molecular Input Line Entry System) to build these structures.
- The second source is databases of, sometimes, purchasable compounds.

SMILES

Examples of such programs are:

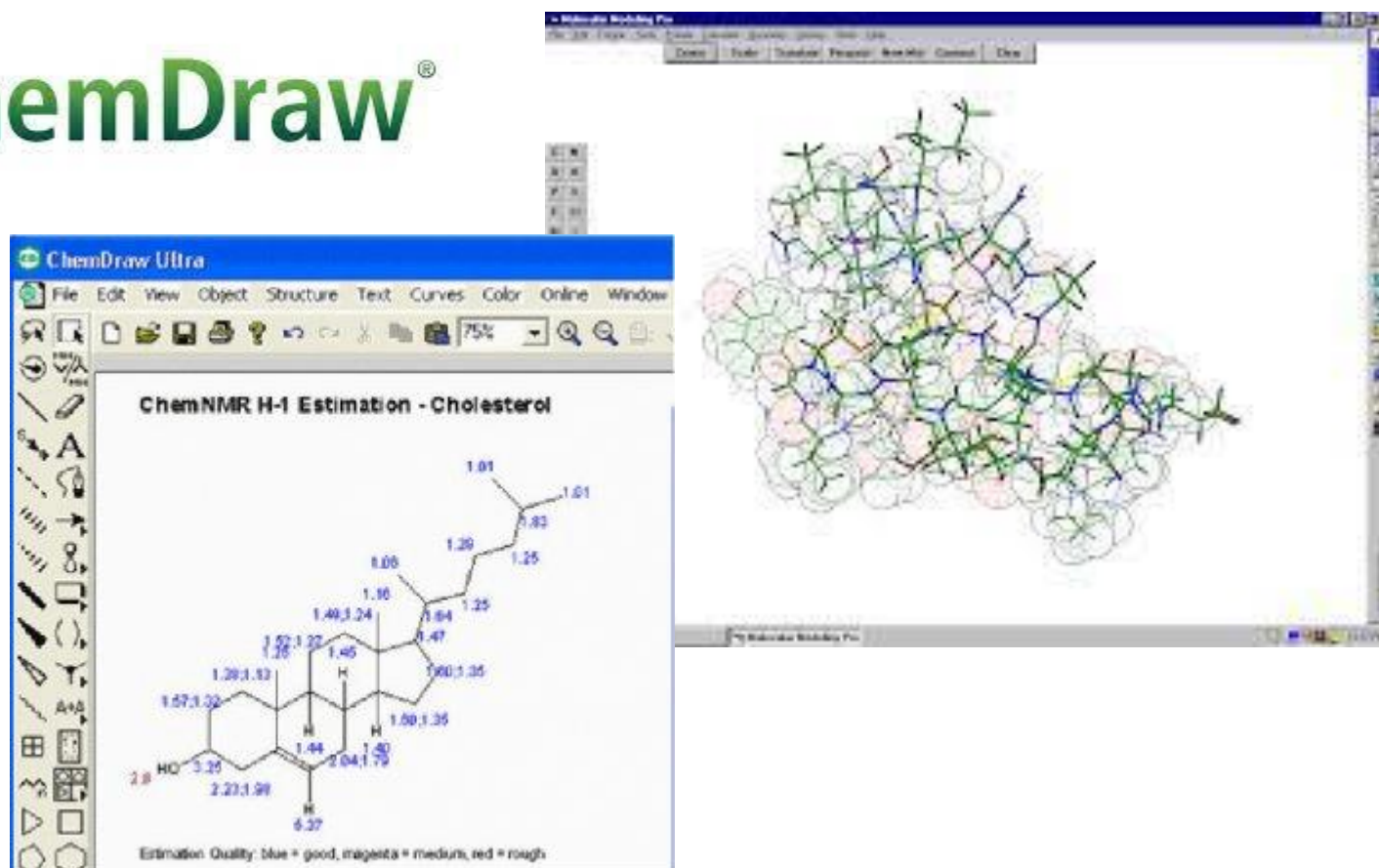
- (a) ChemDraw Ultra™ (CambridgeSoft Corporation, Cambridge MA, USA, <http://www.cambridgesoft.com/>);
- (b) KnowItAll ChemWindow™ (Bio-Rad Laboratories Inc., Hercules CA, USA, <http://www.bio-rad.com/>);
- (c) ISIS/Draw™ (MDL Information Systems, Inc., San Ramon CA, USA, <http://www.mdli.com/>); and
- (d) ACD/ChemSketch™ (Advanced Chemistry Development, Inc., Toronto, Canada, <http://www.acdlabs.com/>).

These programmes are useful only if the set of ligands is limited or if the molecules have not been previously reported.

ChemDraw Ultra



ChemDraw®

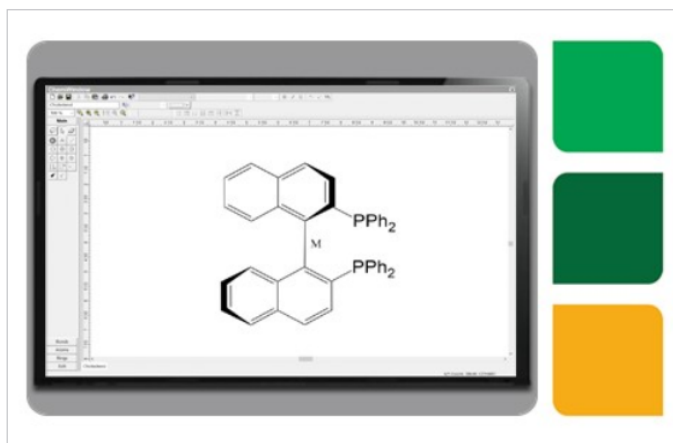


ERASMUS+

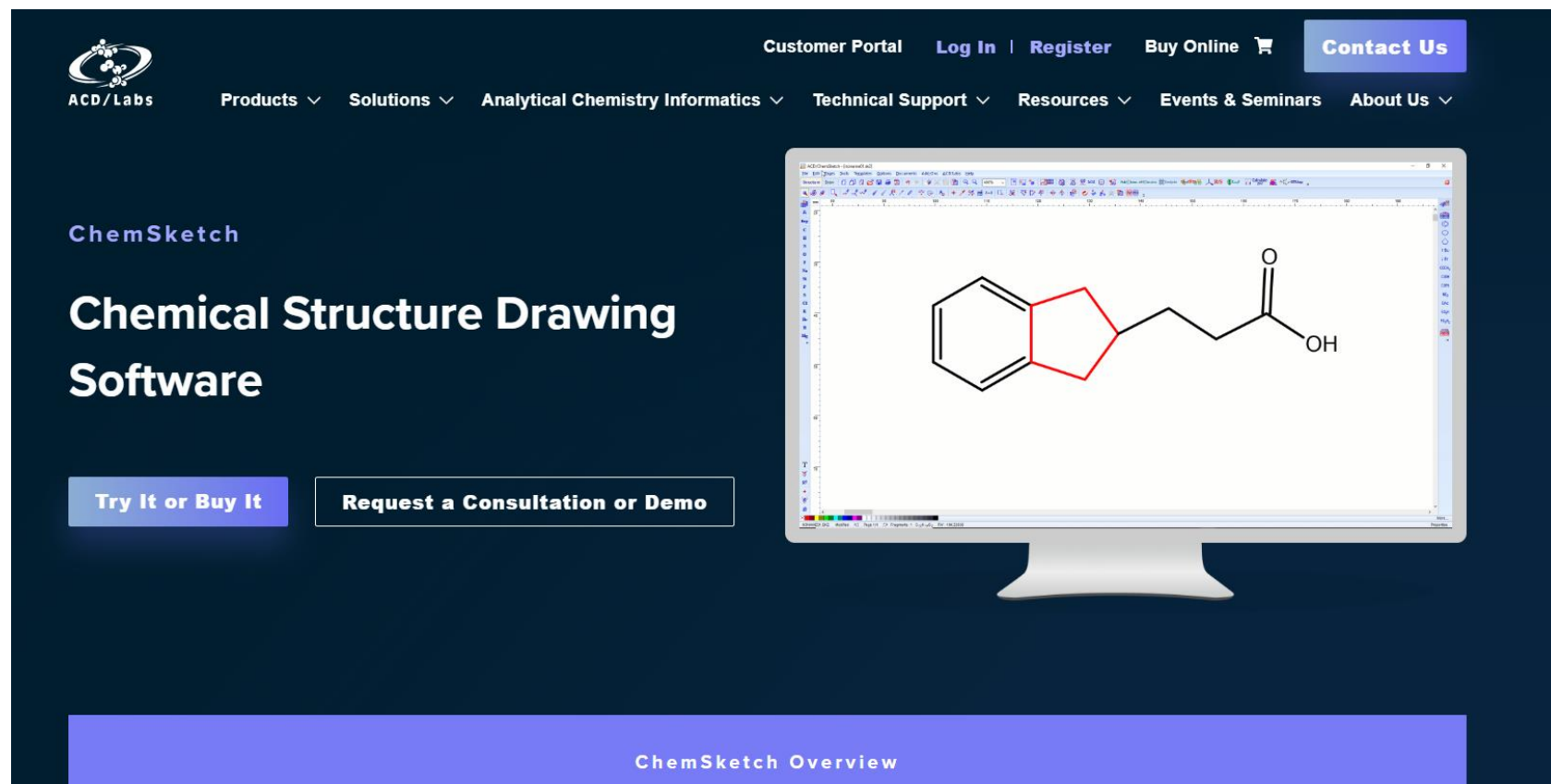
Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

KnowItAll ChemWindow

ChemWindow Chemical Structure Drawing Software



ACD/ChemSketch



The screenshot shows the ACD/ChemSketch website with a dark blue background. At the top, there is a navigation bar with links: Customer Portal, Log In, Register, Buy Online, and Contact Us. Below this is a secondary navigation bar with links: Products, Solutions, Analytical Chemistry Informatics, Technical Support, Resources, Events & Seminars, and About Us. The main content area features the ChemSketch logo and the text "Chemical Structure Drawing Software". Below this are two buttons: "Try It or Buy It" and "Request a Consultation or Demo". On the right side, there is a computer monitor displaying the chemical structure of 2-(1H-inden-3-yl)propanoic acid, which consists of an indene ring system attached to a propanoic acid chain. The structure is drawn in black lines, with the indene ring highlighted in red. The monitor also shows a standard software interface with a toolbar and a status bar at the bottom.

ACD/Labs

Products Solutions Analytical Chemistry Informatics Technical Support Resources Events & Seminars About Us

Customer Portal Log In Register Buy Online Contact Us

ChemSketch

Chemical Structure Drawing Software

Try It or Buy It Request a Consultation or Demo

ChemSketch Overview

Small molecular structures databases

Examples of such chemical databases are:

- (a) the Available Chemicals DirectoryTM (http://www.mdl.com/products/experiment/available_chem_dir), which contains 571000 unique chemicals that can be purchased from over 719 suppliers worldwide;
- (b) the MDL® screening compounds directory (formerly ACD-SCTM; http://www.mdl.com/products/experiment/screening_compounds/) where 3.4 million structures from 46 international chemical suppliers can be found;
- (c) the iResearchTM Library (<http://www.chemnavigator.com/cnc/products/iRL.asp>), which contains over 37.2 million chemical structures (around 20.6 million unique) from more than 252 chemistry suppliers;
- (d) the National Cancer Institute database (<http://129.43.27.140/ncidb2/download.html>) with 250251 compounds;

Small molecular structures databases

Examples of such chemical databases are:

- e) the Ligand.Info database (<http://Ligand.info/>), which is a compilation of various public databases of small molecules [i.e. ChemBank (<http://chembank.broad.harvard.edu/>), ChemPDB (<http://www.ebi.ac.uk/msd-srv/msdchem/cgi-bin/cgi.pl>), KEGG (<http://www.genome.ad.jp/ligand/>), NCI (<http://dtp.nci.nih.gov/>), AKosGmbH (<http://www.akosgmbh.eu/>), Asinex Ltd (<http://www.asinex.com/>), and TimTec (<http://www.timtec.net/>)] and which contains 1159274 entries;
- f) (f) the GDB database (i.e. the Generated Database of Chemical Space of SmallMolecules;<http://www.dcb.unibe.ch/groups/reymond/>), which contains 26.4 million compounds, the vast majority of which have never been synthesized; and
- g) (g) the ZINC database (<http://blaster.docking.org/zinc/>), which contains over 4.6 million compounds in ready-to-dock format from the catalogues of 46 different chemical suppliers.

Small molecular structures databases

Each molecule in the library is ready to be downloaded and used by a number of popular docking programs and it is also annotated with the following properties:

- (a) Molecular weight;
- (b) vendor and original catalog number;
- (c) Calculated LogP;
- (d) number of rotatable bonds;
- (e) number of hydrogen-bond donors;
- (f) number of hydrogen-bond acceptors;
- (g) number of chiral centers;
- (h) number of chiral double bonds (E/Z isomerism);
- (i) polar and apolar desolvation energy (in kcal/mol);
- (j) net charge;
- (k) number of rigid fragments; and
- (l) function or activity (when available).

Review Questions

- Give some examples of the protein databases.
- Give some examples of small molecular structures databases.
- Give some example of software used for building structures.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 2. Where Can Ligands Dock



Contents

- Introduction
- Docking programs

Introduction

In modern drug discovery, protein–ligand or protein–protein docking plays an important role in predicting the orientation of the ligand when it is bound to a protein receptor or enzyme using shape and electrostatic interactions to quantify it. The van der Waals interactions also play an important role, in addition to Coulombic interactions and the formation of hydrogen bonds. The sum of all these interactions is approximated by a docking score, which represents potentiality of binding. In the simplest rigid-body systems, the ligand is searched in a six-dimensional rotational or translational space to fit in the binding site, which can serve as a lead compound for drug design.

AutoDock

AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure. Over the years, it has been modified and improved to add new functionalities, and multiple engines have been developed.

AutoDock has applications in:

- X-ray crystallography;
- structure-based drug design;
- lead optimization;
- virtual screening (HTS);
- combinatorial library design;
- protein-protein docking;
- chemical mechanism studies





DOCK

DOCK 6 is written in C++ and is functionally separated into independent components, allowing a high degree of program flexibility. Accessory programs are written in C and Fortran 77. We provide source code for all programs. The DOCK suite of programs has modest disk space and memory requirements.

The new features of DOCK 6 include: genetic algorithms and de novo design for fragment based ligand searching; additional scoring options during minimization; DOCK 3.5 scoring-including Delphi electrostatics, ligand conformational entropy corrections, ligand desolvation, receptor desolvation; Hawkins-Cramer-Truhlar GB/SA solvation scoring with optional salt screening; and AMBER scoring-including receptor flexibility, the full AMBER molecular mechanics scoring function with implicit solvent, conjugate gradient minimization, and molecular dynamics simulation capabilities. Because DOCK 6 is an extension of DOCK 5, it also includes all previous features.

GOLD



GOLD is the validated, configurable protein–ligand docking software for expert drug discovery. For virtual screening through to lead optimization.

Customize with constraints to guide results towards known features or behaviours, and assess the impact of water molecules on the docking. Or use the wizard for quick protein-ligand docking results.

The extensively validated scoring functions in GOLD are trusted by scientists globally for pose prediction and virtual screening – with results seen across the literature.

Glide

Glide offers the full range of speed vs. accuracy options, from the HTVS (high-throughput virtual screening) mode for efficiently enriching million compound libraries, to the SP (standard precision) mode for reliably docking tens to hundreds of thousands of ligands with high accuracy, to the XP (extra precision) mode where further elimination of false positives is accomplished by more extensive sampling and advanced scoring, resulting in even higher enrichment.

SeeSAR

SeeSAR is a 3D desktop modeling platform. It facilitates interactive, visual, compound prioritization - as well as compound evolution. Structure-based design work ideally supports a multi-parameter optimization to maximize the likelihood of success, rather than affinity alone.



Molegro Virtual Docker

Molegro Virtual Docker is an integrated platform for predicting protein - ligand interactions. Molegro Virtual Docker handles all aspects of the docking process from preparation of the molecules to determination of the potential binding sites of the target protein, and prediction of the binding modes of the ligands.

Molegro Virtual Docker provides

- **High docking accuracy:** the docking engine has been proven to correctly identify binding modes with high accuracy. Molegro Virtual Docker has been shown to outperform other docking programs with regard to identification of correct binding modes.
- **Easy-to-use interface:** the built-in wizards enable the user to easily setup and perform docking runs. Advanced visualization and analysis tools are provided to examine ligand-receptor interactions and fine-tune found docking solutions.
- **Cross-platform:** supported on Linux, Windows and Mac, allowing easy interoperability between platforms.

MPI-Vina

MPI-Vina is an open-source parallelization of AutoDock Vina, which massively reduces the time of virtual screening by using compute clusters or network of computers. It is developed based on MPI and intended for distributed memory environment. The goal of MPI-Vina is to reduce the overall time of screening ligand data set. AutoDock Vina is the primary docking program of MPI-Vina.

In MPI-Vina, protein-ligand docking is distributed into different nodes of a computational cluster where each node performs docking of a single ligand against a target receptor. Distribution of ligand set allows docking multiple ligands in multiple nodes concurrently.

MPI-Vina reduces the overall time of doing virtual screening dramatically than the traditional virtual screening approach by using state-of-the-art parallel processing. It also reduces the magnitude and complexity of the screening problem and focuses on drug discovery and optimization efforts on the most promising leads..

OEDocking

OEDocking is a suite of well-validated molecular docking tools and workflows, each specifically designed to address its own unique aspect of protein-ligand interaction. Specifically, it features POSIT for informed pose prediction as well as FRED and HYBRID as complementary tools for virtual screening.

rDock



rDock is a fast and versatile Open Source docking program that can be used to dock small molecules against proteins and nucleic acids. It is designed for High Throughput Virtual Screening (HTVS) campaigns and Binding Mode prediction studies.

rDock is mainly written in C++ and accessory scripts and programs are written in C++, perl or python languages.

The full rDock software package requires less than 50 MB of hard disk space and it is compilable (at this moment, only) in all Linux computers.



Lead Finder

Lead Finder software is a computational chemistry application for modeling protein-ligand interaction. Lead Finder can be used in molecular docking studies and for the quantitative evaluation of ligand binding and biological activity.

Lead Finder introduces three scoring functions optimized for the accurate prediction of 3D docked ligand poses, protein-ligand binding energy and rank-ordering active and inactive compounds in virtual screening experiments. Lead Finder is designed to satisfy needs of computational and medicinal chemists involved in the drug discovery process, pharmacologists and toxicologists involved in the modeling and evaluation of ADMET properties *in silico*, and biochemists and enzymologists working on enzyme specificity and rational enzyme design.

ADAM

ADAM is an automated docking tool, which can predict the stable binding mode of flexible ligand molecule to target macromolecule. It is able to construct energetically favorable docking models at high speeds, exploring the conformational space of flexible ligand continuously and exhaustively. The docking accuracy and reliability in the actual drug design are the top level in the world (even in the cases of supposing induced-fit motion of target protein). Moreover, high-throughput virtual screening can be performed easily by using ADAM. In a lot of in-house projects and funded researches for drug developments, ADAM virtual screening has already achieved successes. ADAM should greatly contribute to a variety of research areas including the QSAR study and analysis of protein-ligand interaction, as well as the drug design and drug discovery.

PLANTS

In the drug development process, predicting the complex structure of a small ligand with a protein, the so-called protein-ligand-docking-problem, is extensively used in virtual screening of large databases and in lead optimization. Given a protein structure, a ligand structure and a scoring function, the goal is to find a low energy ligand conformation in the protein's binding site that corresponds to the global minimum of the scoring function. The docking algorithm PLANTS is based on a class of stochastic optimization algorithms called ant colony optimization (ACO). ACO is inspired by the behavior of real ants finding a shortest path between their nest and a food source. The ants use indirect communication in the form of pheromone trails which mark paths between the nest and a food source. In the case of protein-ligand docking, an artificial ant colony is employed to find a minimum energy conformation of the ligand in the binding site. These ants are used to mimic the behavior of real ants and mark low energy ligand conformations with pheromone trails. The artificial pheromone trail information is modified in subsequent iterations to generate low energy conformations with a higher probability.

SwissDock

SwissDock is based on the docking software EADock DSS, whose algorithm consists of the following steps:

- many binding modes are generated either in a box (local docking) or in the vicinity of all target cavities (blind docking).
- simultaneously, their CHARMM energies are estimated on a grid.
- the binding modes with the most favorable energies are evaluated with FACTS, and clustered.
- the most favorable clusters can be visualized online and downloaded on your computer.

PatchDock

PatchDock is an algorithm for molecular docking. The input is two molecules of any type: proteins, DNA, peptides, drugs. The output is a list of potential complexes sorted by shape complementarity criteria.

PatchDock algorithm is inspired by object recognition and image segmentation techniques used in Computer Vision. Docking can be compared to assembling a jigsaw puzzle. When solving the puzzle we try to match two pieces by picking one piece and searching for the complementary one. We concentrate on the patterns that are unique for the puzzle element and look for the matching patterns in the rest of the pieces. PatchDock employs a similar technique. Given two molecules, their surfaces are divided into patches according to the surface shape. These patches correspond to patterns that visually distinguish between puzzle pieces. Once the patches are identified, they can be superimposed using shape matching algorithms. The algorithm has three major stages:



DockingServer

DockingServer offers a web-based, easy to use interface that handles all aspects of molecular docking from ligand and protein set-up.

While its user friendly interface enables docking calculation and results evaluation carried out by researchers coming from all fields of biochemistry, DockingServer also provides full control on the setting of specific parameters of ligand and protein set up and docking calculations for more advanced users.

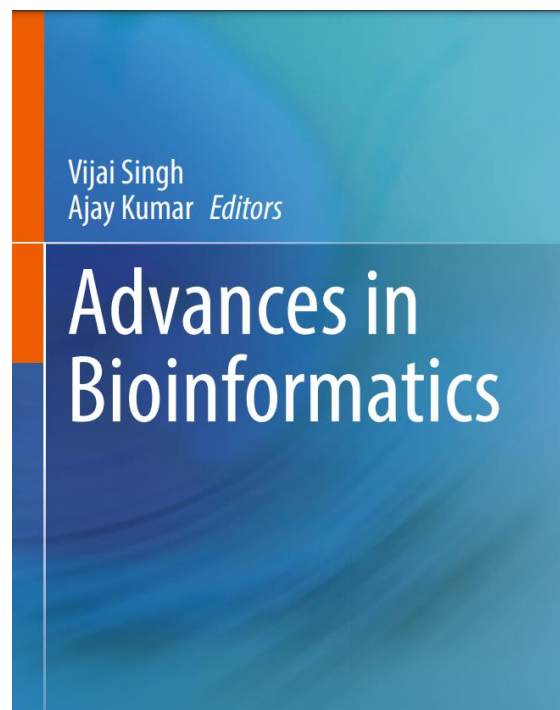
The application can be used for docking and analysis of single ligands as well as for high throughput docking of ligand libraries to target proteins.

DockingServer integrates a number of computational chemistry software specifically aimed at correctly calculating parameters needed at different steps of the docking procedure, i.e. accurate ligand geometry optimization, energy minimization, charge calculation, docking calculation and protein-ligand complex representation. Thus, the use of DockingServer allows the user to carry out highly efficient and robust docking calculations by integrating a number of popular software used in in silico chemistry into one comprehensive web service.



Review Questions

- Give examples of docking programs.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 3. How Does a Docking Program Work



Contents

- Introduction
- Algorithms



Introduction

The ability to produce a large and diverse set of ligand poses (i.e. candidate coordinates for conformations, positions and orientations of the ligand within the protein-ligand complex) is a prerequisite for a docking tool to be useful.

Algorithms

There are two main types of algorithms that allow docking programs to search the conformational space of the ligand in order to find its poses:

- (a) systematic or directed approaches;
- (b) random or stochastic methods.

Simulation methods such as molecular dynamics and quantum mechanics are also able to do this but, at present, they are too computationally expensive to be applied during VLS.

Search algorithms

There are three subtypes of systematic or directed search algorithms:

- (a) conformational search methods;
- (b) fragmentation or incremental construction methods; and
- (c) database methods.

They all have in common that the algorithms try to explore all the degrees of freedom of the ligand; the way they carry out the search, however, is different. In this respect, the conformational search algorithms try to obtain all possible ligand conformations by subjecting all the ligand bonds that can be rotated to a 360° turn by using a fixed increment.

Search algorithms

One of the main problems of this methodology is that the number of ligand conformations that can be generated increases exponentially with the number of rotatable bonds and, therefore, its application in its purest form is very limited (i.e. usually several constraints and restraints on the ligand are needed to reduce the combinatorial explosion problem).

Search algorithms

On the other hand, fragmentation or incremental construction methods are currently used by docking programs such as eHiTS[®], LUDITM (http://www.accelrys.com/products/insight/sbd_modules.html), FlexXTM (<http://www.biosolveit.de/FlexX/>), DOCK (<http://dock.compbio.ucsf.edu/>) and Hammerhead to search for available ligand conformations.

Search algorithms

In these methods, the ligands are incrementally grown in the binding site either by dividing the ligand into several rigid fragments, docking them and finally trying to rebuild the ligand structure by joining the rigid fragments with the flexible segments that join them (this is the so called **place-and-join approach**) or by dividing the ligand into a rigid core fragment that is first docked and the rest of the ligand segments are then successively added (this is the so called incremental approach).

Search algorithms

The last subtype of systematic search algorithms are the database methods that use libraries of pregenerated conformations (so called **conformational ensembles**) that are subsequently subject to a rigid body docking. One example of a docking program that uses this database methodology to deal with the ligand flexibility issue is FLOG.

Random or stochastic search algorithms

There are three subtypes of random or stochastic search algorithms:

- (a) Monte Carlo (MC) methods;
- (b) Genetic Algorithm (GA) methods; and
- (c) tabu search (TS) methods.

They all try to sample the conformational space by performing a random conformational change to the ligand followed by acceptance or rejection of the resulting conformer by using a predefined probability function.

Random or stochastic search algorithms

If the ligand conformation is finally accepted, it is used as the starting point for a new random conformational change. The main limitation of such methods is that the pose that matches the experimental conformation of the ligand in the complex may not be achieved and, therefore, will not be evaluated by the corresponding scoring function.

MC methods

In MC methods, the ligand is randomly placed in the receptor binding site, it is scored and a new conformation is generated by random changes that are applied to:

- (a) the ligand rotatable bonds; and
- (b) the ligand position (i.e. the ligand is randomly rotated and translated).

After each change, the ligand is typically minimized and scored. Then, if the new solution scores better than the previous one, it is immediately accepted. On the other hand, if the latter conformation is not a new minimum, a Boltzmann-based probability function is applied. If the pose succeeds in this probability function test, it is accepted; if not, it is rejected (this is what is called a Metropolis criterion).

MC methods

Docking programs that can deal with ligand conformational searches using MC-based algorithms are AutoDock (<http://autodock.scripps.edu/>) (in fact, it was the first docking program in which a MC simulated annealing algorithm was implemented), PRODOCK, ICMTM (<http://www.molsoft.com/docking.html>), MCDOCK, DockVisionTM (<http://www.dockvision.com/>) and QXP.

Genetic Algorithm (GA) methods

GAs use concepts derived from genetics and the theory of biological evolution to explore the conformational space of the ligands. Unlike MC methods, GAs start from an initial population of different conformations of the ligand and each one is defined by a set of state variables or genes that describe the translation, the orientation, and the conformation of the ligand relative to the receptor. The complete set of these ligand genes is the genotype, whereas the resulting atomic coordinates are the phenotype.

Genetic Algorithm (GA) methods

Docking programs that can use GA or GA-like algorithms to deal with ligand conformational searches are GOLD (http://www.ccdc.cam.ac.uk/products/life_sciences/gold/), AutoDock v3.0 and v4.0, DIVALI, MVDTM and DARWIN. The last subtype of random search algorithms are the TS methods that work by imposing restrictions that prevent already explored areas of the ligand conformational space from being visited again and, therefore, favor the analysis of new conformations. To do so, when a new ligand conformation is available, its root-mean square deviation (i.e. RMSD) relative to the previously visited conformations is calculated.

Genetic Algorithm (GA) methods

When this calculation is finished, the lowest RMSD is compared with a certain threshold value and, if it is higher, then the analyzed conformation for the ligand is accepted and its coordinates are stored and used to accept or reject new conformations. PRO_LEADS and MVDTM are the most popular docking programs that can use a TS algorithm.

Tabu search (TS) methods

Tabu search (TS) is a metaheuristic search method employing local search methods used for mathematical optimization. It was created by Fred W. Glover in 1986 and formalized in 1989.

Local (neighborhood) searches take a potential solution to a problem and check its immediate neighbors (that is, solutions that are similar except for very few minor details) in the hope of finding an improved solution. Local search methods have a tendency to become stuck in suboptimal regions or on plateaus where many solutions are equally fit.

Tabu search (TS) methods

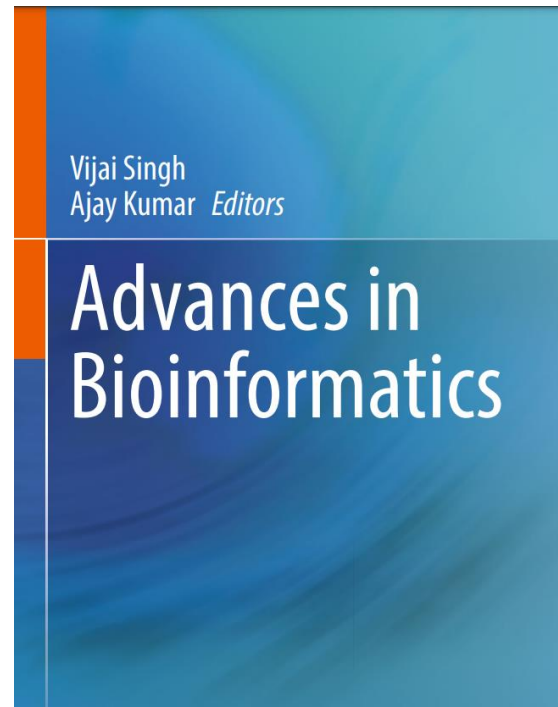
Tabu search enhances the performance of local search by relaxing its basic rule. First, at each step worsening moves can be accepted if no improving move is available (like when the search is stuck at a strict local minimum). In addition, prohibitions (henceforth the term tabu) are introduced to discourage the search from coming back to previously-visited solutions.

The implementation of tabu search uses memory structures that describe the visited solutions or user-provided sets of rules.[2] If a potential solution has been previously visited within a certain short-term period or if it has violated a rule, it is marked as "tabu" (forbidden) so that the algorithm does not consider that possibility repeatedly.



Review Questions

- Explain the way the docking program works?
- What are the most popular algorithms for docking?



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 4. The Scoring Functions



Contents

- Introduction
- Balance between speed and accuracy
- Force field-based scoring functions
- Empirical-based scoring functions
- Knowledge-based scoring functions

Introduction

Once the candidate conformations for the ligand in the complex have been predicted, their binding affinity for the receptor must be scored. This is done by means of a scoring function that evaluates the search results and then gives, ideally, the highest score to the right pose. In fact, if the search algorithm can find the correct pose but the scoring function cannot recognize it, the program will make an invalid and useless suggestion to the scientist. Therefore, the role of the scoring function is critical in every docking protocol.

Balance between speed and accuracy

The lack of a suitable scoring function, both in terms of speed and accuracy, is the major bottleneck in docking. Some recent studies have shown that the scoring function's performance on a particular target protein is largely case-dependent. Therefore, any docking study should start either with an objective evaluation of available scoring functions on the target protein so that the most suitable one can be chosen or with a scoring function that has been tuned according to the binding site characteristics.

Force field-based scoring functions

Force field-based scoring functions are similar to empirical-based functions because they both predict the binding free energy of a protein-ligand complex by adding individual contributions from different types of interactions. Nevertheless, the interaction terms of the former are derived from the theoretical physics that underlie molecular mechanics force fields as opposed to the experimental affinities used to derive the latter. Thus, force field scoring functions use energy functions from classical molecular mechanics and, in general, quantify the sum of two energies:

- (a) the interaction energy between the receptor and the ligand; and
- (b) the internal energy of the ligand.

The binding free energy

The binding free energy of the intermolecular interaction is often approximated as the result of the sum of a van der Waals energy term (described by means of the Lennard-Jones potential function) and an electrostatic energy term (described by means of a Coulombic formulation with a distance-dependent dielectric function that reduces the contribution from charge–charge interactions). Moreover, AutoDock v3.0 and v4.0 and G-Score also consider a hydrogen-bonding term (although with different functional forms) in an attempt to increase the potential of specific molecular recognition.

The internal energy of the ligand

The functions that describe the internal energy of the ligand are usually very similar to the protein–ligand interaction energy because they also contain van der Waals and/or electrostatic contributions (although AutoDock – and optionally GoldScore – also considers a hydrogen bond term and G-Score includes a torsional entropy term).

Limitations

Force field-based scoring functions have the following limitations:

- (a) they have difficulties in considering solvation and entropic terms and either ignore them or deal with them in a cursory fashion (because they were originally formulated to model enthalpic gas-phase contributions to structure and energetics);
- (b) non-bonded interactions can only be dealt with by introducing, in a more or less arbitrary way, cut-off distances that complicate the accurate treatment of the long-range effects involved in binding
- (c) polar interactions are overemphasized; and
- (d) the calculation of atomic partial charges relies on fast but inaccurate methods based on electronegativity indices instead of quantum mechanics methods such as those used in force-field development.

The most commonly used force field scoring functions

the force field scoring functions that are most commonly used by docking programs are D-Score, G-Score (based on the Tripos force field), GoldScore, the ones used by AutoDock v3.0 and v4.0 [based on the AMBER (Assisted Model Building and Energy Refinement) force field but modified with empirical parameters], the full AMBER molecular mechanics scoring function with implicit solvent used by DOCK v6.1 and the one used by single ligand docking in DockVision. Obviously, more rigorous molecular mechanical force-fields such as AMBER, CHARMM (Chemistry at HARvard Macromolecular Mechanics), GROMOS (GRONingen MOlecular Simulation System) and OPLS (Optimized Potentials for Liquid Simulations) can also be used although at the expense of a huge increment in the computational cost.

Empirical-based scoring functions

Empirical-based scoring functions are based on the idea that the binding energy (i.e. ΔG_{bind}) can be obtained by adding several individual and uncorrelated terms and on the pioneer work of Hans-Joachim Böhm at the BASF AG Central Research at Ludwigshafen (Germany). Many of the terms in the empirical scoring functions have equivalences in the force-field scoring functions but the former are usually simpler in form than the latter. Empirical scoring functions contain terms that account for the net contribution of different types of non-bonded interactions (i.e. hydrogen bonds, ionic and hydrophobic interactions with the receptor) to the overall binding energy.

Empirical-based scoring functions

Some of these scoring functions also contain terms that, to some extent, account for:

- (a) non-enthalpic contributions such as the so-called rotor term, which approximates entropy penalties on binding from a weighted sum of the number of rotatable bonds in ligands; and
- (b) solvation and desolvation effects.

Moreover, the terms that correspond to the contribution of a specific type of non-bonded interaction can be function-specific (such as the extra additional term for aromatic interactions in F-Score) or implemented depending on the scoring function.

Empirical-based scoring functions

The main strength of empirical functions is that their terms are often simple to compute. Thus, experimental data such as binding energies or a set of X-ray receptor-ligand complexes are used as the input for a regression analysis that best fits the scoring function to the experimental data.

Empirical-based scoring functions

Their main drawbacks, however, are that:

- (a) they are strongly dependent on the experimental data used in the parameterization process and, therefore, terms from differently fitted scoring functions cannot easily be recombined into a new scoring function; and
- (b) It is unclear whether they are able to predict the binding affinity of ligands that are structurally different from those used in the training set.

Empirical-based scoring functions

Examples of empirical scoring functions are Böhm's scoring function (which comes with LUDI, F-Score, ChemScore, SCORE, Fresno, X-SCORE, PLP and Slide. The scoring function used in Hammerhead and Surflex are fundamentally based on Böhm's approach whereas the empirical-based scoring functions used by GOLD and Glide (<http://www.schrodinger.com/>) are modifications of the Eldridge's ChemScore function. FlexX, on the other hand, implements modifications of both the Böhm and Eldridge functions. Finally, the "MolDock Score" scoring function used by MVDTM is derived from PLP but adds a new hydrogen bonding term and uses new charge schemes. It has been claimed that the empirical scoring functions are the ones that give better results.

Knowledge-based scoring functions

Knowledge-based scoring functions are based on knowing contact preferences and rely on the classical statistical physics idea that observed distributions of geometries can be used to deduce the potential that gives rise to the observed distribution. They were first proposed for studying protein folding but they have also been used in protein ligand docking and to predict protein structures and protein–protein complexes. In the protein-ligand docking field, they are used to score ligand binding poses by means of relatively simple atomic interaction-pair potentials that are built from statistical analyses of experimentally determined protein-ligand structures.

Knowledge-based scoring functions

Knowledge scoring functions try to extract rules by capturing such information as the frequency of occurrence and non-occurrence (i.e. negative data) of different atom–atom pair contacts and other typical interactions in the experimental structures of protein-ligand complexes because it is assumed that:

- (a) interatomic distances occurring more often than some average value should represent favorable contacts, and vice versa; and
- (b) the observed distribution of distances between pairs of different atom types reflects their interaction energy.

In practice, large training sets of protein–ligand structures are analyzed to provide sets of distribution functions that are then converted to sets of atom-pair potentials by means of the inverse Boltzmann law (which provides an energy value for a given state based on observed probabilities).

Knowledge-based scoring functions

Among the strengths of these functions are:

- (a) their simple atomic interactions-pair potentials have a low computational cost which means that large compound databases can be efficiently screened; and
- (b) that they require no experimental binding affinities to be derived.

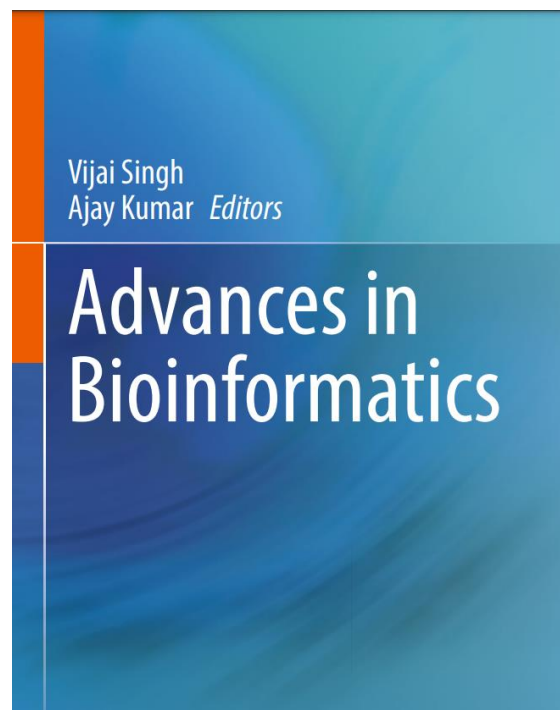
Knowledge-based scoring functions

In contrast, some limitations are that:

- (a) it is difficult to predict their behaviour (i.e they should only be used for VLS when enough ligand and structural information is available to validate the setup of the calculation);
- (b) they are designed to reproduce experimental structures rather than binding energies;
- (c) they are used to identify decoys or in combination with other scoring functions during VLS but not during the optimization phase; and
- (d) They are not general or accurate enough because of the limited number of interactions that can be inferred from crystal structures and the inadequate description of repulsive forces.

Review Questions

- What are the types of scoring functions used in docking software?
- Give some examples and explain their characteristics.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 5. The Importance of Considering Receptor Flexibility During Docking



Contents

- Introduction
- Conformational variability
- Allostery
- Docking with flexible side chains
- Induced fit docking

Introduction

Docking is implemented to predict the binding mode and affinity of small molecules. When docking is applied to a large compound database, it is referred to as virtual screening (VS) and often centered on the prediction of new ligands with high affinity for a target molecule to enrich the compound set for experimental testing. De novo design is performed with the intent of predicting new compounds in novel chemical space. Fragment-mapping techniques that use functional groups to probe binding sites are often applied for this type of approach.

Conformational variability

Our understanding of receptor–ligand binding has significantly advanced from the original lock-and-key model proposed by Fischer (1890). Early experimental studies showed that the act of ligand binding influences the protein conformation, referred to as conformational induction or induced fit. Another model of ligand binding is conformational selection, wherein the ligand chooses a binding partner from among available states in the conformational ensemble, thereby shifting the population distribution.

Conformational variability

the coordinates for a protein that have been obtained by X-ray diffraction and that are available from the PDB correspond to the averages of the coordinates of that protein:

- (1) in the unit cells that are part of the diffracted crystal; and
- (2) during the time that the diffraction experiment is carried out.

Nevertheless, the lines that contain atom coordinates in the PDB files also contain a parameter value (i.e. the temperature factor or B-factor) that can be thought of as a measure of how much the corresponding atom oscillates or vibrates around the position specified by the corresponding coordinates (i.e. the higher the value of the B-factor, the less precise the atom coordinates are).

Allostery

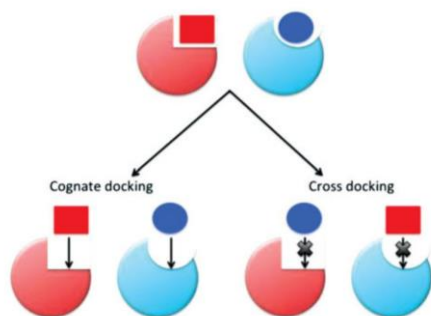
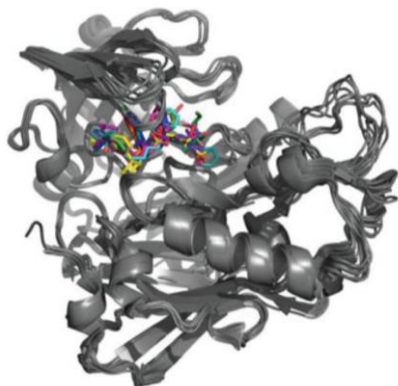
Protein–ligand flexibility upon binding is crucial for proper understanding of allosteric regulation. Nussinov and co-workers postulated that most proteins exist in an ensemble of states, and thus most proteins have the potential for allostery. Based on the experimental literature, they demonstrated that the binding of an allosteric ligand shifted the population of conformational substates, thereby influencing the ability of other ligands to bind an alternate site.



Allostery

Drug resistance is a growing problem that calls for new approaches to drug therapy. Exploring allosteric control in protein targets provides exciting opportunities for finding new modes of action and hence, overcoming emergent resistance, and developing cocktails of drugs to improve treatment.

The cross-docking problem



Research into protein flexibility and allostery has lent support to the importance of representing multiple states in binding studies. Mobley & Dill (2009) noted that binding free energy (ΔG_{bind}) and entropy are influenced by the shape and width of the entire conformational landscape, rather than a single rigid pose. Murray et al. (1999) examined approaches for using rigid receptors in docking studies. When the authors attempted to dock a known ligand into a protein structure solved in the presence of a different ligand (referred to as cross-docking), they found that the active site was biased toward to the native ligand. A variety of differences in the surface of the binding site were identified for the same protein solved with different ligands or in the absence of a ligand.

The cross-docking problem

An underlying assumption in rigid docking efforts is that the complexed state is the lowest free energy state. However, the ligand-bound state is not always the lowest energy state for either the protein or the ligand.

Sources of structural information

Use of a set of similar conformations only generates a finite amount of information on potential binding partners. Additional receptor space should be explored, especially in scenarios where alternate binding modes are of interest. X-ray crystallography, NMR structures, homology models, normal mode analysis (NMA), and molecular mechanics simulations are all potential sources of structural diversity. Advances in technology have allowed for significant increases in available structural information for a vast number of receptor targets.

Sources of structural information

Several research groups, as well as thousands of computer users, have dedicated their computer power to distributed structure prediction of proteins with Folding@HOME, POEM@HOME, and Rosetta@HOME.

Soft docking allows limited discovery of new conformation space while docking with flexible side-chain samples observed conformational space. More recent techniques incorporate ensembles of structures to allow for a greater measure of native flexibility or to reveal new conformations. By relying on several protein conformations, new chemical space can be explored, allosteric sites can be discovered, and more accurate binding conformations can be generated. The accepted metric for a well-docked pose is ≤ 2.0 Å RMSD from the experimentally determined structure.

Soft docking

Protein flexibility can be incorporated into a binding-site model before docking is initiated or it can be allowed post-docking through refinement of the bound complex. Docking can be performed against an average structure or ensemble, a set of receptor conformations (multiple receptor conformations (MRC)), or with conformations generated ‘on the fly’ (induced-fit docking (IFD)).

Relaxation methods

Refinement of the docked complex is another simple approach that adjusts for protein flexibility by modeling induced-fit effects. The incorporation of protein flexibility on the 'back end' can only be done when the docking technique was based on an all-atom structure; it cannot be performed on a grid, or in any other situation where the protein is not explicitly represented. Monte Carlo (MC) or MD simulations are a popular choice for refinement because they enable the optimization of docked poses, investigation of solvent effects, examination of the kinetic stability, and prediction of ΔG_{bind} from physics-based scoring functions. Refinement is frequently performed as a final step in many of the docking approaches.

Docking with flexible side chains

Early efforts at on-the-fly docking focused on the side chains, with the use of a rotamer library based on backbone dihedral angles to describe protein flexibility.

Side-chain motion upon ligand binding has also been approached through continuum sampling methods. Techniques that apply some form of continuum sampling to side-chain motion are AutoDock 4.0, Dynasite/GOLD, FlexX, ICM/MC, Mining Minima, Skelgen, and SLIDE.

Allowing side-chain flexibility is less resource intensive than full flexibility methods and it enabled some conformational variability through the exploration of low-energy orientations of side chains. However, incorporating side-chain flexibility has failed in cases of proteins with large-scale hinge or loop rearrangements or even backbone-dependent movement of side chains, neither of which can be taken into account by discrete or continuum sampling

Induced fit docking

Significant progress has been made since the initial implementation of flexible side chains in docking. Sampling motion of the side chains on the fly increases the potential energy space but can still overlook global conformational shifts. Allowing for conformational changes during docking can be a highly accurate technique for modeling bound poses of protein–ligand complexes. IFD is important because it can allow the docking simulation to search new conformational space; however, this sampling of receptor and ligand degrees of freedom is also quite computationally intensive, which limits its application in large-scale virtual screening studies.

Ensemble docking

Ensemble docking differs from IFD in that protein flexibility is accounted for prior to the actual docking. Two different types of methods exist for representing receptor flexibility during docking: grid-based ensembles and structure-based ensembles. Frequently, alternate protein conformations are represented on a two-body potential grid, enabling fast, inexpensive docking simulations. Flexibility can also be incorporated into binding predictions through the sequential docking to structures in a conformational ensemble or docking to an averaged/united receptor structure. Due to the time required for initial ensemble generation and repeated docking runs to a set of static structures, sequential docking is typically the most computationally intensive approach. However, it avoids the discovery of receptor–ligand complexes that are physically impossible (paradoxical ligands), which are sometimes seen in the results from docking to an average structure or grid.

Grid-based ensembles

The first docking method to use a composite grid was performed through DOCK3.5 in order to evaluate the impact of representing conformational variability as a structural ensemble on binding pose results.

The incorporation of protein flexibility through docking to an interaction grid that represents the receptor ensemble is a common approach for SBDD, and additional methods are those based on AutoDock 3.0, Flog, GRID/CPCA, IFREDA, ISCD, and sets of consensus structures.

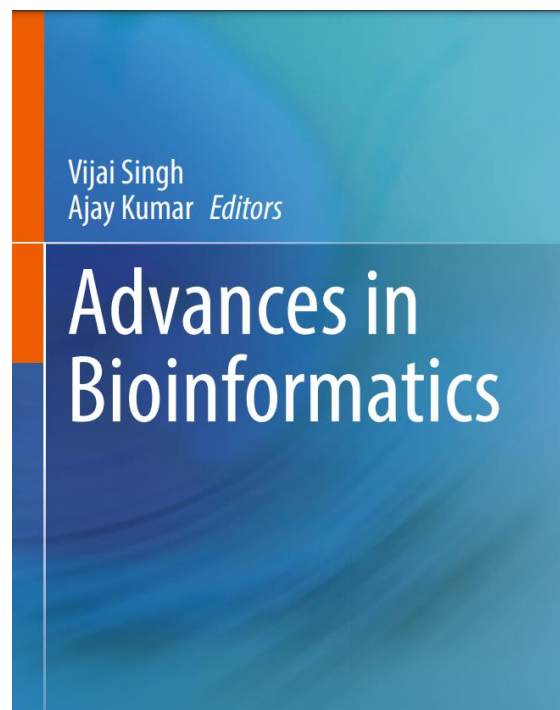
Structure-based ensembles

The multiple copy simultaneous search (MCSS) methodology was the earliest computational approach for mapping binding sites with functional group probes.

Of the variety of techniques that have been developed to improve binding predictions for protein–ligand complexes, the use of a conformational ensemble is perhaps the most common approach.

Review Questions

- How important is to consider receptor flexibility during docking?
- What types of docking procedures exist according to the flexibility of ligands and proteins?



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

Topic 2. Docking Problem

Lesson 6. Main Characteristics of Selected Docking Software



Contents

- Introduction
- Examples of docking software

Introduction

Molecular docking has become an important common component of the drug discovery toolbox, and its relative low-cost implications and perceived simplicity of use has stimulated an everincreasing popularity within academic communities. The goal of ligand-protein docking is to predict the predominant binding mode(s) of a ligand with a protein of known three-dimensional structure. Successful docking methods search high-dimensional spaces effectively and use a scoring function that correctly ranks candidate dockings.

eHiTS® (electronic High Throughput Screening; <http://www.simbiosys.ca/ehits/>

eHiTS® is being developed by SymBioSys Inc. in Toronto (Canada) and its main strengths are:

- (a) it is easy to use;
- (b) it performs very well (it is both quick and accurate); and
- (c) it has a lot of automated features that simplify the drug design workflow and provide innovative solutions to common docking problems (e.g. the protonation state of the ligand/receptor pair; an exhaustive search of the ligand poses; the speed-up of the calculations in VLS; automatic identification of probable binding sites, the capacity to tailor the scoring function to the characteristics of the receptor binding site, etc.).

eHiTS® (electronic High Throughput Screening; <http://www.simbiosys.ca/ehits/>

eHiTS® is not perfect and also has some limitations such as:

- (a) the receptor flexibility is limited to rotations of the -OH or -NH₃⁺ groups from some amino acids (i.e. Ser, Thr, Tyr and Lys);
- (b) All ring systems in the ligands are considered as rigid and therefore, their conformations are not changed during docking (hence, for a complete conformational sampling it is necessary to use multiple ring conformers); and
- (c) no knowledge-based constraints can be imposed on the docking (e.g. a specific ligand atom cannot be forced to be in a specific location in the poses; certain interactions cannot be prevented from occurring; etc.).



GOLD™ (Genetic Optimisation for Ligand Docking;
http://www.ccdc.cam.ac.uk/products/life_sciences/gold/

GOLD™ was originally developed by the University of Sheffield (<http://www.shef.ac.uk/>), GlaxoSmithKline plc (<http://www.gsk.com/>) and the CCDC (Cambridge Crystallographic Data Centre; <http://www.ccdc.cam.ac.uk/>). At present, CCDC is developing and maintaining GOLD™ in an ongoing project with Astex Therapeutics (<http://www.astex-therapeutics.com/>) and Syngenta (<http://www.syngenta.com>).



GOLD™ (Genetic Optimisation for Ligand Docking; http://www.ccdc.cam.ac.uk/products/life_sciences/gold/

GOLD's main strengths are that: (a) some backbone and side chain flexibility (for the moment, up to ten user-defined residues) can be included in the calculations; (b) the program can use userdefined scoring functions and not only the two default ones provided with GOLD™ (and even modify them); (c) the energy functions are partly based on conformational and non-bonded contact information derived from the Cambridge Structural Database (<http://www.ccdc.cam.ac.uk/products/csd/>); (d) a variety of constraint options can defined for the docking; (e) crystallographic water molecules in the ligand binding site can be considered during the docking; (f) it is optimized for parallel execution on processor networks and a distributed version of GOLD™ is available for use on commercial grid systems; (g) it can handles metal atoms automatically if they are set up correctly in the protein input file; and (h) VHTS results can be analyzed and post processed easily with the companion programs SILVER™ or GoldMine™.



GOLD™ (Genetic Optimisation for Ligand Docking; http://www.ccdc.cam.ac.uk/products/life_sciences/gold/

GOLD's main limitations are that: (a) it uses a random method (i.e. a GA) instead of a systematic one to search for the ligand poses in the protein-ligand complex; (b) the default scoring functions have been optimized to predict the ligand poses not the binding affinities; (c) systematic problems in ranking very polar ligands and general ligands in large cavities have been reported; (d) the protein and ligand require a previous set up by the user that includes, for instance, the addition of all the hydrogen atoms; and (e) GOLD™ does not vary tautomeric or ionization states during docking and, therefore, if the user is not sure about the correct tautomeric state of a particular residue, then separate GOLD™ runs must be performed with all the possibilities.



Molegro Virtual Docker (Molecule & Allegro Virtual Docker; <http://www.molegro.com/>)

MVDTM (formerly called MolDock) is being developed by Molegro ApS (Aarhus, Denmark).

Molegro Virtual Docker (Molecule & Allegro Virtual Docker; <http://www.molegro.com/>)

MVD's main strengths are that it: (a) can automatically set up the input structures by assigning charges, bond orders and hybridization and by adding hydrogens; (b) is able to automatically predict potential binding sites in the receptor; (c) can deal with receptor sidechain flexibility by taking into account induced fit interactions; (d) is able to dock in precalculated energy grids (which speeds up the calculations); (e) can deal with user-defined constraints during docking; (f) can benefit from the use of templates (i.e. pharmacophores) during docking; and (g) is able to distribute the calculations on multiple computers.

Molegro Virtual Docker (Molecule & Allegro Virtual Docker; <http://www.molegro.com/>)

MVD's main limitations are that: (a) the stochastic nature of the docking engine implies that more than one docking run may be needed to identify the correct binding mode; and (b) the protonation state of the ligand/receptor pair is not exhaustively searched and prefixed protonation states are used during the docking.



AutoDock (<http://autodock.scripps.edu>).

AutoDock (formerly called AutoDoq since it used quaternions to perform rotations) was the first docking package to model the ligand with full conformational flexibility. It is being developed by Prof. Arthur J. Olson at the Department of Molecular Biology of the Scripps Research Institute in La Jolla (California, USA) and, at present, it is the most cited docking software in the bibliography. The package consists of two programs (i.e. AutoGrid and AutoDock), which are sequentially applied. AutoGrid is initially used to calculate the non-covalent energy of interaction between the rigid part of the receptor and a probe atom that is located at the various grid points of the lattice (where the probe atoms are those that are present in the ligands that will be docked onto the receptor).

AutoDock (<http://autodock.scripps.edu>).

AutoDock's main strengths are that: (a) it can deal with receptor flexibility during docking; (b) it can be used in blind-docking when the exact location of the ligand binding site is unknown; (c) it uses pre-calculated grid maps on a binding site in order to dock ligands with the consequent saving of computational time; (d) the atomic grid maps can also be used as a guide to design new ligands that bind with more affinity to the receptor; (e) it has a free-energy scoring function that is based on a linear regression analysis, the AMBER force field, and a large set of protein-ligand complexes with known inhibition constants; and (f) it provides good correlations between predicted inhibition constants and experimental ones.

AutoDock (<http://autodock.scripps.edu>).

AutoDock's main limitations are that: (a) the protein and the ligand need to be set up before being used by AutoGrid/AutoDock [e.g. hydrogens and partial charges have to be added, non-polar hydrogens and their charges have to be merged with their parent carbon atom, rotatable bonds in the ligand have to be set up, etc.; most of this work can be done with the AutoDockTools (<http://autodock.scripps.edu/downloads/resources/adt>)] ; (b) its different pose search algorithms are stochastic and not systematic; and (c) the protonation state of the ligand/receptor pair is not exhaustively searched and prefixed protonation states are used during the docking.

Glide™ (Grid-based Ligand Docking with Energetics; <http://www.schrodinger.com/>)

Glide™ is being developed by Schrödinger (Portland, Oregon, USA). Its main strengths are that: (a) it not only considers that the receptor is flexible but also that the receptor can change its conformation upon ligand binding due to an induced fit mechanism; (b) it can use userdefined constraints for restricting a ligand atom to lie within a certain region that is defined in relation to the features of the receptor that are responsible for ligand binding; (c) it can be used together with two more of Schrödinger's products (i.e. Liaison™ or Qsite™) to obtain more accurate binding energies for ligand-receptor pairs; (d) it can consider water molecules in the receptor's active site during docking or not; and (e) it has parallel processing or distribution options.



Glide™ (Grid-based Ligand Docking with Energetics; <http://www.schrodinger.com/>)

Glide's main limitations are that: (a) PDB receptors have to be manually prepared (i.e. atom types and bond orders must be assigned, the charge and protonation states must be corrected, side chains reoriented if necessary, incomplete side chains in the active site have to be rebuilt, and steric clashes relieved); and (b) the protonation state of the ligand/receptor pair is not exhaustively searched and prefixed protonation states have to be used during the docking.

Which Docking Software Gives Better Results?

The comparison of the predictive power and the performance of protein-ligand docking programs is currently an active area of research. This predictive power is usually evaluated by studying how the studied software performs relative to other docking programs in: (1) re-docking assays in which crystallographic protein-ligand complexes are split and the resulting ligand is docked into the resulting protein; (2) enrichment studies in which the software has to distinguish between true binders and decoys for a protein target in virtual screening studies; and (3) predicting binding free energies from the best-scored pose. Nevertheless, comparing the performance of docking programs is surprisingly complex and many factors have to be considered before the best one can be chosen. No program has yet been found that offers robust and accurate solutions of general applicability to a majority of the various docking problems.

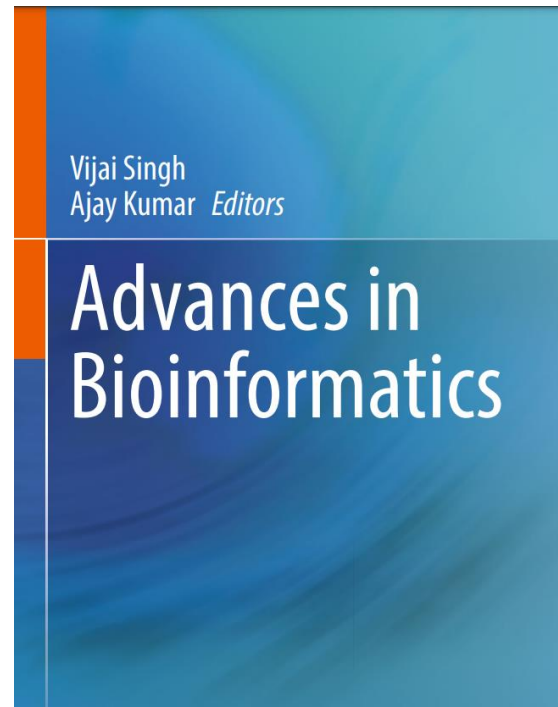
Which Docking Software Gives Better Results?

- (a) all the programs must be validated with the same set of protein complexes (i.e. the so called test set);
- (b) all the programs must search the same 3D space for the ligand pose (this is difficult because not all the programs define in the same way the search space where the ligand docking will be assayed and this is important for the success of the docking;
- (c) the correctness of the top-one poses must be carefully compared with the experimental solutions and not based on blind RMSD measures;
- (d) all ligands must be prepared in the same way; and
- (e) the time allowed to dock a ligand must be the same for all the programs (because one program may be better than another when fast settings are used but worse with slow settings).



Review Questions

- Give some examples of docking software.



Vijai Singh, Ajay Kumar (2021) *Advances in Bioinformatics*
Springer Nature Singapore Pte Ltd



Module 3. Bioinformatic tools

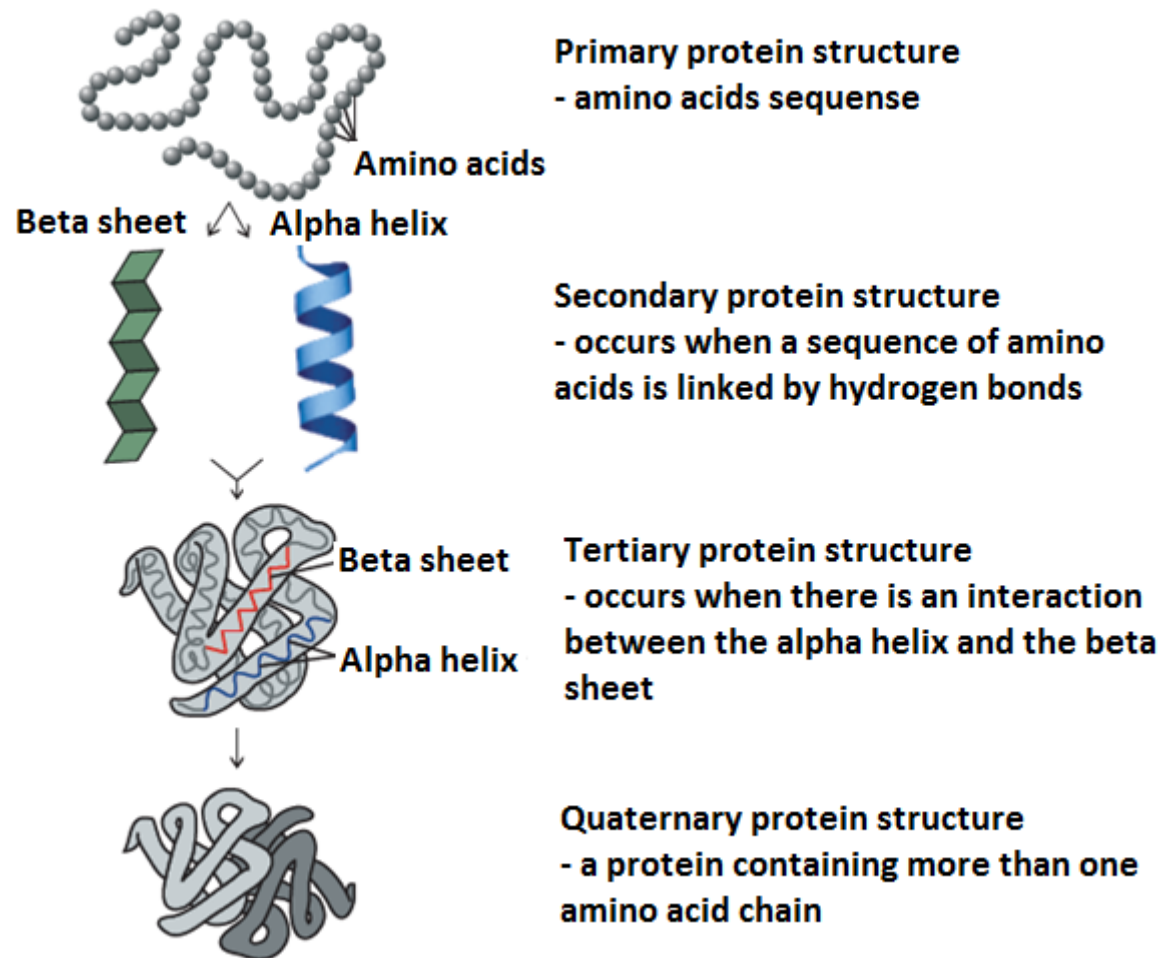
Topic 3. Protein Folding Problem

Lesson 1. Protein folding

Tertiary structure

- Proteins take a key place among the substances associated with life as we know it.
- Proteins have a complex spatial structure and perform a variety of biological functions - from typically structural, protective, transport to catalytic and regulatory.
- The functions of a protein depend on its tertiary structure, which in turn depends on the primary structure of the protein.
- The prediction of the tertiary structure of proteins from their amino acid sequence is one of the most important problems in the field of bioinformatics.

Tertiary structure



Homologous modelling

- Building a homologous model.
 - tertiary protein structure prediction techniques.
- Search for structures that have similar folds with no apparent similarity in protein sequence.
 - Takes the query sequence (protein with unknown structure) and runs it through the coordinates of the target protein (protein with known structure), by X-ray crystallography or MRI (Magnetic resonance imaging).
 - The sequence moves from position to position through the structure, obeying some predetermined physical constraints.
 - The lengths of the secondary structural elements and cyclic areas can be fixed or varied over a range.

Homologous modelling

- In each sequence-structure comparison pairs of hydrophobic interactions between nonlocal residues are determined.
 - These thermodynamic calculations are used to determine the most energetically favorable and conformationally stable alignment of the claimed sequence according to the target structure.
- Homology modelling programs are computationally intensive and require powerful computers.
 - Require knowledge of specialized computer languages.
 - The method is useful when protein sequence-based prediction methods fail to identify a suitable structure.

Homologous modelling

- Homology modeling techniques using the thread model are very powerful tools.
 - The requirements in terms of both hardware and expertise may prove to be a barrier for most biologists.
- There are easy-to-use programs that give the average biologist a good first approximation for comparative protein modeling (WHAT-IF and LOOK, provide advanced capabilities).
 - SWISS-MODEL - a program that performs automated sequence comparisons in two steps
 - SWISS-MODEL - програма, която извършва автоматизирани сравнения на последователности в две стъпки

Homologous modelling

SWISS-MODEL

1. It should be determined whether the sequence can be modeled at all.
 - SWISS-MODEL compares the submitted sequence with the ExPdb crystallographic database and attempts to model it only if there is a homolog of the requested sequence in ExPdb.
 - Selects template structures with at least 25% sequence identity in a region with more than 20 residues.
2. Given one or more suitable entries in ExPdb, atomic models are built and energy minimization is performed to generate the best model.
 - The atom coordinates for the resulting model as well as the structural alignments are returned as an email message.

Homologous modelling *SWISS-MODEL*

```

TARGET      1      QRRQ RTHFTSQQLQ QLEATFQRNR YPDMSTRERI AVWTNLTEAR
11FJL       0      KQRS RTTFSASQLD ELERAPERTQ YPDIYTREEL AQRTNLTEAR
21FJL       1      QRRS RTTFSASQLD ELERAPERTQ YPDIYTREEL AQRTNLTEAR
11B72       1      ARTFDWMKVL RTNFTTRQLT ELEKEPHFNK YLSRARRVEI AATLELNETQ
22HDD       1      KRP  RTAFSSRQLA RLKREPHENR YLTERRRQQL SSELGLINEAQ
12HOA       0      MRKRG RQTYTRYQTL ELEKEPHFNK YLTRRRRIEI AHALSLTERQ
               .  *  ..  *  *  *  *  *  *  ..  *  *  .
TARGET
11FJL                               hhhhhh hhhhhhhhhh hhhhhhhh hhhhh hhhh
21FJL                               hhhhhh hhhhhhhhhh hhhhhhhh hhhhh hhhh
11B72                               hhhhhh hhhhhhhhhh hhhhhhhh hhhhh hhhh
22HDD                               hhhhhh hhhhhhhhhh hhhhhhhh hhhhh hhhh
12HOA                               hhhhhh hhhhhhhhhh hhhhhhhh hhhh  hhhh
  
```



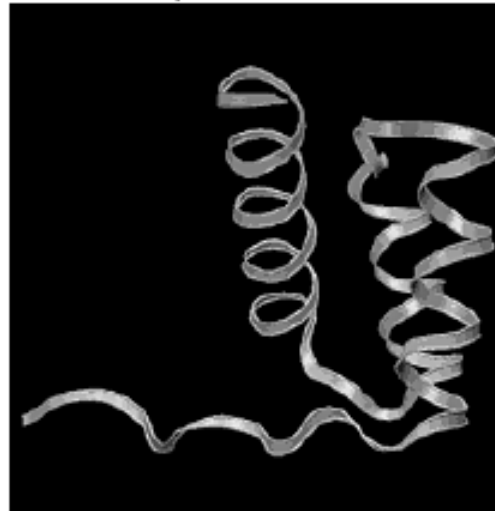
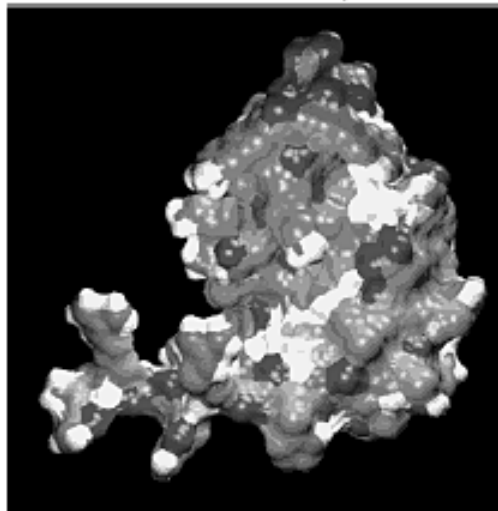
```

ATOM      1  H1  GLN      1      9.226 107.177 13.966 1.00 99.00
ATOM      2  H2  GLN      1     10.769 107.671 13.751 1.00 99.00
ATOM      3  N   GLN      1      9.824 107.785 13.444 1.00 25.00
ATOM      4  H3  GLN      1      9.549 108.738 13.592 1.00 99.00
ATOM      5  CA  GLN      1      9.728 107.473 11.999 1.00 25.00
ATOM      6  CB  GLN      1      8.265 107.520 11.538 1.00 25.00
ATOM      7  CG  GLN      1      7.468 106.270 11.932 1.00 25.00
ATOM      8  CD  GLN      1      8.001 104.970 11.312 1.00 25.00
ATOM      9  OE1 GLN      1      8.748 104.928 10.343 1.00 25.00
ATOM     10  NE2 GLN      1      7.629 103.853 11.899 1.00 25.00
ATOM     11  HE21GLN 1      7.979 103.008 11.502 1.00 99.00
ATOM     12  HE22GLN 1      7.015 103.860 12.683 1.00 99.00
  
```

The input sequence is
the homeodomain area
of human PITX2 protein

Homologous modelling *SWISS-MODEL*

| | | | | | | | | | |
|------|----|------|-----|---|--------|---------|--------|------|-------|
| ATOM | 1 | H1 | GLN | 1 | 9.226 | 107.177 | 13.966 | 1.00 | 99.00 |
| ATOM | 2 | H2 | GLN | 1 | 10.769 | 107.671 | 13.751 | 1.00 | 99.00 |
| ATOM | 3 | N | GLN | 1 | 9.824 | 107.785 | 13.444 | 1.00 | 25.00 |
| ATOM | 4 | H3 | GLN | 1 | 9.549 | 108.738 | 13.592 | 1.00 | 99.00 |
| ATOM | 5 | CA | GLN | 1 | 9.728 | 107.473 | 11.999 | 1.00 | 25.00 |
| ATOM | 6 | CB | GLN | 1 | 8.265 | 107.520 | 11.538 | 1.00 | 25.00 |
| ATOM | 7 | CG | GLN | 1 | 7.468 | 106.270 | 11.932 | 1.00 | 25.00 |
| ATOM | 8 | CD | GLN | 1 | 8.001 | 104.970 | 11.312 | 1.00 | 25.00 |
| ATOM | 9 | OE1 | GLN | 1 | 8.748 | 104.928 | 10.343 | 1.00 | 25.00 |
| ATOM | 10 | NB2 | GLN | 1 | 7.629 | 103.853 | 11.899 | 1.00 | 25.00 |
| ATOM | 11 | HE21 | GLN | 1 | 7.979 | 103.008 | 11.502 | 1.00 | 99.00 |
| ATOM | 12 | HE22 | GLN | 1 | 7.015 | 103.860 | 12.683 | 1.00 | 99.00 |



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Homologous modelling

UCLA

- Another automated method to predict protein folding includes:
 1. Finding sequence-based matches to assign a probable fold to the requested sequence.
 2. Additional prediction of secondary structural information for the requested sequence.
- The correct fold assignment depends on the scored results obtained for the requested sequence.
 - These results are based on the compatibility of the requested sequence with each of the structures in a library of target three-dimensional structures.

Homologous modelling

UCLA

- Further prediction of the secondary structure improves the folding distribution by about 25%.
- The input data for this method is a protein sequence submitted through the web interface.
- A web page containing the results is opened.
 - Results are physically stored on the UCLA server for future reference.

Homologous modelling

DALI

- Compares structures with structures.
- The algorithm searches for similar contact patterns between two proteins, performs an optimization, and returns the best set of solutions to align the structures of the query proteins (Holm and Sander, 1993).
 - Gaps can be of any length and allow alternative connections between aligned segments, thus facilitating the identification of specific areas that are similar in two different proteins, even if the proteins are generally different.

Homologous modelling

DALI

- The web-based DALI interface performed the analysis on two sets of coordinates, using a user-supplied set of coordinates in PDB format.
- If the requested proteins are available in the PDB, then their previously calculated structural neighbors can be found by accessing the FSSP database
 - The FSSP contains structurally aligned families of folded proteins (Holm and Sander, 1994) obtained by an all-against-all comparison of PDB entries.

Homologous modelling

TOPITS

- TOPITS (Rost, 1995)
 1. Creates a searchable database by transforming the three-dimensional structure of PDB proteins into one-dimensional "strings" of secondary structure.
 2. The secondary structure and solvent accessibility for the requested sequence are determined by the PHD method (results are stored as a one-dimensional string).
 3. Both the query and target strings are aligned by dynamic programming to predict the structure of the query protein.
- The results are returned as a ranked list showing the optimal alignment of the requested sequence against the target structure.
 - It also contains a probability score (Z-score) for the accuracy of the prediction.

Protein folding problem

- A large number of scientists are trying to develop mathematical models and algorithms to solve the so-called "Protein Folding Problem".
 - To predict the three-dimensional structure of proteins from basic amino acid sequence information.
- In recent years, scientists have made great strides in developing methods to predict the tertiary structure of proteins using computer.
- Current bioinformatics-based methods are able to predict folding of small globular proteins, which are homologous proteins with known structure.

Protein folding problem

Protein Structure Prediction



Christian Anfinsen, 1961:

denatured RNase refolds into functional state (in vitro)

⇒ no external folding machinery

⇒ Anfinsen's dogma/thermodynamic hypothesis:

all information about native structure is in the sequence

(at least for small globular proteins)

native structure = minimum of the free energy

- unique
- stable
- kinetically accessible

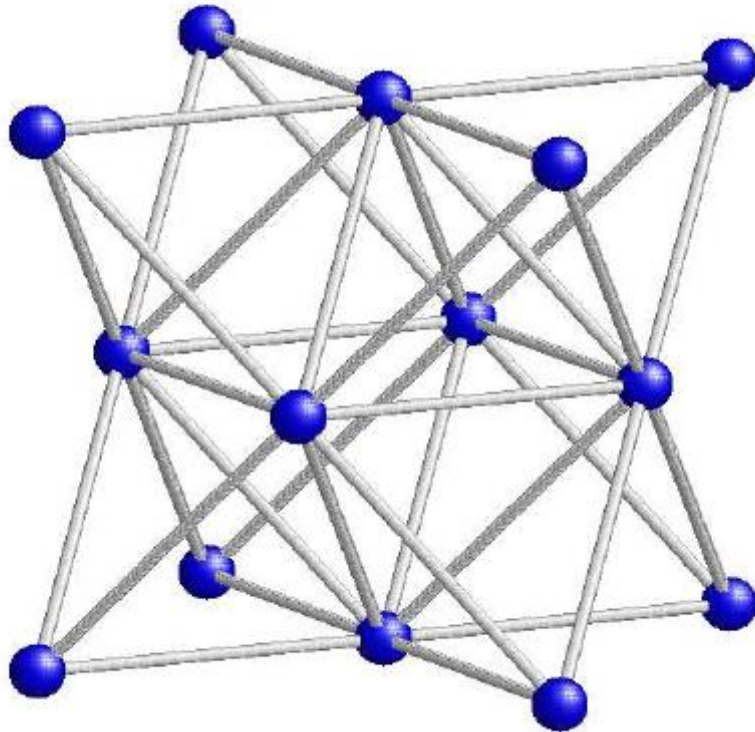
Protein folding problem

- The protein folding problem can be divided into three distinct subproblems:
 1. **folding code** - the question "*How is the natural structure of the protein derived from the intra-atomic forces acting on the amino acid sequence?*" is thermodynamic.
 - The protein sequence must have a uniquely folded conformation that is stable and consistent with the native structure
 2. **protein structure prediction** - a computational problem of how to predict the native structure of a protein from its amino acid sequence;
 3. **folding rate (Levitall's paradox)** - how can the protein fold quickly.
- *It has been proven that even in simplified models, the protein folding problem in 2D and 3D is NP-complete.*

Protein folding problem

Off-lattice model

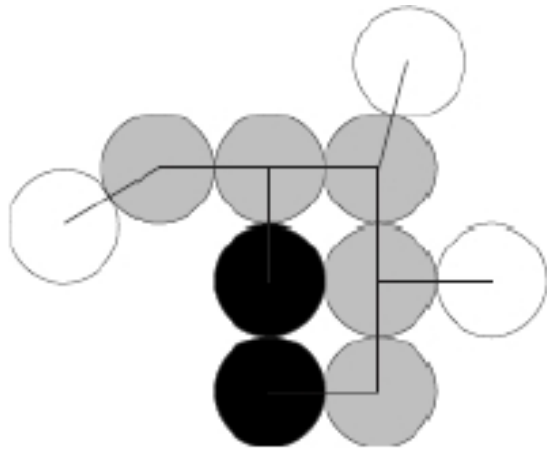
$$\text{FCC} = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{Z}^3 \mid x + y + z \text{ even} \right\}$$



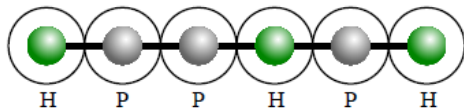
- One of the important questions related to the study of protein folding models is whether **algorithms for grid models** can be generalized to algorithms for **off-grid models**.
 - In 1611. Kepler formulated a hypothesis stating that the highest packing density of identical spheres is obtained by arranging the spheres in a Face-Center-Cubic (FCC) grid.

Protein folding problem

Off-lattice model

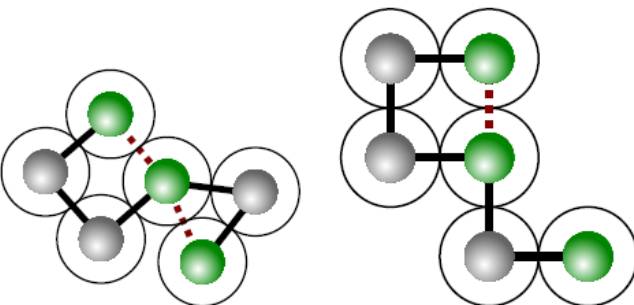


- In 1997, Thomas Hales showed that the densest off-lattice arrangement of amino acids could be made in the FCC grid.
- In 1997, Hart and Istrail introduce an off-lattice model called the *tangent-spherical side-chain HP model*.



- adjacent base and side-chain molecules are represented by identically sized tangent spheres in 3D space.

- Side chains are denoted as hydrophobic (H) or hydrophilic (P), and the conformational energy is the number of H-H tangent spheres.



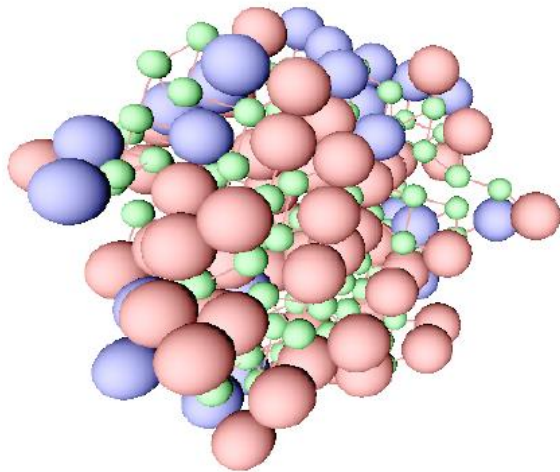
Protein folding problem

Off-lattice model

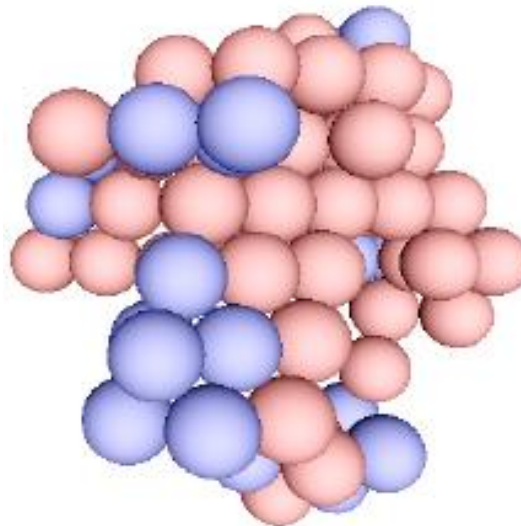
- The approximate algorithm for the side-chain model in the FCC lattice can be inserted into the off-lattice framework and thus provide an off-lattice performance guarantee that is close to optimal (86% of optimal).
- The number of neighbors in the FCC grid and the maximum number of possible neighbors in the space is the same. This suggests that the FCC grid algorithms have good off-lattice performance.
- The following figures show folding in a 2D hexagonal (triangular) lattice, a 3D linear-chain cubic lattice, and a 3D side-chain FCC lattice.

Protein folding problem

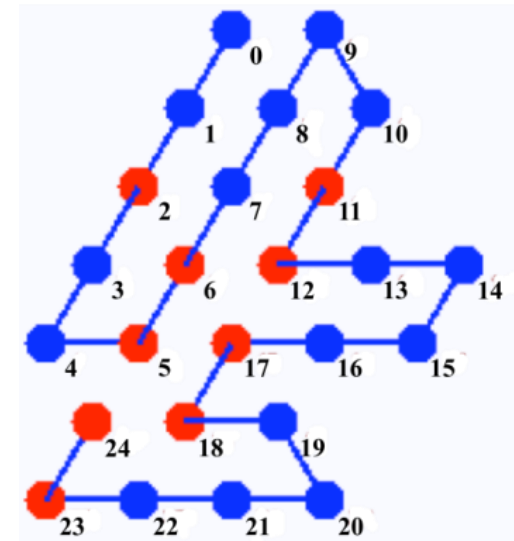
Off-lattice model



*Folding in side-chain model
of FCC lattice*



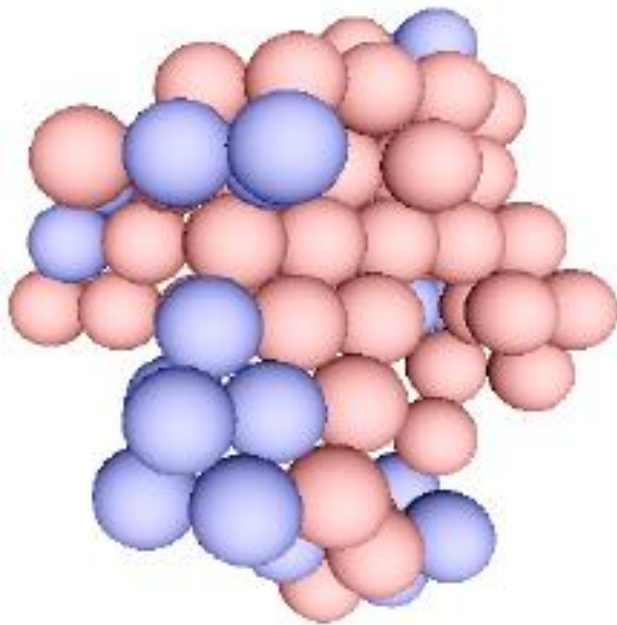
*Folding an 85 length
protein into a 3D FCC
lattice*



*Folding in 2D hexagonal
(triangular) lattice for 25
length protein*

Protein folding problem

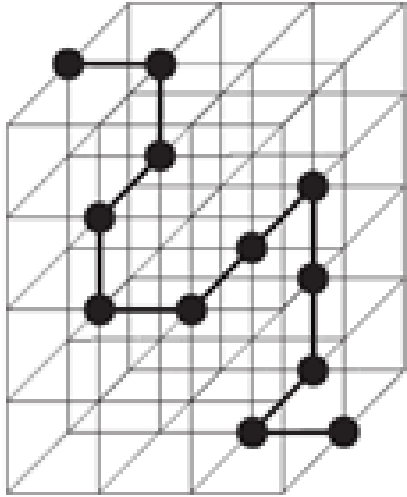
Lattice models



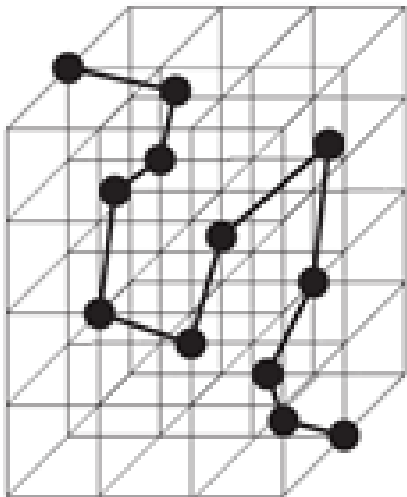
- There are two types of lattice models:
 - The first type describes the basic physical forces that are involved in the protein folding process.
 - The second type was designed to create a realistic folding of real proteins and to parameterize the use of proteins as templates for existing structures.
- In three-dimensional space, the Face-Centered-Cubic (FCC) lattice is very important.
 - In the FCC lattice, the densest packing of identical spheres in three-dimensional space can be achieved.

Protein folding problem

Lattice models



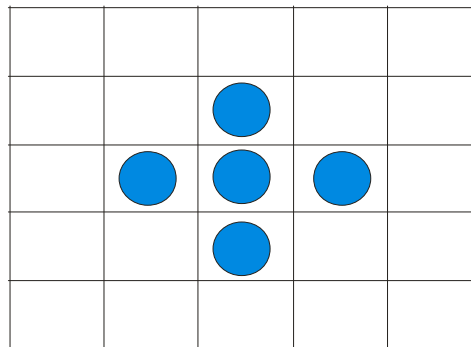
- The lattice model for the protein folding problem places each amino acid at a point on a 2D quadrant or 3D cubic lattice
 - adjacent amino acids in the protein sequence must occupy adjacent points in the lattice in order to maintain protein connectivity.
- The lattice model gives:
 - equal amino acid sizes;
 - equal bond lengths between all pairs of amino acids;
 - orthogonal or linear angle, for all three consecutive amino acids.



Protein folding problem

Lattice models – problem definition

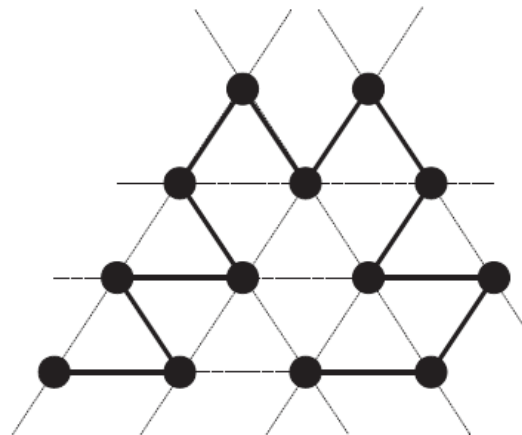
- The purpose is to place all of the amino acids of the protein in a 2D and 3D lattice as:
 - Adjacent amino acids in the protein sequence must occupy adjacent cells in the lattice;
 - We can only put one amino acid in each lattice cell;
 - Each amino acid can occupy exactly one cell of the lattice.



Protein folding problem

Lattice models – problem definition

- Due to the property of the square or cubic lattice model, two amino acids that are an even length apart cannot be adjacent, which is not necessarily true in the real world.
 - Therefore, several different lattice models have been introduced, such as the triangular lattice model and the diagonal lattice model.



Protein folding problem

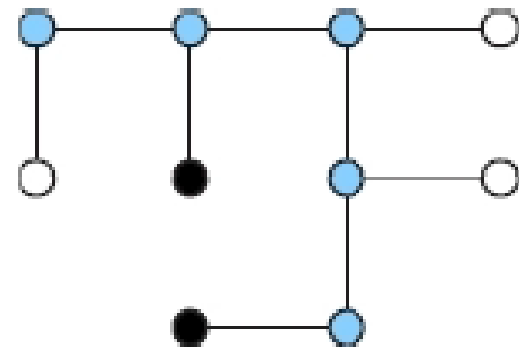
Lattice models

Line-chain

- Hart and Istrail's algorithm as well as Newman's algorithm are approximate algorithms for linear-chain protein folding models.
- Also, Hart and Eastrail develop a basic approximation algorithm as a general method for protein folding in HP models.
 - This algorithm can be referred to a large class of lattice models.

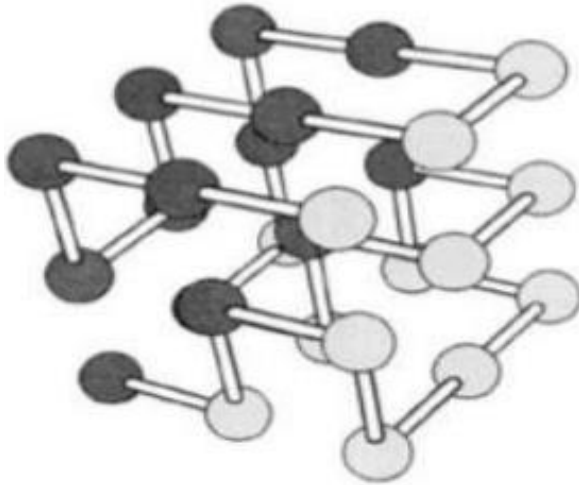
Side-chain

- In order to improve the accuracy of protein structure prediction methods, extended models consider the protein as a core that is formed by multiple peptide bonds connected by side chains.



Protein folding problem

HP Lattice model



- In 1985, Ken Dill proposed the Hydrophobic-Hydrophilic (HP) model
 - The HP model is based on the observation that **hydrophobic bonds between amino acids drive protein folding processes.**
- The hydrophobic effect of amino acids takes a significant portion of the total energy function, so this is the most important force in determining protein structure.

Protein folding problem

HP Lattice models

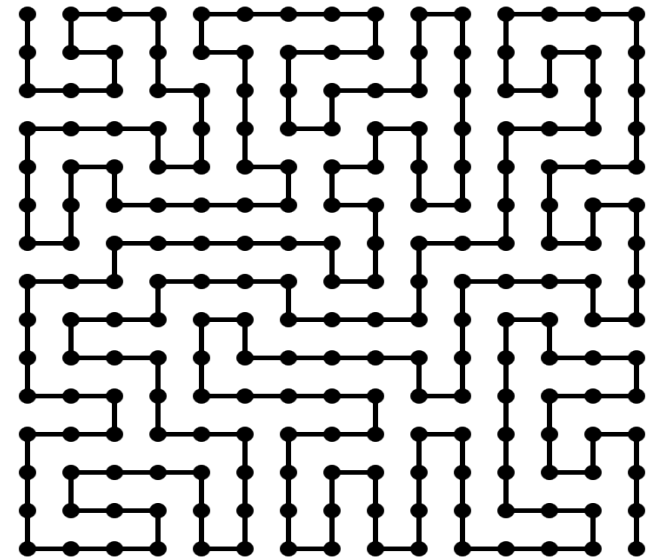
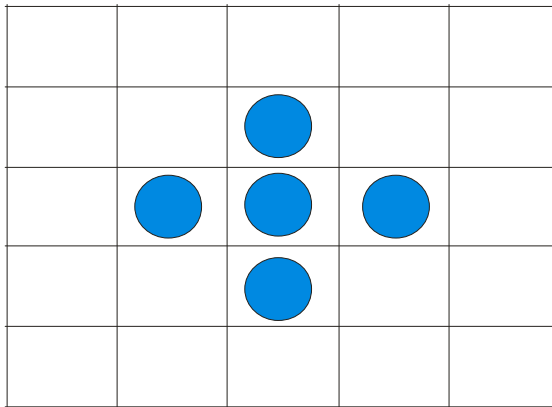
| Name | Symbol | Classification | Name | Symbol | Classification |
|------------------|--------|----------------|---------------|--------|----------------|
| Alanine | A | Hydrophobic | Leucine | L | Hydrophobic |
| Arginine | R | Polar | Lysine | K | Polar |
| Asparagine | N | Polar | Methionine | M | Hydrophobic |
| Aspartic Acid | D | Polar | Phenylalanine | F | Hydrophobic |
| Cysteine | C | Polar | Proline | P | Hydrophobic |
| Glutamic Acid | E | Polar | Serine | S | Polar |
| Glutamine | Q | Polar | Threonine | T | Polar |
| Glycine | G | Polar | Tryptophan | W | Hydrophobic |
| Histidine | H | Polar | Tyrosine | Y | Polar |
| Isoleucine | I | Hydrophobic | Valine | V | Hydrophobic |



Protein folding problem

HP Lattice models

- A self-avoiding walk is a path in a graph from one point (vertex) to another that does not pass through the same point (vertex) more than once.



Protein folding problem

HP Lattice models

- The number of Self-avoiding paths between two diagonal vertices in a lattice sized an $n \times n$ is:

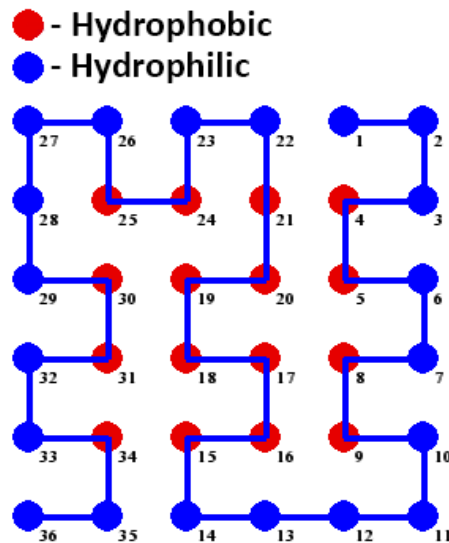
| $N \times N$ | Броя пътища |
|--------------|--|
| 1 | 2 |
| 2 | 12 |
| 3 | 184 |
| 4 | 184 |
| 5 | 1262816 |
| 6 | 575780564 |
| 7 | 789360053252 |
| 8 | 3266598486981642 |
| 9 | 41044208702632496804 |
| 10 | 1568758030464750013214100 |
| 11 | 182413291514248049241470885236 |
| 12 | 64528039343270018963357185158482118 |
| 13 | 69450664761521361664274701548907358996488 |
| 14 | 227449714676812739631826459327989863387613323440 |
| 15 | 2266745568862672746374567396713098934866324885408319028 |
| 16 | 68745445609149931587631563132489232824587945968099457285419306 |
| 17 | 6344814611237963971310297540795524400449443986866480693646369387855336 |
| 18 | 1782112840842065129893384946652325275167838065704767655931452474605826692782532 |
| 19 | 1523344971704879993080742810319229690899454255323294555776029866737355060592877569255844 |
| 20 | 10^{97} |
| 21 | 10^{107} |



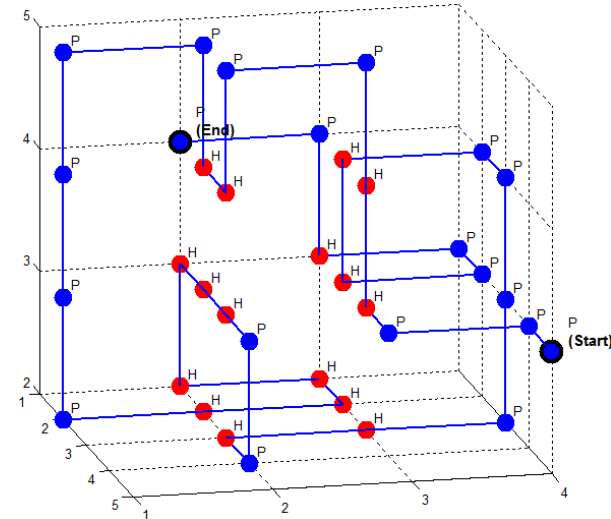
Protein folding problem

HP Lattice models

- The optimal conformation when folding a protein in the HP model is the conformation that has the maximum number of contacts between the hydrophobic amino acids, which gives the lowest energy value of the protein.



14 contacts



18 contacts

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Protein folding problem

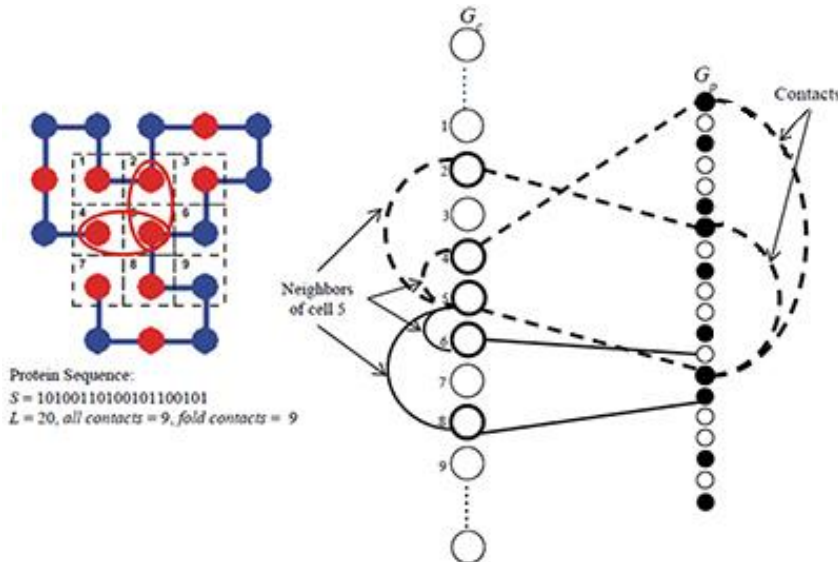
Algorithms to solve the problem

- The combinatorial methods used to solve the protein folding problem are approximate algorithms and combinatorial optimization algorithms.
 - The approximate algorithms are polynomial-time algorithms that, for each protein sequence, generate a folding for which the number of contacts is nearly optimal.
 - Combinatorial optimization algorithms are exponential algorithms that, for some protein sequences, generate a folding for which the number of contacts can be proved to be optimal.
- Heuristic algorithms are the third category of combinatorial methods.

Protein folding problem

Algorithms to solve the problem

- Key elements in solving the problem are:
 - A sublattice;
 - HP sequence.
- A graph problem that can be solved, either by integer optimization or by Graph Theory.
- In the 2D case the sublattice is a square, and in the 3D case it is a cube with vertices colored black (the set V_b) and white (the set V_w).



Protein folding problem

Algorithms to solve the problem in 2D

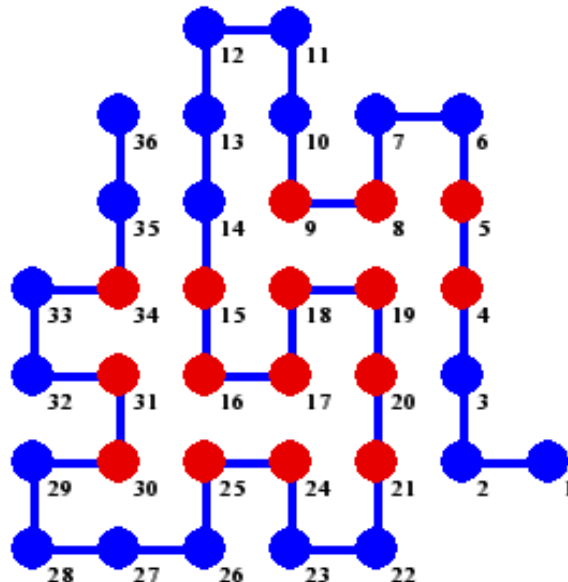
- We divide the sequence S into segments of a predefined size.
- We take the i -th part of S and generate all possible folds. We select the fold with the maximum number of contacts and place it on the square grid.
- Add $i+1$ part of S to the already obtained fold, find all possible folds with respect to the already chosen fold.
- From the new obtained folds, select the one with the maximum number of contacts and place it on the square grid.

Protein folding problem

Algorithms to solve the problem in 2D

- The algorithm stops in two cases:
 - All possible folds have been exhausted;
 - We have found a folding at which the number of contacts is maximal ($C_{2D}(S) = 2 \min\{|ODD|, |EVEN|\}$).

#Contacts: 13



*Protein folding with length 36
amino acids, constructed from
heuristic algorithm*

Protein folding problem

Algorithms to solve the problem in 2D

Algorithm GREEDYLIKE:

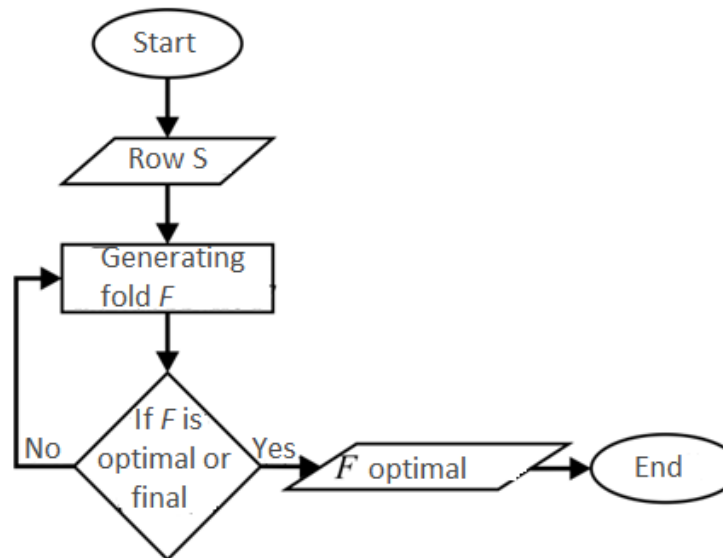
```

1: for  $i$  in  $\{1, 2, \dots, k\}$  do
2:    $rec := 0$ 
3:    $PATHFINDER(i, 0)$ 
4: end for
def  $PATHFINDER(i, m)$   / Generates all substrings with a defined length
5: if  $m > len(S_i)$  then
6:   if  $cont(S_{i,m}) > rec$  then           / * cont() counts the contacts for
7:      $rec := cont(S_{i,m})$                  the obtained conformation. */
8:     Saves  $S_{i,m}$ 
9:   end if
10:  return
11: end if
12: for  $a$  in  $\{s, r, l\}$  do
13:   $st[m] := a$ 
14:  if  $check(S_{i,m})$  then           / * check() checks whether the resulting
15:     $PATHFINDER(i, m+1)$          conformation is a self-avoiding path. */
16:  end if
17: end for
  
```

Protein folding problem

Algorithms to solve the problem in 2D

- Each fold starts with 2 amino acids fixed in the center of a square lattice.
- From each fixed cell you can go left (-1), right (+1), down (-N) and up (+N).



Protein folding problem

Algorithms to solve the problem in 3D

- The main components of the algorithms described below are:
 - Creating the model - create a *.lp* file for the model
 - MIP (Mixed Integer Programming) Solver as CPLEX or GUROBI
- The common input is a cubic sub-lattice $d_1 \geq x \geq 0, d_2 \geq y \geq 0, d_3 \geq z \geq 0$ connected with the length n of the input HP sequence S . For the **HE-1** and **HE-2** heuristic algorithms, we divide S into segments of a predefined length, i. e. , $S = S_1 \cup S_2 \cup \dots \cup S_k, S_i \cap S_{i+1} = \emptyset$.
- **Exact Algorithm - EA:**
 - If** $\text{length}(S) < t$ **then**
 - Step 1: Create the model.
 - Step 2: Solve the model using MIP Solver.
 - End if**

Protein folding problem

Algorithms to solve the problem in 3D

- Heuristic Algorithm-1 – HE-1

For $i = 1, 2, \dots, k$ **do:**

If $i = 1$ **then**

We create and solve the model for the input S_i and $z=i$ (2D sub-lattice)

Else:

Let $v_j^* = 1$ for $j \in J$ in the optimal solution for S_{i-1} .

Step 1: Create the model for the input S_i and $z = i$ (2D sub-lattice).

Add $\sum v_j$ to the target function (1). To preserve connectivity we add $x_{s1} = 1$ to the constraint set, where s is the row number (in the aligned graph or just the coordinates) for the last symbol of S_{i-1} in the optimal solution.

Step 2: Solve the model.

End if

End for

Protein folding problem

Алгоритми за решаване на проблема в 3D

- Heuristic Algorithm-2 – HE-2

For $i = 1, 2, \dots, k$ **do:**

If $i = 1$ **then**

Create and solve the model for input S_i .

Else:

Step 1: Create the model for input $T_i = S_1 + S_2 + \dots + S_i$.
For each $x_{ik} = 1$ for the optimal solution for input T_{i-1}
add (to the set of constraints for input T_i) $x_{ik} = 1$.

Step 2: Solve the model.

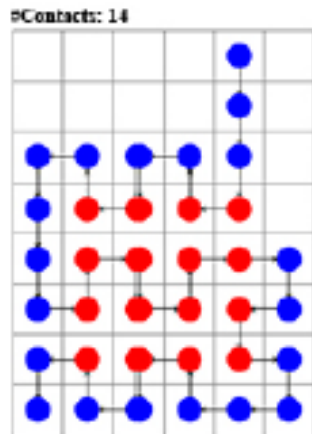
End if

End for

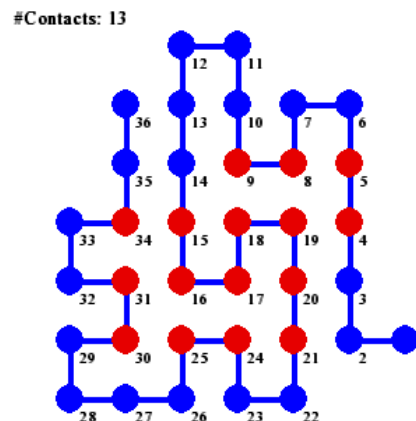
Protein folding problem

Results

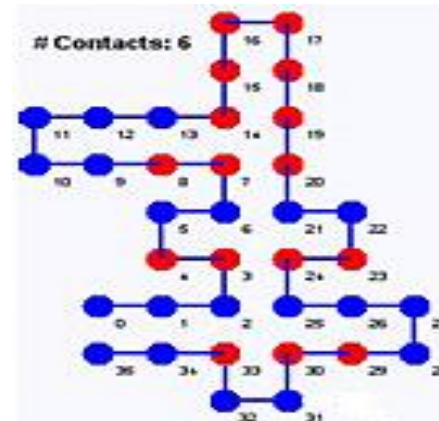
- For a test sample, we use a small human protein with 36 amino acids. The amino acid sequence for this protein is PPPHHPPHHPPPPHHHHHHHPPHHPPPPHHPPHPP :



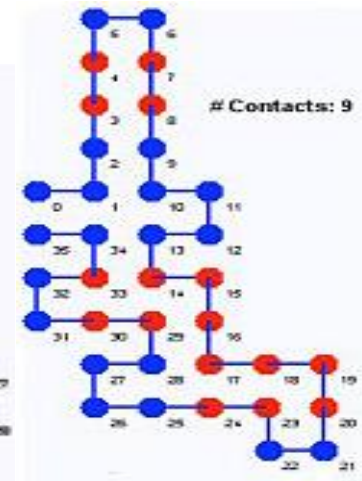
Optimal solution



Heuristic algorithm



Hart –Istrail algorithm



Newman algorithm

Protein folding problem

Results

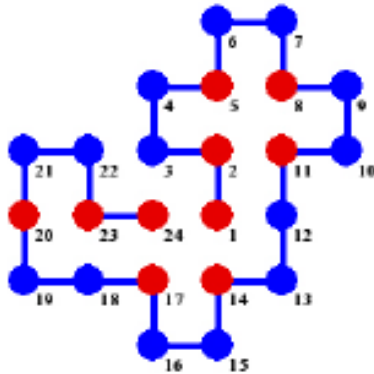
- Computational experiments using proteins of different lengths have been implemented.

| Number | HP Sequence | Length |
|--------|---|--------|
| 1 | HPHPPHHPPHPPHPPHPPHPPH | 20 |
| 2 | HHPHPPHPPHPPHPPHPPHPPHPPH | 24 |
| 3 | PPHPPHHPPPPHHPPPPHHPPPPHH | 25 |
| 4 | PPPHHPPHHPPPPPPHHHHHHHHPPHHPPPPPHHPPHPP | 36 |
| 5 | PPHHHPPHHHPPPPHPPHHPPHHPPHHHHHPPHPPHHHHHPPHPPHHPPHHPP | 46 |
| 6 | PPHPPHHPPHHPPPPPPHHHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHHH | 48 |
| 7 | HHPHPPHPPHHHHPPHPPPPHPPPPHPPPPHPPPPHPPHHHHPPHPPHPPH H | 50 |
| 8 | P(PH ₃) ₂ H ₅ P ₃ H ₁₀ PHP ₃ H ₁₂ P ₄ H ₆ PH ₂ PHP | 60 |
| 9 | H ₁₂ PHPPH ₂ H ₂ P ₂ H ₂ P ₂ HP ₂ H ₂ P ₂ H ₂ P ₂ HP ₂ H ₂ P ₂ H ₂ P ₂ HPHPPH ₁₂ | 64 |
| 10 | PH ₂ P ₅ H ₂ P ₂ H ₂ PHP ₂ HP ₇ HP ₃ H ₂ PH ₂ P ₆ HP ₂ HPHPP ₂ HP ₅ H ₃ P ₄ H ₂ PH ₂ P ₅ H ₂ P ₄ H ₄ P HP ₈ H ₅ P ₂ HP ₂ | 102 |

Protein folding problem

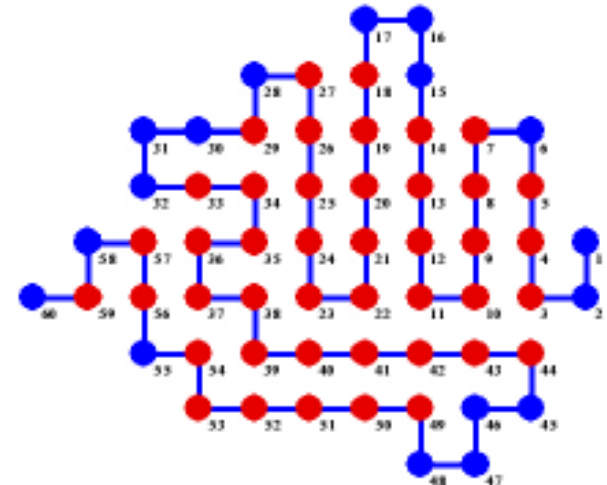
Results

#Contacts: 9



- Folding a protein of length 24.
The number of contacts is 9:
HHPPHPPHPPHPPHPPHPPHPPHH

#Contacts: 35

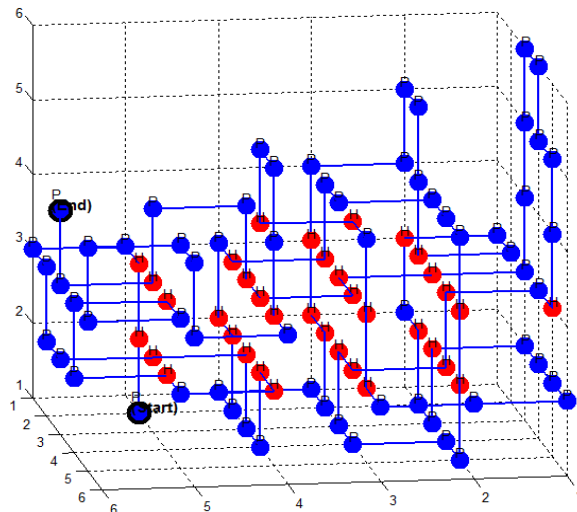


- Folding a protein of length 60
The number of contacts is 35:
 $P(PH_3)_2H_5P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$

Protein folding problem

Results

- The figure below shows the solution for a protein of length 102 amino acids in analytical form, where x/x^- , y/y^- or z/z^- represent plus/minus one to the corresponding coordinate of the preceding amino acid and in graphical form.



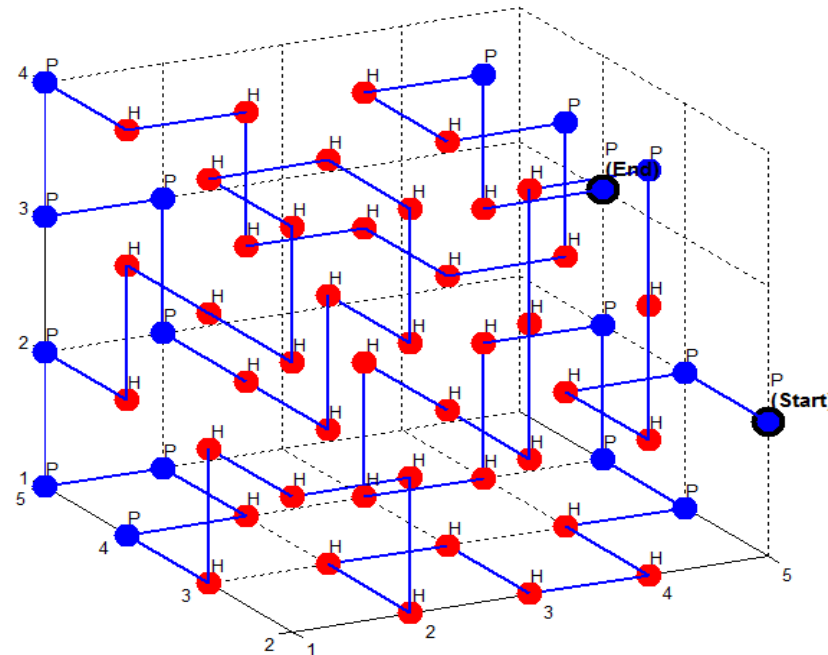
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| P | H | H | P | P | P | P | P | H | H | P | P | H | H | P | H | P | P | H | P | P | P | P | P | P | P | H | P | P | P |
| | | | y | x | y | z | y | x | x | z | y | z | y | x | y | x | z | x | y | x | z | x | y | z | x | y | z | z | y |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| H | H | P | H | H | P | P | P | P | P | P | H | P | P | H | P | H | P | P | H | P | P | P | P | P | H | H | H | P | P |
| | | x | y | z | z | y | x | x | y | y | x | z | y | z | y | z | x | y | x | z | z | y | z | x | z | z | y | z | y |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| P | P | H | H | P | H | H | P | P | P | P | P | H | H | P | P | P | P | H | H | H | H | P | H | P | P | P | P | P | P |
| | | | y | z | y | x | z | y | x | y | y | z | z | z | y | x | y | z | y | x | z | x | y | z | z | y | z | y | z |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | | | | | | | | | | | | | | | | | | |
| P | P | H | H | H | H | H | P | P | H | P | P | | | | | | | | | | | | | | | | | | |
| | | z | z | x | x | y | z | x | z | x | z | | | | | | | | | | | | | | | | | | |

Protein folding problem

Results

60 amino acids

PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHHH
HHHHHHHHHPPPPHHHHHHHHPHHPHP



ERASMUS+

Thank you for your attention!



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness



Module 3. Bioinformatic tools

Topic 3. Protein Folding Problem

Lesson 2. Sequence Alignment

Sequence pairwise alignment

- In this lecture we will show some of the methods for aligning two protein, DNA or RNA sequences:
 - Are two sequences homologous?
 - Do given sequences have common domains or regions?
 - Determine the exact location of common features, such as disulfide bridges or catalytically active sites.
 - Comparison of two genes and/or their products.
- Typically, biologists use Pairwise alignment techniques for protein, DNA or RNA to refine results obtained from a specific database search and perform detailed analysis.

Sequence pairwise alignment

- The most delicate point is the selection of the sequences themselves for comparison.
- Do not use pairwise alignment methods to find a sequence that might be homologous to a sequence you already have.
 - Database search programs simply compare the queried sequence to all sequences in the database, so there is no real need to do further pairwise comparisons.
 - Programs such as BLAST search databases using methods that are optimized for search speed rather than alignment accuracy.

Selection of appropriate sequences

- The reason for aligning two sequences is the assumption that these sequences are homologous and have a common origin.
 - They often have similar 3D structures and functions.
- Select sequences according to the following (conservative) criteria:
 - DNA Sequence: **At least 70% identity** along more than 100 bases between the requested and searched sequence or presence of an E-value lower than 10^{-4} .
 - Protein Sequence: **More than 25% identity** in length of more than 100 amino acids between the requested and the searched sequence or presence of an E-value lower than 10^{-4} .

Selection of appropriate sequences

- Searching a database is like searching "new apartment listings" 😊
- Choosing a sequence from a BLAST result is like making a compromise between new information and security.
 - The desired object must be a broadly annotated Swiss-Prot sequence that satisfies the query with E-value much lower than 10^{-4} .
 - One possibility is to **select your sequences from the result obtained from the database.**
 - Another possibility is to **select two sequences using experimental criteria**, e. g. to have the same function or other similar properties.
 - **For this purpose, you can use the Sequence Retrieval System (SRS), which allows you to identify sequences by keywords.**

Selection of appropriate sequences

- If you have only one sequence, it is possible (and sometimes advisable) to do the alignment with the sequence itself.
 - Leads to the discovery of interesting properties:
 - Repeated domains;
 - Regions with a repeated small motif (low level of complexity);
 - Palindromes - DNA fragments that repeat in opposite directions;
 - Potential secondary structures in RNA.

Choosing an appropriate alignment method

- One of the most difficult tasks for biologists is the sequence alignment.

| Method name | Situation |
|------------------|---|
| Dot plot | General survey of your sequence. Repeat detection. Finding long insertion/deletions. Extraction of parts of sequences to perform multiple alignment. |
| Local alignment | Comparison of sequences with partial homology. Comparing parts of a sequence. Making quality alignments. Performing residual versus residual type analyses. |
| Global alignment | Comparing two sequences along their entire length. Identification of long insertions/deletions. Data Verification. Identify each mutation in the sequence. |

Choosing an appropriate alignment method

- Dot plot are very powerful tools that give an instant overall picture in a single alignment.
 - An ideal tool for deciding the next step in your analysis.
 - Dot plot is not sufficient for a more accurate study.
 - Dot plot doesn't really conduct alinement.
 - Gives general instructions.
 - It doesn't clearly show how amino acids or nucleotides correspond to each other between the two sequences.
 - In order to extract this information, an alignment must be made.

Choosing an appropriate alignment method

- There are two types of alignment:
 - Global alignment - the two sequences are compared along their entire length.
 - Local alignment - only the most similar parts of the two sequences are compared.
- A global alignment is only performed if the two sequences are linked along their entire length and if they do not contain very long insertions or deletions.
- If you don't know how to choose between global and local alignment, it is always better to use local methods.
- When comparing two sequences, start with a Dot plot, comparing each sequence to itself.

Dot plot alignment

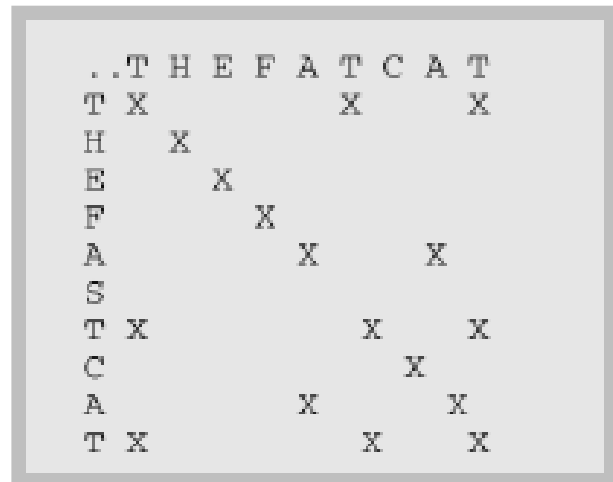
- **Example 1.** Imagine you want to compare the following two sequences:
 - Sequence 1: THEFASTCAT
 - Sequence 2: THEFATCAT
- To make a dot plot take a piece of paper, draw a grid and:
 1. Write the individual letters of sequence 1 one above the other (**in a column**).
 2. Write sequence 2 from left to right (**in a row**).



Dot plot alignment

- **Example 1.**

3. Examine each cell one by one and cross it out if the top sequence and the corresponding one in the row contain the same symbol.



4. Longest diagonal wins! - Shows segments of similarity between the two sequences.

Dot plot alignment

- Dot plot techniques are closely related to sliding window methods.
 - The window is part of a sequence that is compared to part of another sequence of similar size.
- Dot plot has two main advantages:
 - It is easy to implement and no biological hypothesis is required;
 - It's hard to make a mistake.
- Dot plot interpretation relies on the finest detection device, exactly the eyes.

Dotlet alignment

- Dotlet is very friendly because:
 - Easy to use;
 - Free of charge;
 - No installation required;
 - Works on (almost) any computer that has Internet access.
- Dotlet is ideal for sequences with length less than 10,000 amino acids or nucleotides.
 - If you want to browse longer sequences using online tools, then you will need to use Dnadot, which is a tool designed for Dot plot between long sequences.
- If your sequences are longer than 100,000 characters, there is no suitable online tool available.

Dotlet alignment

- Different Dot plot programs

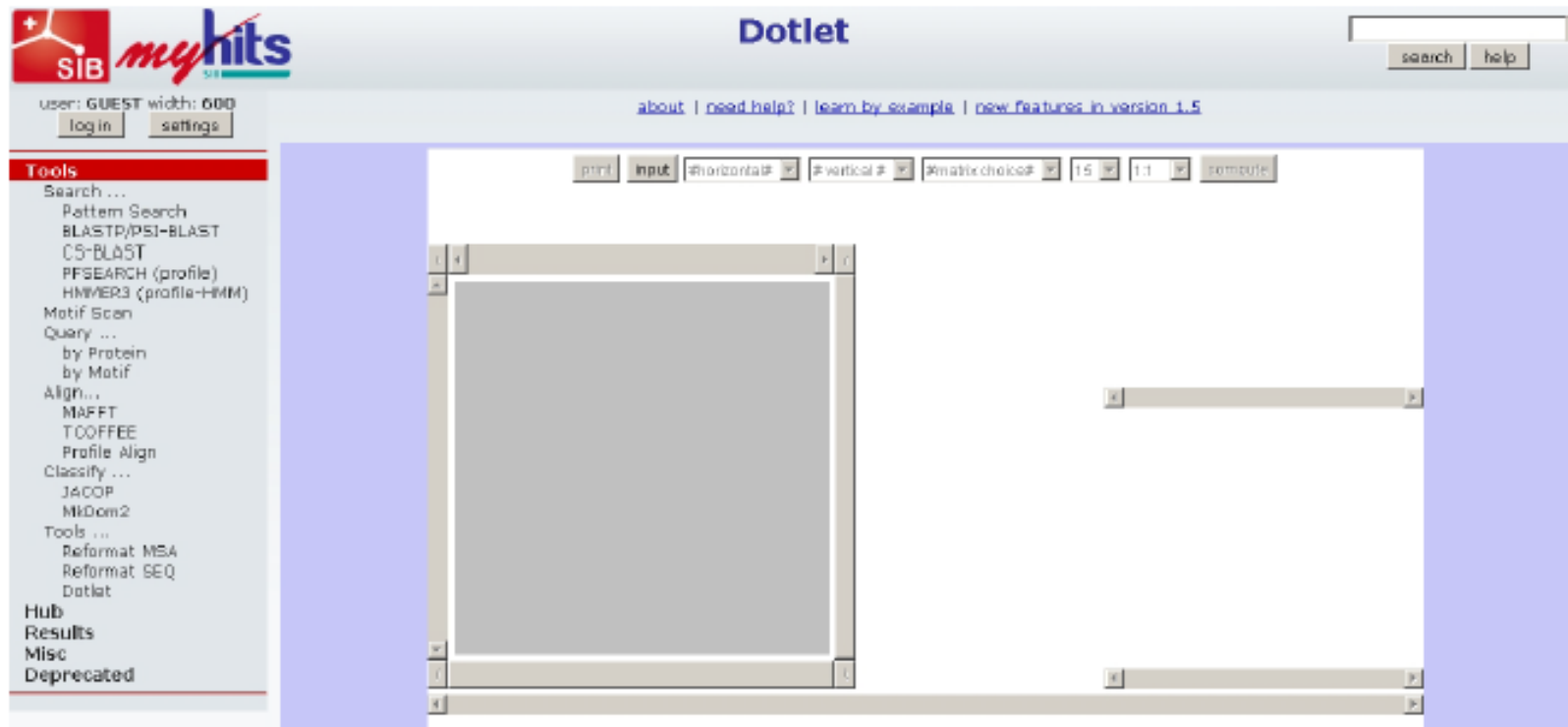
| Name | Used for: | Range: | URL: | Platform: |
|--------|-----------------------|----------|---|-------------------------|
| Dotlet | proteins, DNA | 10000 | http://www.ch.embnet.org/ | всички (Java) |
| Dnadot | proteins, DNA | 100000 | http://arbl.cvmbs.colostate.edu/molkit/dnadot/ | всички (Java) |
| Dotter | proteins, DNA | 100000 | http://arbl.cvmbs.colostate.edu/molkit/dnadot/ | Unix, Linux, Windows |
| Dottup | Whole genomes, DNA | > 100000 | http://emboss.sourceforge.net/ | Unix, Linux |

Dotlet alignment

- Dotlet usage - a Java applet that you can access on the EMBnet (European Molecular Biology Network) server.
 - Applet is a small program that loads automatically into your browser and then runs on your computer.
 - You don't need to install anything as long as you have Java.
- Dotlet is not a complex program, but it has many functions.
- Change the parameters until you get the desired result.
 - Basic rule when using Dotlet
 - You want to find a result, you will have to change various parameters until you get it.

Dotlet alignment

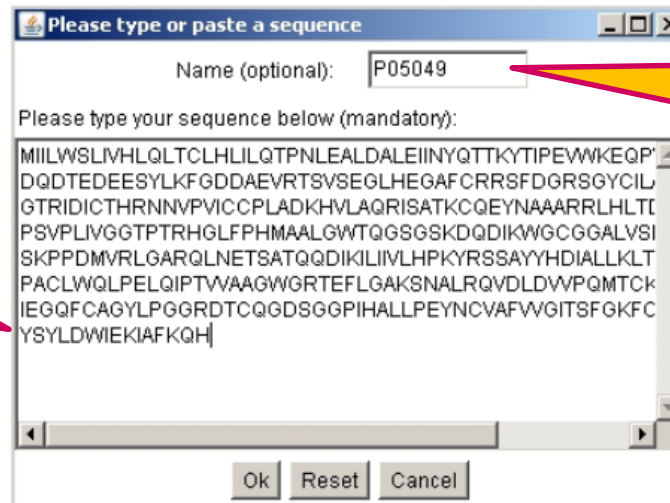
1. Starting Dotlet - Enter the following address in your browser:
<http://myhits.isb-sib.ch/cgi-bin/dotlet>



Dotlet alignment

2. Sequence input to Dotlet - two sequences that contain a specific kinase domain are compared:
 1. Open a new browser window and enter the address below:
<http://www.uniprot.org/uniprot/P05049.fasta>
 2. Copy the sequence shown in the second browser, open the first browser, click the *Input* button, paste the sequence into a window and click the *Ok* button.

The sequence is
set in raw
format



Please type or paste a sequence

Name (optional): P05049

Please type your sequence below (mandatory):

```
MILWSLIVHLQLTCLHLILQTPNLEALDALEIINYQTTKYTIPEVWKEQP'  
DQDTEDEESYLKFGDDAEVRTSVSEGLHEGAFRRSFDGRSGYCL  
GTRIDICTHRNNVPVICPLADKHVLAQRISATKCQEYNAAARRLHLTI  
PSVPLIVGGTPTRHGLFPHMAALGWTQSSGSKDQDIKWGCGGALVSI  
SKPPDMVRLGARQLNETSATQQDIKILIVLHPKYRSSAYYHDIALKLT  
PACLWQLPELQIPTVVAAGWGRTEFLGAKSNALRQVDLDVVPQMTCK  
IEGQFCAGYLPGGRTCCQGDSSGGPIHALLPEYNCAVFWGITSFQKFC  
YSYLDWIEKIAFKQH|
```

Ok Reset Cancel

You can specify a
sequence name

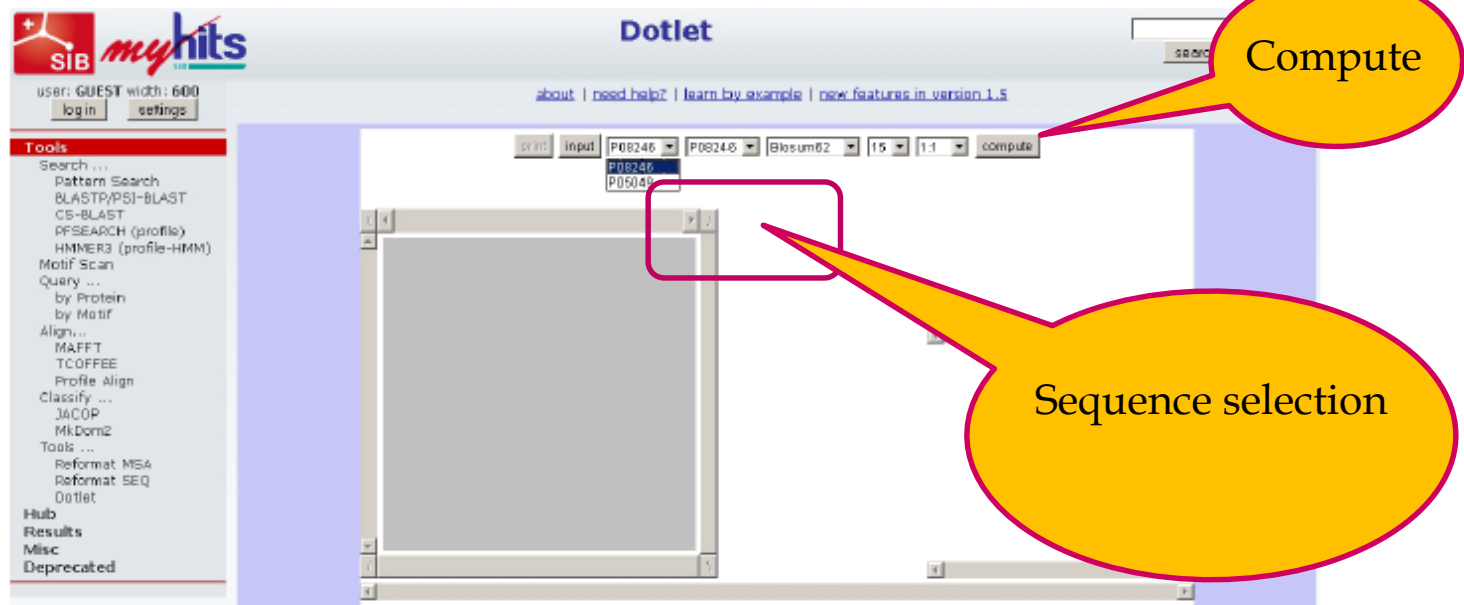
ERASMUS+

Dotlet alignment

2. Sequence input to Dotlet - two sequences that contain a specific kinase domain are compared:
3. Load <http://www.uniprot.org/uniprot/P08246.fasta> in the second window you opened in step 1.
4. Click the *Input* button, copy the sequence and paste it into the Dotlet window.
5. Click the *OK* button in the window.
6. In the Dotlet window, select the two sequences you want to compare using the two drop-down lists that are located near the *Input* button.

Dotlet alignment

2. Sequence input to Dotlet - two sequences that contain a specific kinase domain are compared:
7. In the Dotlet window, select the two sequences. . . .



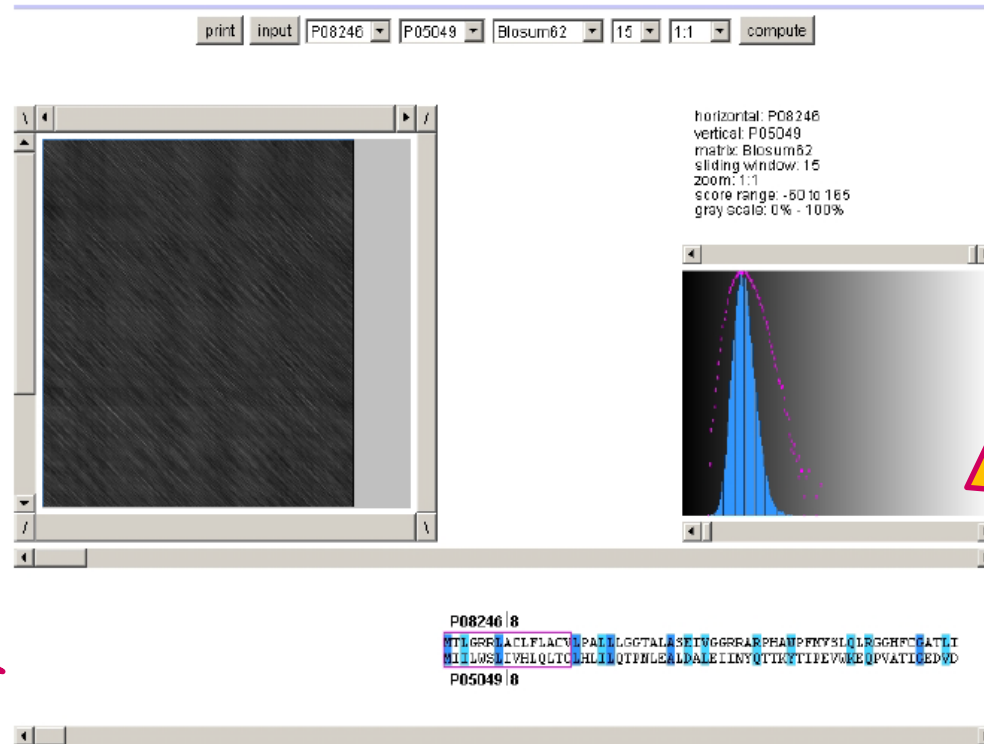
8. In the Dotlet window, click on the *Compute* button located on the far right.

Dotlet alignment

2. Sequence input to Dotlet - two sequences that contain a specific kinase domain are compared:
 - The result looks like this

Dot plot window

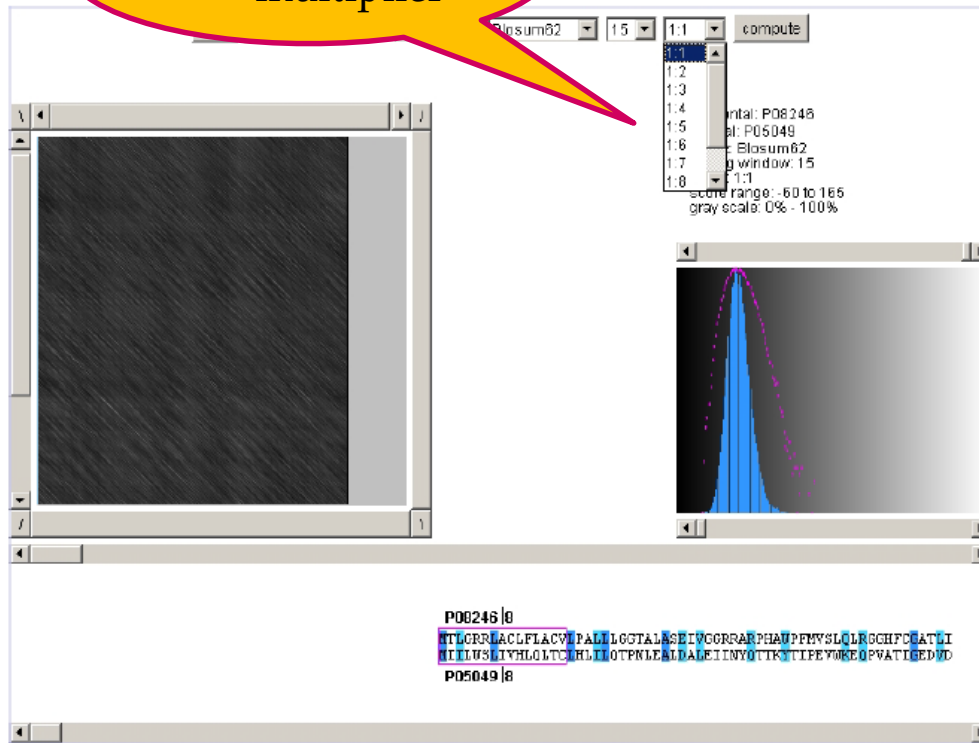
Прозорец на
алайнмънта



Threshold window
- shows the
sensitivity of the
Dot plot graph

Dotlet Detailed Setup

Scaling
multiplier



- The instructions described follow a up-down approach.

Scaling multiplier determination

- By default, Dotlet has a scaling multiplier of 1:1 (1 pixel corresponds to 1 residual).

- When selecting the scaling from the drop-down menu to the left of the *Compute* button, select 1:2
- Click on the *Compute* button.

Dotlet Detailed Setup

2. *Selecting the window size* - When Dotlet compares two amino acids or two nucleotides, it also compares the amino acids or nucleotides in immediate proximity. **The window size controls the size of these areas:**

1. Select 51 from the *Window* drop-down menu (located to the left of the scaling box).
2. Click on the *Compute* button.
 - Long windows make your Dot plot more clear.
 - The window size should be within the size range of the items you are searching.
 - Shorter windows are more sensitive but contain quite a bit of noise.
 - **Start with a large window and scale in until the signal you are searching appears.**

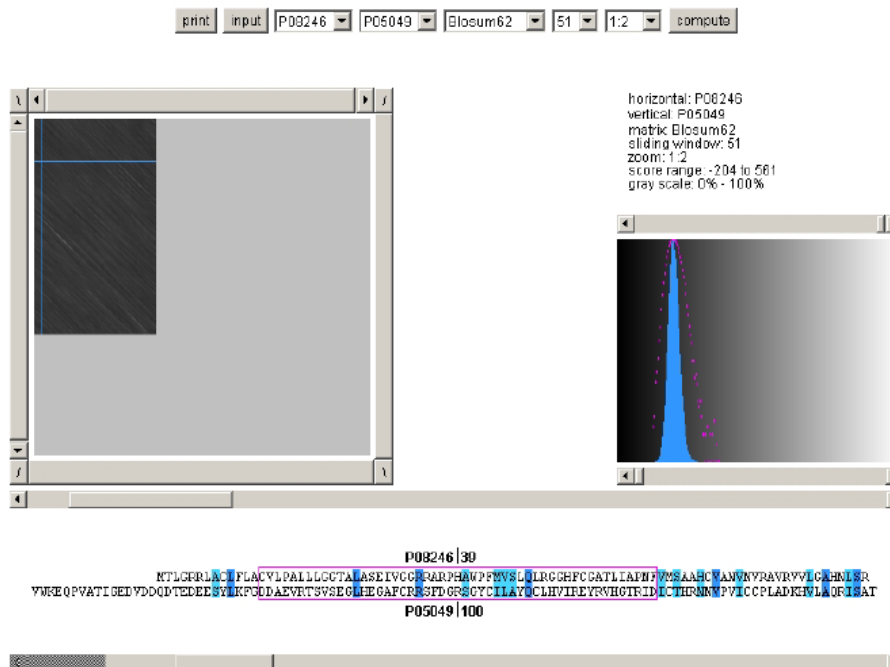
Dotlet Detailed Setup

3. *Threshold adjustment* - The threshold is a value that determines the color of each dot in the Dot plot window. **Points whose score is above the threshold are white and those with a score below the threshold are black:**
- The horizontal axis of the threshold window represents the result. **The low ones are left, the high ones are right.**
 - The large peak on the curve indicates the most common result for two residuals chosen at random.
 - You can imagine the threshold as two vertical lines passing through the top and bottom of the marker. Its position controls the appearance of the Dot plot window:
 - Points with a score below the lower threshold are in black;
 - Points with a score above the upper threshold are in white;
 - Points with a score between the two thresholds are in gray.

Dotlet Detailed Setup

3. Threshold adjustment:

1. Move the bottom cursor of the threshold window all the way to the right - This makes the Dot plot on the left all black.
2. Take the right side of the threshold window to the middle and slowly drag it to the left, keeping the right mouse button pressed.
 - When the top and bottom cursors are at the top, the dots in the Dot plot are either black or white.



Dotlet Detailed Setup

4. *Dot plot saving:*

1. Press the *PrntScrn* button on your keyboard (usually located in the upper right corner).
2. Start Power Point Presentation/Paint.
3. Use the keyboard shortcuts Ctrl+X and Ctrl+V to grab and paste the information into the presentation.

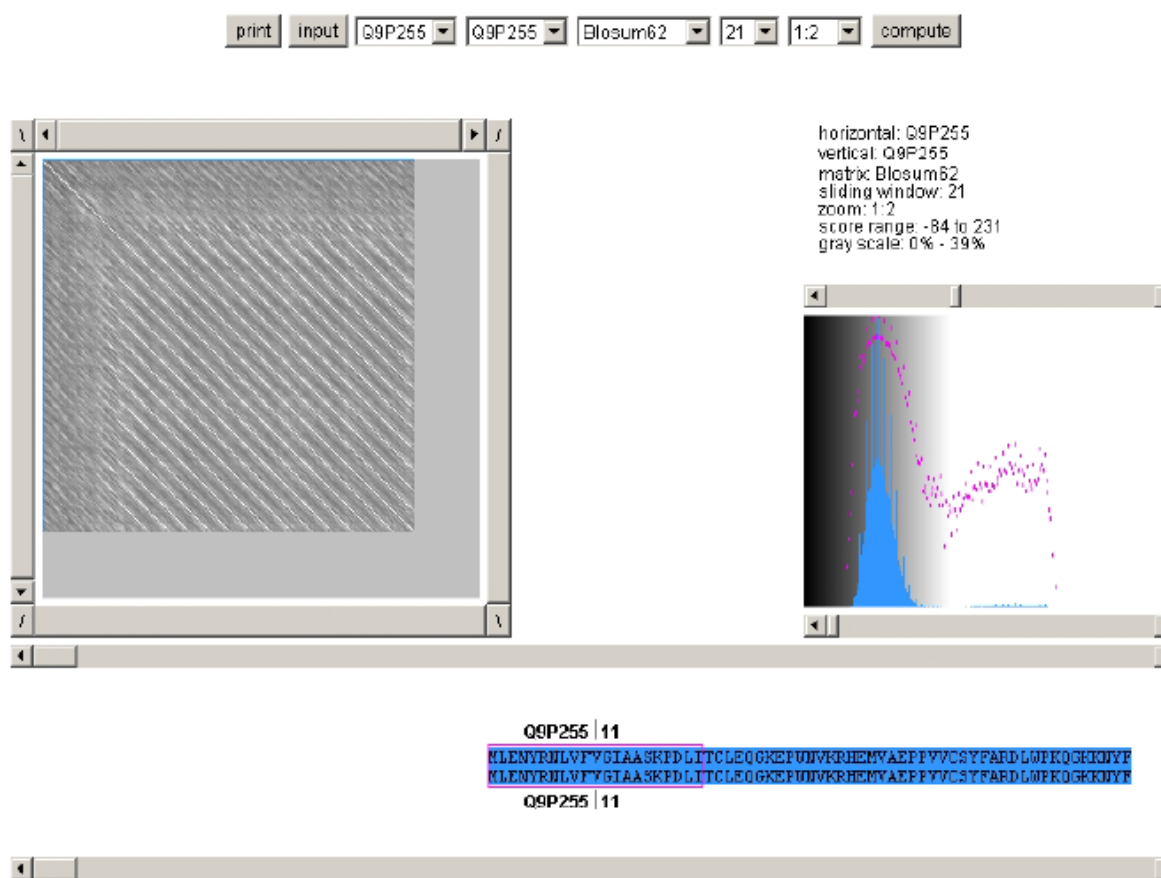
Interpretation of results

- The two proteins considered in the example (P05049 (serine protease) and P08246) are highly distantly related.
 - If you run a BLAST between them you will get an E-value 10^{-4} match.
- Conducting a Dot plot is appropriate to check whether two sequences are homologous in a particular area.
- Identifying conserved domains between two proteins (even if you know nothing about their functions) is a good approach to find a specific region responsible for a given function.

Biological analysis using Dot plot

- Dot plot allows representation of important biological relationships.
- *Defining tandem repeats* - Proteins very often contain short multiple repeat domains.
 - Internal duplication is a tool often used by evolution to create new proteins or ones that function more efficiently.
- Dot plot is the best way to identify repeated domains.

Biological analysis using Dot plot



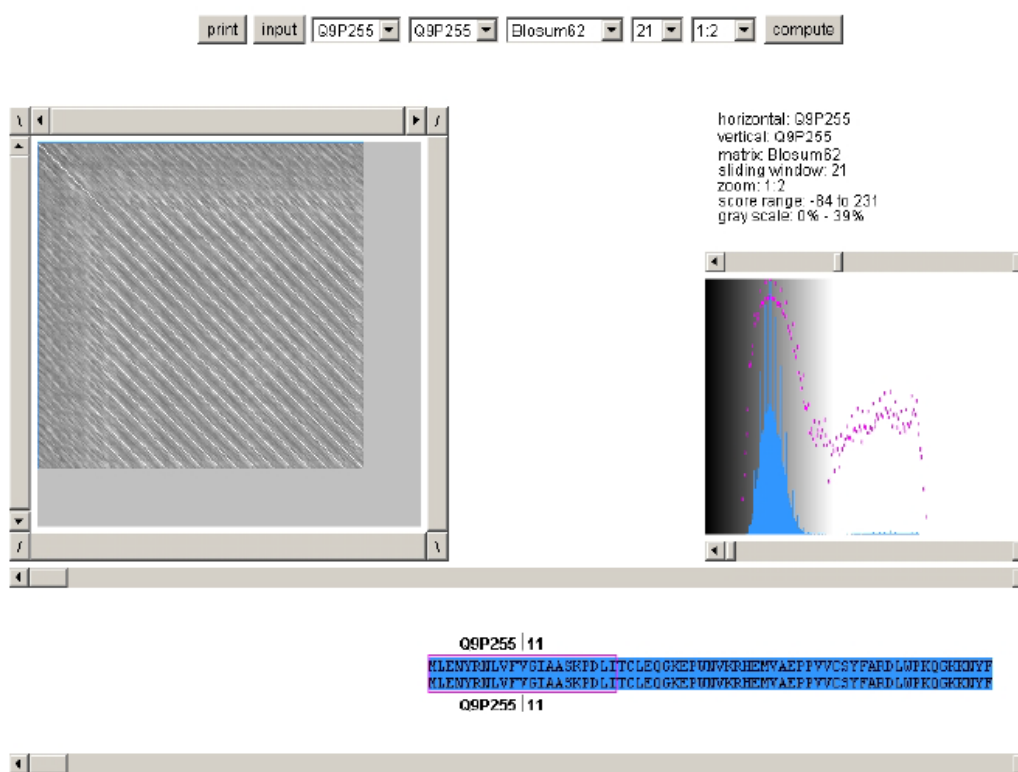
YE73 (human protein) is a potential transcription factor (involved in the RNA transcription) that contains 13 domains with repeated *Zn-ring*

Biological analysis using Dot plot

- To analyze YE73, follow these steps:
 1. Load the following address in your browser:
<http://www.uniprot.org/uniprot/Q9P255.fasta>
 2. Enter the sequence in Dotlet.
 3. Set the following window and threshold settings:
 1. Set window size 21,
 2. Scale 1:2,
 3. 39% for the grey scale, for the upper and lower cursor of the threshold window.
- The main diagonal represents the sequence opposite itself;
- The repeats appear as long continuous diagonals above and below the main diagonal;
- The diagonals are evenly distributed;
- The diagonals are restricted by a square shape.

Biological analysis using Dot plot

- Three conclusions:

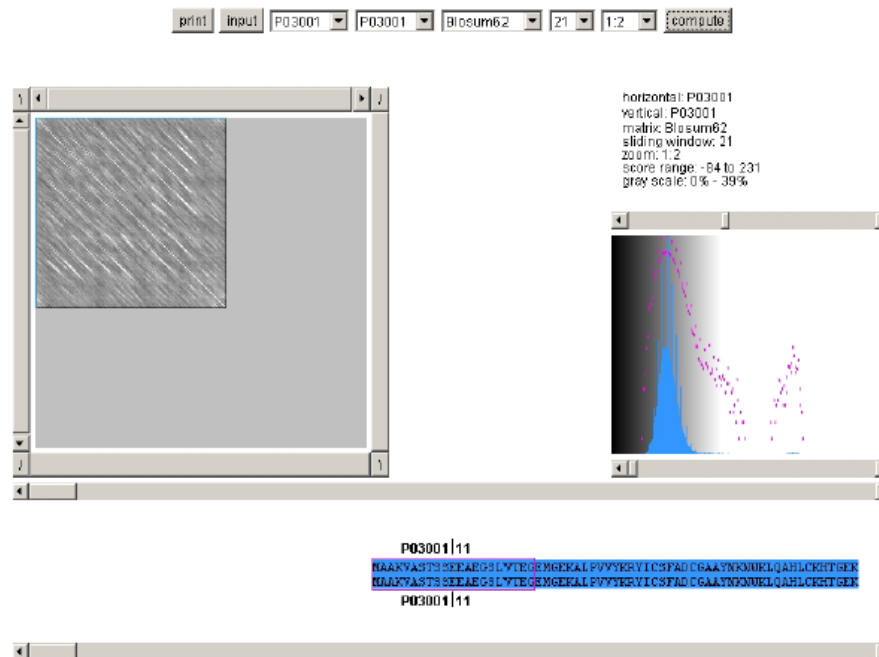


- The number of repeats is equal to the number of diagonals below (or above) the main diagonal (including the main diagonal itself);
- The distance between two adjacent diagonals represents the size of the repeat;
- The shortest diagonal gives the coordinates of a single repeat unit.

Biological analysis using Dot plot

- Proteins that contain distantly related repeats produce more complex Dot plot patterns.
- Tf3a is a transcription factor that also contains tandem *Zn-ring* domains:

<http://www.uniprot.org/uniprot/P03001.fasta>



ERASMUS+

Biological analysis using Dot plot

- Tandem domains are not the only form of repeat domains in proteins and DNA.
 - The description of repeats is the same for different forms of repeat domains (proteins/DNA): **A diagonal that lies off the main diagonal when you align a sequence to itself.**
- If you find a repeat domain with unknown function and no similarity to the proteins already characterized, then you should do the following:
 1. Extract each repeated unit;
 2. Do multiple domain alignment;
 3. Identify conservative locations in the domain;
 4. Include your domain in the PROSITE model or profile;
 5. Search the Swiss-Prot to check if this pattern is associated with certain features.

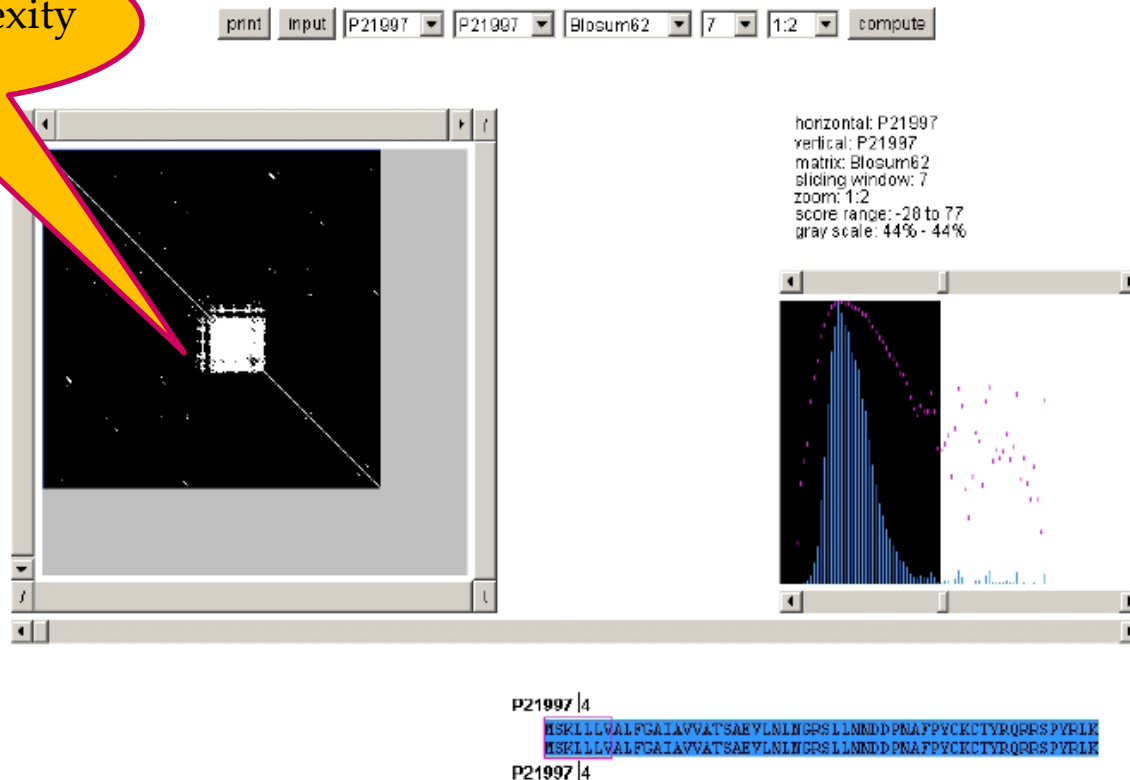
Finding low complexity regions in proteins

- *Low complexity regions* – fragments that contain certain amino acids more frequently than expected in normal proteins.
 - For example, a segment of 100 amino acids that contains 45 prolines is definitely a fragment of low complexity.
 - They often have biological functions such as protein-protein interaction (Leucine Zipper) or non-specific DNA/RNA binding (ARG-rich regions).
- When aligning a sequence with itself, regions of low complexity appear as squares.

Finding low complexity regions in proteins

- Identifying low complexity segments with Dotlet

Low complexity
region



Finding low complexity regions in proteins

- To identify low complexity regions follow these steps:
 1. Load the following address into your browser:
<http://www.uniprot.org/uniprot/P21997.fasta>
 - P21997 is the accession number of sulfated glycoprotein 185.
 2. Enter the sequence in Dotlet.
 3. Set the following settings:
 - Window size 7;
 - Scale 1:2;
 - 44% for the gray scale, for the upper and lower cursors of the threshold window.

Analyzing the nucleic acids with Dotlet

- Dot plot is very suitable for mapping genes.
 - Dotlet can be quite slow for applications that require routine alignment of sequences greater than 10,000 residues in length.
- When analyzing nucleotide sequences, Dotlet contains two useful features:
 - When aligning a protein sequence to a nucleotide sequence, Dotlet automatically translates the nucleotide sequence into all three of its possible reading frames.
 - When aligning a nucleotide sequence to itself, Dotlet automatically replaces one of the sequences with its complementary sequence.
- Online guide:

https://myhits.sib.swiss/util/dotlet/doc/dotlet_help.html

Performing a local alignment

- Two types of alignment:
 - **Global** - the two sequences are aligned along their entire length
 - **Local** - the program compares only the most similar parts of the two sequences and ignores the rest.
- If you have a Dot plot showing that your two sequences are connected along their entire length, you don't need to do a **local alignment**, you can go directly to a **global** one.
- Two main reasons to analyze your sequences using **local alignment**:
 1. Compare two distantly related sequences that share only a few non-contiguous domains/regions;
 2. To analyze repeated elements within the same sequence.

Performing a local alignment

- Local alignment methods do what their name implies: two sequences are set, as a result an alignment of the most similar parts of those sequences returns.
 - Automatically removes regions from amino acids and nucleotides that cannot be compared.
- In theory, the local alignment corresponds to one of the diagonals that appear in a Dot plot.
- In practice, you may see a Dot plot signal that does not correspond to any local alignment.
- Occasionally, local alignment programs report alignments that do not appear in the Dot plot.

Suitable local alignment selection

- Two basic methods for conducting **local alignment**:
 - Fast, heuristic method integrated into the BLAST program
 - Slower and more accurate, the Lalign program.

| Ability | BLAST | Lalign |
|------------------|----------------------|-----------------------------------|
| Speed | Very fast | Slower |
| Sequences lenght | Very long sequences | Shorter sequences |
| Result | E-value | E-value |
| Alignments | Reports the best one | Reports the best ten (or more) |
| Sequence type | Best with DNA | Best with proteins |

Suitable local alignment selection

- BLAST can also be used to search in a given database.
 - BLAST comparing two sequences (bl2seq).
 - It has been adapted so that you can limit it to only two sequences.
 - It does not generate alignments other than the BLAST results you get when you search in a given database.

<https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE%20TYPE=BlastHome>

Using Lalign to find ten best local alignments

- Lalign is an ideal addition to BLAST.
 - Slower, but more accurate, and gives as a result as many local alignments as you want (best, next, etc up to the number you wanted and specified).
 - Suitable for analysis of complex proteins with many repeats.

Find multiple matching subsegments in two sequences

This is William Pearson's *lalign* program. A manual page for this program is available [here](#). The lalign program implements the algorithm of Huang and Miller, published in Adv. Appl. Math. (1991) 12:337-357.

This program is part of the FASTA package of sequence analysis program.

Usage: Paste your two sequences in one of the supported **formats** into the sequence fields below and press the "Run lalign" button.

Make sure that both format buttons (next to the sequence fields) shows the correct formats

Choose the alignment method:
☒ local (default) ☐ global ☐ global without end-gap penalty

Number of reported sub-alignments: E-value threshold: (default 10.0)

Scoring matrix:

Opening gap penalty: (default -12) Extending gap penalty: (default -2)

First sequence title (optional):

Input sequence format:

1st Query sequence:
or ID or AC or GI
(see above for valid formats)

Second sequence title (optional):

ERASMUS+

Using Lalign to find ten best local alignments

- **Example 2.** Extract local alignments from two distantly related sequences containing serine proteases domains:
 1. Load the following address in your browser:
https://embnet.vital-it.ch/software/LALIGN_form.html
 2. Select *Local* from the options for alignment methods.
 3. Select the number of sub-alignments displayed:
 - You must choose a number that corresponds approximately to the number of diagonals you observed in the Dot plot.
 4. Select an evaluation matrix - controls the value of mutations when Lalign compares sequences.

Using Lalign to find ten best local alignments

Find multiple matching subsegments in two sequences

This is William Pearson's *lalign* program. A manual page for this program is available [here](#). The *lalign* program implements the algorithm of Huang and Miller, published in Adv. Appl. Math. (1991) 12:337-357.

This program is part of the FASTA package of sequence analysis program.

Usage: Paste your two sequences in one of the supported [formats](#) into the sequence fields below and press the "Run lalign" button.
Make sure that both format buttons (next to the sequence fields) shows the correct formats

Choose the alignment method: ☒ local (default) ☐ global ☐ global without end-gap penalty

Number of reported sub-alignments:

E-value threshold: (default 10.0)

Scoring matrix:

Opening gap penalty: (default -12) Extending gap penalty: (default -2)

First sequence title (optional):

Input sequence format:

1st Query sequence:
or ID or AC or GI
(see above for valid formats)

Second sequence title (optional):

- **Example 2.** Extract local alignments from two related distant sequences containing serine proteinase domains.

Using Lalign to find ten best local alignments

- **Example 2.** Extract local alignments from two distantly related sequences containing serine proteinase domains:
 5. Set size for *Opening gap penalty* option - defines the possibility of opening a "hole" in one of the sequences when aligning.
 6. Specify a value for *Extending gap penalty* - The value for *Extending gap penalty* + the value for *Opening gap penalty* characterize the sequence alignment gap.
 - The value for the *Extending gap penalty* must be at least ten times lower than the value for the *Opening gap penalty*.
 7. Select *Swiss-Prot ID or AC* from the *Input Sequence Format* drop-down menu.

Using Lalign to find ten best local alignments

- **Example 2.** Extract local alignments from two distantly related sequences containing serine proteinase domains:
 8. Enter *P05049* in the first field.
 9. Select *Swiss-Prot ID or AC* from the *Input Sequence Format* drop-down menu.
 10. Enter *P08246* in the second field.
 11. Click the *Run Lalign* button.
 12. Save Results - To save the result to a file, use the *File→Save As* commands as an option on your browser.

Interpretation of Lalign results

- Lalign gives as a result the number of alignments you have specified, arranged by their rating.

Readseq version 2.1.30 (12-May-2010) Readseq version 2.1.30 (12-May-2010)

lalign output for P05049 vs. P08246

[EMBnet-Server] Date: Tue Oct 5 3:46:51 2021

```
# bin/lalign36 -E 10.0 -f -12 -g -2 29292.1.seq 29292.2.seq -J -K 5
LALIGN finds non-overlapping local alignments
version 36.3.5e Nov, 2012(preload8)
Please cite:
X. Huang and W. Miller (1991) Adv. Appl. Math. 12:373-381

Query: 29292.1.seq
1>>>sp|P05049|SNAK_DROME (snk)RecName: Full=Serine protease snake; EC=3.4.21.-; Flags: Precursor;[Drosophila melanogast - 435 aa
Library: 29292.2.seq
267 residues in 1 sequences

Statistics: (shuffled [500]) MLE statistics: Lambda= 0.1852; K=0.03785
statistics sampled from 1 (1) to 500 sequences
Threshold: E() < 10 score: 33
Algorithm: Smith-Waterman (SSE2, Michael Farrar 2006) (7.2 Nov 2010)
Parameters: BL50 matrix (15:-5), open/ext: -12/-2
Scan time: 0.010

>>sp|P08246|ELNE_HUMAN (ELANE)RecName: Full=Neutrophil e (267 aa)
Waterman-Eggert score: 181; 53.1 bits; E(1) < 1.2e-11
25.5% identity (53.0% similar) in 251 aa overlap (186-430:30-245)

      190      200      210      220      230      240
sp|P05  IVGGTPTRHGLFPMAALGWTQSGSGKDQDIKWGCGGALVSELYVLTAAHCATSGSKPPD
      :::  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:  .:
sp|P08  IVGGRRARPHAWPFMVSLQLRGGHF-----CGATLIAPNFVMSAAHCVANVNVRAV
      30      40      50      60      70      80
```

Interpretation of Lalign results

- Lalign alignments are in a BLAST-like format, arranged according to their E-values.
- The following characteristics are on the first line:
 - *Identity Percentage* - The fraction of identical residues compared to each other.
 - *Local Alignment Length (Overlap)* - This is the total length of the local alignment.
 - *Score (Comparison Score)* - The higher this value is, the better the alignment is.
 - *E-value* - shows how many times by chance one would expect to find such a good alignment for two sequences.
 - Note that the **E-value in this method is of much less importance than in BLAST, involved with searching a database.**
 - A good E-value should be below 10^{-4}

Interpretation of Lalign results

- The alignment itself contains three types of information
 - *Residue index* - located on the row above the sequence.
 - *The alignment itself* - the gaps in the sequence are represented with dashes.
 - *Identity and similarity* - the row between the two sequences being compared.
 - *(_) symbol means identity* - Identity can be observed when comparing protein and DNA sequences
 - *(.) means similarity* - can only exist in the protein alignment.
 - Similarity gives the fraction of amino acids that have similar physicochemical characteristics.
- Two residues are similar when their Substitution score is greater than 0.

Interpretation of Lalign results

- To assess the quality of your alignment, pay attention to the specified size.
 - Low fragment complexity, low levels of repetition in protein sequences, repetitive models of hydrophobic/hydrophilic residues, etc.
- **Bioinformatics programs are not perfect.** They can make mistakes and give incorrect results.
 - The most upper alignment that Lalign generates illustrates this limitation well. Because of the information that Dot plot provides, we trust this alinement.
 - The two sequences are so distant and somewhat similar that we have to be quite cautious in the interpretations we make.

Performing a global alignment

- A global alignment is what the name implies - **an alignment of every amino acid or nucleotide in your sequences.**
- Important in multiple alignments.
- Global alignments are not useful in detecting similarities between two sequences, since the statistical method of E-value estimation does not apply to them.
- In a global alignment, there are no amino acids or nucleotides that mysteriously disappear.

Performing a global alignment

- Three reasons to prefer using a global alliance:
 - Checking for small differences between two sequences.
 - This can happen with sequences that you have manipulated and possibly changed. Global scaling is the best way to locate potential problems.
 - Analyzing polymorphisms (e. g. SNPs) between closely related sequences.
 - Comparing two sequences that partially overlap.
- Using Lalign to conduct a global alignment is like conducting a local alignment.
 - The only additional thing you need to do is click on the *Global without End-Gap Penalty* button.

Protein alignment with DNA sequences

- The alignment tools we have examined are only effective when sequences of a similar type are compared - protein to protein or DNA to DNA.
- Sometimes, it is necessary to compare a protein with a DNA fragment (for example, its original gene).
 - Institut Pasteur (<http://mobyle.pasteur.fr/cgi-bin/portal.py?#forms>)
 - dr. Peer Bork and his group at the European Molecular Biology Laboratory in Heidelberg (<http://www.bork.embl.de/pal2nal/>).
- Both servers require you to have the protein and DNA sequences beforehand.
 - Protogene (<http://www.tcoffee.org/>) automatically extracts the DNA sequence corresponding to a given protein.

Thank you for your attention!



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness



Module 3. Bioinformatic tools

Topic 3. Protein Folding Problem

Lesson 3. Multiple sequence alignment



Multiple alignment

- Specialized tools for sequence assembly are used, such as:
 - Phred и Phrap (<http://www.phrap.org/>)
 - Fault-Tolerance Synthesizer
(<http://genome.cs.mtu.edu/Tools.html>).
- It is not required to run a multiple alignment when the sequence of interest has no homology to any sequence in the database.
 - You can try to find multiple sequences using functional criteria or BLAST search.
- You can create a correct multiple alignment only if there are appropriate sequences.

Multiple alignment

- The basic idea behind a multiple alignment is **placing amino acids or nucleotides in the same column** based on similarities according to some criterion.

- Three main criteria for building multiple alignment

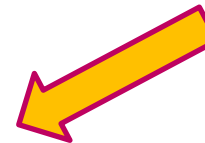


| Criterion | Meaning |
|-----------------------|---|
| Structural similarity | Amino acids with the same role in each structure are in one column. Precision positioning programs are the only ones that use this criterion. |
| Functional similarity | Amino acids or nucleotides with the same function are in one column. There is no automatic program that explicitly uses this criterion, but if the information is available, you can force some programs to follow it, or you can edit your alignment manually. |
| Sequence similarity | Amino acids in one column are those that give an alignment with maximum similarity. Most programs use sequence similarity because it is the easiest criterion. When sequences are closely related their structural and functional similarities are equivalent to sequence similarity. |

Multiple alignment

| Application | Procedure |
|--------------------------|---|
| Extrapolation | A good multiple alignment can convince you whether an unknown sequence is a real member of a protein family. |
| Phylogenetic analysis | If you correctly select the sequences for a given multiple alignment, you can rewrite the history of these proteins. |
| Motives Identification | By detecting many conserved positions, you can define a region that is common for a particular feature. |
| Domains identification | It is possible for a multiple alignment to become a profile that describes a protein family or protein domain (PSSM). You can use this profile to scan databases for new family members. |
| DNA regulatory elements | You can convert a multiple DNA alignment into a transcription factor matrix and scan other DNA sequences for potentially similar binding sites. |
| Prediction of structures | A good multiple alignment can give you a perfect prediction of secondary structure, both protein and RNA. Sometimes it can help in building a 3-D model. |
| SNP analysis | A number of gene alleles often have different amino acid sequences. Multiple alignment can help predict whether nonsynonymous polymorphisms in single nucleotides are beneficial or harmful. |
| PCR analysis | A multiple alignment can help you identify less degenerate parts of a protein family, which would allow the discovery of new members of that family by PCR (polymerase chain reaction). Or at least make your primers more optimal. |

- Basic applications of multiple alignment.




- No matter what your choice or motivation is, you should keep in mind that the reason to use multiple alignment is that it **gives an instant picture of the forces that shape the evolution!**

Multiple alignment

- A possible hypothesis in multiple alignment:
 - conserved positions (columns where all sequences contain the same amino acid or nucleotide) are more important for the function than non-conserved positions (columns where sequences contain different amino acids or nucleotides).
- Multiple alignment is a great way to present your results.
 - Allows you to put a lot of information into a single model and easily to locate inconsistencies or potential problems.

Selection of appropriate sequences

| Problem | Solution |
|--------------------------------|---|
| Proteins or DNA | Use proteins when possible. You can convert them back to DNA after performing multiple alignment |
| Multiple sequentions | Start with 5-10 sequences, avoid alignment with more than 30. |
| Multiple different sequentions | There is often a problem with sequences that are less than 30 percent identical to more than half of the other sequences in the group. |
| Identical sequentions | There is never any use for them. If you have a reason to work with such sequences, exclude those that are more than 90% identical from other sequences in the group. |
| Incomplete sequentions | Multiple alignment programs prefer sequences that are relatively uniform in length. They often have trouble comparing whole sequences and incomplete fragments. |
| Multiple repeated domains | Sequences with multiple repeated domains usually cause problems for multiple alignment programs (especially if the number of domains is different). In that case, it might be better to extract the domains with Dotlet or Lalign and do multiple alignment on those fragments. |

- Carefully select the sequences you want to align.
- 

Using DNA or protein sequences

- If you're interested in non-coding sequences, you obviously have no choice but to use DNA.
 - Non-coding DNA sequences are difficult for alignment!
- You can use local multiple alignment methods such as Gibbs sampler or Pratt.
- Multiple alignment methods are more suitable for proteins.
- In phylogenetic analysis of multiple coding DNA sequences:
 1. Convert your DNA sequences into proteins.
 2. Perform multiple alignment of these proteins.
 3. Convert the proteins used in the multiple alignment back into DNA using pal2nal
(<http://www.bork.embl.de/pal2nal/>) или Protogene
(<http://www.tcoffee.org/>).

Selection of correct number of sequences

- There is no absolute answer.
- Start with a relatively small number of sequences.
- If you start with hundreds of sequences, you can get into trouble right away:
 - Calculating a large alignment is difficult.
 - Creating a large alliance is also difficult.
 - Showing a big alignment is awkward.
 - The use of a large alignment is inefficient.
 - Making an accurate and large alignment is almost impossible.
- Start with less and gradually increase the number of sequences you run multiple alignments with until you include all the sequences you are interested in.

Compromise between similarity and new information

- If you think that very similar sequences give very good alignments, you are right!
- Working with sequences that are highly distant in relationship is also not appropriate.
- Two things that multiple alignment programs really don't like:
 - Sequences that are very different from the rest in the group;
 - Sequences that require long insertions/deletions for a proper alignment.
- Sequences should be related as strongly as possible without requiring too many breaks when conducting an alignment.

Compromise between similarity and new information

- Basic steps in sequence selection:
 1. Select several sequences.
 2. Perform multiple alignment using one of the servers.
 3. If your alignment looks good, keep the sequences.
 4. If your alignment is difficult to interpret:
 - Look at the sequences in more detail - try to remove "offending" sequences, those that are most distant in relationship or those that cause long insertions/deletions.
 - Run the alignment again with fewer sequences.
 - Store the initial set of sequences before obtaining satisfactory, easy-to-interpret results.

Setting the sequences name

- Multiple alignment programs do not have standard ways of handling sequence names:
 - Do not use blank positions when naming sequences;
 - Do not use special characters. Keep using regular letters, numbers, and underscore (_) to replace blank spaces. Avoid all other characters, especially those that are most inviting for special sequences (such as @, #, _, L, etc.);
 - Do not use names longer than 15 characters;
 - Do not give the same name to two different sequences contained in a set.

Setting the sequences name

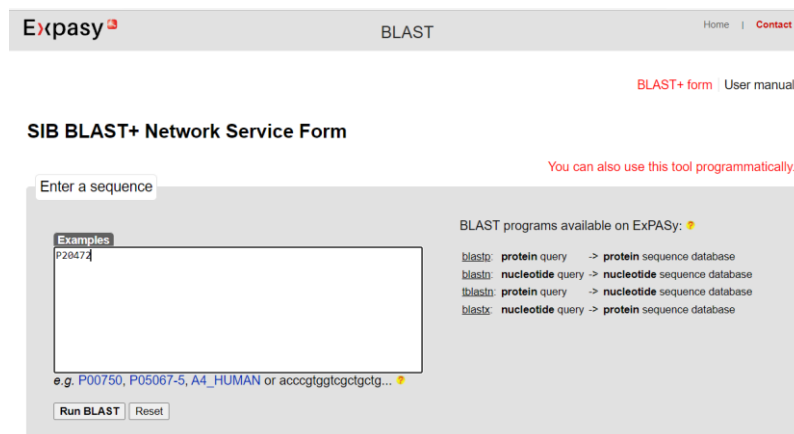
- There are two types of sequences you can integrate into a single multiple alignment:
 - Characterized/ Annotated Sequences - These are sequences for which there are good annotations and experimental data to support them.
 - Unknown sequences - This category includes your sequences as well as sequences from different databases.
- Using BLAST, you can search a database for sequences that are homologous (similar) to your query.

| Address | Opportunities |
|---|---|
| http://www.expasy.ch/tools/blast/ | Extract whole sequences, export sequences to FESTA format, request sequences to ClustalW, Tcoffee or MAFFT |
| http://blast.ncbi.nlm.nih.gov/ | Extract whole sequences, extract fragments, export sequences in FESTA format, request sequences to ClustalW |
| http://srs.ebi.ac.uk/srsbin/cgi-bin/wgetz?+page+srsq2+-noSession | Заявяване на секвенции към ClustalW |

ERASMUS+

Selecting sequences via the ExPASy server

- You can use ExPASy to download protein sequences.
 - If you are interested in collecting DNA sequences, use NCBI, BLAST (<http://blast.ncbi.nlm.nih.gov>).
 - The BLAST server on the PBIL site is very similar to ExPASy.
- 1. Enter the following address in your browser:
<http://web.expasy.org/blast/>
- 2. Enter sequence identification number **P20472**.



The screenshot shows the ExPASy BLAST web interface. At the top, there's a header with 'Expasy' and 'BLAST' links. Below that, a section titled 'SIB BLAST+ Network Service Form' contains a text input field labeled 'Enter a sequence'. Inside this field, 'Examples' are listed: 'P20472'. Below the input field, there are buttons for 'Run BLAST' and 'Reset'. To the right of the input field, a list of BLAST programs available on ExPASy is shown, including 'blastp', 'blastn', 'tblastn', and 'blastx', each with a description of the query and database types. A link to 'You can also use this tool programmatically...' is also present.

Selecting sequences via the ExPASy server

- You can use ExPASy to download protein sequences.
Keep the default options (*Complete database*) for the **UniProt Knowledgebase (UniProtKB)** drop-down list.
 3. Go to the **Select options** tab and set the number of best sequences displayed to 1000.
 4. In the same tab, set the number of best alignments displayed to 1000.
 5. Click the **Run BLAST** button.
 6. Choose the sequences you want.
 - There is no absolute rule when choosing sequences, but you can use the following guidelines:

Selecting sequences via the ExPASy server

- You can use ExPASy to download protein sequences only.
 7. Scroll down the page to select the sequences you want.
 - There is no absolute rule when choosing sequences, but you can use the following guidelines:
 - Select the top sequence.
 - On first analysis, you should select ten sequences or less. Ideally, the ten sequences should have an E-Value between 10^{-40} and 10^{-5} .
 - Before selecting a sequence, check that it is similar along the entire length of your query.

Selecting sequences via the ExPASy server

- You can use ExPASy to download protein sequences only.
 8. From the **Send Selected Sequences** drop-down menu, select the method by which you want to export the selected sequences:
 - **FASTA** – Generates a file that contains your sequence in FASTA format.
 - **ClustalW**, **Tcoffee** and **MAFFT** – These are multiple-alignment packages running on the EMBnet server.
 - **Reduce redundant repeats** – This option will extract the most significant sequences from a data set - a good option if too many sequences are available and you don't know which ones to select.
 - **Pratt** – Scans for conserved motives in the stated sequence without including them in the alignment.

Selecting sequences via the ExPASy server

- You can use ExPASy to download protein sequences only.
- 8. From the **Send Selected Sequences** drop-down menu, select the method by which you want to export the selected sequences

List of the matches

Clustal Omega - Multiple Sequence Alignment (MSA) [\[More on these tools...\]](#)

Clustal Omega - Multiple Sequence Alignment (MSA)
T-COFFEE - Multiple Sequence Alignment (MSA)
MAFFT - Multiple Sequence Alignment (MSA)
Decrease redundancy
JACOP - Clusters builder
PRATT - Patterns finder
Retrieve entries in Swiss-Prot format
Retrieve sequences in FASTA format
Retrieve accession numbers

| | | Description | Score | E-value | | |
|--------------------------|----|--|----------------|--|----------------|-------|
| | | Parvalbumin, isoform CR... | 184 bits (467) | 3e-58 | | |
| | | Parvalbumin alpha OS=Homo sapiens OX=... | 184 bits (467) | 3e-58 | | |
| <input type="checkbox"/> | 3 | G3RTP2 (GORGO) | 110 | Parvalbumin OS=Gorilla gorilla gori... | 183 bits (464) | 9e-58 |
| <input type="checkbox"/> | 4 | A0A2J8QPJ7 (PANTR) | 110 | Parvalbumin OS=Pan trog... | 183 bits (464) | 9e-58 |
| <input type="checkbox"/> | 5 | A0A6D2WVV0 (PANTR) | 110 | PVALB isoform 1 OS=Pan ... | 183 bits (464) | 9e-58 |
| <input type="checkbox"/> | 6 | G1RX85 (NOMLE) | 110 | Uncharacterized protein OS=Nomascus... | 181 bits (458) | 6e-57 |
| <input type="checkbox"/> | 7 | A0A2J8UVX4 (PONAB) | 110 | Uncharacterized protein... | 181 bits (458) | 6e-57 |
| <input type="checkbox"/> | 8 | A0A2K5PMF8 (CEBIM) | 110 | Parvalbumin OS=Cebus im... | 181 bits (458) | 6e-57 |
| <input type="checkbox"/> | 9 | A0A6D2XAW5 (PONAB) | 110 | PVALB isoform 1 OS=Pong... | 181 bits (458) | 6e-57 |
| <input type="checkbox"/> | 10 | A0A0D9R690 (CHLSB) | 110 | Uncharacterized protein... | 179 bits (453) | 3e-56 |

Collecting a famous collection of Swiss-prot sequences

- If you already know the name or the access number of each of the sequences you want to include in the multiple alignment, and if they are in Swiss-Prot or TrEMBL, you can access them directly using special online ExPASy tools.
 1. Enter the following address in your browser:
<http://www.uniprot.org/batch/>
 2. Enter the access numbers for the desired sequences in the **Provide your identifiers** window.
 3. Select the desired sequences, click the **Download** button, select **FASTA** in the **Format** drop-down list and click the **Go** button.
 4. Save the results on your computer.

Selecting an appropriate multiple alignment method

- Constructing a good multiple alignment requires practice.
- ClustalW (the latest version is called ClustalOmega) - the most widely used multiple alignment package.
- Tcoffee - one of the latest created multiple alignment programs:
 - Combining sequences and structures,
 - Evaluating an alignment,
 - Merging several alternative multiple alignments into a single result.

Selecting an appropriate multiple alignment method

- ClustalOmega / ClustalW - the most commonly used program for performing multiple alignments.
- ClustalOmega/ClustalW uses incremental methods in constructing the alignment.
 - Instead of comparing all the sequences at once, it adds them one by one.
- There are many ClustalOmega / ClustalW servers.
- ClustalOmega/ClustalW is a typical example of a program that can produce a good result with its default settings.
 - Change settings only if you want to change the program output format.

Selecting an appropriate multiple alignment method

- How does ClustalOmega/ClustalW work?
 - The method used by the program is fast and simple - progressive alignment.
 - Initially, all sequences are compared 2 by 2, and the program arranges them similarly.
 - This grouping looks like a phylogenetic tree, is called a **Guide tree**, and is stored in a file with the extension **.dnd**.
- The **W** in ClustalW comes from **Weights** - each sequence in the alignment is given a "weight" proportional to the new information it contributes.

Run the EBI ClustalOmega server

- Before you start working with ClustalOmega/ClustalW, you need to collect all the sequences you want to work with.
 - The most convenient way is to enter the sequences in FASTA format, but ClustalOmega/ClustalW accepts other formats, including Swiss-Prot and PIR, as well as already aligned sequences in the most common formats after multiple alignment.
- If you want to use ClustalOmega/ClustalW effectively, keep the following in mind:
 - If you specify a group of sequences that have already passed multiple alignment, ClustalOmega/ClustalW does not remove the existing "gaps".
 - The order of sequences in the file you submit for processing can sometimes affect the final alignment.

Working with the ClustalOmega server

1. Go to the EBI ClustalOmega server page:
<https://www.ebi.ac.uk/Tools/msa/clustalo/>
2. Enter the sequences you have assembled in the appropriate field or upload the file that contains them in the *Upload a file field*.
3. Select an *Output Format*.
 - The most secure way is to use "*Aln Without Numbers*"
 - To choose the most appropriate for you, do NOT restart the entire alignment!
 - You can easily reformat your alignments using one of the online reformatting tools (e. g. EMBOSS Seqret).

Working with the ClustalOmega server

4. Select the *Input* from *Output Order* menu.
 - If you select the *Input* option, the ClustalOmega/ClustalW alignment file displays sequences in the original order, i. e. the order in which they were entered.
 - If you select the *Aligned* option, the sequences appear in the order the software arranged them when it built the guide tree - *closely related sequences are close together*.
 - It's best to pre-arrange your sequences in the most informative way in the file you'll be submitting and then select the *Input* option.
5. Click the *Submit* button.

Working with the ClustalOmega server

STEP 1 - Enter your input sequences

Enter or paste a set of

PROTEIN

sequences in any supported format:

Or, upload a file: [Разглеждане...](#) Не е избран файл.

[Use a example sequence](#) | [Clear sequence](#) | [See more example inputs](#)

STEP 2 - Set your parameters

OUTPUT FORMAT

ClustalW

DEALIGN INPUT SEQUENCES

no

MBED-LIKE CLUSTERING GUIDE-TREE

yes

MBED-LIKE CLUSTERING ITERATION

yes

NUMBER of COMBINED ITERATIONS

default(0)

MAX GUIDE TREE ITERATIONS

default

MAX HMM ITERATIONS

default

ORDER

input

STEP 3 - Submit your job

☐ Be notified by email (Tick this box if you want to be notified by email when the results are available)

Submit

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Working with the ClustalOmega server

6. Save your results - The results appear in a table with several hyperlinked sections. Clicking on them opens:
 - *Output file* - Pairwise scores.
 - *Alignments* - multiple alignment - you can preview this section on the screen or save it as a text file.
 - *Guide tree file* - Most ClustalOmega/ClustalW Web servers (including the one at EBI) display the guide tree as a *.dnd* file.
 - *Your input file* - the file with the sequences you entered in ClustalW.

Working with the ClustalOmega server

- ClustalOmega results page.

Results for job clustalo-R20211207-121640-0007-3474410-p2m

| | | | | | |
|--|-----------------------|------------|-------------------|-----------------|--------------------|
| Alignments | Result Summary | Guide Tree | Phylogenetic Tree | Results Viewers | Submission Details |
| Input Sequences | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.input | | | | | |
| Tool Output | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.output | | | | | |
| Alignment in CLUSTAL format with base/residue numbering | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.clustal_num | | | | | |
| Guide Tree | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.dnd | | | | | |
| Phylogenetic Tree | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.ph | | | | | |
| Percent Identity Matrix | | | | | |
| clustalo-R20211207-121640-0007-3474410-p2m.pim | | | | | |

Change in ClustalW parameters

| Parameter | Effect |
|-----------------------------|---|
| Substitution matrix | These matrices estimate the value of mutations in the alignment. If you select a category such as PAM or BLOSUM, ClustalW automatically selects the most adapted index. It is difficult to predict the effect of changing the matrix and there is no ideal matrix. If your sequences have high identity, changing the matrix will have no effect. If the alignment you get is difficult to interpret, it may be useful to change the matrix from BLOSUM to PAM. These matrices measure the probability of the bases changing/mutating over the course of evolution, thus providing an estimate when comparing them between sequences in an alignment. |
| Gap-opening penalty (GOP) | This parameter evaluates the opening of "gaps"/sequence breaks in the alignment. The larger the set value, the harder it is to insert a "gap" into your alignment. The GOP is applying once for opening each gap. If you change it, it won't have much effect, because ClustalW re-sets these values automatically. |
| Gap-extension penalty (GEP) | This parameter controls the size of the gaps in the alignment. It is not possible to predict the optimal GOP/GEP pair, but for each protein family, for example, one exists and can only be determined by experience. |

- Three parameters you can change in your alignment
 - *Substitution matrices,*
 - *Gap-opening penalties*
 - *Gap-extension penalties*

Change in ClustalOmega/ClustalW parameters

- Never change parameters just to get ClustalOmega/ClustalW to produce the alignment you know is correct!
 - You can then edit it with one of the programs for this purpose.
- Change parameters make sense only if you want to understand how small changes can improve the overall appearance of your alignment as blocks or weakly conserved positions.

Comparing sequences and structures with Tcoffee

- Tcoffee is a modern method for multiple alignment of sequences.
 - It uses a principle somewhat similar to that of ClustalOmega, but gives much more accurate alignments at the expense of a slightly longer run time.
- Tcoffee does progressive alignment, but compares segments from a whole set of sequences.
- Besides accuracy, other great advantages of Tcoffee are:
 - Its ability to compare sequences and structures (EXPRESSO);
 - the ability to evaluate the accuracy of the alignment (CORE);
 - the ability to combine many alternative multiple alignments into one (Mcoffee).

Comparing sequences and structures with Tcoffee

- Programs available at www.tcoffee.org.

| Program | With it you can |
|----------|--|
| TCOFFEE | make multiple alignments with Tcoffee |
| CORE | evaluate the reliability of an existing multiple alignment |
| MCOFFEE | run any multiple alignment analyses and combine all results into one final alignment |
| EXPRESSO | include any available structural information in your alignment. If there are known structures, it makes the best alignment of sequences. |

Multiple Alignments with Tcoffee

- To do multiple alignments on the usual (Regular) Tcoffee server you will only need to place the desired sequences in the appropriate window.
 1. Go to the main page of the Tcoffee server: www.tcoffee.org.
 2. Press the T-Coffee button.
 3. Put the sequences in the *Sequences to align* window.
 4. Click on *Submit*.
 5. Take a look at your results.

[illegible]

Combining sequences and structures with EXPRESSO

- EXPRESSO is the latest invention to Tcoffee, replacing 3D-Coffee (old version).
 - When you run Expresso, the program uses BLAST to search the PDB (protein structure database) for structures whose sequences are similar to used ones.
 - Then it uses these structures to support the alignment.
 - EXPRESSO is slower than Tcoffee, but if it finds enough structures, it produces the most accurate alignments of sequences currently known.
 - EXPRESSO compares structures with SAP (a Taylor & Orengo program), and sequences as well as structures using FUGUE (software developed at the University of Cambridge).
 - To launch EXPRESSO, click the EXPRESSO button on the www.tcoffee.org page.

Combining sequences and structures with EXPRESSO

- In the results section, look for a file called `template_list`.
 - Contains a list of all the structures that the program has been able to associate with the sequences you have submitted.
 - If this file is empty, then no structure matched the sequences used. Then the EXPRESSO alignment has just executed a standard alignment with Tcoffee.
- Tcoffee can assess the quality of your alignment to help understand which parts of it can be trusted.
- These assessments are empirical only - they don't substitute the E-value. However, it's useful to know that all bases or AAs shown on a yellow/orange/red background have an index above 5 (up to 10) - that's over an 80% probability of having been compared correctly.

Processing large data sets with MUSCLE

- MUSCLE is a new member of the family of multiple alignment programs, but it is very efficient software, both fast and high quality.
- MUSCLE is ideal if you want to do an alignment of several hundred sequences at once.
- You can run them from many servers, including a page at www.drive5.com/muscle.
- Working with MUSCLE is very simple - all you have to do is paste your sequences into the appropriate window.

Interpreting your multiple alignment

- E-values, which indicate how reliable a similarity search is, do not yet exist for multiple alignments, so assessing their correctness is often a matter of experience and guesswork.
- The alignments of DNA sequences are the most difficult to interpret.
 - A DNA block is informative only if it contains a group of several identical columns.

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Recognition of good sections in a protein alignment

- How to know if a block is good?
 - When you look at an alignment made with ClustalOmega, MUSCLE or Tcoffee, you will notice that the last line contains strange characters - (*), (:) or (.):
 - (*) asterisk indicates a fully conserved column.
 - (:) the two points denote a column in which all AA residues have approximately the same size and hydrophilicity/hydrophobicity properties.
 - (.) dot denotes a column in which size and hydrophilicity/hydrophobicity properties have been conserved in evolution
- For example, an average good block is a compact unit of at least 10-30 AAs with no discontinuities, having at least 1-3 (*), a few more (:) located near the asterisks, and also a few dots scattered throughout.

Recognition of good sections in a protein alignment

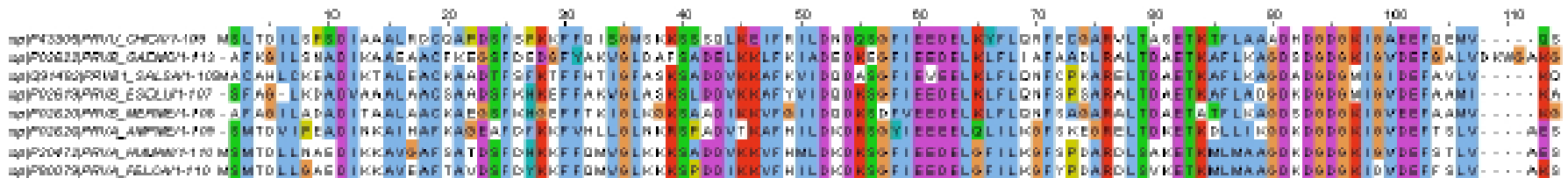
- The magic of multiple alignments is that 4 or 5 conserved positions out of 50+ compared AAs may be enough to get a true signal.
 - That's under 10% identity!
- Another criterion for a useful multiple alignment is to know which types of amino acids you should expect to be conserved.
- conserved columns in a multiple alignment only matter when the surrounding columns are not conserved.
 - The following table lists the most common properties associated with some conserved columns that you will encounter in your alignment.

Recognition of good sections in a protein alignment

| Amino-acid | Characteristic |
|------------|--|
| W, Y, F | Conservative columns with tryptophan meet very often. It has a large hydrophobic residue that is ofetdeep in the core of proteins and is very important for thier stability. When tryptophan mutates (which is rare), it is usually replaced by another hydrophobic AA, such as phenylalanine or tyrosine. Conservative columns with aromatic AAs are the most common signs to recognise protein domain. |
| G, P | Conservative columns of glycine or proline are common in multiple alignment. These two AAs often coincide with the ends of well-structured beta chains or alpha coils. |
| C | Cysteine is known for its ability to make C-C (disulfide) bridges. Conservative columns of cysteines are commonly found and usually indicate such bridges, and when at a certain distance give a good cue for the recognition of protein domains and folds. |
| H, S | Histidine and serine are often participants in catalytic sites, especially of proteases. These AAs are good candidates for active site participants. |
| K, R, D, E | Charged AA lysine, arginine, aspartic acid and glutamic acid are often involved in ligand binding. Highly conserved columns involving them may also signify a salt bridge in the core of the protein. |
| L | Leucines are rarely very conservative unless involved in protein-protein interactions, e. g. "leucine zipper" |

Recognition of good areas in a protein alignment

- Multiple alignment indicates important positions in proteins, the conserved amino acids - amino acids that are not mutated even in distantly related proteins.



- One possible strategy to show the evolutionary constraints on our protein is to include in the alignment exactly those sequences that the BLAST search output as borderline hits in the homology search.

Recognition of good areas in a protein alignment

1. Go to <http://www.uniprot.org/batch/>
2. In the **Provide your identifiers** window, enter the following sequence access numbers:
 - *P20472, P80079, P02626, P02619, P43305, Q91482, P02620, P02622, P02586.*
3. Click on **Submit** button.
4. Download the sequences in FASTA format (fasta file)
5. Open FASTA file and copy.

Recognition of good areas in a protein alignment

- When you've collected all the sequences, it's time for alignment again.
- 1. Go to EBI's ClustalW server:
<http://www.ebi.ac.uk/Tools/msa/clustalo/>
- 2. Place the collected sequences in the **Sequence** window.
- 3. Select the Output Format you want.
- 4. Select Input from the Output Order menu.
- 5. Press the Submit button.
- 6. Save your results.

Recognition of good areas in a protein alignment

- Multiple alignment of long-distance proteins.

CLUSTAL O(1.2.4) multiple sequence alignment

```

sp|P20472|PRVA_HUMAN      -----MSMTDLLNAE-DIKKAV 16
sp|P80079|PRVA_FELCA      -----MSMTDLLGAE-DIKKAV 16
sp|P02626|PRVA_AMPME      -----SMTDVIPEA-DINKAI 15
sp|P02619|PRVB_ESOLU      -----SFAG-LKDA-DVAAAL 14
sp|P43305|PRVU_CHICK      -----MSLTDLSPS-DIAAAL 16
sp|Q91482|PRVB1_SALSA      -----MACAHLCKEA-DIKTAL 16
sp|P02620|PRVB_MERME      -----AFAGILADA-DITAAL 15
sp|P02622|PRVB_GADMC      -----AFKGILSNA-DIKAAE 15
sp|P02586|TNNC2_RABIT     MTDQQAEARSYLSEEMIAEFKAAFDMFDADGGGDISVKELGTVMRMLGQTPTTKEELDAAI 60
                                     ::

sp|P20472|PRVA_HUMAN      G--AFSATDSFDHKKFFQMV-----GLKKKSADDVKKVFHMLDKDKSGFIEEDELGFIL 68
sp|P80079|PRVA_FELCA      E--AFTAVDSFDYKKFFQMV-----GLKKKSADDVKKVFHMLDKDKSGFIEEDELGFIL 68
sp|P02626|PRVA_AMPME      H--AFKAGEAFDFKKFVHLL-----GLNKRSPADVTKAFHILDKDRSGYIEEELQLIL 67
sp|P02619|PRVB_ESOLU      A--ACSAADSFKHKFFFAKV-----GLASKSLDDVKKAFYVIDQDKSGFIEEDELKLFL 66
sp|P43305|PRVU_CHICK      R--DCQAPDSFSPKFFQIS-----GMSKSSSLKLEIFRILDNQSGFIEEDELKYFL 68
sp|Q91482|PRVB1_SALSA      E--ACKAADTFSPKTFHTI-----GFASKSADDVKKAFKVIDQDASGFIEVEELKLFL 68
sp|P02620|PRVB_MERME      A--ACKAEGSFKHGFEFFTKI-----GLKGKSADIKKVFGIIDDQDKSDFVEEDELKLFL 67
sp|P02622|PRVB_GADMC      A--ACFKEGSFDEDDGFYAKV-----GLDAFSADELKKLKFIADEDKEGFIEEDELKLFL 67
sp|P02586|TNNC2_RABIT     EEVDEDGSGTIDFEFFLVMMVRQMKEDAKGKSEELAECEFRIFDRNADGYIDAEELAEI- 119
                                     ::      *      .      *      :      *      *      :      :      :      :      :      :      :

sp|P20472|PRVA_HUMAN      KGFSPDARDLSAKETKMLMAAGDKDGDGKIGVDEFSTLVAES---- 110
sp|P80079|PRVA_FELCA      KGFYPDARDLSVKETKMLMAAGDKDGDGKIDVDEFFSLVAKS---- 110
sp|P02626|PRVA_AMPME      KGFSGREGRELTDRKETDLLIKGDKDGDGKIGVDEFTSLVAES---- 109
sp|P02619|PRVB_ESOLU      QNFSPSARALTDKETKAFKADGDKDGDGMIGVDEFAAMIKK---- 107
sp|P43305|PRVU_CHICK      QRFECGARVLTASETKTFLAAADHDGDGKIGAEFFQEMVQS---- 109
sp|Q91482|PRVB1_SALSA      QNFCPKARELTDAETKAFKAGDADGDMIGIDEFAVLVQK---- 109
sp|P02620|PRVB_MERME      QNFSAGARALTDKETATFLKAGSDGDGKIGVEEFAAMVKG---- 108
sp|P02622|PRVB_GADMC      IAFAADLRALTDKETKAFKAGSDGDGKIGVDEFGALVDKNGAKG 113
sp|P02586|TNNC2_RABIT     --FRASGEHVTDEEIESLMKDGDKNNDRIDFDEFKMMEGVQ--- 160
                                     *      .      :      *      :      .      *      :      :      :      :      :      :

```


Recognition of good areas in a protein alignment

- At this point, it would be a good idea to add another long-range sequence to the resulting alignment.
 - The aim is to verify that these small highly conserved regions are indeed conserved throughout the protein family.
- To do this, you need to go through the two previous algorithms again, this time using one more sequence:
 - P20472, P80079, P02626, P02619, P43305, Q91482, P02620, P02622, P02586, P19123

Internet resources for multiple alignment

- Large number of online resources for multiple alignment.
- When choosing which one to work with, consider the following advices:
 - Use reliable resources and always do some simple test to see if they do what you expect them to do.
 - Never blindly trust resources over which you have no control.
 - If you want to do a lot of alignments, or for some reason don't want to "post" your sequences on the Internet, you will need to install such programs locally on your computer.

Discover your preferred method for multiple alignment

- The table below lists addresses of multiple alignment servers.

| Method | Description/ Advantage | Web page |
|---------|--|--|
| Tcoffee | Exact combination of sequences and structures | www.tcoffee.org http://tcoffee.vital-it.ch/cgi-bin/Tcoffee/tcoffee_cgi/index.cgi http://www.ebi.ac.uk/Tools/msa/tcoffee/ |
| MUSCLE | Fast and accurate method | www.drive5.com/muscle/ http://www.ebi.ac.uk/Tools/msa/muscle/ |
| Kalign | Fast method | http://msa.sbc.su.se/cgi-bin/msa.cgi |
| MAFFT | Fast and accurate method using - Fast Fourier Transforms | http://align.bmr.kyushu-u.ac.jp/mafft/software/ |
| Dialign | Ideal for sequences with local homology | http://bibiserv.techfak.uni-bielefeld.de/dialign/ |

Comparison of sequences on which no alignment can be made

- Sometimes it is necessary to compare sequences that do not necessarily have a common origin or are so distantly related that they are difficult to recognize as homologs.
- In such situations, do not expect miracles from bioinformatics. In most cases, your first impression that there is nothing to get out of these sequences turns out to be correct.
 - Gibbs sampler, examines multiple sequences simultaneously for short, partially preserved segments without breaks.
 - Pratt, searches for flexible models - a special category of segments that may contain breaks and are conserved on only certain positions.

How to make multiple local alignments with Gibbs sampler

- Gibbs sampler is a stochastic method:
 - first shuffles the sequences,
 - then compares them randomly until a good result appears, then continues shuffling to further improve that result.
- Stochastic methods are the most powerful in bioinformatics so far.
 - For example, if you analyze the same set of sequences twice with the Gibbs sampler, you may not get exactly the same results.
- However, if you assume slightly more unusual logic you may find that the Gibbs sampler provides quite good solutions for extremely complex problems.

<http://ccmbweb.ccv.brown.edu/gibbs/gibbs.html>

Search for conserved motives

- The Gibbs sampler is only useful if the sequences you are looking for have the same length, as is the case with Helix-Turn-Helix domains.
- If your sequences contain a similar motif but are not or very weakly linked, the only way to analyze them is to use a pattern-finding motif.
- There are several tools to detect such common poorly preserved motifs.
 - One of the most powerful is Pratt because it allows some flexibility of space between conserved positions. You can also use TEIRESIAS, MEME or SMILE.

Resources for searching motifs and patterns

- The table below lists some of the most frequently used online resources for searching common motifs in sequences that are too poorly related to have multiple alignments.

| Method | Address |
|---------------|--|
| Gibbs Sampler | http://mobyli.pasteur.fr/cgi-bin/portal.py http://ccmbweb.ccv.brown.edu/gibbs/gibbs.html |
| Pratt | http://www.ebi.ac.uk/Tools/pfa/pratt/ |
| eMotif | http://motif.stanford.edu/projects.html |
| MEME | http://meme.sdsc.edu/meme/ |
| TEIRESIAS | http://cbcsrv.watson.ibm.com/Tspd.html |
| Improbizer | http://users.soe.ucsc.edu/~kent/improbizer/improbizer.html |

Thank you!



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness



Module 3. Bioinformatic tools

Topic 4. Artificial Neural Networks

Lesson 1. General Concepts of ML and ANN

Artificial Neural Networks

Lesson 1

General - Introduction

Εισαγωγή στο AI, ML και DL

Τεχνητή νοημοσύνη (TN):

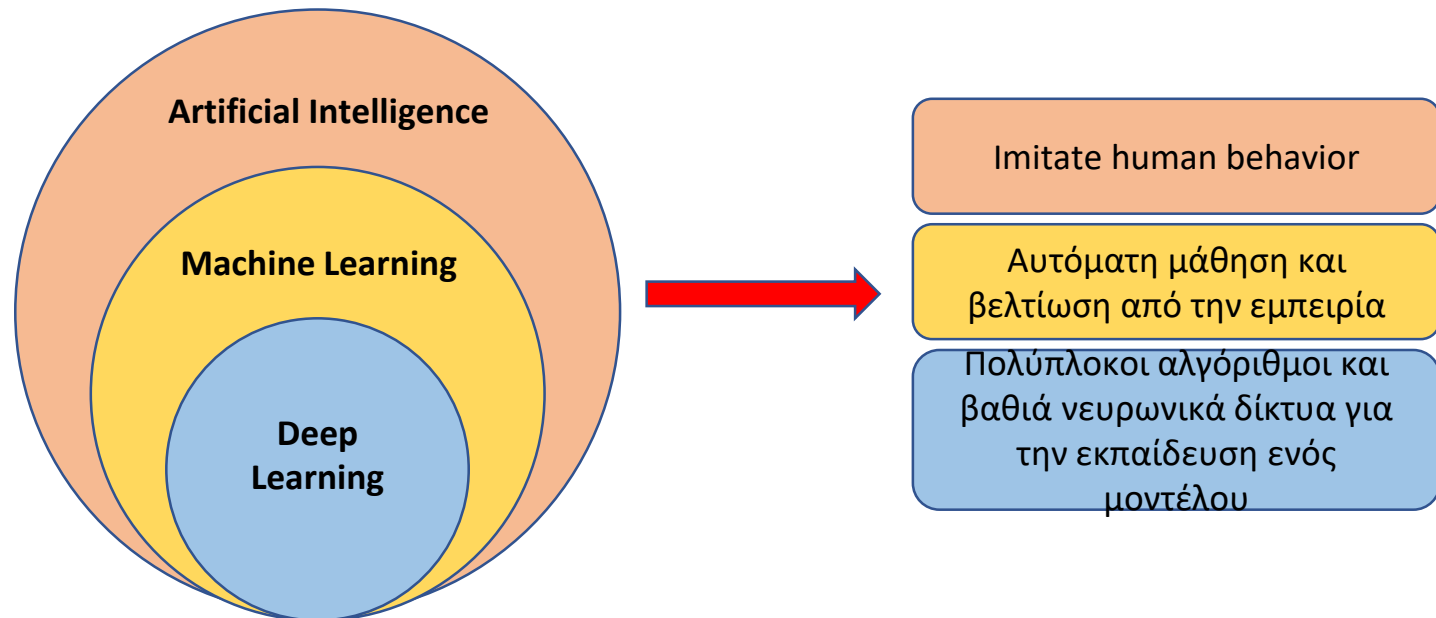
- Η ευρύτερη έννοια ότι οι μηχανές είναι σε θέση να εκτελούν εργασίες με τρόπο που θα θεωρούσαμε «έξυπνο» ή «έξυπνο».

Μηχανική Μάθηση (MM):

- A subset of AI that allows computers to learn from data, instead of being explicitly programmed to perform a task

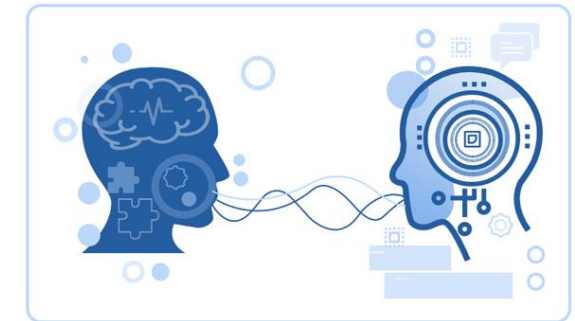
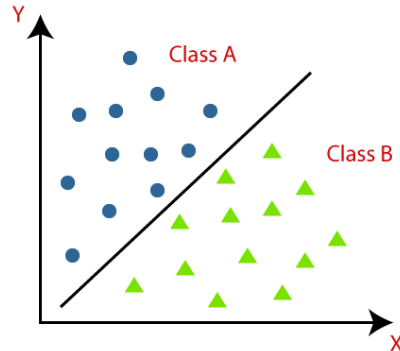
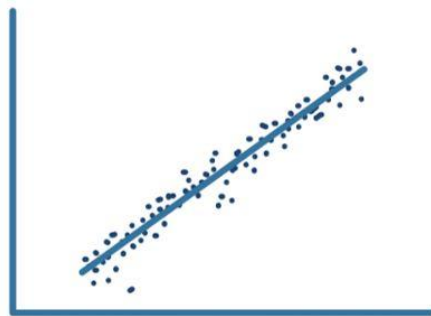
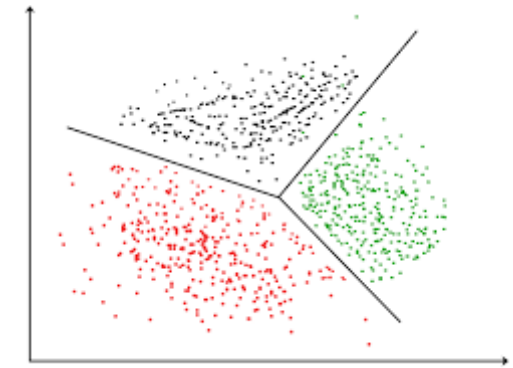
Βαθιά Μάθηση (BM):

- A subset of ML that uses neural networks with many layers (hence "deep").



Problems Addressed by Machine Learning

- Regression
- Classification
- Image recognition
- Natural Language
- Clustering
- Reinforcement Learning
- Anomaly Detection
- Recommendation Systems

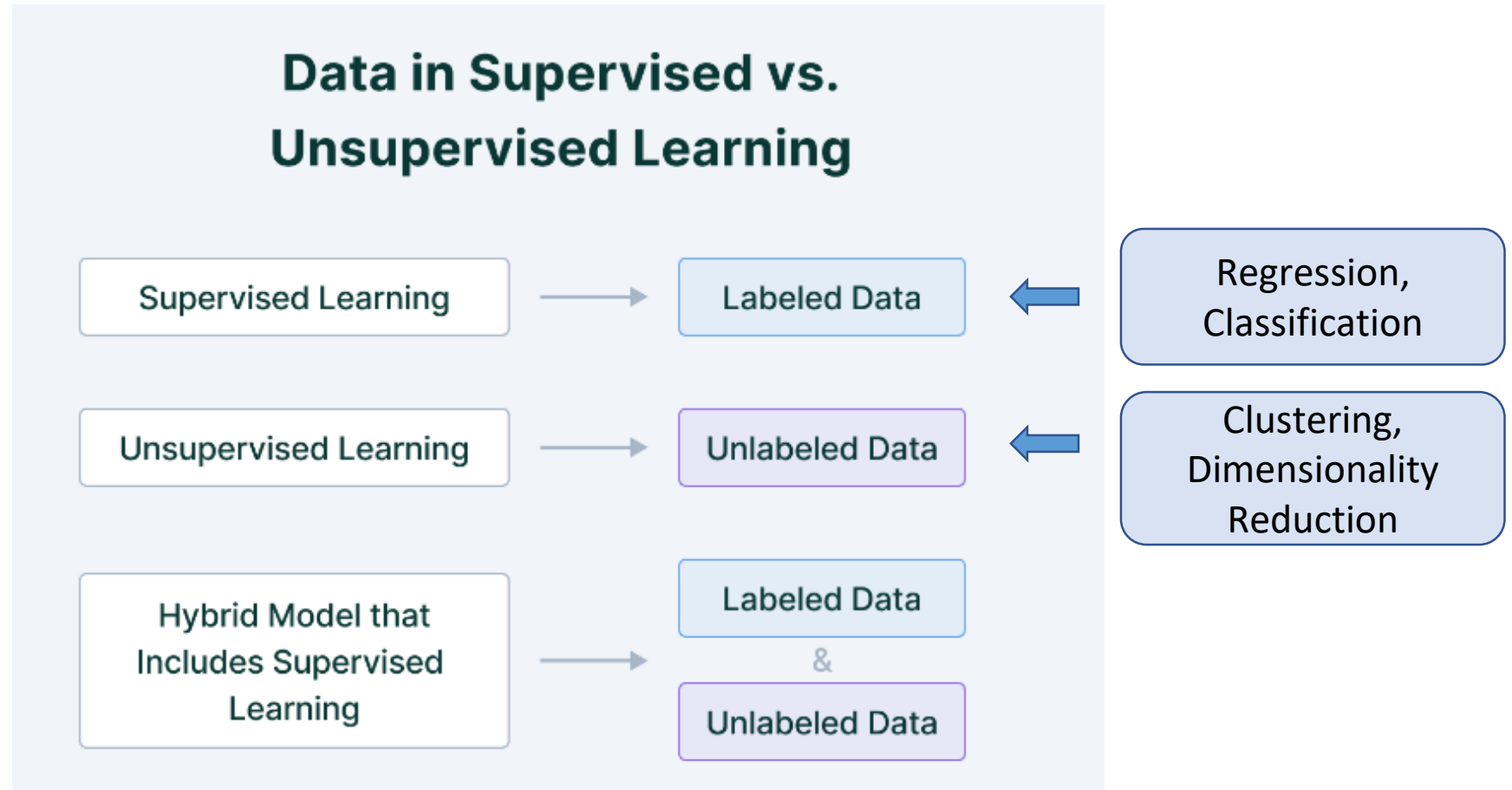


ERASMUS+

Supervised vs Unsupervised Learning

Key Differences:

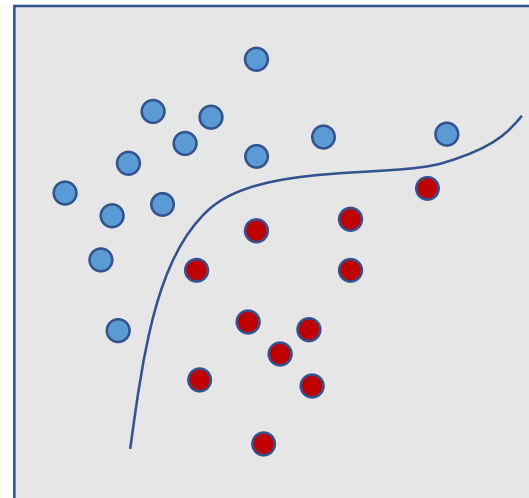
1. **Data Type:** Supervised uses labeled data, while unsupervised uses unlabeled data.
2. **Goal:** Supervised aims to predict outcomes, while unsupervised aims to find patterns or groupings.
3. **Feedback Loop:** Supervised has a feedback loop (correction based on label), while unsupervised does not.



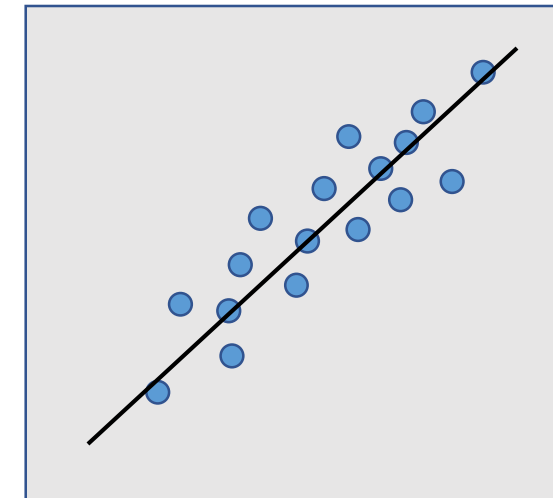
Regression and Classification

Key Differences:

1. **Output Type:** Classification predicts categories, while regression predicts numerical values
2. **Evaluation Metrics:** Classification uses accuracy, precision, recall, etc., while regression uses mean squared error, R-squared, etc.



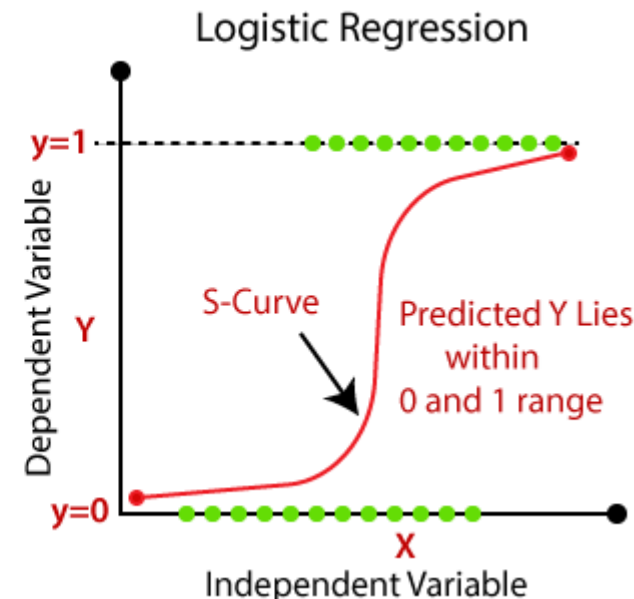
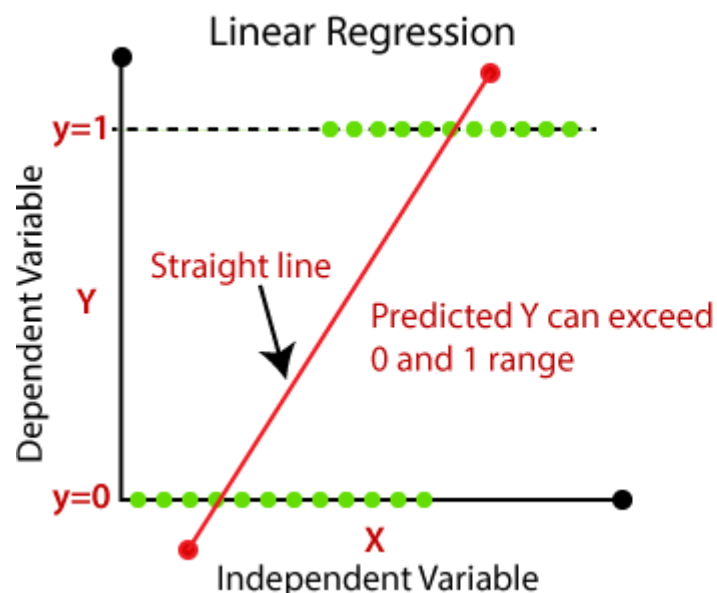
Classify data points in
different categories



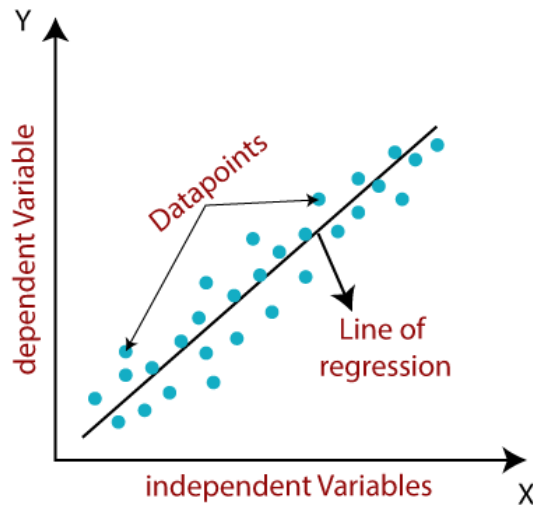
Predict the value of the dependent
variable given a series of independent
variables

Linear and Logistic Regression (1)

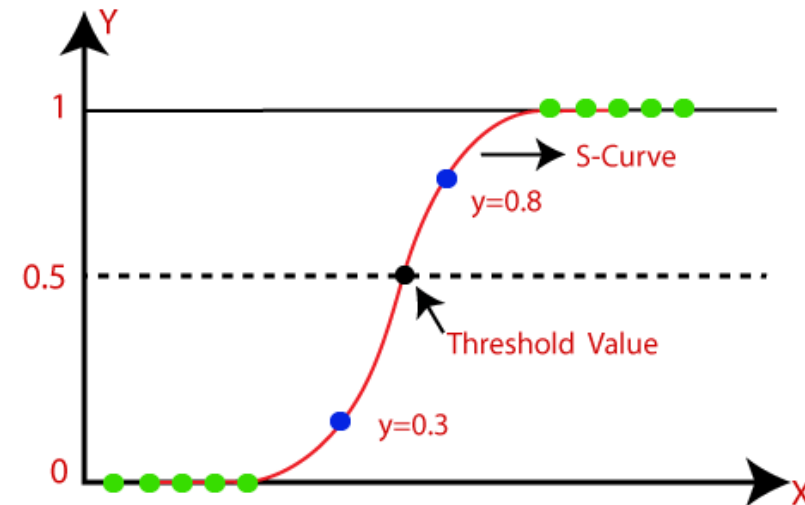
- *Linear regression* is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables
- *Logistic regression* estimates the probability of an event occurring, based on a given dataset of independent variables



Linear and Logistic Regression (2)



$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots$$



$$\log\left(\frac{y}{1-y}\right) = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots$$

Find coefficients $a_i, i = 1, 2, \dots$ that best describe the problem

Image Recognition Classification

Detection



bicycle

cat

strawberr

y



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

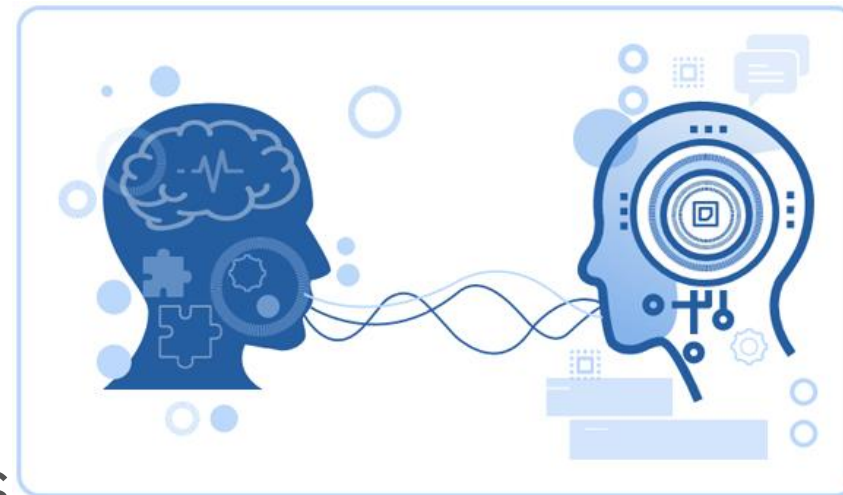
Action Type KA226 - Partnerships for Digital Education Readiness

Natural Language Processing (NLP)

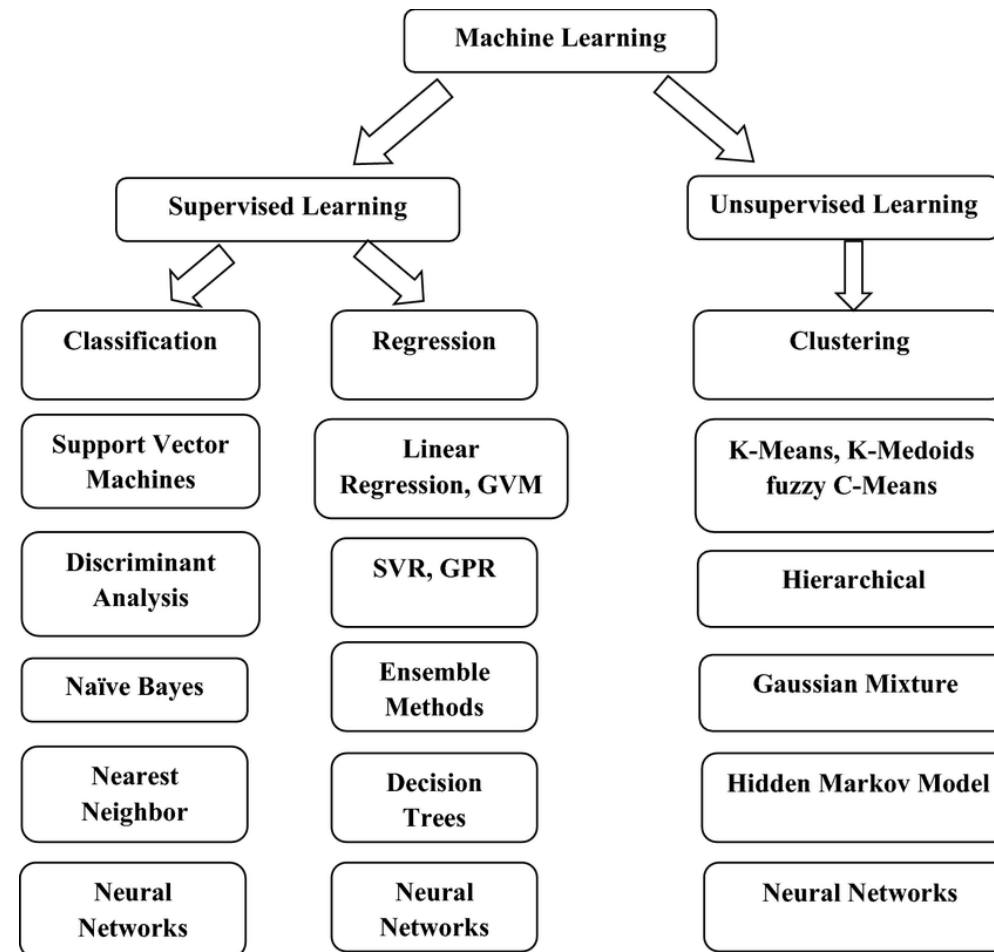
Speech recognition, Grammatical tagging, Word sense disambiguation, Sentiment analysis, Natural language generation

Common Tasks:

- Spam detection
- Machine translation
- Virtual agents and chatbots
- Social media sentiment analysis
- Text summarization

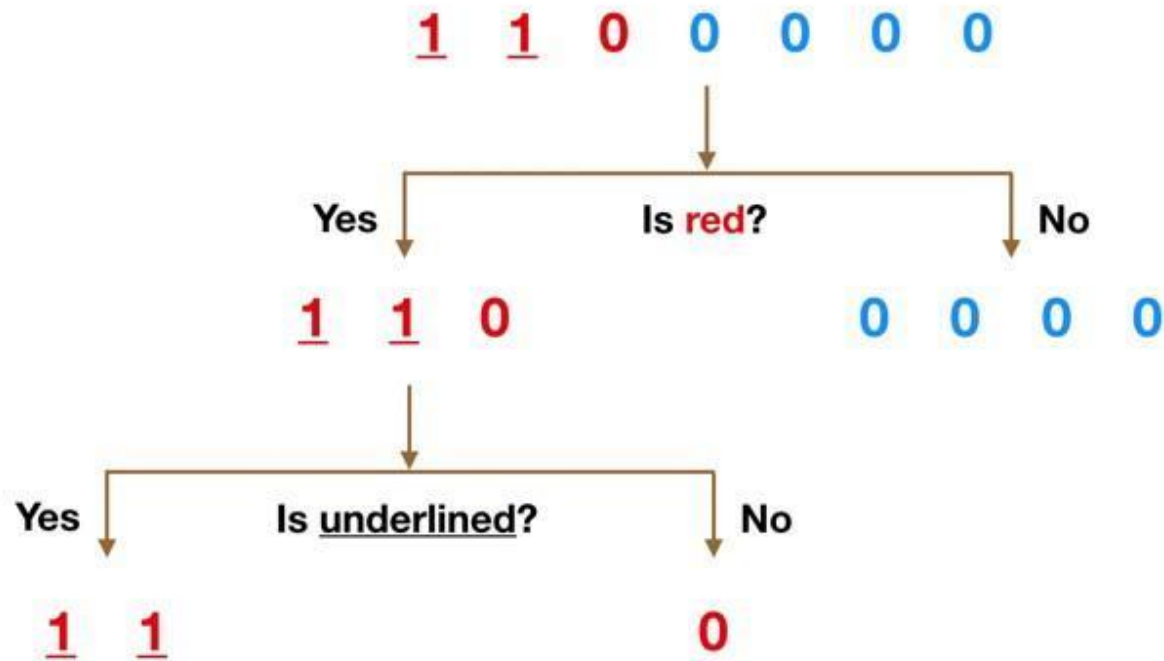


Machine Learning Models

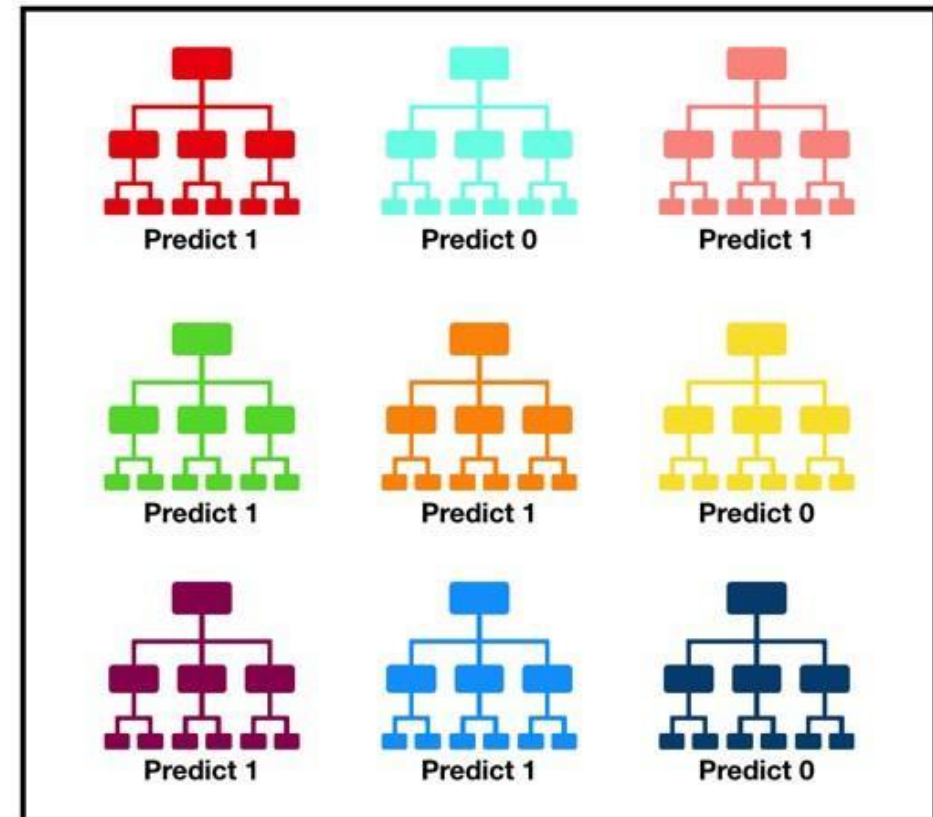


Decision Trees – Random Forests

Decision
Tree

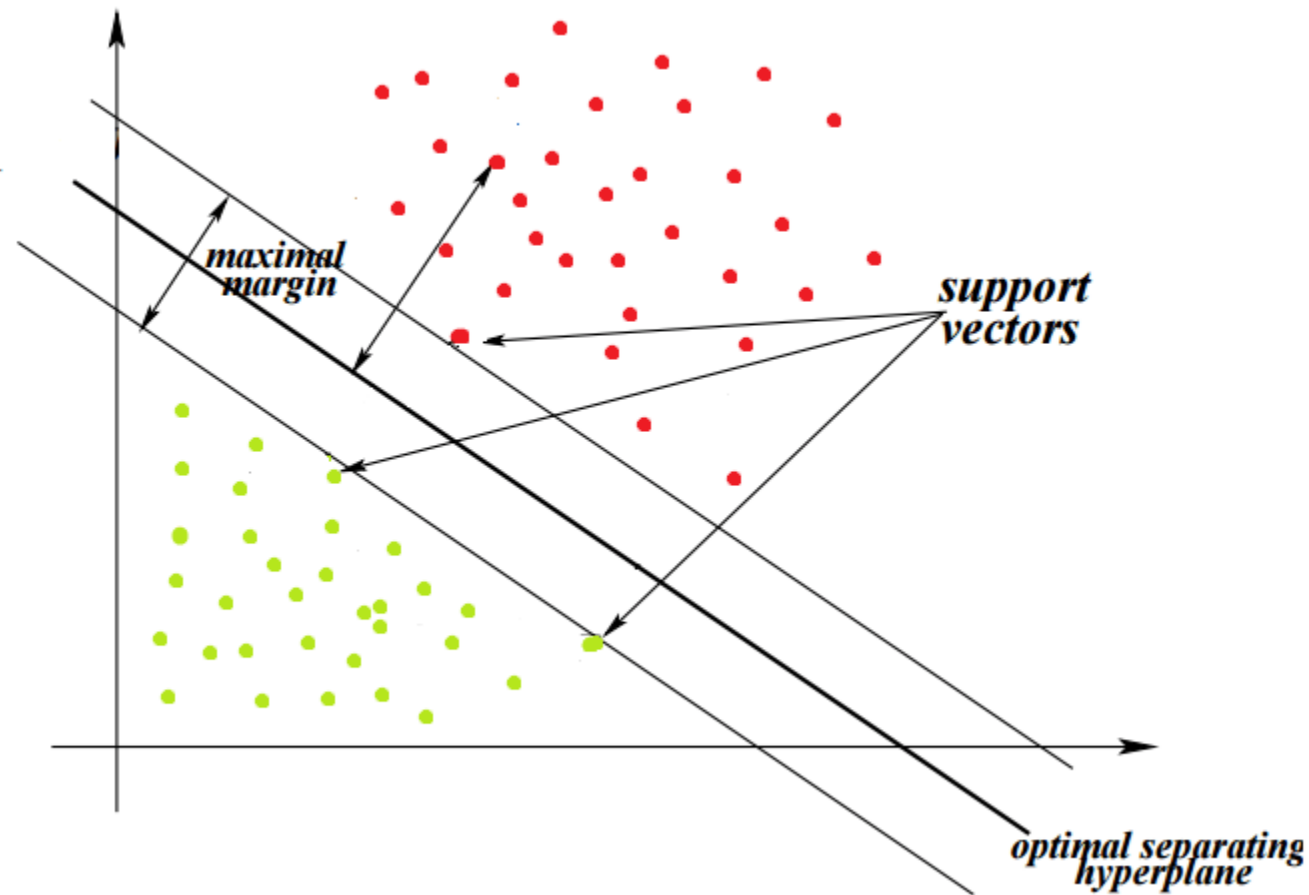


Random Forrest

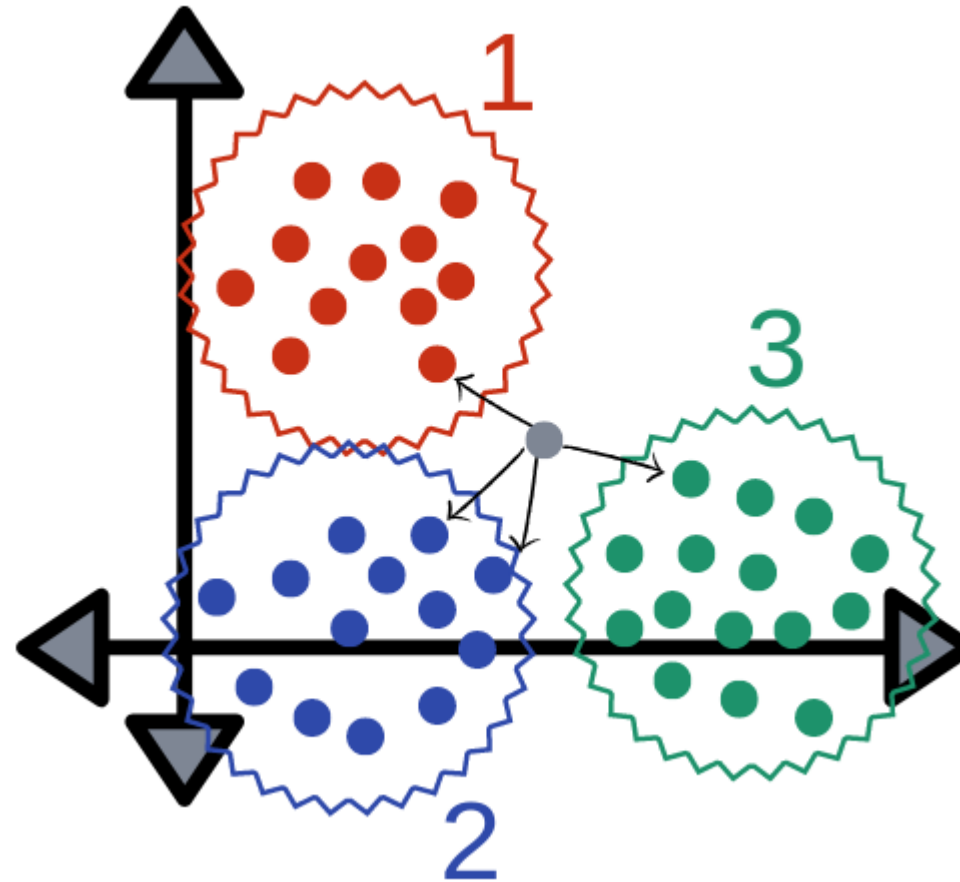


Tally: Six 1s and Three 0s
Prediction: 1

Support Vector Machines (SVM)

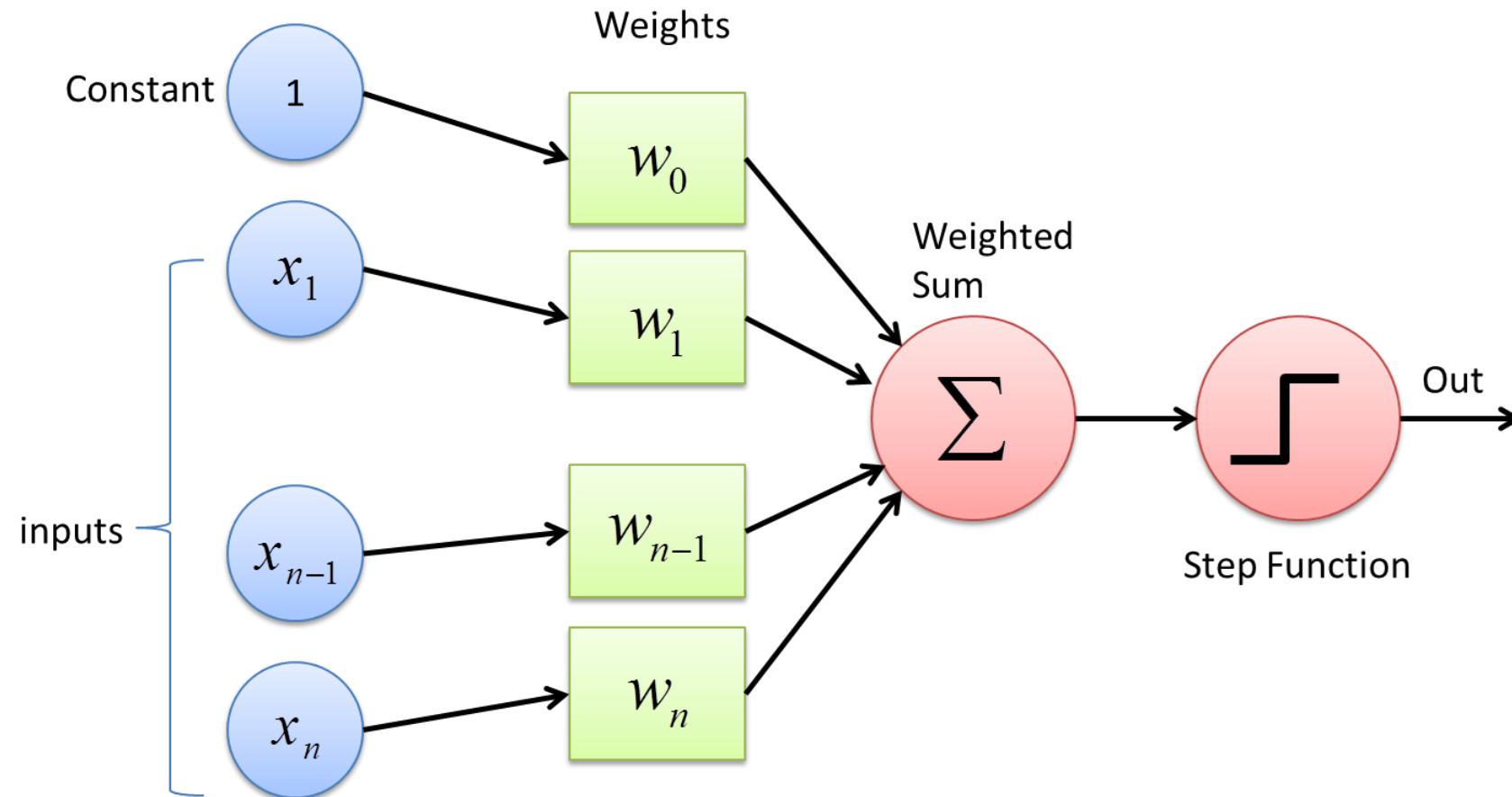


k Nearest Neighbors (kNN)

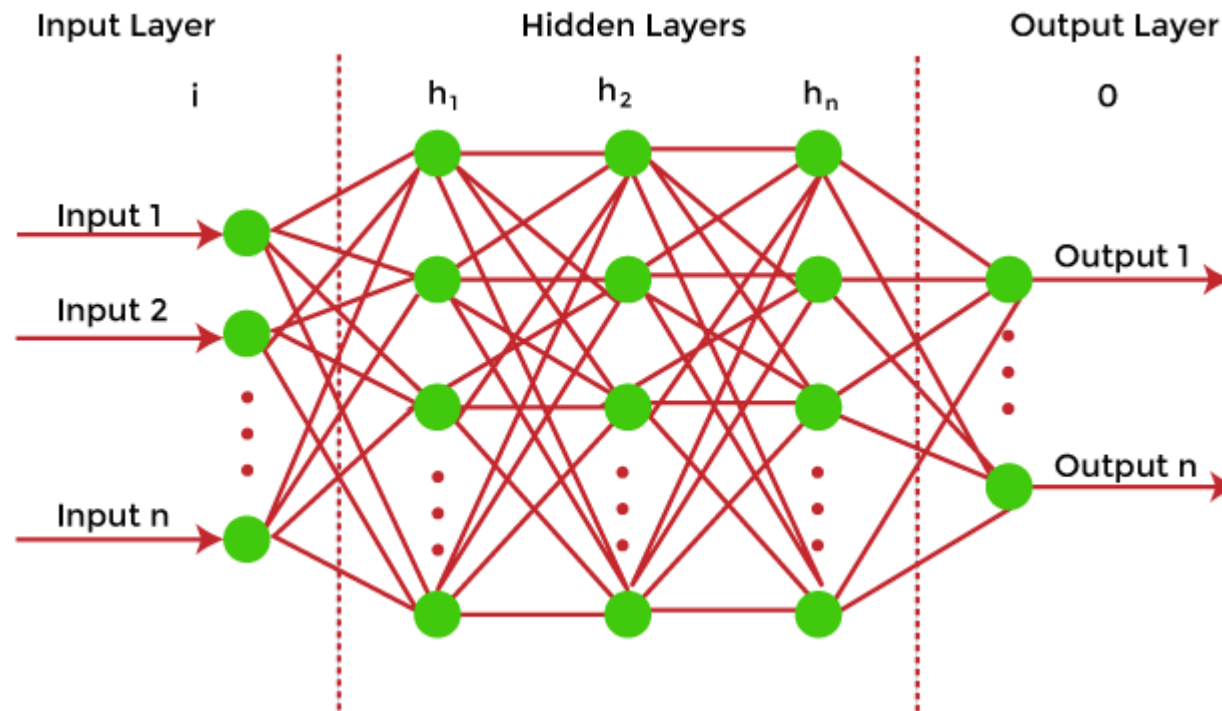


ERASMUS+

Perceptron



Multi Layer Perceptrons (MLP)



Ensemble Methods in Machine Learning

Types of Ensemble Methods:

1. Bagging (Bootstrap Aggregating):

- Uses different subsets of training data to train multiple models.
- Averages the result (for regression) or takes a majority vote (for classification).
- Example: Random Forests.

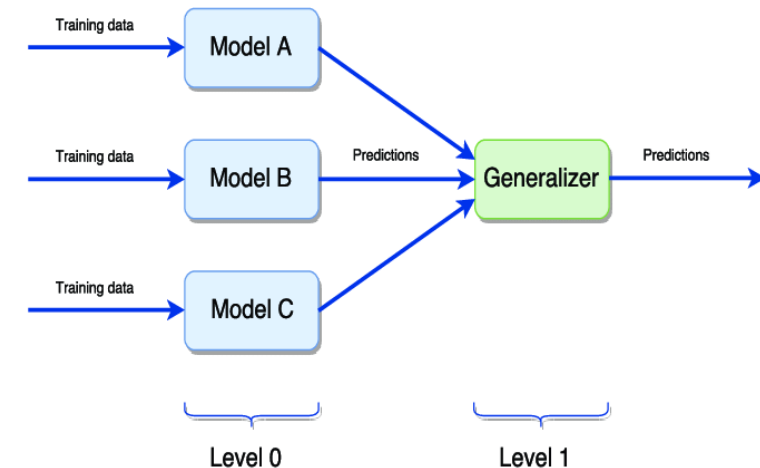
2. Boosting:

- Trains models sequentially, where each model corrects the errors of its predecessor.
- Focuses on instances that were misclassified by previous models.
- Examples: AdaBoost, Gradient Boosting, XGBoost.

3. Stacking:

- Combines multiple classification or regression models via a meta-classifier or a meta-regressor.
- The base level models are trained on a complete training set, while the meta-model is trained on the outputs of the base level model as features.

- **Definition:** Ensemble methods combine multiple machine learning models to produce one optimal predictive model.
- **Why Use Ensemble?:**
 - Improve accuracy.
 - Reduce overfitting.
 - Enhance model robustness.



Conclusion and Resources

- **Recap:**

1. Explored the foundational concepts of AI, ML, and DL.
2. Delved into the core problems ML aims to solve.
3. Unpacked key ML techniques and their applications.

- **Why ML Matters:**

1. Machine Learning is at the heart of many modern technological advancements.
2. Its applications span various domains, from healthcare to finance, making it a crucial skill for the future.

- **Resources for Further Learning:**

1. **Books:**

- "Pattern Recognition and Machine Learning" by Christopher Bishop.
- "The Hundred-Page Machine Learning Book" by Andriy Burkov.

2. **Online Courses:**

- Coursera: "Machine Learning" by Andrew Ng.
- Udacity: "Intro to Machine Learning".

3. **Websites:**

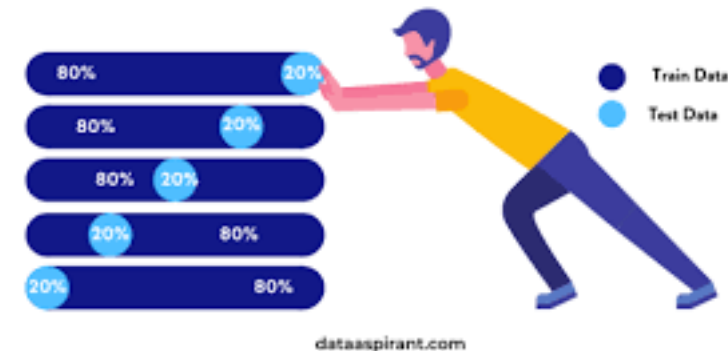
- [Towards Data Science](#)
- [Machine Learning Mastery](#)

Validation, Metrics, and Feature Engineering in AI

An Introduction to Model Evaluation and Optimization

| | | Predicted | |
|--------------|----------|----------------|----------------|
| | | Positive | Negative |
| Ground-Truth | Positive | True Positive | False Negative |
| | Negative | False Positive | True Negative |

Cross Validation



Introduction to Model Evaluation and Optimization

1. Model Evaluation:

- The process of determining how well a machine learning model performs on unseen data.
- Essential for understanding the model's generalization capabilities.

2. Model Optimization:

- Refining the model to improve its performance.
- Involves techniques like hyperparameter tuning, feature selection, and more.

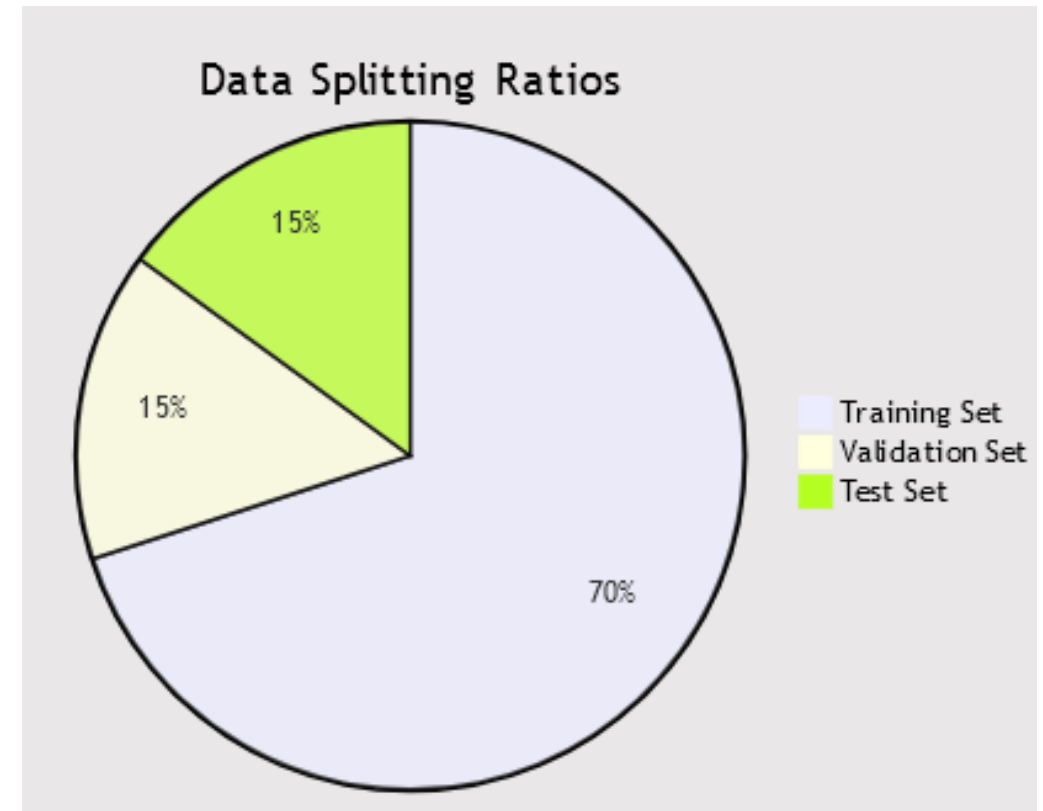
3. Relevance to Artificial Neural Networks (ANNs):

- ANNs are powerful but can easily overfit or underperform without proper evaluation and optimization.

○ Proper evaluation ensures that the neural network model is robust and reliable.
Just as building a model is crucial, evaluating its performance and optimizing it for better results is equally important, especially in the context of complex models like ANNs.

Data Splitting

1. **Importance of Data Splitting:**
 - Ensures that the model can generalize well to new, unseen data.
 - Helps in assessing the real-world performance of the model.
2. **Training Set:**
 - Used to train the machine learning model.
 - Typically the largest portion of the dataset.
3. **Validation Set:**
 - Used to tune hyperparameters and make decisions about the model's architecture.
 - Helps in preventing overfitting during the training phase.
4. **Test Set:**
 - Used to evaluate the model's final performance after training and validation.
 - Should never be used during the training phase to ensure unbiased evaluation.



Proper data splitting is foundational for building a reliable and robust machine learning model. It ensures that we have a fair assessment of our model's capabilities and weaknesses

Understanding Validation and Test Sets

1. Validation Set:

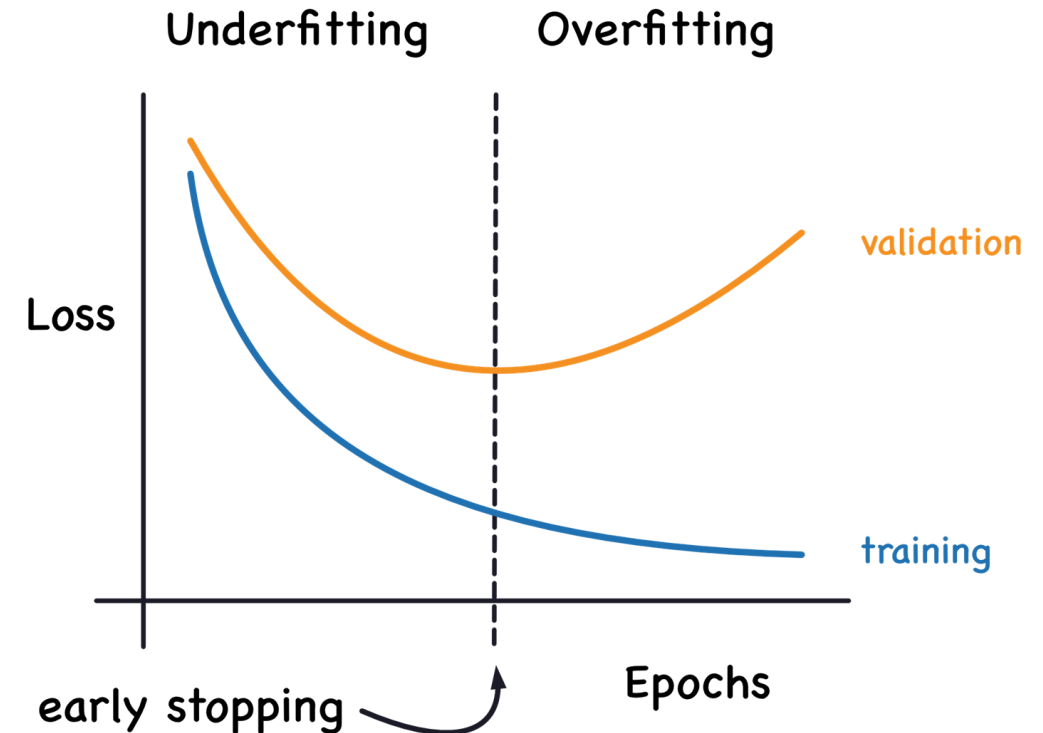
- Definition: A subset of the dataset used to evaluate the performance of the model during training and make decisions about its architecture.
- Purpose: Helps in tuning model parameters, selecting features, and preventing overfitting.

2. Test Set:

- Definition: A subset of the dataset used to assess the model's performance after it has been trained and validated.
- Purpose: Provides an unbiased evaluation of the model's final performance on unseen data.

3. Overfitting vs. Underfitting:

- Overfitting: When the model performs exceptionally well on the training data but poorly on new, unseen data.
- Underfitting: When the model performs poorly on both the training data and new data.
- Importance: Proper use of validation and test sets helps in identifying and preventing both overfitting and underfitting.



Validation and test sets play a crucial role in ensuring the robustness and reliability of machine learning models. They help in achieving a balance between model complexity and performance.

Cross Validation

1. What is Cross Validation?

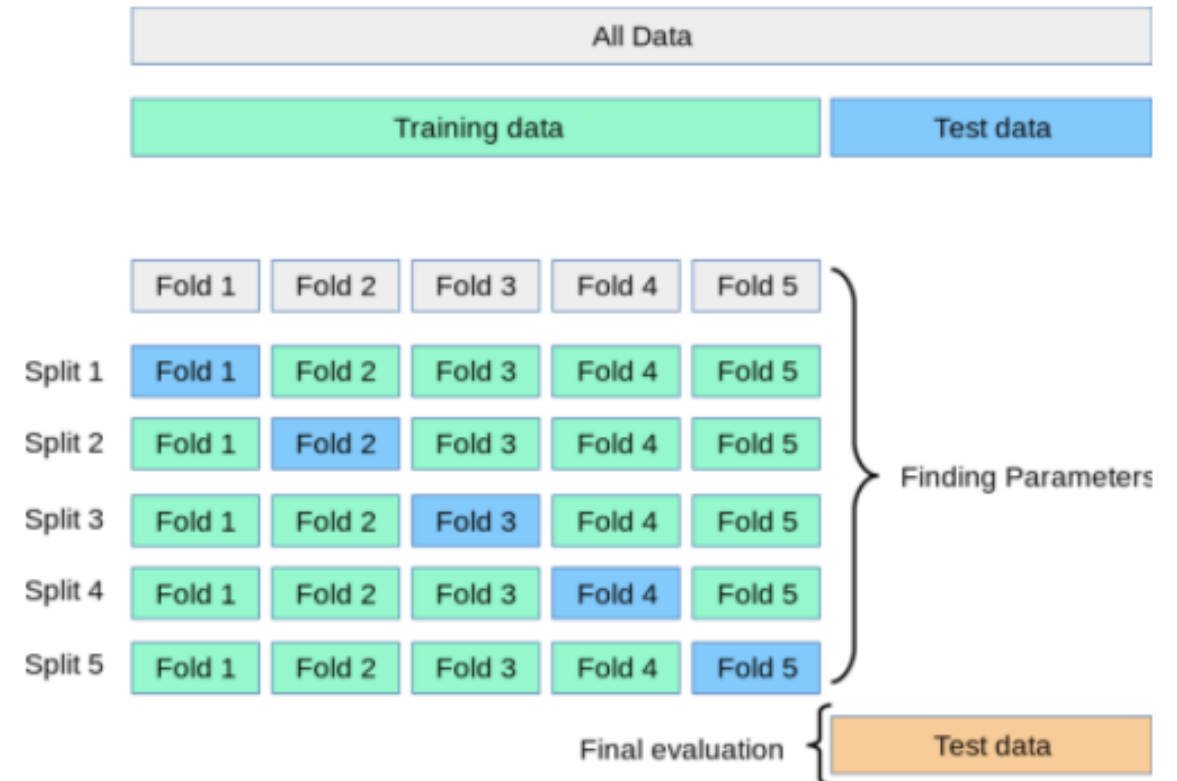
- Definition: A technique used to assess the performance of machine learning models by partitioning the original sample into a training set to train the model, and a validation set to evaluate it.
- Purpose: To reduce overfitting and obtain a more generalized model.

2. K-Fold Cross Validation:

- Explanation: The original sample is randomly partitioned into 'K' equal-sized subsamples. Of the 'K' subsamples, a single subsample is retained as validation data, and the remaining 'K-1' subsamples are used as training data. The process is repeated 'K' times.

3. Benefits of Cross Validation:

- More reliable estimate of model performance.
- Utilizes the entire dataset for both training and validation.
- Helps in identifying the best model hyperparameters.



Cross Validation is a robust method for evaluating the performance of machine learning models, ensuring that the model is well-trained and **generalized**, and not just **memorizing** the training data.

Benefits of Cross Validation

1. **Robust Model Evaluation:**
 - Provides a more comprehensive assessment of a model's performance than a single train-test split.
 - Helps in understanding the model's behavior across different subsets of data.
2. **Mitigates Overfitting:**
 - By training on different subsets and validating on different sets, the chances of the model memorizing specific data patterns are reduced.
 - Ensures the model generalizes well to unseen data.
3. **Optimal Utilization of Data:**
 - Every data point gets to be in both training and validation sets.
 - Especially beneficial when the dataset size is limited.
4. **Hyperparameter Tuning:**
 - Helps in identifying the best set of hyperparameters for the model.
 - Can be combined with grid search or random search for optimal model parameters.

Cross Validation is not just a technique for model evaluation; it's a powerful tool that ensures the creation of a robust, well-generalized, and optimized machine learning model.

Metrics Introduction

1. Why Metrics?

- Importance: Metrics provide quantitative measures to assess the performance of machine learning models.
- Role: They guide the optimization process and help in model selection.

2. Types of Metrics:

- **Classification Metrics:** Used for models that predict categorical outcomes. Examples include Accuracy, Precision, Recall, F1-score, etc.
- **Regression Metrics:** Used for models that predict continuous outcomes. Examples include Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, etc.

3. Choosing the Right Metric:

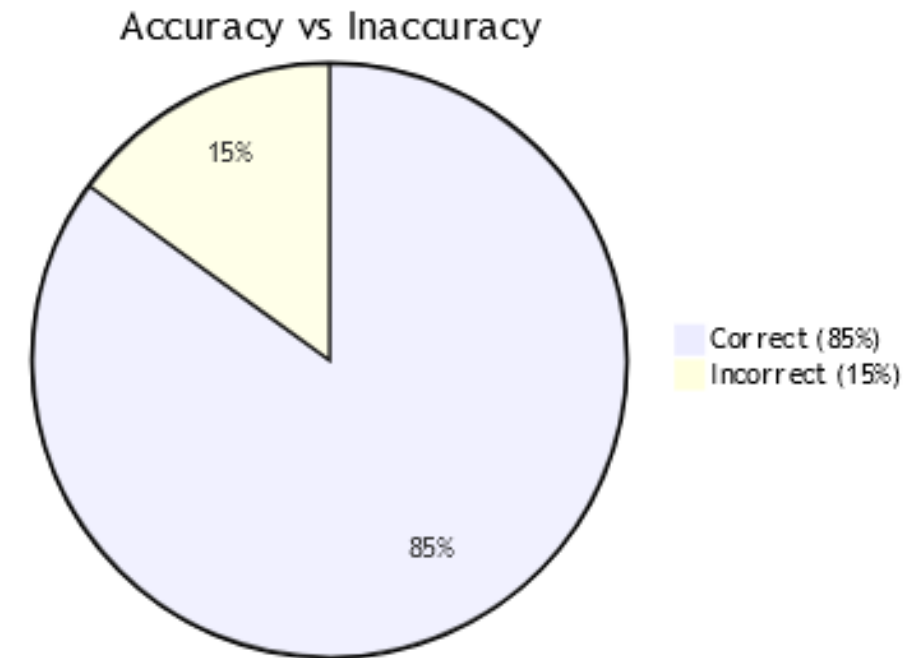
- Depends on the problem at hand: For instance, in medical diagnoses, false negatives might be more costly than false positives.
- Balancing trade-offs: Precision vs. Recall, Bias vs. Variance.

Model evaluation metrics are crucial in the machine learning pipeline. They provide insights into the model's strengths and weaknesses, guiding improvements and ensuring the model aligns with the specific goals of the task.

Accuracy

- Definition of Accuracy:**
 - Formula: $\text{Accuracy} = (\text{Number of Correct Predictions}) / (\text{Total Number of Predictions})$
 - Interpretation: Proportion of total predictions that are correct.
- When to Use Accuracy:**
 - Suitable for balanced datasets where the number of samples in each class is roughly the same.
 - Provides a quick snapshot of overall model performance.
- Limitations of Accuracy:**
 - Can be misleading for imbalanced datasets.
 - Doesn't provide insight into the types of errors made (false positives vs. false negatives).

Accuracy is a straightforward and commonly used metric. However, it's essential to be aware of its limitations and ensure it's the right metric for the task at hand.



Sensitivity (Recall) and Specificity

1. Definitions:

- **Sensitivity (True Positive Rate):** The ability of a model to correctly identify positive cases among all actual positive cases.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

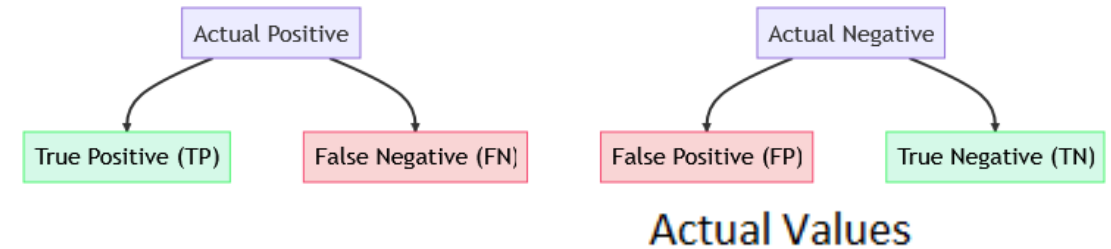
- **Specificity (True Negative Rate):** The ability of a model to correctly identify negative cases among all actual negative cases.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

2. Importance:

- **Medical Tests:** Sensitivity is crucial to ensure that diseases are not missed (**minimizing false negatives**), while Specificity ensures that healthy individuals are not misdiagnosed (**minimizing false positives**).
- **Imbalanced Datasets:** In datasets where one class significantly outnumbers the other, Sensitivity and Specificity provide a clearer picture of model performance than Accuracy.

Sensitivity and Specificity are essential metrics, especially in scenarios where the consequences of misclassification are significant. They provide a more nuanced understanding of a model's performance, especially in the context of binary classification problems.



| | | Positive (1) | Negative (0) |
|------------------|--------------|--------------|--------------|
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

Precision and Recall

Definitions:

- **Precision:** Out of all the predicted positive instances, how many were actually positive? Formula: $\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$

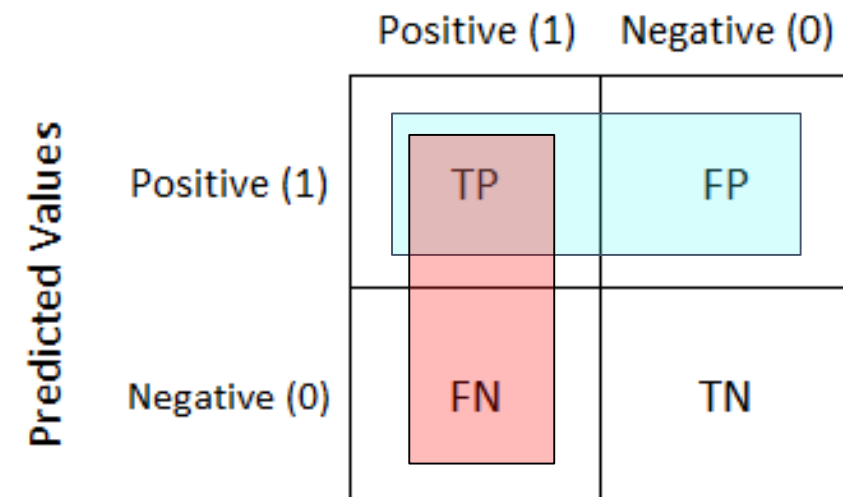
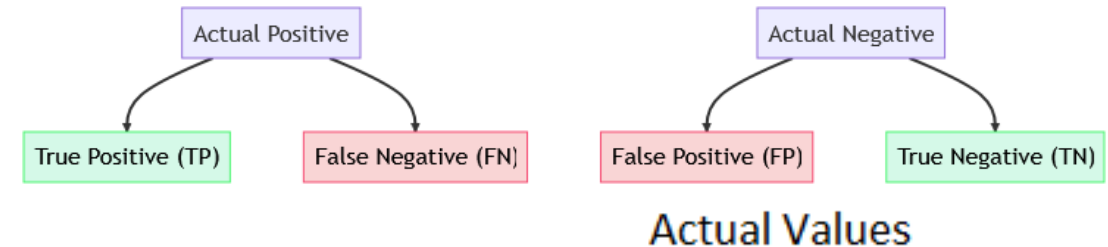
Balancing Act:

- Precision and Recall (Sensitivity) are often inversely related. Improving one might reduce the other.
- The choice between prioritizing Precision or Recall depends on the specific problem and its implications.

Use Cases:

- **High Precision:** When the cost of False Positives is high. E.g., Email spam detection where legitimate emails being classified as spam can be problematic.
- **High Recall:** When the cost of False Negatives is high. E.g., Disease diagnosis where missing an actual positive case can have severe consequences.

Precision and Recall are crucial metrics that provide insights into the performance of classification models, especially in imbalanced datasets. Understanding their definitions, trade-offs, and implications is essential for building and evaluating robust models.



F1-Score

1. Definition of F1-Score

- Formula: $F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$

- Interpretation: Harmonic mean of Precision and Recall, providing a balance between the two.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (1)$$

2. Why F1-Score?

- Importance in imbalanced datasets: When one class significantly outnumbers the other, Accuracy can be misleading. F1-Score provides a more balanced measure.
- Combines both false positives and false negatives into a single metric.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

3. When to Use F1-Score:

- When both false positives and false negatives are crucial.
- When there's an uneven class distribution.

Dimensionality Reduction

1. What is Dimensionality Reduction?

- Definition: The process of reducing the number of random variables under consideration by obtaining a set of principal variables.
- Purpose: Simplify data without losing much information, making it easier to visualize, analyze, and process.

2. Why Dimensionality Reduction?

- **Data Visualization:** Reducing dimensions (e.g., from hundreds to 2 or 3) can help in visualizing data patterns.
- **Speed up Algorithms:** Less data means faster training times for machine learning models.
- **Avoid Overfitting:** Reducing redundant features can help in preventing models from overfitting.

3. Common Techniques:

- **PCA (Principal Component Analysis):** Linear technique that finds the axes in the data that maximize variance.
- **t-SNE (t-Distributed Stochastic Neighbor Embedding):** Non-linear technique suitable for visualizing high-dimensional data in 2D or 3D.
- **LDA (Linear Discriminant Analysis):** Supervised method used to find a linear combination of features that best separates two or more classes.

Dimensionality Reduction is a powerful tool in data science and machine learning, aiding in data visualization, speeding up algorithms, and preventing overfitting. It's essential to choose the right technique based on the nature and purpose of the data.

Feature Selection

1. What is Feature Selection?

- Definition: The process of selecting a subset of relevant features (variables, predictors) for use in model construction.
- Purpose: Improve model performance, reduce overfitting, and enhance model interpretability.

2. Why Feature Selection?

- **Enhanced Performance:** By removing irrelevant or redundant features, models can often perform better.
- **Faster Training:** Fewer features mean reduced computational cost and faster training times.
- **Improved Interpretability:** A simpler model with fewer features is easier to understand and interpret.

3. Common Techniques:

- **Filter Methods:** Based on the intrinsic properties of the data. Examples include correlation coefficient scores and chi-squared tests.
- **Wrapper Methods:** Use a subset of features and train a model using them. Examples include forward feature selection, backward feature elimination, and recursive feature elimination.
- **Embedded Methods:** Algorithms that have built-in feature selection methods. Examples include LASSO and decision trees.

Feature Selection is a critical step in the machine learning pipeline. It not only enhances model performance but also ensures that models are efficient and interpretable. The choice of feature selection method should align with the nature of the data and the specific goals of the task.

Benefits of Dimensionality Reduction & Feature Selection

1. **Enhanced Model Performance:**
 - Removing irrelevant or redundant data can lead to better model accuracy and generalization.
 - Helps in reducing the noise in the data.
2. **Efficiency & Speed:**
 - Reduced computational costs due to fewer data points and features.
 - Faster training and prediction times for machine learning models.
3. **Improved Data Visualization:**
 - Lower-dimensional data can be easily visualized, aiding in understanding patterns and relationships.
 - Helps in identifying clusters or groups in the data.
4. **Reduced Risk of Overfitting:**
 - Simplified models with fewer features are less likely to overfit to the training data.
 - Ensures that the model captures the underlying patterns rather than noise.
5. **Enhanced Interpretability:**
 - Simpler models are easier to understand and explain.
 - Important for applications where model transparency is crucial, such as in medical or financial domains.

Both Dimensionality Reduction and Feature Selection play pivotal roles in the data preprocessing phase. They ensure that machine learning models are not only accurate but also efficient, interpretable, and generalizable.

Integrating with Neural Networks

1. **Pre-Processing for Neural Networks:**

- Neural networks often benefit from simplified and standardized input data.
- Dimensionality reduction and feature selection can serve as preprocessing steps before feeding data into a neural network.

2. **Enhanced Training Efficiency:**

- Reduced data dimensions and features can lead to faster convergence during training.
- Helps in efficient utilization of computational resources, especially in deep learning models with numerous parameters.

3. **Improved Generalization:**

- Simplified data can reduce the risk of overfitting in neural networks.
- Ensures that the network learns meaningful patterns rather than noise.

4. **Feature Extraction Layers:**

- Some neural network architectures, especially convolutional neural networks (CNNs), inherently perform feature extraction.
- Combining traditional feature selection with neural-based feature extraction can lead to robust models.

5. **Embeddings & Latent Representations:**

- Neural networks, especially autoencoders, can be used for dimensionality reduction by learning compact latent representations of the data.
- These embeddings can capture non-linear relationships in the data.

Integrating dimensionality reduction and feature selection with neural networks can lead to efficient, accurate, and robust models. The synergy between traditional data simplification techniques and neural architectures can unlock powerful modeling capabilities.

Summary

1. **Understanding Data Complexity:**
 - Emphasized the importance of understanding the complexity and dimensions of our data.
 - Recognized that not all features or dimensions are equally informative.
2. **Dimensionality Reduction & Feature Selection:**
 - Highlighted the significance of simplifying data through dimensionality reduction and selecting relevant features.
 - Discussed various techniques and their applications.
3. **Metrics & Model Evaluation:**
 - Revisited the importance of various metrics like Accuracy, Sensitivity, Specificity, and F1-Score.
 - Stressed the need for comprehensive model evaluation.
4. **Neural Network Integration:**
 - Discussed the synergy between data simplification techniques and neural network architectures.
 - Explored the potential of combining traditional methods with deep learning for robust modeling.
5. **The Road Ahead:**
 - Encouraged continuous exploration and adaptation of techniques based on specific data and problem requirements.
 - Emphasized the dynamic nature of the AI and machine learning field, urging the need for lifelong learning.

The journey through dimensionality reduction, feature selection, metrics, and neural network integration has equipped us with a comprehensive toolkit for AI and machine learning. As we move forward, it's essential to adapt, experiment, and continuously refine our approaches to stay at the forefront of this ever-evolving field.



Discussion

Recommended Books:

- "Pattern Recognition and Machine Learning" by Christopher M. Bishop
- "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Online Resources:

- Coursera: Machine Learning by Andrew Ng
- edX: Principles of Machine Learning
- Fast.ai: Practical Deep Learning for Coders

Research Papers:

- "A Few Useful Things to Know About Machine Learning" by Pedro Domingos
- "Understanding Machine Learning: From Theory to Algorithms" by Shai Shalev-Shwartz and Shai Ben-David

The journey of understanding and mastering AI and machine learning is continuous. There's always more to learn, explore, and discuss. Engage in discussions, dive into further readings, and stay updated to remain at the cutting edge of this exciting field.

Module 3. Bioinformatic tools

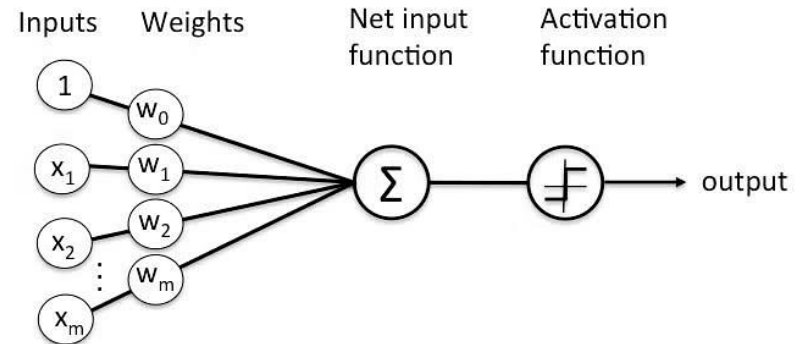
Topic 4. Artificial Neural Networks

Lesson 2. Perceptron

Introduction to The Perceptron

Welcome to the World of Perceptrons!

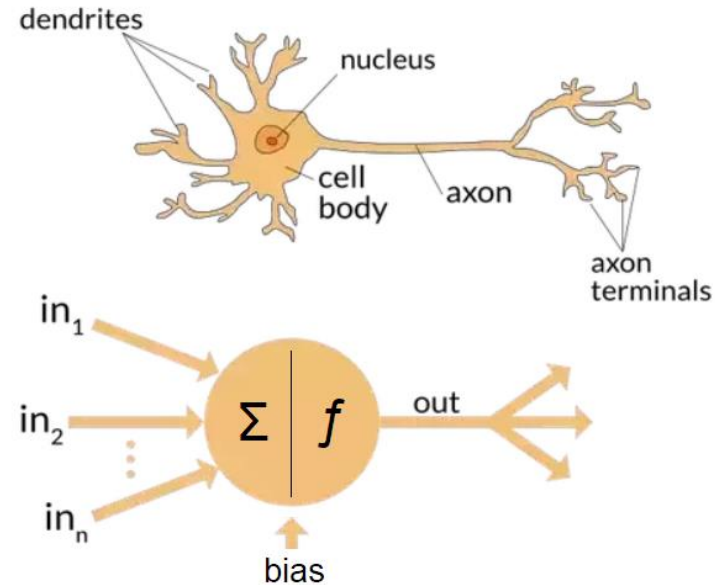
- **Definition:** The perceptron is one of the simplest artificial neural network architectures, serving as a fundamental building block for more complex neural networks.
- **Historical Context:** Introduced in the late 1950s by Frank Rosenblatt, the perceptron was inspired by the information processing of a single biological neuron.
- **Significance:**
 - Acts as a binary classifier, making decisions based on linear predictors.
 - Lays the foundation for multi-layer neural networks and deep learning.
 - Provides insights into the basic principles of machine learning, such as training, weights, and bias.



The perceptron is a foundational concept in neural networks and machine learning. Its simplicity offers clarity, making it an ideal starting point for understanding more complex neural architectures.

Inspiration - Biological Neuron

Nature's design of the biological neuron, with its intricate structure and sophisticated information processing capability, serves as the foundational inspiration for the perceptron in artificial neural networks.



Biological Neuron

- **What is a Neuron?**
 - Fundamental unit of the brain and nervous system.
 - Responsible for receiving, processing, and transmitting information through electrical and chemical signals.
- **Components of a Neuron:**
 - **Dendrites:** Branch-like structures that receive signals from other neurons.
 - **Cell Body (Soma):** Contains the nucleus and processes the received signals.
 - **Axon:** A long projection that transmits signals to other neurons or muscles.
 - **Synapses (axon terminals):** Junctions where the neuron connects to other neurons or cells, facilitating signal transmission.
- **Neuron's Working Mechanism:**
 - Receives signals through dendrites.
 - Processes signals in the cell body.
 - If the processed signal exceeds a certain threshold, the neuron fires, sending the signal down the axon to other neurons or cells.
- **Relation to Perceptron:**
 - The perceptron mimics the functioning of a biological neuron.
 - Inputs in a perceptron are analogous to dendrites, weights resemble synaptic strengths, and the activation function is inspired by the neuron's firing mechanism.

Bridging Biology and Computation

The Inspiration:

- The perceptron's design is directly inspired by the biological neuron's structure and function.
- Both systems process inputs, weigh their importance, and produce an output based on a threshold.

Key Analogies:

- **Dendrites ↔ Inputs:** Just as dendrites receive signals in a neuron, perceptrons receive multiple input values.
- **Synaptic Strengths ↔ Weights:** In neurons, synapses have varying strengths. Similarly, perceptrons assign different weights to inputs.
- **Neuron Firing ↔ Activation Function:** A neuron fires if the input signal surpasses a threshold. In a perceptron, the activation function determines if the combined input exceeds a certain value to produce an output.

The perceptron, as a foundational concept in artificial neural networks, beautifully encapsulates the essence of the biological neuron. This synergy between biology and computation has paved the way for the advancements we see in modern AI and deep learning.

Differences:

- While biological neurons use electrochemical signals, perceptrons use mathematical functions.
- The perceptron's design is a simplified abstraction of the biological neuron, focusing on the core principles of input processing and output generation.

Evolution of the Concept:

- The perceptron laid the groundwork for more advanced neural network architectures.
- Modern neural networks consist of multiple layers of interconnected perceptrons (or neurons), allowing for complex data processing and pattern recognition.

Anatomy of a Perceptron

Core Components:

- **Inputs (x_1, x_2, \dots, x_n):** Values received by the perceptron, analogous to the signals received by a neuron's dendrites.
- **Weights (w_1, w_2, \dots, w_n):** Assigned to each input, determining the importance or influence of that input. Comparable to synaptic strengths in a neuron.
- **Bias (b):** A constant value added to the weighted sum of inputs, allowing the perceptron to adjust its output independently of its inputs.
- **Activation Function (f):** A mathematical function that processes the weighted sum of inputs and bias to produce the perceptron's output. Determines whether the perceptron "fires" or not.

The perceptron's structure, though simple, encapsulates the core principles of neural processing. Each component plays a crucial role in determining the perceptron's output, making it a fundamental building block for more complex neural network architectures.

The Perceptron in Action

- **Working Mechanism:**
 - a. The perceptron computes the weighted sum of its inputs and adds the bias.
 - b. This sum is then passed through the activation function to produce the output.
- **Mathematical Representation:**
 - a. $\text{Output} = f(\sum (\text{weight} * \text{input}) + \text{bias})$
 - b. For instance, with a step function as the activation function, the perceptron outputs 1 if the sum exceeds a threshold, and 0 otherwise.

The perceptron's decision-making process is a harmonious interplay of weighing inputs, summing values, and applying an activation function. This systematic approach allows it to classify inputs based on learned patterns.

Activation Function

What is an Activation Function?

- A mathematical equation that determines the output of a perceptron or neuron.
- It introduces non-linearity into the output of a neuron, enabling neural networks to learn from error and make adjustments, which is essential for learning complex patterns.

Role in a Perceptron:

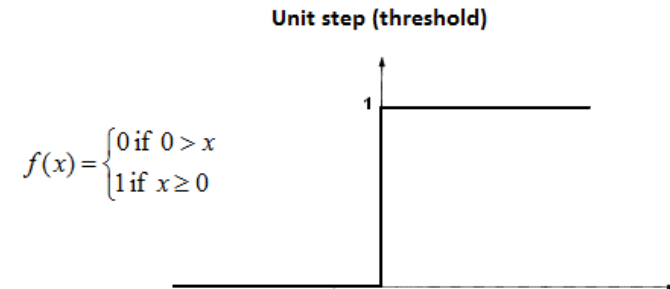
- Decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias.
- Helps in normalizing the output, ensuring it's in a format suitable for further layers or as a final output.

Why is it Important?

- Without an activation function, a perceptron would simply be a linear regression model, limiting its ability to learn and perform more complex tasks.
- It allows perceptrons to capture non-linear patterns in the data.

The Step Function:

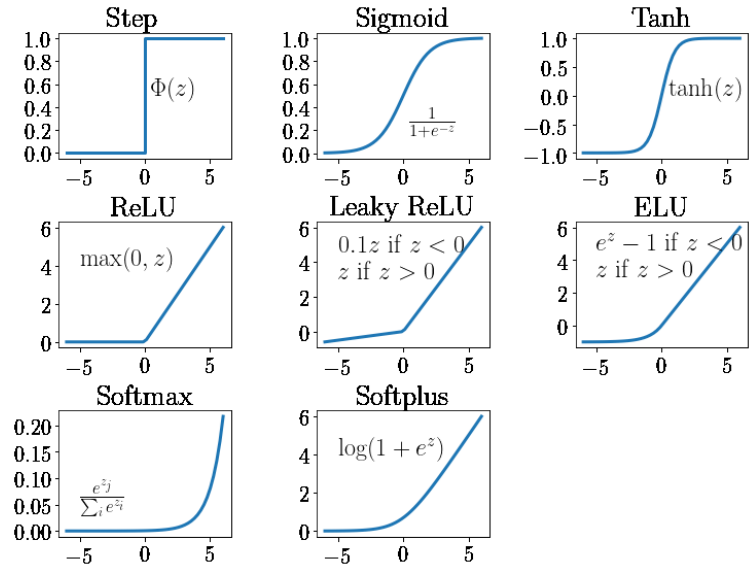
- The simplest form of activation function, producing a binary output based on a threshold.
- If the input value surpasses the threshold, the function outputs a 1 (or 'fires'). Otherwise, it outputs a 0



Activation functions play a pivotal role in shaping the output of a perceptron. They introduce the necessary **non-linearity**, enabling the perceptron to learn and adapt. The step function, as a basic example, offers a glimpse into the world of activation functions.

Common Activation Functions

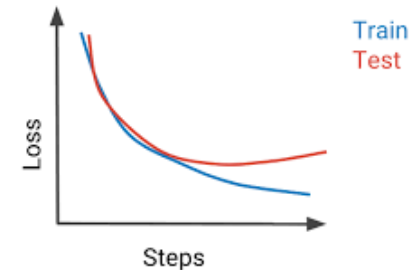
- **Step Function:**
 - Produces a binary output based on a threshold.
 - If the input value surpasses the threshold, the function outputs a 1. Otherwise, it outputs a 0.
- **Sigmoid Function:**
 - Outputs values between 0 and 1.
 - Smooth gradient, preventing sudden jumps in output values.
- **ReLU (Rectified Linear Unit):**
 - Allows positive values to pass through unchanged, while negative values are set to zero.
 - Commonly used in hidden layers of neural networks.
- **Tanh (Hyperbolic Tangent):**
 - Outputs values between -1 and 1.
 - Similar to the sigmoid but can output negative values.
- **Softmax:**
 - Used in the output layer of classification problems.
 - Converts a vector of numbers into a vector of probabilities.



Different activation functions serve different purposes and are chosen based on the specific requirements of a neural network model. Their diverse characteristics enable neural networks to capture a wide range of patterns and complexities in data.

The Role of Error (Loss) Functions

- **What is an Error (Loss) Function?**
 - A mathematical equation that quantifies how well the prediction of the model aligns with the actual data.
 - It measures the difference between the predicted output and the actual output.
- **Purpose in Neural Networks:**
 - Provides a metric to optimize during training.
 - The goal of training is to minimize this error, leading to better model prediction
- **Why is it Important?**
 - It guides the adjustment of model parameters (like weights and biases).
 - A well-chosen loss function can accelerate the training process and improve model accuracy.
- **Common Types:**
 - **Mean Squared Error (MSE):** Used for regression problems. It calculates the square of the difference between predicted and actual values.
 - **Cross-Entropy Loss:** Used for classification problems. It measures the difference between two probability distributions.



Error (Loss) functions play a crucial role in the training of neural networks. They provide a measure of how far off predictions are from actual values, guiding the adjustments made to the model's parameters for improved performance.

Common Activation Functions

- **Mean Squared Error (MSE):**

- **Usage:** Commonly used for regression problems.
- **Characteristics:** Penalizes large errors more due to the squaring of differences.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- **Cross-Entropy Loss (Log Loss):**

- **Usage:** Ideal for binary and multi-class classification problems.
- **Characteristics:** Measures the performance of a classification model whose output is a probability value between 0 and 1.

$$L(y, t) = - \sum_i t_i \ln y_i$$

- **Hinge Loss (Max Margin Loss):**

- **Usage:** Used for "maximum-margin" classification, mainly for support vector machines.
- **Characteristics:** Intended for use with binary classification where the target values are in the set $\{-1, 1\}$.

$$\ell(y) = \max(0, 1 - t \cdot y)$$

- **Huber Loss:**

- **Usage:** Useful when outliers are present, but a squared error is desired.
- **Characteristics:** Combines properties of MSE and mean absolute error.

$$L_{\delta}(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta \cdot (|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

Different activation functions serve different purposes and are chosen based on the specific requirements of a neural network model. Their diverse characteristics enable neural networks to capture a wide range of patterns and complexities in data.

Paving the Path to Learning: The Delta Rule

What is the Delta Rule?

- Also known as the Widrow-Hoff learning rule: A gradient descent learning rule used for adjusting the weights of a perceptron.

Purpose in Neural Networks:

- Provides a mechanism to minimize the error of the perceptron by adjusting its weights.
- Helps the model converge to the optimal solution by iteratively reducing the error.

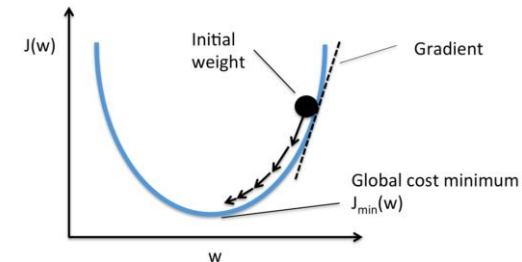
How Does It Work?

- The weights are adjusted in the direction that decreases the error.
- The change in weights is proportional to the negative gradient of the error with respect to the weights.

The Delta Rule is a foundational learning algorithm for perceptrons, guiding the adjustments made to the weights to minimize the error. By following the gradient of the error, it ensures that the perceptron moves towards better performance with each iteration.

$$\Delta w = \alpha(t-o)x$$

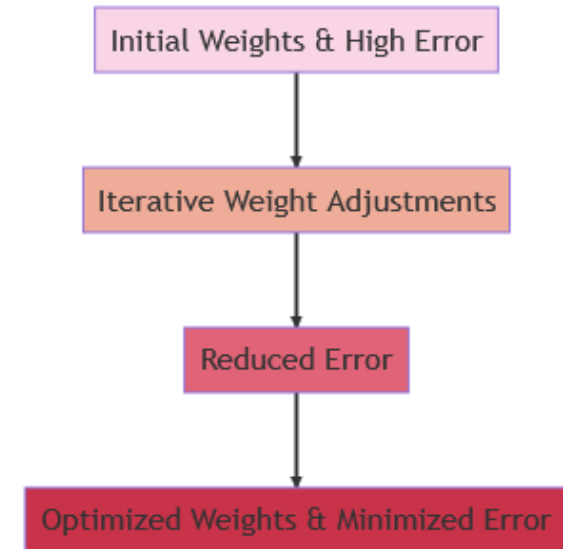
- Where:
 - Δw is the weight change.
 - α is the learning rate.
 - t is the target output.
 - o is the actual output.
 - x is the input.



Delta Rule in Action

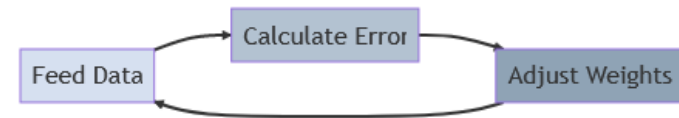
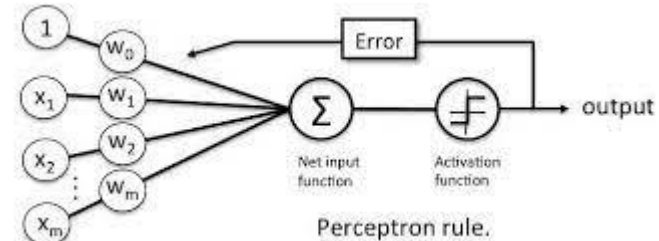
- **Iterative Learning:**
 - The Delta Rule is applied iteratively, adjusting weights after each training example.
 - Over time, these adjustments lead to a perceptron that makes more accurate predictions.
- **Weight Adjustment:**
 - If the perceptron produces an error (difference between predicted and actual output), the weights are adjusted using the Delta Rule.
 - The magnitude and direction of adjustment depend on the error and the input values.
- **Learning Rate (α):**
 - A hyperparameter that determines the step size during weight adjustment.
 - A high learning rate might overshoot the optimal solution, while a low learning rate might converge slowly.
- **Convergence:**
 - With the right learning rate and sufficient iterations, the Delta Rule ensures that the perceptron converges to a solution where the error is minimized.

The Delta Rule's practical application showcases its power in refining a perceptron's performance. Through iterative adjustments, it steers the perceptron towards an optimized state, ensuring that it learns and adapts to the training data effectively.



Training a Perceptron

- **Feeding Data:**
 - The perceptron is presented with training examples, which consist of inputs paired with correct outputs.
 - Each input is multiplied by a weight, and the results are summed up.
- **Calculating Error:**
 - Once the perceptron produces an output, it's compared to the correct output to determine the error.
 - The error is simply the difference between the predicted output and the actual output.
- **Adjusting Weights:**
 - Based on the calculated error, the perceptron's weights are adjusted.
 - The Delta Rule is applied to guide these adjustments, ensuring the perceptron moves towards better performance.
 - The process is repeated for multiple iterations or until the error is minimized to an acceptable level.



Training a perceptron is an iterative process that involves presenting data, evaluating performance, and refining the model. Through continuous adjustments, the perceptron learns to make accurate predictions, showcasing the power of machine learning.

Perceptron Convergence

Convergence Defined:

- The point at which the perceptron's weights stabilize, and the error no longer decreases significantly.
- Indicates that the perceptron has learned the patterns in the training data to the best of its ability.

Guaranteed Convergence:

- For **linearly separable** data, the perceptron is guaranteed to converge.
- The number of iterations required can vary based on the factors mentioned above.

Factors Influencing Convergence:

- **Learning Rate:** Determines the step size during weight adjustments. A suitable learning rate ensures steady convergence.
- **Training Data:** Quality and quantity of data can impact how quickly and effectively a perceptron converges.
- **Initial Weights:** Starting weights can influence the path and speed of convergence.

Challenges:

- For non-linearly separable data, a perceptron **may never** converge using standard training algorithms.
- This limitation led to the development of more advanced neural network architectures.

Convergence is a critical concept in perceptron training. It signifies the point where the perceptron has effectively learned from the training data. Understanding the factors that influence convergence can help in designing better training algorithms and ensuring effective learning.

Applications of Perceptrons

Binary Classification:

- Perceptrons are inherently binary classifiers, making them suitable for tasks like spam detection, sentiment analysis, and more.

Image Recognition:

- Early applications of perceptrons were in recognizing simple patterns in images, such as identifying shapes or characters.

Linear Decision Boundaries:

- For datasets that have a linear decision boundary, perceptrons can be an effective tool. For instance, classifying data points based on two linearly separable features.

While perceptrons have their limitations, they have found applications in various domains due to their simplicity and efficiency. Understanding these applications provides insights into the versatility of this foundational machine learning model.

Linear Regression:

- With appropriate activation functions, perceptrons can be used for predicting continuous values, making them suitable for simple linear regression tasks.

Control Systems:

- Perceptrons can be used in adaptive control systems where the system learns to control an operation based on the input it receives.

Neural Network Foundations:

- Besides their limitations, they laid the groundwork for multi-layer neural networks and deep learning models that can handle more complex tasks.

Module 3. Bioinformatic tools

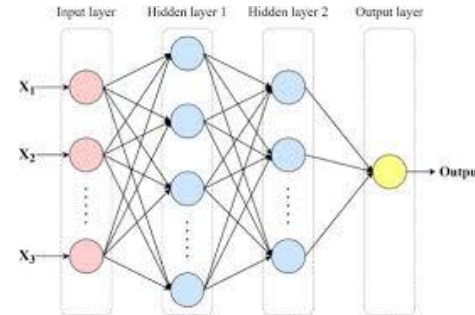
Topic 4. Artificial Neural Networks

Lesson 3. Multi-Layer Perceptron - Backpropagation

Introduction to MLP

Unveiling the Multi-Layer Perceptron (MLP)

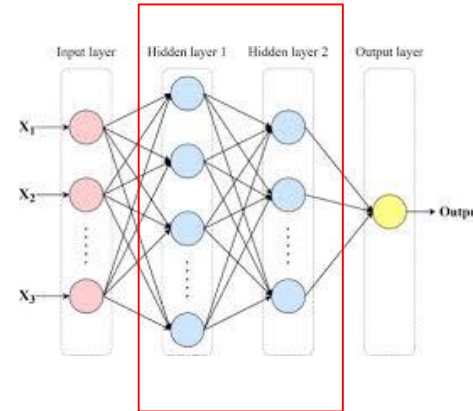
- **What is an MLP?**
 - A type of artificial neural network.
 - Comprises multiple layers of nodes or neurons.
 - Can model complex, non-linear relationships.
- **Why is MLP Important?**
 - Serves as the foundation for deep learning.
 - Capable of handling a wide range of tasks, from classification to regression.
 - Offers flexibility in design with varying depths and widths.
- **Basic Structure:**
 - **Input Layer:** The initial layer that receives data.
 - **Hidden Layers:** Intermediate layers that process data. The number can vary.
 - **Output Layer:** Produces the final prediction or classification.



The Multi-Layer Perceptron is a foundational concept in neural networks and deep learning. Its multi-layered structure allows it to learn and represent intricate patterns in data, making it a versatile tool in the AI toolkit.

Hidden Layers

- **What are Hidden Layers?**
 - Layers situated between the input and output layers.
 - Composed of multiple neurons or nodes.
 - Responsible for extracting and processing features from input data.
- **Depth vs. Width in MLPs:**
 - **Depth:** Refers to the number of hidden layers. More layers can capture complex patterns but may risk overfitting.
 - **Width:** Refers to the number of neurons in each hidden layer. A wider layer can represent more features but may increase computational cost.
- **Significance of Hidden Layers:**
 - They enable the MLP to model non-linear relationships.
 - Each layer can capture different levels of abstraction, from basic to complex features.
 - Act as a transformation space, converting input data into a format that makes it easier for the output layer to make predictions.



Hidden layers are the heart of an MLP, allowing it to learn intricate patterns and relationships in the data. Their design, in terms of depth and width, plays a crucial role in the performance and efficiency of the network.

Activation Functions

What is an Activation Function?

- A mathematical function applied to a neuron's output.
- Determines the neuron's output based on its input.
- Introduces non-linearity into the network.

Why is it Important?

- Without activation functions, MLPs would be equivalent to linear regression models, limiting their capacity to model complex relationships.
- They allow neurons to make decisions, enabling the network to learn from errors and make adjustments.

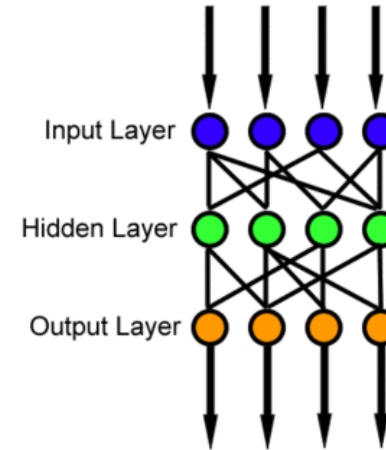
Common Activation Functions for MLPs:

- **Step Function:** Binary output based on a threshold.
- **Sigmoid:** Produces outputs between 0 and 1, often used in binary classification.
- **ReLU (Rectified Linear Unit):** Allows positive values to pass through unchanged while setting negative values to zero. Commonly used in hidden layers.

Activation functions are pivotal in introducing non-linearity to neural networks, enabling them to capture and model intricate patterns in data. Choosing the right activation function can significantly impact the performance of an MLP.

Feed Forward Mechanism

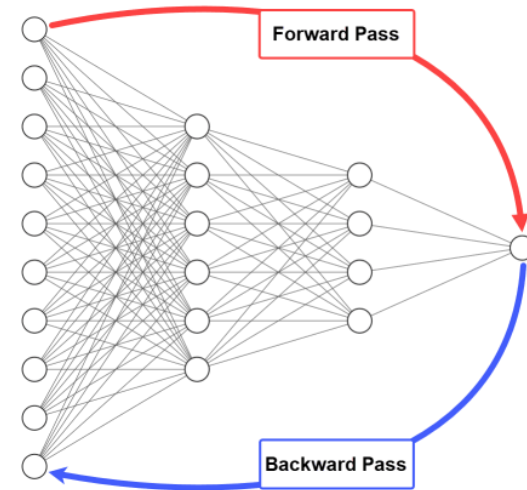
- **What is Feed Forward?**
 - a. The initial phase where input data is passed through the network.
 - b. Data flows from the input layer, through hidden layers, to the output layer.
 - c. Each neuron processes the data, applies the activation function, and passes it to the next layer.
- **Steps in Feed Forward:**
 - a. **Input Reception:** Neurons in the input layer receive the data.
 - b. **Weighted Sum:** Each neuron in the hidden layer calculates a weighted sum of its inputs.
 - c. **Activation:** The activation function is applied to the weighted sum to produce the neuron's output.
- **Role in Prediction:**
 - a. The final output of the feed forward process is the network's prediction.
 - b. For classification tasks, this could be a class label or probability distribution over classes.
 - c. For regression tasks, it could be a continuous value.



The feed forward mechanism is the essence of how an MLP makes predictions. It's a systematic process where data is transformed and processed layer-by-layer, culminating in the network's final prediction.

Back Propagation

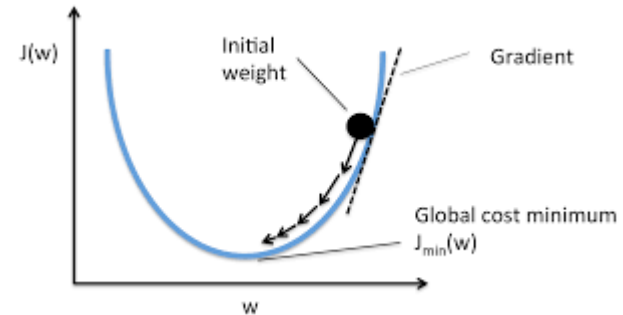
- **What is Back Propagation?**
 - a. The process by which an MLP learns and adjusts its weights.
 - b. Based on the error between the predicted output and the actual target.
 - c. Involves computing the gradient of the error with respect to each weight.
- **Why is it Crucial?**
 - a. Enables the network to minimize the error in its predictions.
 - b. By adjusting weights in the right direction, the network improves its performance over time.
 - c. It's the primary mechanism that allows neural networks to learn from data.
- **Basic Steps:**
 - a. **Error Calculation:** Determine the difference between the predicted output and the actual target.
 - b. **Gradient Computation:** Calculate how much each weight contributed to the error.
 - c. **Weight Adjustment:** Update weights in the opposite direction of the gradient to reduce the error.



The feed forward mechanism is the essence of how an MLP makes predictions. It's a systematic process where data is transformed and processed layer-by-layer, culminating in the network's final prediction.

Gradient Descent - The Optimization Engine

- **What is Gradient Descent?**
 - An optimization algorithm used to minimize the error in neural networks.
 - Adjusts the weights of the network in the direction of steepest decrease of the error.
 - Iteratively refines weights until the error reaches a minimum value.
- **How Does it Work?**
 - **Compute Gradient:** Calculate the gradient of the error with respect to each weight. This indicates the direction and magnitude of change needed.
 - **Update Weights:** Adjust weights by a fraction (learning rate) of the gradient.
 - **Iterate:** Repeat the process until the error stops decreasing or decreases very slowly.
- **Learning Rate:**
 - A hyperparameter that determines the step size during weight updates.
 - Too large: Might overshoot the minimum and cause oscillations.
 - Too small: Convergence might be very slow.



Gradient Descent is the driving force behind the learning process in MLPs. By navigating the error landscape and making strategic weight adjustments, it ensures the network converges to an optimal solution.

The Mathematics Behind Learning

1. Weight Update Equation:

- $W_{new} = W_{old} - \alpha \times \nabla E$
- W_{new} : Updated weight
- W_{old} : Previous weight
- α : Learning rate
- ∇E : Gradient of the error

2. Error Gradient:

- For a given weight w and error E , the gradient is given by:
- $\nabla E_w = \frac{\partial E}{\partial w}$
- Represents how much the error changes with a small change in weight.

3. Chain Rule & Back Propagation:

- The gradient is often computed using the chain rule of calculus.
- This allows the error from the output layer to be propagated backward through the network, adjusting weights in all layers.

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l} \quad \text{chain rule}$$

$$z_j^l = \sum_{k=1}^m w_{jk}^l a_k^{l-1} + b_j^l \quad \text{by definition}$$

m – number of neurons in $l-1$ layer

$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \quad \text{by differentiation (calculating derivative)}$$

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} a_k^{l-1} \quad \text{final value}$$

The weight update process in MLPs is governed by mathematical equations. These equations ensure that weights are adjusted in a manner that systematically reduces the error, driving the learning process.

Batches & Mini-Batches

What are Batches?

- Grouping of training data used to update network weights.
- Types:
 - **Batch Gradient Descent:** Uses the entire dataset to compute the gradient and update weights once per epoch.
 - **Stochastic Gradient Descent (SGD):** Uses one data point at a time to update weights.
 - **Mini-Batch Gradient Descent:** Strikes a balance by using a subset of the dataset for each update.

Why Mini-Batches?

- Faster convergence: More frequent weight updates can lead to quicker learning.
- Memory efficiency: Suitable for large datasets that don't fit in memory.
- Can escape local minima due to the noise in the gradient estimation.

Choosing the Right Batch Size:

- Too small: May lead to noisy weight updates.
- Too large: Can be computationally expensive and might get stuck in local minima.
- A typical range is between 32 and 512, but it's problem-dependent.

The choice of batch size can significantly influence the efficiency and effectiveness of the learning process. Mini-batch gradient descent often provides a good balance between speed and stability.

Batches vs. Mini-Batches vs. Stochastic

Batch Gradient Descent:

- **Definition:** Uses the entire dataset to compute the gradient.
- **Pros:** Stable convergence, consistent gradient direction.
- **Cons:** Can be slow, especially for large datasets. Memory-intensive. Might get stuck in local minima.
- **Use Case:** Ideal for smaller datasets or when a convex error surface is guaranteed.

Stochastic Gradient Descent (SGD):

- **Definition:** Uses one data point at a time for weight updates.
- **Pros:** Faster convergence for large datasets. Inherent randomness can escape local minima.
- **Cons:** Noisy convergence due to frequent updates. More susceptible to oscillations.
- **Use Case:** When computational resources are limited or when the dataset has a lot of redundancy.

Mini-Batch Gradient Descent:

- **Definition:** A middle ground between Batch and SGD.
- **Pros:** Balances speed and stability. Suitable for parallel processing on hardware like GPUs.
- **Cons:** Requires careful selection of batch size. Might still face challenges of both SGD and Batch methods.
- **Use Case:** Most real-world applications, especially deep learning, due to its balance of benefits.

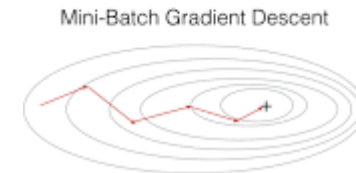
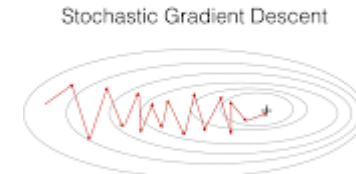
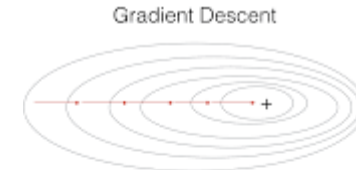
Additional Considerations:

- **Learning Rate:** Crucial for all methods. Adaptive learning rate techniques can help optimize convergence.
- **Epochs vs. Iterations:** An epoch is one complete forward and backward pass of all training examples. An iteration is a pass of a single batch. SGD and Mini-Batch require more iterations per epoch.

The choice between Batch, SGD, and Mini-Batch depends on the dataset size, memory constraints, and desired convergence properties. Often, Mini-Batch offers a good trade-off between speed and stability.

Stochastic Gradient Descent

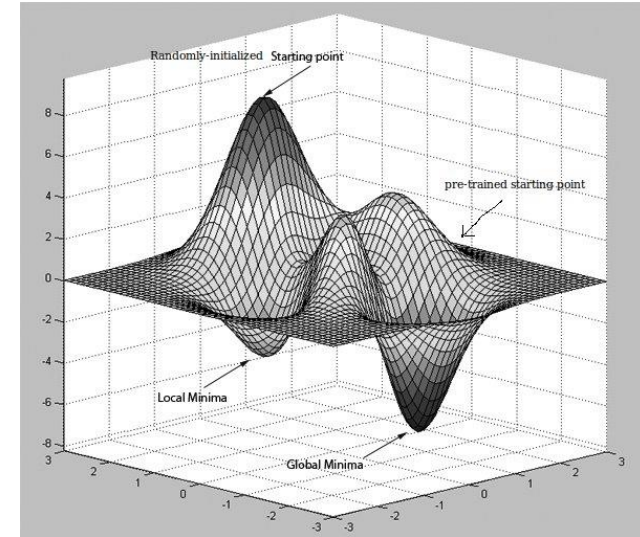
- **What is SGD?**
 - a. A variant of gradient descent where weights are updated after each training example.
 - b. Introduces randomness in the learning process, as updates are based on individual data points.
- **Advantages of SGD:**
 - a. **Faster Convergence:** Frequent updates can lead to quicker learning.
 - b. **Escape Local Minima:** The noisy nature of updates can help escape shallow local minima.
 - c. **Memory Efficiency:** Doesn't require the entire dataset in memory at once.
- **Challenges & Solutions:**
 - a. **Noisy Updates:** Can lead to oscillations in learning. Solution: Use a decaying learning rate or momentum.
 - b. **Sensitive to Learning Rate:** Requires careful tuning. Solution: Adaptive learning rate techniques like Adagrad, RMSprop, or Adam.



SGD offers a dynamic approach to learning, with frequent updates based on individual data points. While it introduces randomness, with the right techniques, it can lead to efficient and effective learning.

Error Surface & Optimization Landscape

- **What is the Error Surface?**
 - a. A graphical representation of the error as a function of the network's weights. Visualizes the optimization landscape that gradient descent algorithms navigate.
- **Features of the Error Surface:**
 - a. **Local Minima:** Points where the error is lower than neighboring points but not the lowest overall.
 - b. **Global Minimum:** The point with the lowest error across the entire surface.
 - c. **Saddle Points:** Locations where the error is flat in some directions but sloped in others.
- **Challenges in Optimization:**
 - a. Getting stuck in local minima or saddle points.
 - b. Oscillations or slow convergence due to the shape of the surface.
- **Enhancing Optimization:**
 - a. **Momentum:** Helps the algorithm move faster through flat regions and avoid getting stuck.
 - b. **Adaptive Learning Rates:** Adjusts the learning rate based on the landscape, e.g., Adam or RMSprop.



The error surface provides insights into the challenges faced during optimization. By understanding its features, we can employ techniques to navigate it more effectively and achieve better learning outcomes.

Practical Weight Adjustments

The Balancing Act:

- Adjusting weights is a balance between learning quickly and not overshooting the optimal values.
- Importance of the learning rate: Too high can cause divergence, too low can cause slow convergence.

Overcoming Plateaus:

- During training, the model might encounter regions where the error doesn't change much (plateaus).
- Techniques like momentum can help push through these regions.

Avoiding Oscillations:

- If weights keep oscillating back and forth without settling, it indicates a high learning rate.
- Adaptive learning rate techniques can help reduce oscillations.

Regularization Techniques:

- Methods like L1 and L2 regularization can prevent weights from becoming too large, which can lead to overfitting.
- They add a penalty to the loss function based on the magnitude of weights.

Weight Initialization:

- Starting weights can influence the training process.
- Techniques like Xavier and He initialization can set weights in a way that aids faster convergence.

Adjusting weights in neural networks is not just about following mathematical rules. It's about understanding the challenges and employing strategies to navigate them effectively.

Advanced Optimization Techniques

Momentum:

- **Concept:** Incorporates the direction of the previous gradient to accelerate convergence.
- **Benefit:** Helps in faster convergence and escaping local minima.

Adagrad (Adaptive Gradient Algorithm):

- **Concept:** Adapts the learning rates of all model parameters, scaling them inversely proportional to the square root of the sum of historical squared values.
- **Benefit:** Allows different learning rates for each parameter.

RMSprop (Root Mean Square Propagation):

- **Concept:** Modifies Adagrad to use a moving average of squared gradients.
- **Benefit:** Addresses Adagrad's radically diminishing learning rates.

Adam (Adaptive Moment Estimation):

- **Concept:** Combines ideas from Momentum and RMSprop. Keeps an exponentially decaying average of past gradients.
- **Benefit:** Balances the benefits of both Momentum and RMSprop.

Nesterov Accelerated Gradient (NAG):

- **Concept:** A tweak to momentum where the gradient is calculated after the current momentum step.
- **Benefit:** Provides a more accurate estimate of the gradient.

Advanced optimization techniques provide refined ways to navigate the error landscape, ensuring faster and more stable convergence in complex scenarios.

Conclusion & Future Outlook

- **Recap:**
 - **MLP's Power:** Multi-layer perceptrons offer a deeper and more intricate way to model complex relationships in data compared to single-layer perceptrons.
 - **Key Concepts:** Hidden layers, activation functions, error functions, and advanced optimization techniques form the core of MLPs.
- **Future Outlook:**
 - **Deep Learning:** MLPs are foundational to deep neural networks, which are revolutionizing fields like computer vision, natural language processing, and more.
 - **Customization:** As research progresses, we'll see more tailored activation functions, optimization techniques, and architectures for specific tasks.
 - **Challenges:** While MLPs are powerful, they come with challenges like overfitting, vanishing gradients, and computational demands. Ongoing research aims to mitigate these.
- **Continuous Learning:**
 - **Stay Updated:** The field of neural networks is rapidly evolving. Continuous learning and staying updated with the latest research is crucial.
 - **Hands-on Practice:** Theoretical knowledge is vital, but practical implementation and experimentation will solidify understanding.

MLPs have paved the way for the deep learning revolution. Embracing their concepts and understanding their intricacies is crucial for anyone diving into the world of artificial neural networks.

Multi-Layer Perceptrons

Celebrating the Success of MLPs

- **Journey Overview:**
 - From their inception, MLPs have been at the forefront of many breakthroughs in AI. This lesson celebrates their achievements and pivotal moments.
- **Why Celebrate MLPs?**
 - They've overcome significant challenges, like the XOR problem, setting the stage for more advanced neural networks.
 - Their adaptability has led to successes in diverse fields, from healthcare to finance.
- **A Glimpse Ahead:**
 - While we'll start with the XOR triumph, we'll also explore modern successes, showcasing the enduring relevance of MLPs.



The story of MLPs is one of resilience, adaptability, and continuous evolution. Join us as we journey through their remarkable successes.

The XOR Problem - A Turning Point

The XOR Challenge:

- The XOR (exclusive OR) problem was a significant challenge for single-layer perceptrons. It represents a scenario where data isn't linearly separable, meaning a straight line can't separate the classes.
- Example: For binary inputs (0 or 1), XOR outputs 1 only when the inputs are different.

MLP's Triumph:

- While single-layer perceptrons failed to solve the XOR problem, introducing a hidden layer in MLPs made it possible.
- This success demonstrated the power of depth in neural networks, paving the way for more complex architectures.

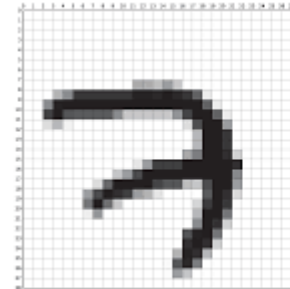
Significance:

- The ability to solve the XOR problem was a **watershed moment** in AI. It showcased the potential of neural networks to tackle non-linear problems.
- It also highlighted the limitations of simpler models and the need for more sophisticated architectures.

The XOR problem served as both a challenge and a validation for neural networks. MLP's success in solving it marked a significant advancement in the field.

MLPs in Image Recognition

- **Early Days of Image Recognition:**
 - a. Before the deep learning era, MLPs were among the primary tools used for image recognition tasks.
 - b. They were employed to recognize handwritten digits, basic shapes, and other simple visual patterns.
- **How MLPs Work in Image Recognition:**
 - a. Images are broken down into pixel values, which serve as inputs to the MLP.
 - b. The network learns to recognize patterns and features, such as edges and textures, through its layers.
- **Achievements & Milestones:**
 - a. MLPs achieved notable success in datasets like MNIST (handwritten digits).
 - b. They set the foundation for more advanced architectures like Convolutional Neural Networks (CNNs) that dominate the field today.



(a) MNIST sample belonging to the digit '7'.

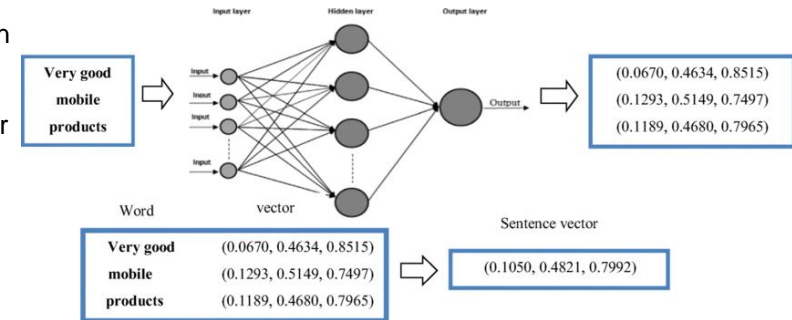


(b) 100 samples from the MNIST training set.

While modern deep learning models have surpassed MLPs in image recognition, the foundational work done by MLPs in this domain is undeniable and paved the way for future advancements.

MLPs in Natural Language Processing (NLP)

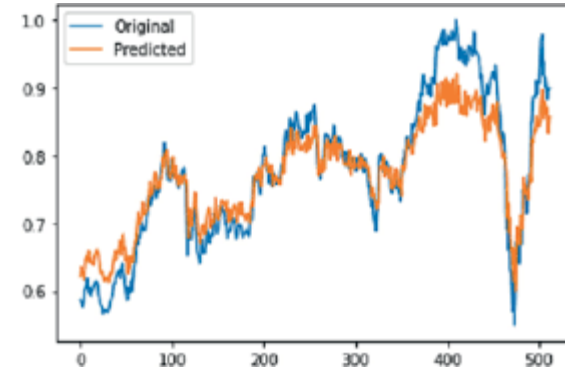
- **NLP - A Brief Overview:**
 - Natural Language Processing involves making computers understand, interpret, and generate human language.
 - It's a complex task due to the nuances, idioms, and structures in languages.
- **MLPs in Early NLP Tasks:**
 - Before the rise of models like transformers, MLPs were used for basic NLP tasks.
 - They were employed for text classification, sentiment analysis, and basic language modeling.
- **How MLPs Process Text:**
 - Text data is converted into numerical vectors using techniques like TF-IDF or word embeddings.
 - These vectors serve as inputs to the MLP, which then learns patterns and relationships between words and phrases.



MLPs played a foundational role in early NLP tasks, helping bridge the gap between human language and machine understanding, setting the stage for more advanced NLP models

MLPs in Financial Forecasting

- **Financial Forecasting - The Challenge:**
 - Predicting stock market movements, currency exchange rates, and other financial metrics is inherently complex due to numerous influencing factors.
 - Accurate predictions can lead to significant financial gains, making it a sought-after application of AI.
- **Role of MLPs:**
 - MLPs have been used to model and predict financial time series data.
 - They capture patterns, trends, and relationships in historical data to make future predictions.
- **Features & Data Processing:**
 - Financial data, like stock prices, trading volumes, and technical indicators, are fed into MLPs.
 - Data normalization and preprocessing are crucial to ensure the network effectively learns from past trends.



While modern algorithms and models have further refined financial forecasting, MLPs laid the groundwork, showcasing the potential of neural networks in predicting complex financial metrics.

MLPs in Healthcare Diagnostics

- **Healthcare Diagnostics - The Importance:**
 - Accurate and timely diagnosis is crucial in healthcare, impacting treatment decisions and patient outcomes.
 - AI has the potential to assist medical professionals, ensuring better patient care.
- **MLPs in Medical Imaging & Diagnostics:**
 - Before the advent of more advanced deep learning models, MLPs were used to analyze medical images like X-rays, MRIs, and CT scans.
 - They helped in identifying anomalies, tumors, and other medical conditions.
- **Data & Features:**
 - Medical images are processed and converted into a format suitable for MLPs.
 - Features such as edges, textures, and patterns are extracted and used for diagnosis.



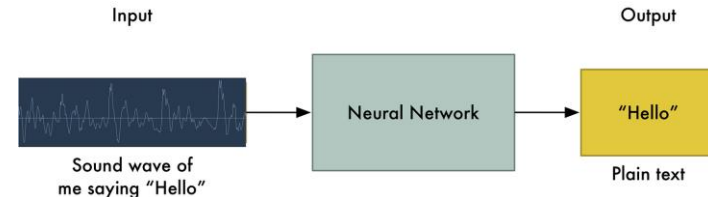
MLPs have played a pivotal role in early medical diagnostics using AI, enhancing the accuracy and efficiency of disease detection and patient care.

MLPs in Voice Recognition

- **Voice Recognition - The Revolution:**
 - The ability to recognize and interpret human voice has revolutionized user interfaces, leading to voice assistants, voice-activated systems, and more.
 - It involves understanding speech patterns, accents, and nuances.
- **MLPs in Early Voice Recognition:**
 - Before deep learning models like RNNs and LSTMs took over, MLPs were among the tools used for basic voice recognition tasks.
 - They were employed to detect specific voice commands, recognize speakers, and more.

- **Data Processing & Features:**

- Voice data is transformed into spectrograms or other representations suitable for MLPs.
- The network learns from the frequency patterns, pitch variations, and other voice features.



MLPs were instrumental in the early stages of voice recognition technology, setting the foundation for the sophisticated voice-activated systems we have today.

MLPs in Gaming & Simulations

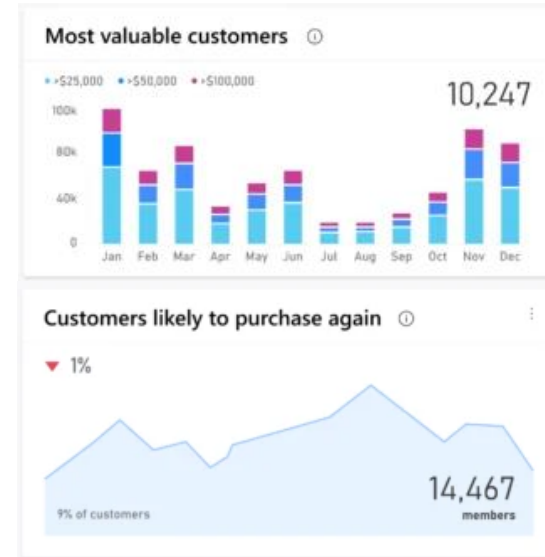
- **Gaming & Simulations - The Digital Playground:**
 - Video games and simulations offer immersive experiences, often requiring AI to create realistic and dynamic environments.
 - AI-driven characters (NPCs) and events enhance gameplay and user engagement.
- **MLPs in Game AI:**
 - In the early days of gaming AI, MLPs were used to drive the behavior of NPCs, making them react to player actions.
 - They helped in pathfinding, decision-making, and creating adaptive challenges for players.
- **Data & Learning:**
 - Game data, such as player actions, game states, and environment variables, are inputs for the MLP.
 - The network learns to predict player behavior and adapt the game environment accordingly.



MLPs have contributed to the evolution of gaming AI, making virtual worlds more interactive, unpredictable, and engaging.

MLPs in Customer Analytics

- **Customer Analytics - The Business Edge:**
 - Understanding customer behavior, preferences, and patterns is vital for businesses to tailor their offerings and improve customer satisfaction.
 - Predictive analytics can forecast customer needs, enhancing marketing and sales strategies.
- **Role of MLPs in Early Customer Analytics:**
 - MLPs were used to analyze customer data, segment markets, and predict purchasing behavior.
 - They helped businesses identify potential high-value customers, churn rates, and more.
- **Data Insights & Features:**
 - Data from sales, customer interactions, feedback, and online behavior are processed by the MLP.
 - The network identifies patterns, trends, and segments, providing actionable insights for businesses.



By analyzing vast amounts of customer data, MLPs provided businesses with valuable insights, shaping marketing campaigns, product developments, and customer service strategies.

Challenges Overcome by MLPs

Overfitting:

- **Challenge:** MLPs, with their capacity to fit complex data, often memorized training data, leading to poor generalization on unseen data.
- **Solution:** Techniques like dropout, regularization, and early stopping were introduced. These methods prevent the network from becoming too complex and help in achieving better generalization.

Vanishing & Exploding Gradients:

- **Challenge:** During backpropagation, gradients can become too small (vanish) or too large (explode), making training unstable or stagnant.
- **Solution:** Initialization techniques, like He or Xavier initialization, and activation functions like ReLU, helped mitigate these issues.

Local Minima:

- **Challenge:** MLPs could get stuck in local minima during training, preventing them from reaching the best possible solution.
- **Solution:** Optimizers like momentum and adaptive learning rates helped navigate the error surface more effectively.

Computational Challenges:

- **Challenge:** Training large MLPs required significant computational resources.
- **Solution:** Hardware accelerators like GPUs and distributed training techniques made it feasible to train larger and more complex MLPs.

Despite facing numerous challenges, innovations and techniques emerged to enhance the training and performance of MLPs, solidifying their place in the AI landscape.

The Modern Era - Deep Learning

Evolution into Deep Learning:

- While MLPs laid the foundation, the introduction of deep neural networks (DNNs) marked a significant leap in AI capabilities.
- DNNs consist of multiple hidden layers, allowing them to model even more complex relationships in data.

Advancements in Hardware:

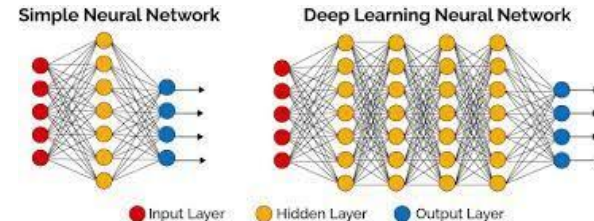
- The rise of GPUs and TPUs facilitated the training of deeper networks, handling millions of parameters efficiently.
- Parallel processing capabilities made it feasible to train networks on vast datasets.

Breakthroughs in Various Domains:

- Deep learning has achieved state-of-the-art results in areas like image recognition, natural language processing, and reinforcement learning.
- Innovations like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) stemmed from the foundational concepts of MLPs.

Challenges & Ongoing Research:

- Despite their power, DNNs come with challenges like interpretability and computational costs.
- Research in areas like transfer learning, attention mechanisms, and transformers continues to push the boundaries of what's possible.



The principles of MLPs paved the way for the deep learning revolution, transforming the landscape of AI and leading to breakthroughs in numerous domains.

The Road Ahead for MLPs and Deep Learning

Expanding Domains:

- As technology advances, the application of MLPs and deep learning will expand into newer domains like quantum computing, neuromorphic engineering, and more.
- Integration with other AI techniques, such as reinforcement learning and generative models, will lead to more holistic solutions.

Addressing Current Challenges:

- Research is ongoing to make deep learning models more interpretable, energy-efficient, and robust against adversarial attacks.
- Efforts are being made to reduce the data dependency of models, making them more generalizable with fewer examples.

Ethical and Responsible AI:

- As AI becomes more integrated into daily life, there's a growing emphasis on creating models that are ethical, unbiased, and transparent.
- Regulations and guidelines will play a crucial role in ensuring AI benefits all of humanity.

Collaborative AI:

- The future will see humans and AI working more closely, with models augmenting human capabilities rather than replacing them.
- Emphasis on human-in-the-loop systems, where AI provides insights and humans make the final decisions.

The journey of MLPs and deep learning is far from over. As we look to the future, the potential for growth, innovation, and positive impact is immense.

Module 3. Bioinformatic tools

Topic 4. Artificial Neural Networks

Lesson 4. Deep Learning with Python

Introduction to Deep Learning in Python

Unleashing the Power of Neural Networks

What is Deep Learning?

- A subset of machine learning that uses neural networks with many layers (hence "deep") to analyze various factors of data.
- Capable of automatically learning data representations without manual feature extraction.

Python's Role in Deep Learning:

- Python has emerged as the leading language for deep learning due to its **simplicity**, **readability**, and **extensive libraries** tailored for data science and AI.
- The vast ecosystem of Python, combined with its dynamic community, has led to the development of powerful frameworks like TensorFlow, Keras, and PyTorch.

Course Overview:

- Dive into the core concepts of deep learning using Python.
- Explore popular frameworks, data handling, visualization, model evaluation, and deployment.

With Python's versatility and the power of deep learning, we can solve complex problems, ranging from image recognition to natural language processing and beyond.



The Dominance of Python in Deep Learning

Simplicity and Readability:

- Python's syntax is clear and intuitive, making it accessible for beginners and efficient for experts.
- Allows for rapid prototyping and iterative development.

Extensive Libraries:

- Python boasts a rich ecosystem of libraries tailored for data science, machine learning, and deep learning (e.g., **TensorFlow, Keras, Pandas, NumPy**).
- These libraries simplify complex tasks, from data preprocessing to model training.

Supportive Community:

- A dynamic and growing community that contributes to open-source projects, shares knowledge, and provides support through forums like *Stack Overflow*.
- Regular updates, tutorials, and documentation for deep learning frameworks.

Interoperability:

- Python seamlessly integrates with other languages like C, C++, and Java, allowing for performance optimization and utilization of legacy code.
- Offers bindings to other platforms and tools, enhancing its utility.

Platform Agnostic:

- Python is cross-platform, meaning it runs on various operating systems without modification.
- Facilitates deployment and scaling of deep learning applications.

Python's combination of simplicity, powerful libraries, community support, and versatility makes it the premier choice for deep learning endeavors.

Setting Up the Environment

Python Installation:

- Ensure you have the latest version of Python installed.
- Use official Python distributions or tools like Anaconda for a comprehensive data science setup.

Virtual Environments:

- Utilize **venv** or **conda** to create **isolated environments** for different projects.
- Helps manage dependencies and avoid version conflicts.

Key Libraries and Frameworks:

- Install essential libraries: **NumPy**, **Pandas**, **Matplotlib** for data manipulation and visualization.
- Deep learning frameworks: **TensorFlow**, **Keras**, **PyTorch**. Choose based on project needs.

Integrated Development Environments (IDEs):

- Tools like Jupyter Notebook, PyCharm, or Visual Studio Code enhance coding efficiency.
- **Jupyter** is especially popular for interactive data analysis and model training.

GPU Setup (Optional but Recommended):

- For intensive computations, setting up a GPU can drastically reduce training times.
- Libraries like CUDA and cuDNN help in leveraging NVIDIA GPUs with TensorFlow or PyTorch.

Testing the Setup:

- Run a simple deep learning script or notebook to ensure all installations work seamlessly. Verify GPU utilization if set up.

A well-configured environment is crucial for efficient deep learning development. Ensure all tools are correctly installed and integrated.

Importing Data in Python

1. Sources of Data:

- Local files (CSV, Excel, SQL databases).
- Web sources (APIs, web scraping).
- Cloud storage (AWS S3, Google Cloud Storage).

2. Using Pandas for Data Import:

- **CSV:** `data = pd.read_csv('path_to_file.csv')`
- **Excel:** `data = pd.read_excel('path_to_file.xlsx')`
- **SQL:**

```
import sqlite3

conn = sqlite3.connect('database.db')

data = pd.read_sql_query('SELECT * FROM table_name', conn)
```

1. Preparing Data for Splitting:

- **Features (X):** `X = data.drop('target_column', axis=1)`
- **Target (y):** `y = data['target_column']`

Splitting Data - Train and Test Sets

1. Why Split?

- Train the model on one subset (**training set**).
- Evaluate the model's performance on another subset (**test set**).

2. Using `train_test_split` from Scikit-Learn:

python

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

3. Considerations:

- **Stratification:** Ensuring the train and test sets have similar proportions of class labels.
- **Random State:** For reproducibility of results.
- **Test Size:** Typically between 20% - 30% of the data.

Introduction to TensorFlow

- **What is TensorFlow?**
 - An open-source deep learning framework developed by Google Brain.
 - Enables building, training, and deploying machine learning models with ease.
- **Core Features:**
 - **Computational Graphs:** Represents computations as directed graphs, allowing for efficient parallel processing.
 - **Tensor Operations:** Handles multi-dimensional data arrays (tensors) for operations like matrix multiplications.
 - **Auto Differentiation:** Automatically computes gradients, simplifying optimization tasks.
- **Flexibility and Scalability:**
 - Supports a wide range of neural network architectures.
 - Scales seamlessly from a single CPU to multiple GPUs and even TPUs.
- **TensorBoard:**
 - A visualization tool bundled with TensorFlow.
 - Allows for monitoring model training, visualizing computational graphs, and analyzing metrics.
- **Integration with Keras:**
 - TensorFlow 2.x integrated Keras as its high-level API, making model building more intuitive.
 - Offers pre-defined layers, optimizers, and loss functions.
- **Ecosystem and Community:**
 - A vast ecosystem of tools, libraries, and extensions (e.g., TF Lite, TF.js, TF Hub).
 - Strong community support with regular updates, tutorials, and resources.



TensorFlow provides a comprehensive platform for building and deploying sophisticated machine learning models, backed by a robust ecosystem and community.

Building a Simple Model in TensorFlow (1)

```
import tensorflow as tf
from tensorflow import keras

model = keras.Sequential([
    keras.layers.Dense(128, activation='relu', input_shape=(784,)),
    keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(train_data, train_labels, epochs=10)
```

- Import TensorFlow and other necessary libraries.
- Define the Model: Use the Sequential API for stacking layers.
- Compile the Model: Specify the optimizer, loss function, and metrics.
- Train the Model

TensorFlow provides a comprehensive platform for building and deploying sophisticated machine learning models, backed by a robust ecosystem and community.

Building a Simple Model in TensorFlow (2)

Evaluating and Predicting:

```
test_loss, test_acc = model.evaluate(test_data, test_labels)
predictions = model.predict(new_data)
```

For simplicity, we have omitted the initial phase, where the data is imported and then splitted to train set and test set.

```
model.save('model_path.h5')
new_model = tf.keras.models.load_model('model_path.h5')
```

- Assess the model's performance on test data
- Make predictions.

Save or Load the model for future use or deployment

A concise walkthrough of building a basic neural network model in TensorFlow, from defining the architecture to training and evaluation. The code snippets offer a practical glimpse into the process.

Introduction to Keras

What is Keras?

- A high-level neural networks API, written in Python.
- Built to enable fast experimentation with deep neural networks.
- Runs on top of TensorFlow, Theano, or CNTK.



Why Keras?

- **Simplicity:** Intuitive and user-friendly API.
- **Flexibility:** Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- **Modularity:** Models are constructed using configurable building blocks (layers, optimizers, activation functions).
- **Integration:** Seamlessly integrates with TensorFlow, allowing for advanced features and optimizations.

Key Features:


- **Pre-processed Datasets:** Comes with built-in datasets like CIFAR-10, MNIST, etc.
- **Pre-trained Models:** Access to models like VGG, ResNet, and more for transfer learning.
- **Callbacks:** Set of functions applied during training to monitor progress or trigger actions.

Building a Simple Model in Keras (1)

1. Sequential Model:

- The simplest type of model, a linear stack of layers.

python


 Copy code

```
model = keras.Sequential()
```

2. Adding Layers:

- Easily add layers using the `add()` method.

python


 Copy code

```
model.add(Dense(128, activation='relu', input_shape=(784,)))  
model.add(Dense(64, activation='relu'))  
model.add(Dense(10, activation='softmax'))
```

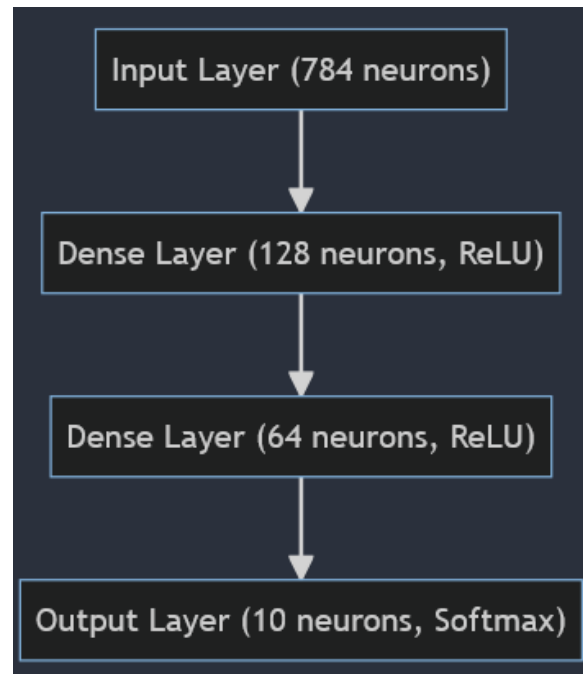
3. Configuring the Learning Process:

- Once the model is defined, configure its learning process with `compile()`.

python

 Copy code

```
model.compile(optimizer='adam',  
              loss='categorical_crossentropy',  
              metrics=['accuracy'])
```




ERASMUS+

Building a Simple Model in Keras (2)

4. Training the Model:

- Use the `fit()` method to train the model on data.

python


 Copy code

```
model.fit(x_train, y_train, epochs=10, batch_size=32)
```

5. Evaluation and Prediction:

- Assess the model's performance and make predictions.

python

 Copy code

```
loss, accuracy = model.evaluate(x_test, y_test)  
predictions = model.predict(new_data)
```

Visualization with Matplotlib and Seaborn

1. Why Visualization?

- Understand data distribution.
- Identify patterns and outliers.
- Make informed decisions during model building.

2. Matplotlib:

- Fundamental plotting library in Python.
- Versatile and provides a base layer for many other visualization libraries.
- Example:

```
import matplotlib.pyplot as plt
plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Sample Plot')
plt.show()
```

3. Seaborn:

- Built on top of Matplotlib.
- Provides a high-level interface for drawing attractive statistical graphics.
- Comes with several built-in themes and color palettes to make plots more visually appealing.
- Example:

```
import seaborn as sns
sns.histplot(data, bins=30, kde=True)
```

Model Evaluation (1)

Why Model Evaluation?

- Ensure the model's generalization to unseen data.
- Compare the performance of different models.
- Identify areas of improvement.

Common Classification Metrics:

- **Accuracy:** Overall correctness of the model.
- **Precision:** Correctness of positive predictions.
- **Recall (Sensitivity):** Ability to detect all positive instances.
- **F1-Score:** Harmonic mean of precision and recall.

Confusion Matrix:

A table used to evaluate the performance of a classification model. Shows true positives, true negatives, false positives, and false negatives.

| | | Predicted | |
|--------------|----------|----------------|----------------|
| | | Positive | Negative |
| Ground-Truth | Positive | True Positive | False Negative |
| | Negative | False Positive | True Negative |

Proper model evaluation is crucial for understanding its strengths and weaknesses, guiding further refinement, and ensuring its reliability in real-world applications.

Model Evaluation (2)

Common Regression Metrics:

- **Mean Absolute Error (MAE):** Average of absolute differences between predictions and actual values.
- **Mean Squared Error (MSE):** Average of squared differences.
- **R-squared:** Proportion of variance explained by the model.

Regression Metrics in Scikit-learn:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
y_pred = model.predict(X_test)
print("MAE:", mean_absolute_error(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))
```

Python, with its rich ecosystem of libraries, offers a straightforward and efficient way to evaluate machine learning and deep learning models.

Model Evaluation (3)

Classification Metrics in Scikit-learn:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1-Score:", f1_score(y_test, y_pred))
```

Evaluating Keras Models:

```
loss, accuracy = model.evaluate(X_test, y_test)
print("Loss:", loss)
print("Accuracy:", accuracy)
```

Python, with its rich ecosystem of libraries, offers a straightforward and efficient way to evaluate machine learning and deep learning models.

Saving & Loading (1)

Why Save & Load Models?

- **Efficiency:** Avoid retraining from scratch.
- **Deployment:** Use trained models in production environments.
- **Reproducibility:** Ensure consistent results across different platforms and times.

Saving & Loading with Scikit-learn:

- **Pickle:**

```
import pickle
# Save
with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)
# Load
with open('model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)
```

- **Joblib (recommended for larger models):**

```
from joblib import dump, load
# Save
dump(model, 'model.joblib')
# Load
loaded_model = load('model.joblib')
```

Saving and loading models is crucial for practical machine learning workflows, allowing for seamless transitions between training, evaluation, and deployment phases.

Saving & Loading (2)

Saving & Loading with Keras:

```
# Save entire model (architecture, weights, optimizer)
model.save('model.h5')
```

```
# Load the model
from keras.models import load_model
loaded_model = load_model('model.h5')
```

- Ensure compatibility: Models saved in one version of a library might not be compatible with another version.
- Security: Be cautious when loading models from untrusted sources

When you load the model using the respective methods, it will be in the **exact state** it was when saved, including both its architecture and the trained weights.

This ensures that you can immediately use the loaded model for predictions, further training, or evaluation without needing to retrain it.

Saving and loading models is crucial for practical machine learning workflows, allowing for seamless transitions between training, evaluation, and deployment phases.

Deployment of Deep Learning Models

1. What is Deployment?

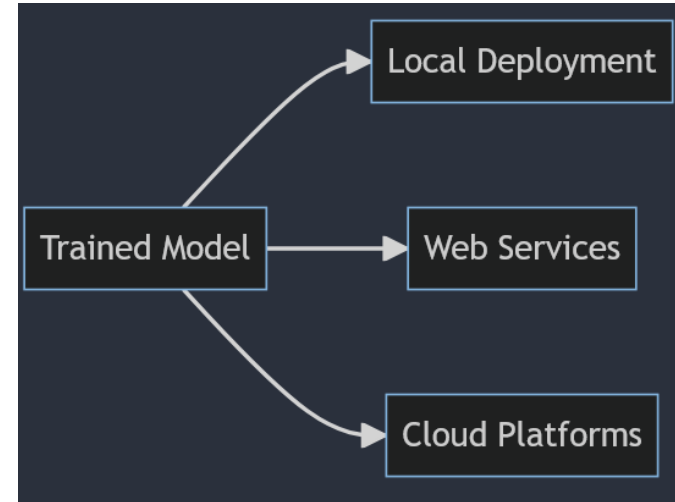
- Transitioning a trained model into an application where it can take in new data and make predictions in real-time.

2. Why is Deployment Important?

- **Utilization:** The real value of a trained model is realized when it's used in real-world applications.
- **Accessibility:** Allows end-users, who might not be familiar with ML, to benefit from the model's capabilities.

3. Deployment Options in Python:

- **Local Deployment:** Using the model within local applications or scripts.
- **Web Services:** Deploying the model as an API using frameworks like Flask or FastAPI.
- **Cloud Platforms:** Leveraging platforms like AWS SageMaker, Google AI Platform, or Azure Machine Learning for scalable deployments.



Deployment bridges the gap between model development and real-world impact, making it a crucial phase in the machine learning lifecycle.

Deployment Steps and Challenges

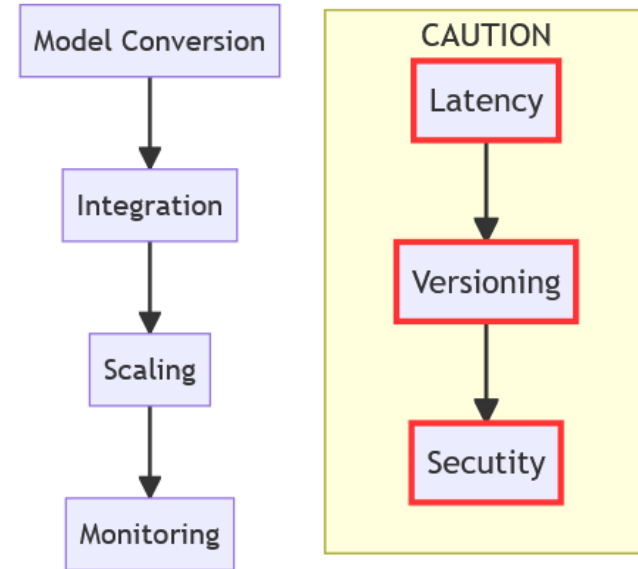
Steps for Deployment:

1. **Model Conversion:** Convert the model to a format suitable for deployment (e.g., TensorFlow Serving, ONNX).
2. **Integration:** Embed the model into the desired application or platform.
3. **Scaling:** Ensure the deployment can handle the expected number of requests.
4. **Monitoring:** Track the model's performance and health in the deployed environment.

Challenges in Deployment:

- **Latency:** Ensuring real-time predictions with minimal delay.
- **Versioning:** Managing updates to the model without disrupting the service.
- **Security:** Protecting the model and data from potential threats.

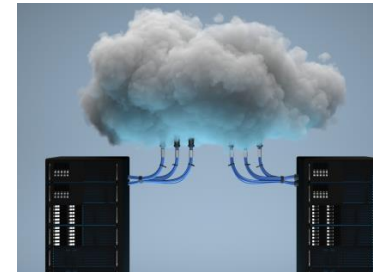
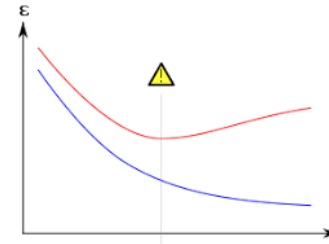
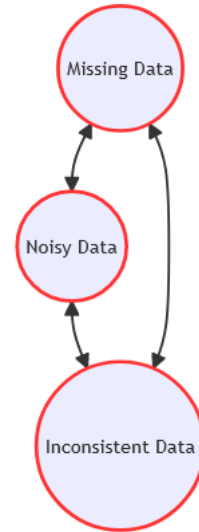
Deployment bridges the gap between model development and real-world impact, making it a crucial phase in the machine learning lifecycle.



Common Challenges in Deep Learning

Common Challenges

1. **Overfitting:** When a model performs well on the training data but poorly on unseen data.
2. **Data Quality:** Inaccurate or inconsistent data can lead to poor model performance.
3. **Computational Resources:** Deep learning models, especially complex ones, require significant computational power and memory.



Overfitting, poor data quality, and computational demands are primary challenges in deep learning. Ensuring data integrity and having adequate computational resources are crucial for model success.

Best Practices for Deep Learning in Python

1. **Regularization:** Techniques like dropout or L1/L2 regularization can help prevent overfitting.
2. **Data Augmentation:** Enhancing the training dataset by applying random modifications. Especially useful for image data.
3. **Early Stopping:** Monitor the validation performance and stop training once it starts to degrade.
4. **Use Pre-trained Models:** Leveraging models that have already been trained on large datasets can save time and resources.



Techniques like regularization, data augmentation, and early stopping enhance model robustness and efficiency. Utilizing pre-trained models can expedite training and improve performance, especially with limited data.

Conclusion and Resources

1. Conclusion:

- Deep Learning in Python offers a powerful toolkit for building sophisticated models.
- While challenges exist, following best practices ensures robust and efficient models.
- Continuous learning and staying updated with the latest techniques is key to success in the ever-evolving field of deep learning.

2. Resources for Further Learning:

- **Books:**
 - i. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
 - ii. "Python Deep Learning" by Ivan Vasilev and Daniel Slater.
- **Online Courses:**
 - i. Coursera's "Deep Learning Specialization" by Andrew Ng.
 - ii. Udacity's "Deep Learning Nanodegree".
- **Websites:**
 - i. TensorFlow Official Documentation.
 - ii. Keras Official Documentation.
 - iii. Python Data Science Handbook by Jake VanderPlas (available online).

3. Communities and Forums:

- Stack Overflow (Deep Learning tags).
- Reddit r/MachineLearning.
- TensorFlow community forums.



Module 3. Bioinformatic tools

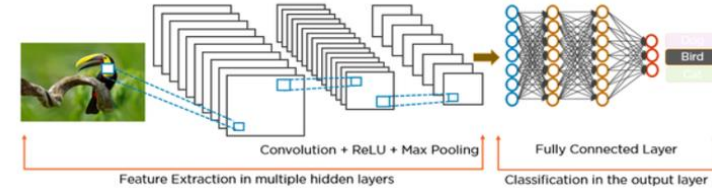
Topic 4. Artificial Neural Networks

Lesson 5. Convolutional Neural Networks

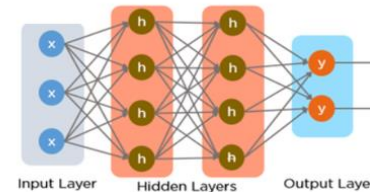
Introduction to Convolutional Neural Networks (CNNs)

- **Definition:**
 - A type of deep neural network designed to recognize patterns in data, especially visual data.
- **Key Features:**
 - Hierarchical layer structure.
 - Ability to automatically and adaptively learn spatial hierarchies of features.
- **Importance:**
 - Revolutionized the field of computer vision.
 - Powers image and video recognition, facial recognition, and many real-world applications.

Convolutional Neural Network



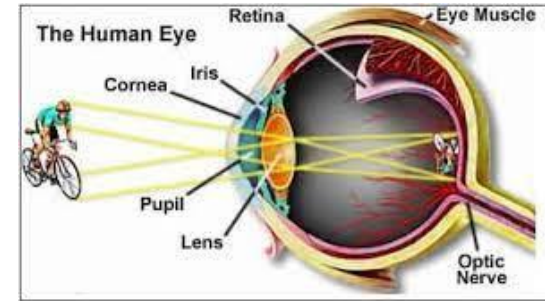
Artificial Neural Network



CNNs are specialized neural networks that have proven highly effective in recognizing and classifying visual patterns in data.

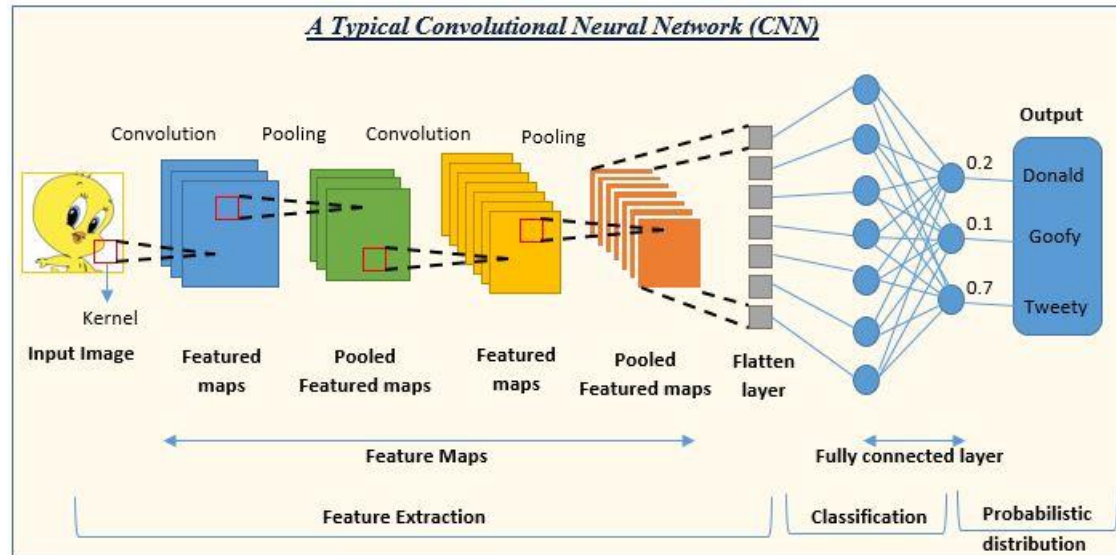
Inspiration - Human Visual System

- **The Eye as a Camera:**
 - The human eye captures light and processes it, similar to how a camera captures an image.
- **Processing in Layers:**
 - Visual information is processed in stages, from detecting simple features like edges to recognizing complex objects.
- **Neurons Specialized for Vision:**
 - Different neurons in the visual cortex respond to different visual stimuli, such as colors, shapes, and movements.
- **Hierarchical Processing:**
 - Lower layers detect simple features, while higher layers combine these features to recognize more complex patterns.



The design of CNNs is inspired by the hierarchical and layered processing of the human visual system, where different layers of neurons process different aspects of visual information.

Basic Structure of a CNN (1)



CNNs are composed of a series of layers, each designed to process and transform the input data in a specific way, ultimately leading to accurate image classification.

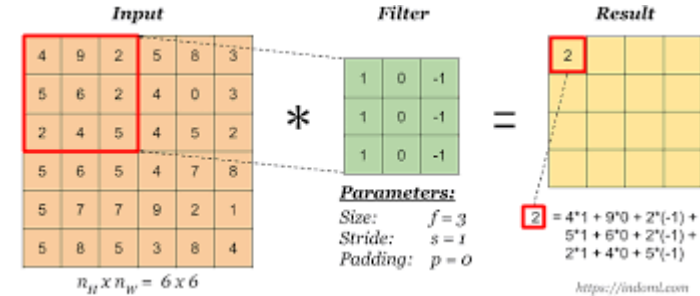
Basic Structure of a CNN (2)

- **Input Layer:**
 - Receives the raw pixel values of the image.
- **Convolutional Layers:**
 - Apply convolutional filters to the input data to detect features like edges, textures, and patterns.
 - Each filter focuses on a specific feature, and its output is a feature map.
- **Pooling (Subsampling) Layers:**
 - Reduce the spatial dimensions of the feature maps, making the network less computationally intensive.
 - Common methods: Max pooling, Average pooling.
- **Fully Connected Layers:**
 - After several convolutional and pooling layers, the data is flattened and passed through one or more fully connected layers.
 - These layers perform the final classification based on the detected features.
- **Output Layer:**
 - Produces the final classification result, often using a softmax activation function to give probabilities for each class.

CNNs are composed of a series of layers, each designed to process and transform the input data in a specific way, ultimately leading to accurate image classification.

Convolutional Layer - Introduction

- **Role of the Convolutional Layer:**
 - Primary function is to detect local patterns, textures, edges, and specific features from the input image.
 - Operates over a small, local region of the input data (a receptive field) but extends through the full depth of the input volume.
- **Filters/Kernels:**
 - Small, learnable weight matrices.
 - Slide over the input data (like a sliding window) to produce a feature map.
 - Each filter is responsible for detecting a specific kind of feature.
- **Feature Maps:**
 - Resulting 2D activation maps that show where in the image the features of the filter were detected.
 - Higher values in the feature map indicate the presence of the desired feature in the corresponding location of the input.



Convolution in action: Using a filter to detect features and produce a feature map.

The convolutional layer uses filters to scan the input image and generate feature maps, capturing essential patterns and details.

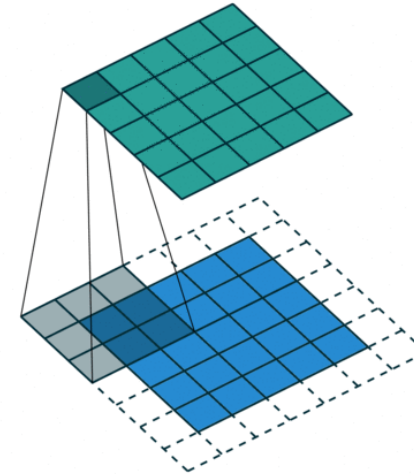
The Convolution Operation

Stride:

- The number of pixels by which the filter slides over the input matrix.
- A stride of 1 means the filter moves one pixel at a time, while a stride of 2 means it jumps two pixels.
- Affects the size of the output feature map.

Padding:

- Adding extra pixels around the input image.
- Helps in preserving the spatial dimensions of the input.
- Two types: Valid (no padding) and Same (padding to keep output size same as input).



Convolution involves sliding a filter over the input, with stride determining the step size and padding ensuring spatial dimensions are maintained.

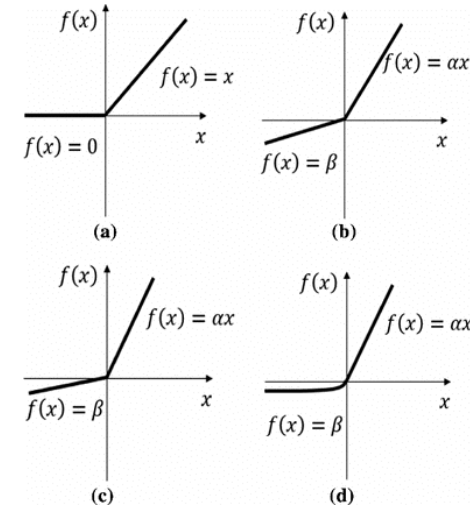
Activation Functions in CNNs

ReLU (Rectified Linear Unit):

- Mathematical Expression: $f(x) = \max(0, x)$
- Advantages: Helps in mitigating the vanishing gradient problem, computationally efficient.

Variants of ReLU:

- **Leaky ReLU:** Allows a small gradient for negative values.
- **Parametric ReLU (PReLU):** A type of leaky ReLU where the slope for negative values is learned during training.
- **Exponential Linear Unit (ELU):** Tries to make the mean activations closer to zero which speeds up learning.



Activation functions, especially ReLU and its variants, play a pivotal role in CNNs by introducing non-linearity, allowing the network to capture intricate patterns in data.

Pooling Layer

Purpose of the Pooling Layer:

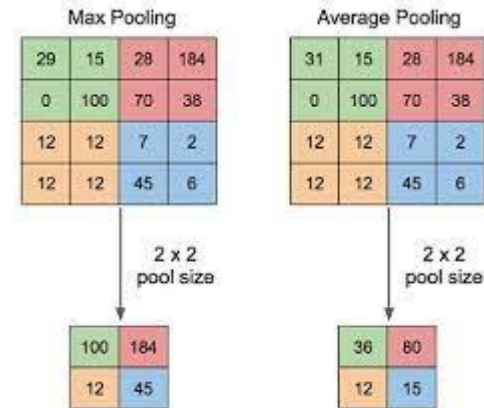
- Reduces the spatial dimensions (width & height) of the input volume.
- Helps in reducing the computational complexity.
- Provides a form of translation invariance to the model.

Types of Pooling:

- **Max Pooling:** Takes the maximum value from a portion of the input.
- **Average Pooling:** Computes the average value from a portion of the input.
- **Min Pooling:** Takes the minimum value from a portion of the input.

Stride and Window Size:

- **Stride:** The step size or the number of pixels shifted when moving the pooling window.
- **Window Size:** The size of the pooling operation (e.g., 2x2, 3x3).



Pooling layers simplify the information, retaining only the most important features for further processing

Fully Connected Layer: Bridging Features to Predictions

Role of the Fully Connected Layer:

- Connects every neuron from the previous layer to the neurons in the current layer.
- Acts as a classifier on top of the extracted features from previous layers.

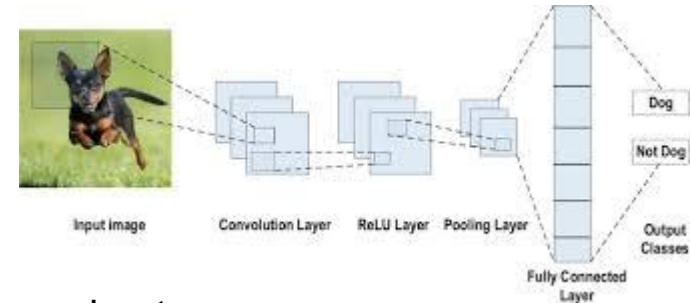
Flattening Feature Maps:

- Before connecting to the fully connected layer, the 2D or 3D feature maps are flattened into a single long continuous linear vector.
- This prepares the data for traditional neural network processing.

Making Predictions:

- The fully connected layer uses the flattened feature maps to make predictions about the input data.
- Often, the final fully connected layer has neurons equal to the number of classes in the classification problem.

The Fully Connected Layer is crucial in CNNs as it takes the spatial features, flattens them, and uses them to make final predictions, effectively acting as the decision-making body of the network.



Importance:

- It combines all the learned features from previous layers to predict the final class of the input image.
- Provides the ability to learn non-linear combinations of high-level features.

CNNs with Python (Keras)

1. Import Necessary Libraries:

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

2. Initialize the CNN:

```
model = Sequential()
```

3. Add Convolutional Layer:

```
model.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
```

4. Add Pooling Layer:

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

5. Flatten & Fully Connect:

```
model.add(Flatten())
model.add(Dense(units=128, activation='relu'))
model.add(Dense(units=1, activation='sigmoid'))
```

6. Compile the CNN:

```
model.compile(optimizer='adam',
              loss='binary_crossentropy', metrics=['accuracy'])
```

7. Train the Model:

```
model.fit(training_set, epochs=25,
          validation_data=test_set)
```

Keras provides a user-friendly and modular platform to build, train, and evaluate deep learning models, making it a popular choice for implementing CNNs in Python.

CNN Architectures

1. LeNet (1998):

- Developed by Yann LeCun.
- Primarily used for digit recognition tasks.
- Consists of 2 convolutional layers followed by 2 fully connected layers.

2. AlexNet (2012):

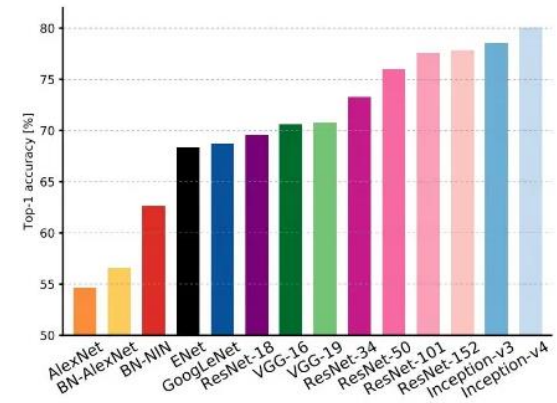
- Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton.
- Winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012.
- Introduced the use of ReLU activations and had 5 convolutional layers.

3. VGG (2014):

- Developed by the Visual Geometry Group at Oxford.
- Known for its simplicity and depth, with configurations having 16 to 19 layers.
- Uses small 3x3 convolutional filters throughout the network.

4. ResNet (2015):

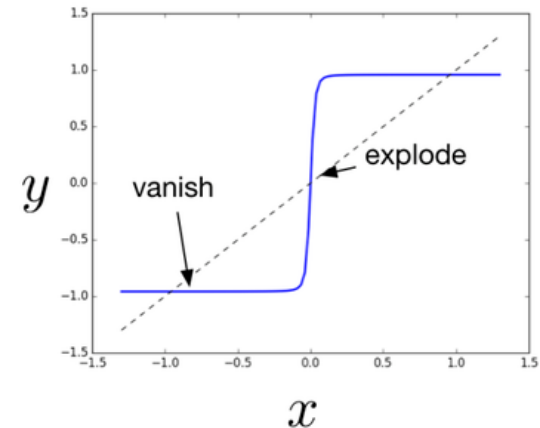
- Developed by Kaiming He and team.
- Introduced "residual blocks" to combat the vanishing gradient problem in deep networks.
- Winner of ILSVRC 2015 with architectures going up to 152 layers.



CNN architectures have evolved significantly over the years, with each new architecture introducing novel techniques and structures to improve performance and tackle challenges in deep learning.

Training a CNN

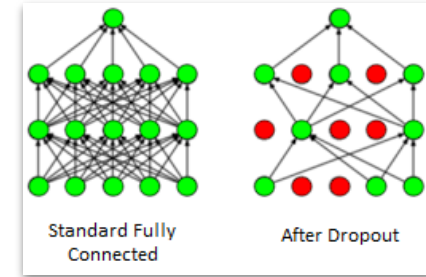
- **Backpropagation in CNNs:**
 - In CNNs, gradients flow backward from the output through all layers.
 - Each filter's weights are updated to minimize the loss.
- **Challenges in Training:**
 - **Vanishing Gradient:** As gradients are passed back through deep networks, they can become very small, causing weights to update very slowly and potentially stalling training.
 - **Exploding Gradient:** Conversely, gradients can grow exponentially in deep networks, causing large weight updates and unstable training.
- **Solutions:**
 - Use of ReLU or its variants to mitigate vanishing gradients.
 - Batch normalization to stabilize activations.
 - Gradient clipping to prevent gradients from becoming too large.



Training CNNs requires careful management of gradients. While backpropagation is a powerful tool, challenges like the vanishing and exploding gradient problems necessitate additional techniques and considerations.

Regularization Techniques in CNNs

- **Dropout:**
 - A technique where randomly selected neurons are ignored during training.
 - Helps in preventing overfitting by ensuring that the network remains robust and doesn't rely too heavily on any single neuron.
 - Typically applied on the fully connected layers.
- **Batch Normalization:**
 - Normalizes the activations of a given input volume before passing it through the layer.
 - Helps in stabilizing learning and mitigating the vanishing/exploding gradient problem.
 - Often used after convolutional and fully connected layers.
- **Data Augmentation:**
 - Process of artificially increasing the size of the training dataset.
 - Common techniques: rotating, zooming, flipping, and cropping images.
 - Helps in improving model generalization and preventing overfitting.



$$\begin{aligned}\mu_B &\leftarrow \frac{1}{m} \sum_{i=1}^m x_i && // \text{ mini-batch mean} \\ \sigma_B^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 && // \text{ mini-batch variance} \\ \hat{x}_i &\leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} && // \text{ normalize} \\ y_i &\leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) && // \text{ scale and shift}\end{aligned}$$

Regularization techniques like dropout, batch normalization, and data augmentation are essential tools in the deep learning toolkit, ensuring CNNs are both accurate and robust

Transfer Learning

Definition of Transfer Learning:

- Leveraging knowledge from one domain to improve performance in another related domain.

Benefits of Transfer Learning:

- **Speed:** Faster training as the model is already trained on a similar task.
- **Data Efficiency:** Useful when there's limited labeled data for the specific task.
- **Performance:** Can achieve better accuracy, especially for tasks with limited data.

Using Pre-trained Models:

- Models trained on large datasets can be used as a starting point.
- Layers from these models can be frozen to retain their learned patterns.

Transfer learning allows us to leverage pre-trained models on new tasks, significantly reducing training time and data requirements. By fine-tuning these models, we can achieve high performance even with limited data.

Fine-tuning:

- Adjusting the weights of the pre-trained model by continuing the backpropagation.
- Can be done on the entire model or only specific layers, depending on the task and data availability.

Common Pre-trained Models:

- VGG16, VGG19, ResNet, Inception, MobileNet, etc.

Applications:

- Image classification, object detection, segmentation in domains where data might be scarce.

Applications of CNNs

Image Classification:

- Assigning a label to an image from a set of predefined categories.
- Example: Distinguishing between images of cats and dogs.

Object Detection:

- Identifying objects within images and providing a bounding box around them.
- Example: Detecting and labeling multiple objects like cars, pedestrians, and traffic lights in a street scene.

Image Generation:

- Creating new images that resemble a set of given images.
- Techniques like Generative Adversarial Networks (GANs) are often used.

Real-world Use Cases:

- **Medical Imaging:** Detecting tumors, anomalies, or diseases in X-rays, MRIs, etc.
- **Autonomous Vehicles:** Recognizing pedestrians, other vehicles, and traffic signs to navigate safely.
- **Facial Recognition:** Identifying or verifying a person from an image.
- **Augmented Reality:** Overlaying digital information on the real world

Convolutional Neural Networks have revolutionized image processing tasks, from basic image classification to complex real-world applications like medical imaging and autonomous driving.

Challenges in CNNs

1. Need for Large Datasets:

- **Training Data:** CNNs, especially deeper architectures, require vast amounts of labeled data to train effectively. Without sufficient data, they can easily overfit.
- **Data Diversity:** It's not just about quantity. The quality and diversity of the training data play a crucial role in the model's ability to generalize.

2. Computational Demands:

- **Hardware Requirements:** Training CNNs, especially state-of-the-art architectures, demands powerful GPUs or TPUs. This can be a barrier for individual researchers or small organizations.
- **Training Time:** Even with powerful hardware, training a CNN on a large dataset can take days or even weeks.
- **Model Complexity:** Deeper networks with millions of parameters can be computationally intensive, affecting deployment in real-time applications.

While CNNs offer remarkable performance in image-related tasks, their need for extensive data and computational power can pose challenges, especially for those without access to high-end resources. It's essential to balance model complexity with available resources and the specific task at hand.

Future of CNNs

1. Evolution of Architectures:

- **Efficient Networks:** As computational resources become a concern, there's a shift towards designing efficient architectures that provide good performance with fewer parameters.
- **Capsule Networks:** Proposed as an alternative to CNNs, they aim to recognize patterns in an image regardless of their spatial orientation.

2. Beyond Image Data:

- **Video Analysis:** CNNs combined with RNNs or 3D convolutions for video classification and anomaly detection.
- **Multimodal Learning:** Integrating data from different sources (e.g., image and text) for richer representations.

3. Ethical and Fair AI:

- **Bias Mitigation:** Ensuring CNNs are trained on diverse datasets to avoid biases and ensure fairness.
- **Explainability:** Making CNN decisions more interpretable for users, especially in critical applications like healthcare.

4. Integration with Augmented Reality (AR) and Virtual Reality (VR):

- **Real-time Processing:** Using CNNs for real-time image and scene recognition in AR/VR applications.
- **Interactive Learning:** Training CNNs in an interactive environment where they can learn from user feedback.

The future of CNNs is not just about improving accuracy but also about making them more efficient, interpretable, and ethical. As technology evolves, CNNs will find applications in areas we haven't even imagined yet, making them a cornerstone of AI's future.

Conclusion and Resources

- **Pioneering Technology:** CNNs have revolutionized the field of computer vision, making tasks like image classification, object detection, and even image generation possible.
- **Continuous Evolution:** As with all technologies, CNNs continue to evolve, with researchers constantly pushing the boundaries to improve performance and efficiency.
- **Ubiquitous Application:** From healthcare to entertainment, CNNs have found applications in almost every domain, showcasing their versatility.
- **Books:**
 - a. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
 - b. "Neural Networks and Deep Learning" by Michael Nielsen.
- **Online Courses:**
 - a. [Coursera: Convolutional Neural Networks by Andrew Ng](#)
 - b. [Udacity: Deep Learning Nanodegree](#)
- **Websites:**
 - a. [Distill: Feature Visualization](#)
 - b. [CS231n: Convolutional Neural Networks for Visual Recognition](#)

CNNs are a testament to the power of deep learning. As we continue to harness this power, it's essential to stay updated with the latest advancements and best practices in the field. The provided resources are a great starting point for anyone looking to delve deeper into the world of CNNs.



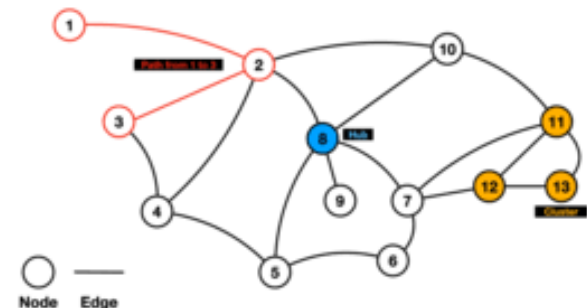
Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 1. Biological Networks

Biological Networks

- Biological networks are a way to represent and analyze complex biological systems, such as molecular interactions, signaling pathways, and metabolic pathways.
- A biological network is a collection of nodes (or vertices) and edges (or links) that connect them.
- Nodes represent biological entities, such as genes, proteins, metabolites, or cells, while edges represent the relationships or interactions between them.



- Are distinguished between
 - Networks in the cell
 - Networks outside the cell



Nature Reviews | Genetics

- These networks can be represented mathematically as graphs and can be analyzed using graph theory and network science techniques.
- Understanding the structure and dynamics of biological networks is crucial for understanding complex biological phenomena and for developing new therapeutic strategies.

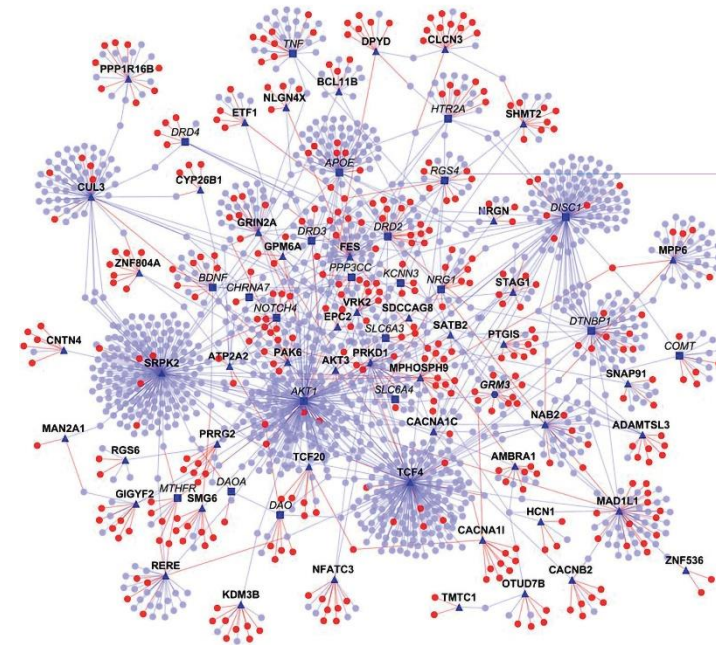


Biological networks on bioinformatics

- The study of biological networks has become a major focus of bioinformatics research in recent years.
- It can lead to
 - a better understanding of biological systems
 - identification of disease biomarkers
 - and development of new drugs.
 - Identifying key nodes, critical for the network, and its functioning (potential targets).
- Overall, the use of biological networks in bioinformatics has revolutionized the way we study and understand complex biological systems, and holds great promise for advancing our knowledge of the underlying mechanisms of life.

Protein – Protein Interaction

- Protein-protein interaction networks (PPINs) are graphs that represent the physical interactions between proteins in a cell.
- Analyzing PPINs can provide insights
 - into the roles of individual proteins
 - their interactions in disease processes.



Algorithms on Protein Protein Networks

- Algorithms applied to PPINs can provide valuable insights into disease mechanisms and potential drug targets
- Steps of constructing such an algorithm
 - Construction of a PPIN
 - Identification of network modules or clusters of highly interconnected proteins
 - Integration of drug-target information and gene expression data
 - Prioritization of potential drug targets based on their network centrality and association with disease-related pathways

Metabolic Networks

- Metabolic networks are a type of biological network that represents the interconnected set of metabolic reactions within a cell or organism.
- Directed graphs
 - Nodes correspond to metabolites
 - Edges correspond to reactions between metabolites.
- Examples
 - *Escherichia coli*
 - *Saccharomyces cerevisiae*





Importance of Metabolic networks

- Important for understanding the complex interactions between metabolites and enzymes that govern metabolic processes within cells.
- Prediction of the behavior of metabolic systems under different conditions
- Identification of potential drug targets for metabolic diseases.
- However, the analysis of metabolic networks is challenging due to their complexity and the need to integrate data from multiple sources, including enzyme kinetics, gene expression, and metabolomics data.

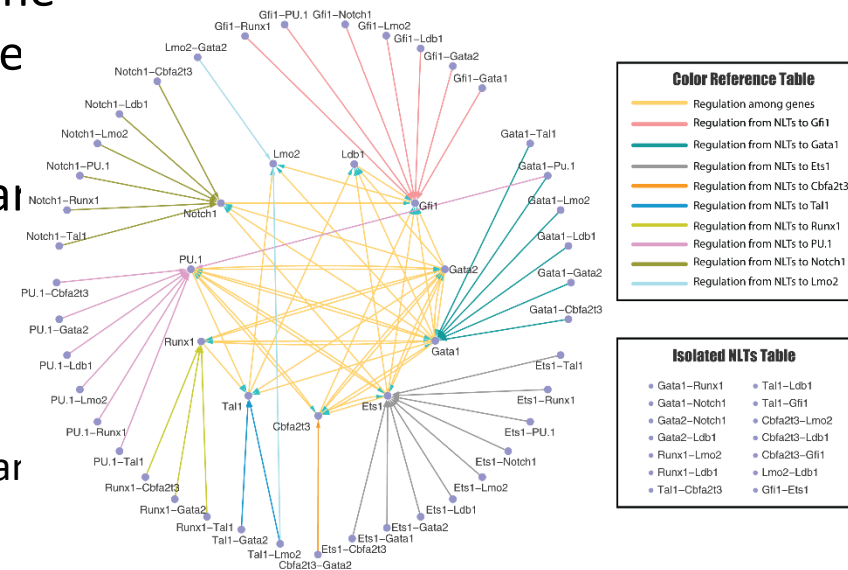


Clustering on metabolic networks

- Clustering algorithms can be applied to identify groups of metabolites or enzymes that are functionally related and may be involved in specific biological pathways.
- Overall, clustering algorithms provide a powerful tool for the analysis of metabolic networks
 - Identification of functionally related metabolites and enzymes
 - providing insights into key metabolic pathways and potential drug targets.

Gene Regulatory networks

- Gene regulatory networks are a type of biological network that model the interactions between genes and the regulators.
- Used to understand how genes are regulated and how their expression is controlled.
- Examples of gene regulatory networks
 - interactions between transcription factors and their target genes
 - interactions between microRNAs and their target mRNAs.





Gene Regulatory networks importance

- Crucial role in understanding the regulation of gene expression in various biological processes.
- They are essential for understanding the molecular mechanisms that underlie development, differentiation, and disease.
- Identification of key regulatory genes and pathways, potential drug targets, and biomarkers for disease diagnosis and treatment.
- Identification of genetic variants associated with disease susceptibility and drug response, as well as predicting the effects of mutations and environmental factors on gene expression.



Regulatory networks in bioinformatics

- Clustering algorithms have also been applied to gene regulatory networks to identify groups of genes with similar functions.
- One example of this is the use of hierarchical clustering on gene expression data to identify co-regulated genes.
- Clusters could be used to predict the function of previously uncharacterized genes.
- This approach has the potential to accelerate the discovery of new regulatory mechanisms and therapeutic target



Exercise

- Define biological networks
- Name 3 categories of biological networks
- How can these types of networks be applied to bioinformatics? Name one of their applications
- Give one network example of each category

Summary

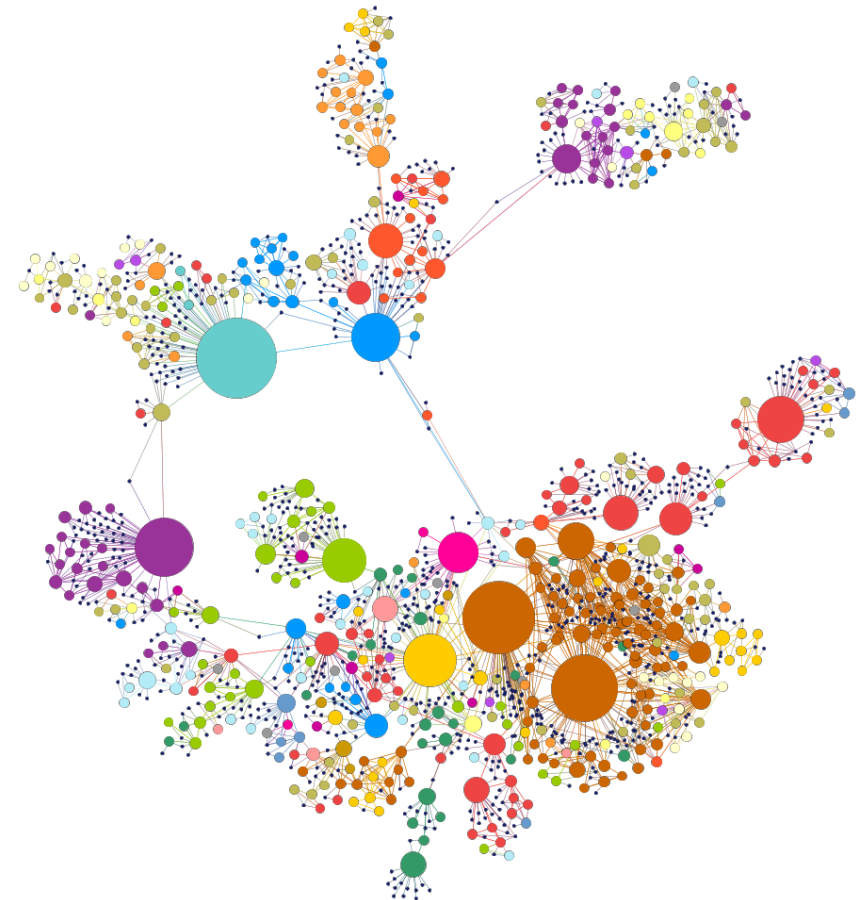
- Biological networks are a way to represent and analyze complex biological systems, such as molecular interactions, signaling pathways, and metabolic pathways.
- They are a collection of nodes (or vertices) and edges (or links) that connect them.
- Examples of biological networks
 - Protein-Protein
 - Metabolic
 - Gene Regulatory
- Crucial for research

Disease Networks

- Disease Networks are graphical representations of
 - the complex relationships among diseases
 - their genetic components
 - and their molecular mechanisms.
- They are constructed by integrating various types of biological data, such as
 - gene expression
 - protein-protein interactions
 - and pathways.

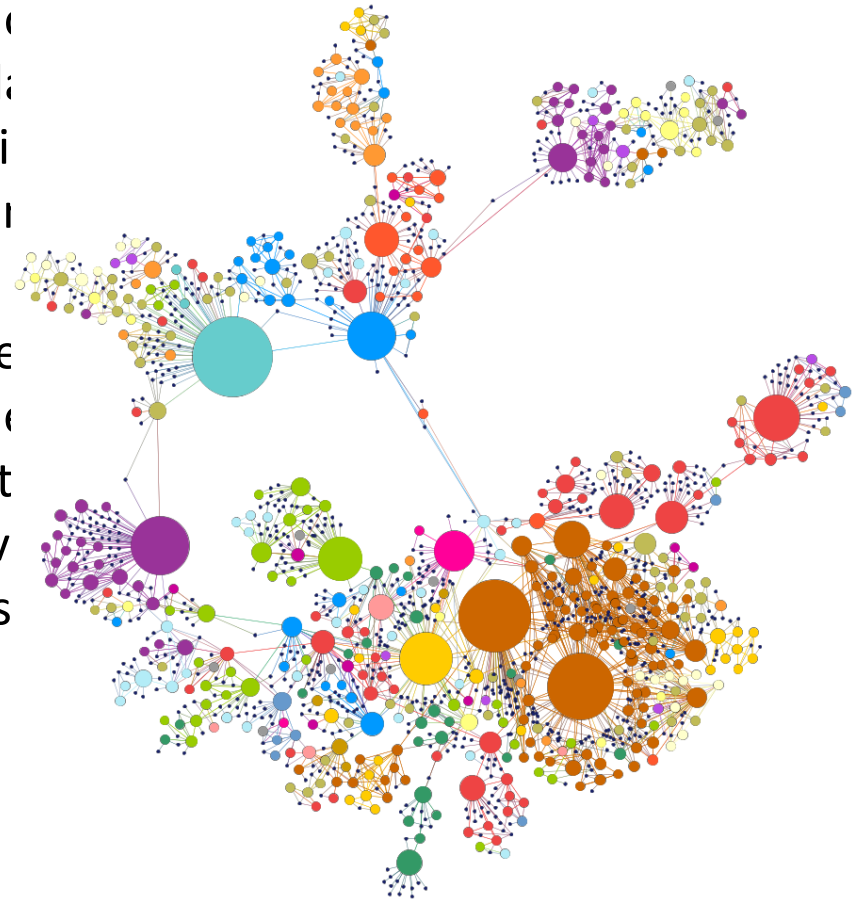
Human Disease Network

- Nodes represent diseases
- Diseases are connected to each other if they share at least one gene in which mutations are associated with both diseases
- In this example
 - visibly clustered according to major disease classes



Disease Networks in Bioinformatics

- Network analysis reveals clusters of diseases with similar molecular mechanisms, suggesting potential shared drug targets and treatment strategies.
- Key diseases and genes identified through network analysis are further investigated in wet-lab experiments leading to the discovery of a novel therapeutic target for a previously untreatable disease.



Biological Neural Networks

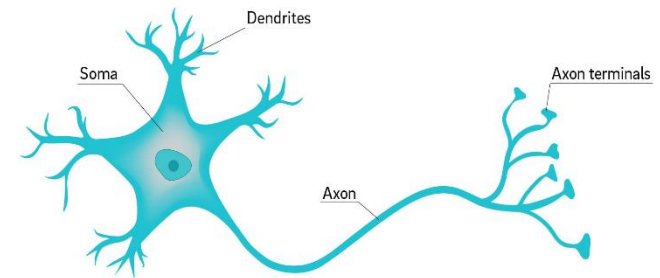
- Biological neural networks refer to the networks of interconnected neurons present in the nervous system of living organisms.
- These networks play a crucial role
 - in processing information,
 - controlling behavior,
 - regulating bodily functions.

- The brain is the most well-known example of a biological neural network, consisting of billions of neurons and trillions of connections.
- Other examples include the neural networks found
 - in the spinal cord,
 - the retina of the eye,
 - and the enteric nervous system in the gut.

Neural Networks in Bioinformatics

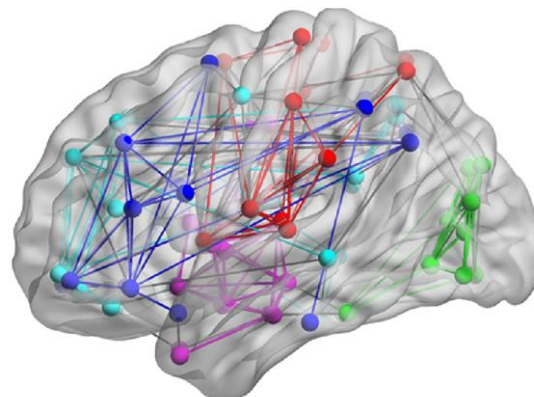
- Studying biological neural networks can help us understand how the brain works and how it processes information.
- This knowledge can lead to the development of more advanced machine learning algorithms and artificial intelligence.
- Researchers are also studying neural networks to gain insights into neurological disorders and develop treatments for them.

Neuron



Brain networks

- Brain networks refer to the complex networks of neurons and their connections in the brain that underlie cognitive function and behavior.
- Structural networks: these are the physical connections between neurons and brain regions. They can be studied using diffusion tensor imaging (DTI) or tractography methods.
- Functional networks: these are the patterns of coordinated activity between brain regions. They can be studied using functional magnetic resonance imaging (fMRI) or electroencephalography (EEG).



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness



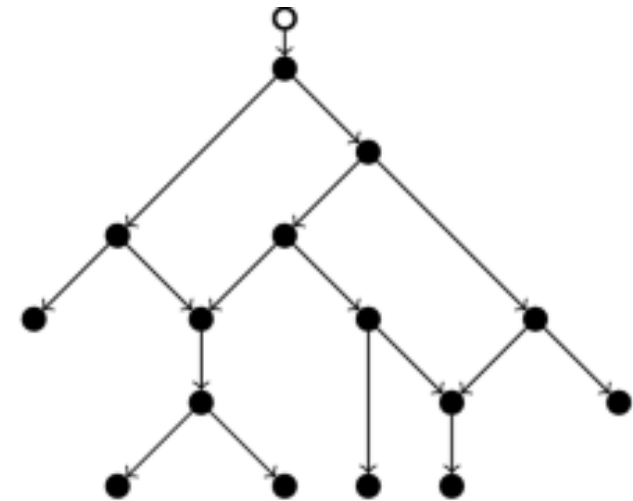
Brain networks in Bioinformatics

- Brain networks are a key focus of neuroinformatics, which is the study of the organization and function of the brain at the systems level.
- Understanding the structure and function of brain networks is crucial for understanding how the brain works and for developing treatments for neurological and psychiatric disorders.
- Network analysis methods can be applied to brain networks to identify patterns of connectivity that are associated with specific cognitive or behavioral functions.

- One of the main applications is the classification of different brain states
- In many occasions the metrics of brain networks produced by people with the same condition (epilepsy, AD, ADHD) differ from healthy people
- After training a large dataset, we can conclude by the network of a person if he is sick or not.

Phylogenetic Networks

- Type of biological network that represents evolutionary relationships between species or genes that cannot be captured by traditional tree-based approaches.
- They are used to study evolutionary events like hybridization, horizontal gene transfer, and recombination that create networks with reticulate structure.





Importance of phylogenetic Networks

- They provide a more accurate representation of the evolutionary history of species or genes.
- Traditional tree-based approaches assume that evolutionary relationships are strictly bifurcating, but this is not always the case.
- Can help researchers gain a more complete understanding of the evolution of biological systems.



Exercise

- Name the 4 categories of biological networks mentioned in the lesson
- How can these types of networks be applied to bioinformatics? Name one of their applications
- Give one network example of each category

Summary

- In this lesson we explored certain types of biological networks and their importance on bioinformatics. These networks are
 - Disease Networks
 - Complex relationships among diseases
 - Biological Neural Networks
 - Connections between neurons
 - Brain networks
 - Connections between regions of the brain/neurons
 - Phylogenetic networks
 - Evolutionary trees



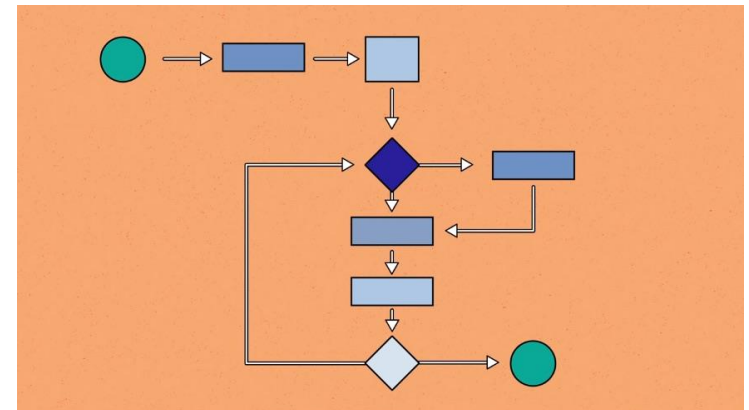
Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 2. Algorithms

Algorithms

- Algorithms are procedures for solving problems or performing tasks.
- They are essential in computer science and related fields, as they provide a systematic way to solve complex problems.
- In bioinformatics, algorithms are used to analyze biological data, model biological systems, and simulate complex biological processes.
- Examples of algorithms on bioinformatics
 - sequence alignment algorithms for comparing DNA and protein sequences,
 - network analysis algorithms for studying biological networks
 - Machine learning algorithms for predicting biological outcomes

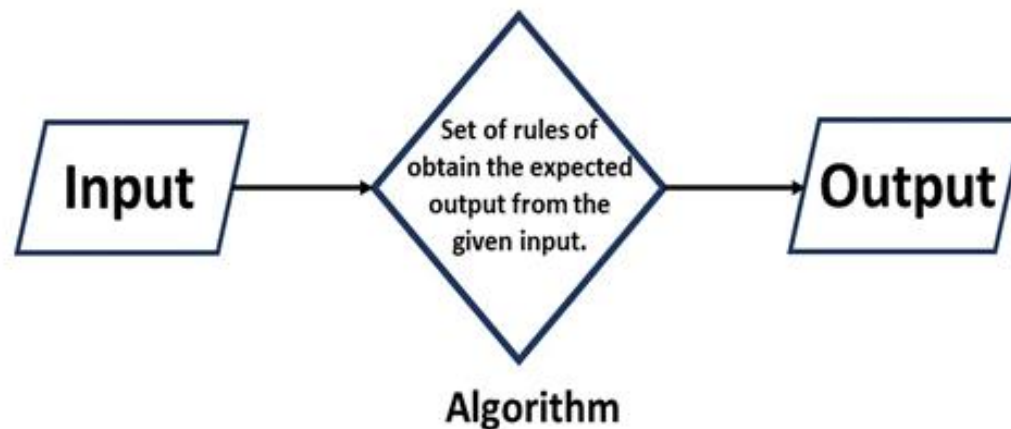


Goal of the course

- The goal of this course is to introduce students to algorithms and their applications in bioinformatics.
- We will cover different types of algorithms,
- their properties
- complexity
- and how they can be applied to solve biological problems.

Algorithms, Definition

- Algorithms are a set of well-defined instructions for solving a problem or performing a task.
- They consist of a series of steps that transform inputs into outputs.



Steps

- Inputs: The data or information that the algorithm operates on.
- Outputs: The result of the algorithm.
- Control structures: The decision-making mechanisms that control the flow of the algorithm, such as conditional statements and loops.
- Operations: The actions that the algorithm performs on the input data to transform it into the output data

Applications

- Algorithms are used to solve problems in computer science, engineering, mathematics, and other fields.
- They are used to automate tasks, optimize processes, and make decisions based on data.
- Examples
 - Search algorithms for finding information in a database or on the internet.
 - Sorting algorithms for organizing data in a particular order.
 - Encryption algorithms for securing data transmission.
 - Optimization algorithms for minimizing cost or maximizing efficiency.

Example

- Making Coffee
- Inputs:
 - Coffee, filter, coffee maker, mug
- Outputs
 - Cup of Coffee
- Steps
 - Insert the coffee to the coffee maker
 - Insert the filter
 - Turn on the machine
 - When the coffee is ready, turn it off
 - Place it in a mug

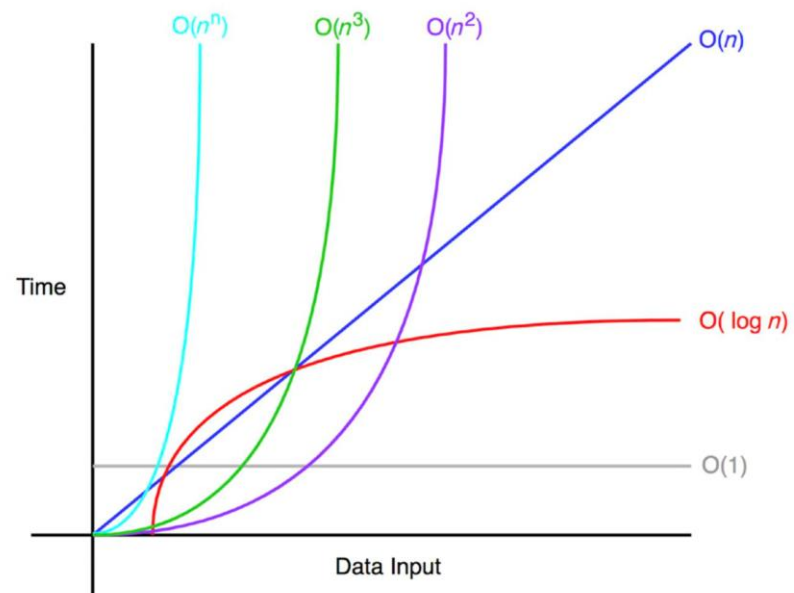


Analysis of Algorithms

- It's important to analyze algorithms to ensure they are efficient and scalable for large inputs in terms of time and space complexity.
- Time complexity refers to the amount of time an algorithm takes to run as the input size increases.
- Space complexity refers to the amount of memory an algorithm uses as the input size increases.

Big – O notation

- One way to analyze algorithms is using Big-O notation, which expresses the upper bound of the running time or space usage of an algorithm.
- Common time complexities and their Big-O notations include:
 - Constant time: $O(1)$
 - Linear time: $O(n)$
 - Quadratic time: $O(n^2)$
 - Exponential time: $O(2^n)$

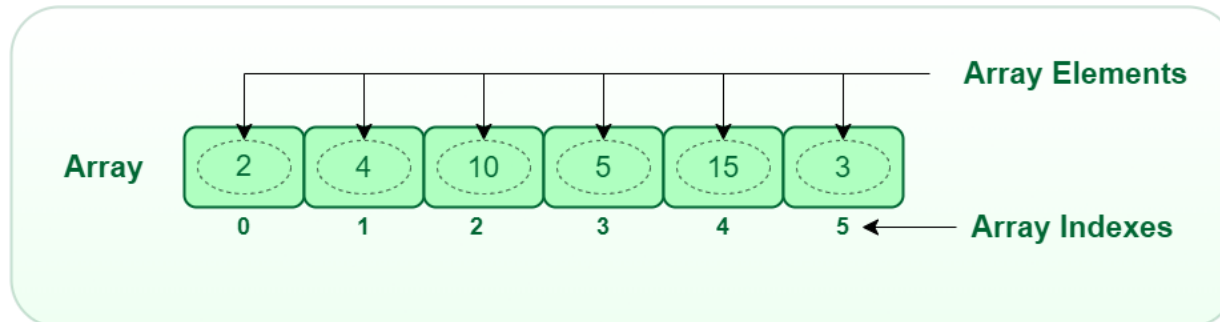


Algorithm Design Techniques

- Divide and Conquer: breaking a problem into subproblems, solving them recursively, and combining the solutions.
 - Merge sort
- Dynamic Programming: breaking a problem into smaller subproblems and solving each subproblem only once.
 - Example: Fibonacci sequence.
- Greedy Algorithms: A technique that makes the locally optimal choice at each step in the hope of finding a global optimum.
 - Huffman coding.
- Backtracking: searching for a solution by building candidates incrementally and rejecting them as soon as they can't possibly lead to a valid solution.
 - Example: Sudoku solver.

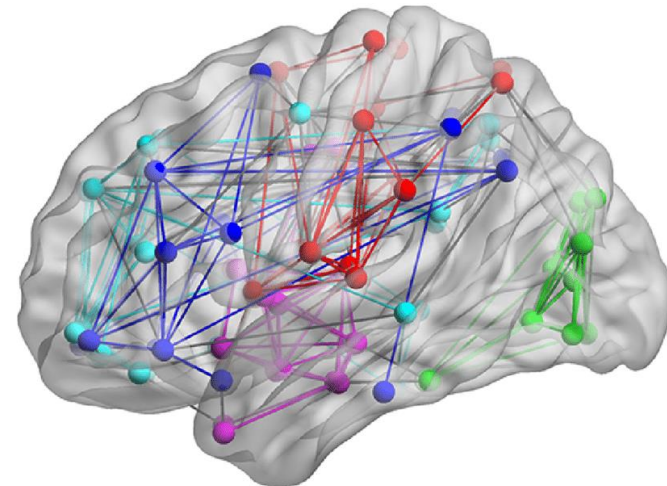
Data Structures

- Data Structures: Arrays, linked lists, stacks, queues, trees, and graphs.
- Arrays: A collection of elements of the same type that are stored in contiguous memory locations.
- Linked Lists: A collection of elements, each containing a reference to the next element in the list.
- Stacks: A collection of elements that supports two operations: push (add an element to the top of the stack) and pop (remove the top element from the stack).
- Graphs: A collection of vertices (nodes) and edges that connect them.



Link to Bioinformatics

- Overview of how algorithms are used in bioinformatics and computational biology.
 - Use of algorithms to
 - Visualize,
 - Analyze,
 - interpret
- biological data.
- Importance of algorithmic techniques for identifying patterns in biological data.



- Algorithmic Analysis of Biological Networks:
 - Overview of how algorithms can be used to analyze biological networks.
 - Examples of algorithms used in network analysis, such as clustering algorithms, theoretical network modelling, network alignment.
- Takeaway:
 - Algorithms play a critical role in the analysis of biological data, especially in the context of biological networks. Understanding how algorithms can be used in bioinformatics can lead to new insights and discoveries in the field.



Exercise

- **Exercise 1**

Let's assume a real life procedure. You are making a toast. State the inputs, outputs and the steps that are needed.

- **Exercise 2**

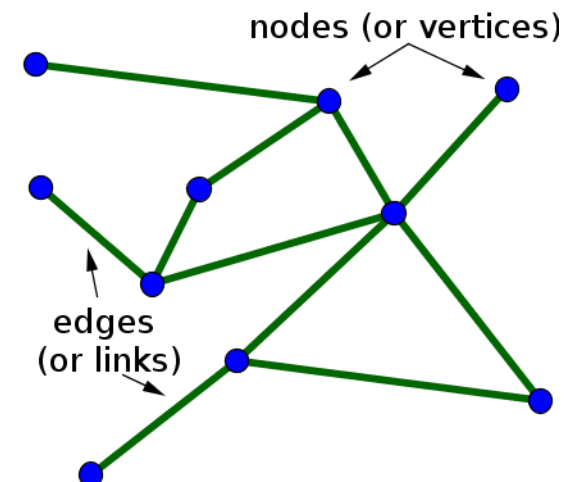
Create your own example and follow the same process.

Summary

- Algorithms are defined procedures, important in computer science and related fields
- An algorithm has Inputs, Outputs, control structures and operations
- Technique Design
 - Divide and conquer
 - Dynamic Programming
 - Greedy Algorithms
- Applications on Bioinformatics, where the algorithms help us
 - Visualize, Analyze and Interpret the Biological data
 - Network analysis of Biological networks

Graph Algorithms and Bioinformatics

- Graph algorithms play a crucial role in bioinformatics, as biological data can often be represented as graphs
- Graph algorithms are a collection of algorithms that are designed to solve problems on graphs
- Graphs
 - Mathematical structures that consist of nodes and vertices
 - Represent the interaction between nodes

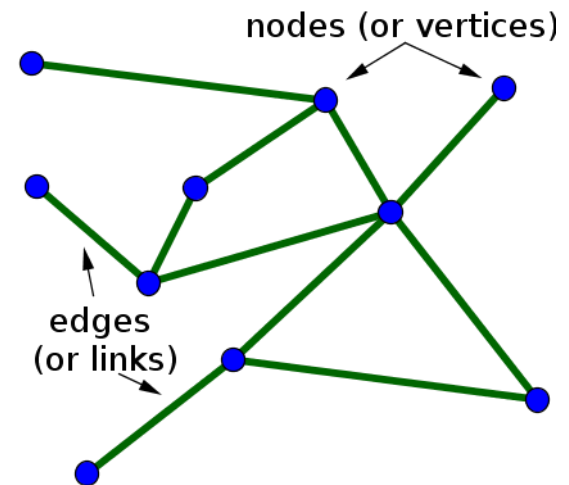


- In bioinformatics, graphs can be used to represent a variety of biological data such as protein-protein interaction networks, gene regulatory networks, and metabolic pathways.
- Analyzing these biological networks using graph algorithms, researchers can gain insights into the underlying biological processes and interactions.
- In the following slides, we will explore some common graph algorithms used in bioinformatics and their applications.



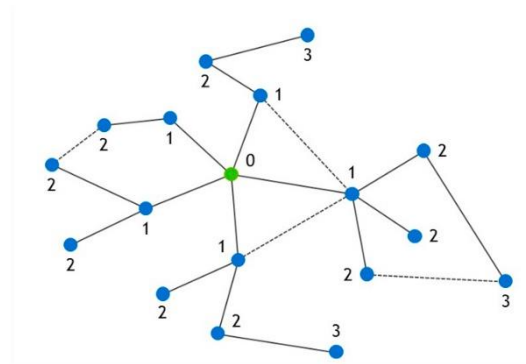
Examples on Graphs

- Dynamic programming is a powerful algorithmic technique that can be used to solve complex optimization problems
- It is often applied to graph problems that arise in the analysis of biological data.
- Examples of dynamic programming in Graphs:
- Finding the
 - Shortest
 - Longest paths
- Between two vertices in a graph



Example

- These algorithms work by computing the distance between nodes in a graph,
- Distance is defined as the number of the edges along the path between the nodes.
- The shortest path algorithm finds the path with the minimum distance
- longest path algorithm finds the path with the maximum distance.



Dijkstra's algorithm

- Create a set of unvisited nodes and set the distance of the start node to 0.
- While there are unvisited nodes, select the node with the smallest distance and mark it as visited.
- For each neighbor of the selected node, calculate the distance from the start node through the selected node.
- If the calculated distance is smaller than the current distance of the neighbor, update the distance.
- Repeat steps 2-4 until the end node is visited or there are no more unvisited nodes.
- The distance of the end node is the shortest path from the start node.

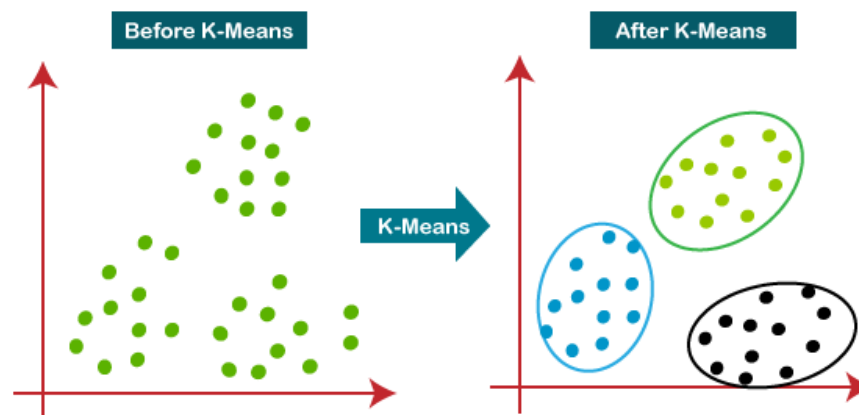
Randomized algorithms

- Randomized algorithms are algorithms that make use of random numbers to solve computational problems.
- These algorithms are useful in situations where the input data is too large or too complex for deterministic algorithms to handle efficiently.
- In bioinformatics, randomized algorithms can be used for tasks such as clustering and classification.
- Fast and memory efficient.
- less predictable and harder to interpret than those of deterministic algorithms.



K-Means Clustering

- One example of a randomized algorithm in bioinformatics is k-means clustering.
- This algorithm is used to partition a set of data points into k clusters, where each data point belongs to the cluster with the closest mean.
- The algorithm works by randomly selecting k initial cluster centroids, assigning each data point to the closest centroid,
- Then updating the centroids based on the mean of the points assigned to each cluster.



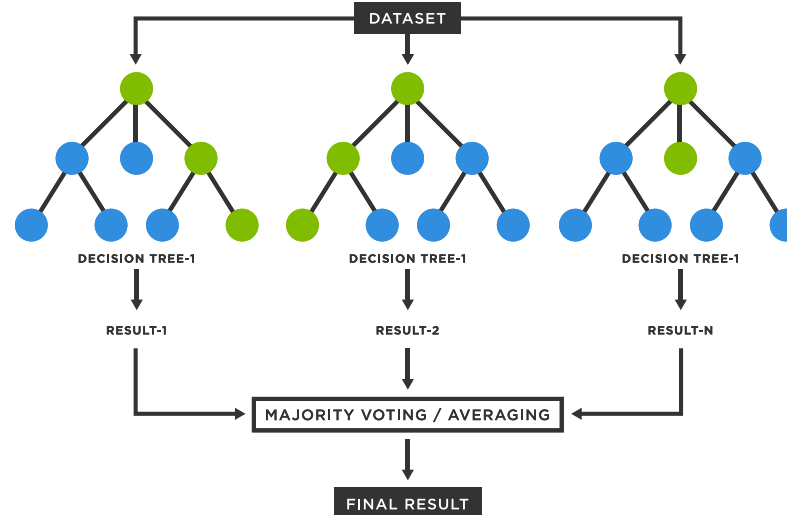
ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Random Forest

- Another example of a randomized algorithm in bioinformatics is the Random Forest algorithm for classification.
- This algorithm uses an ensemble of decision trees, each trained on a randomly selected subset of the data, to classify new data points.
- The algorithm combines the predictions of the individual trees to make a final classification.

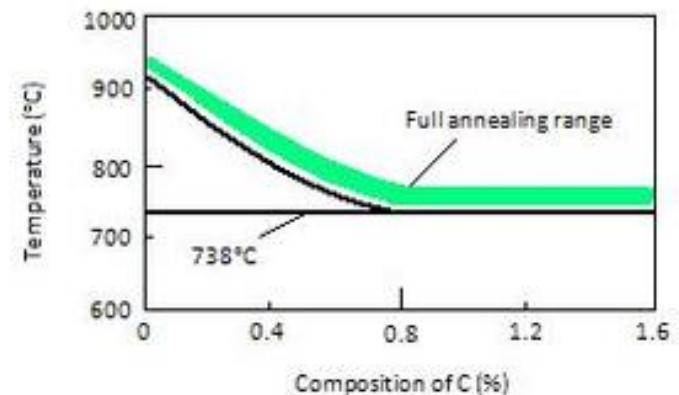


Approximation algorithms

- Approximation algorithms are algorithms that find an approximate solution to a problem that is difficult or impossible to solve exactly.
- In bioinformatics, these algorithms are used to solve optimization problems related to protein structure prediction, gene function prediction, and other areas.
- Often sacrificing accuracy for speed.
- Approximation algorithms can provide a near-optimal solution by using a heuristic approach to explore a subset of the possible conformations.

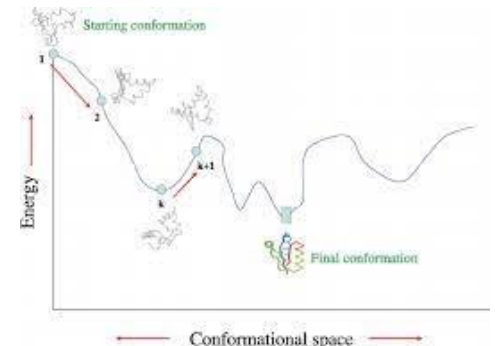
Simulated Annealing

- Another example is the simulated annealing algorithm
- Randomized algorithm that can be used to find near-optimal solutions to complex optimization problems.
- This algorithm is inspired by the process of annealing in metallurgy
- where a metal
 - heated
 - then slowly cooled
- to increase its strength and durability



Simulated annealing

- It can be applied to a wide range of optimization problems in bioinformatics, such as protein structure prediction, DNA sequencing, and genome assembly.
- In protein structure prediction, simulated annealing is often used to find the conformation of the protein that has the lowest energy.
- The algorithm starts with a random conformation
- iteratively searches for better conformations



Steps

- Formulate the problem: Define the graph problem you want to solve, such as graph clustering.
- Define the objective function: Create an objective function that quantifies the quality of a solution to the graph problem
- Initialize the solution: Start with an initial solution, which could be random
- Define the neighborhood: Define a set of possible solutions, that can be reached from the current solution. In a graph context, this could involve modifying the assignment of values to graph nodes or edges.
- Accept or reject new solutions: Compare the quality of the new solution with the current solution. If the new solution improves the objective function value, accept it as the new current solution. If it is worse, accept it with a certain probability based on the Simulated Annealing schedule
- Perform iterations: Iteratively explore the solution space by generating new solutions from the current solution. At each iteration, select a random neighboring solution and evaluate its quality using the objective function.

Exercise

- Combine what you learned and
 - Define the categories of algorithms mentioned in the lesson
 - Name one similarity or one difference between them
 - Pick one algorithm example and mention its steps

Summary

- Algorithms are essential tools of bioinformatics
- Graph algorithms are important tools which help us understand and visualize the structure of biological entities
- Categories of algorithms
 - Dynamic Programming (breaking down into smaller problems)
 - Shortest Paths
 - Longest Paths
 - Randomized (use of random numbers to solve)
 - K-means
 - Random Forest
 - Approximation (Find approximate solutions)
 - Simulated Annealing



Module 3. Bioinformatic tools

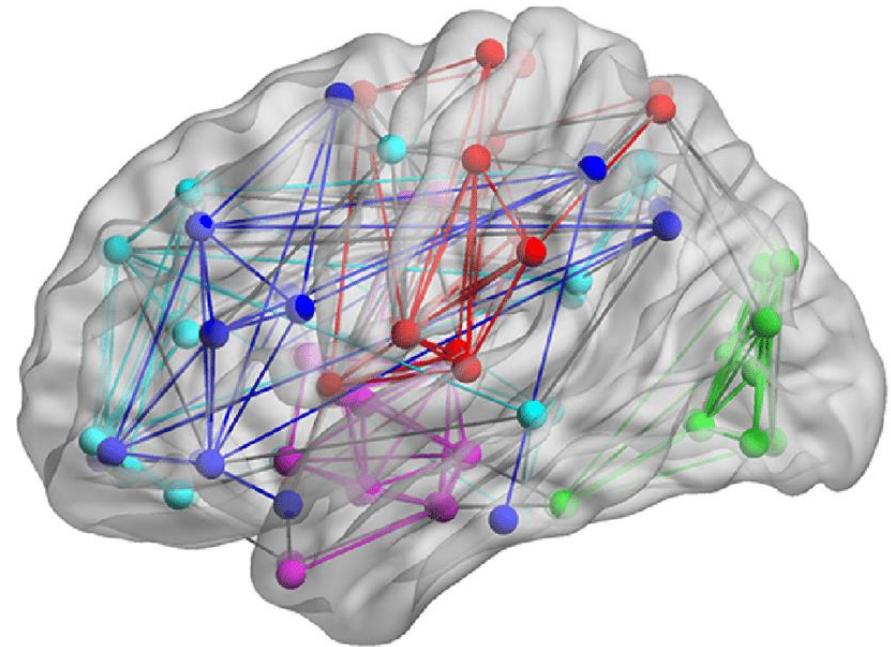
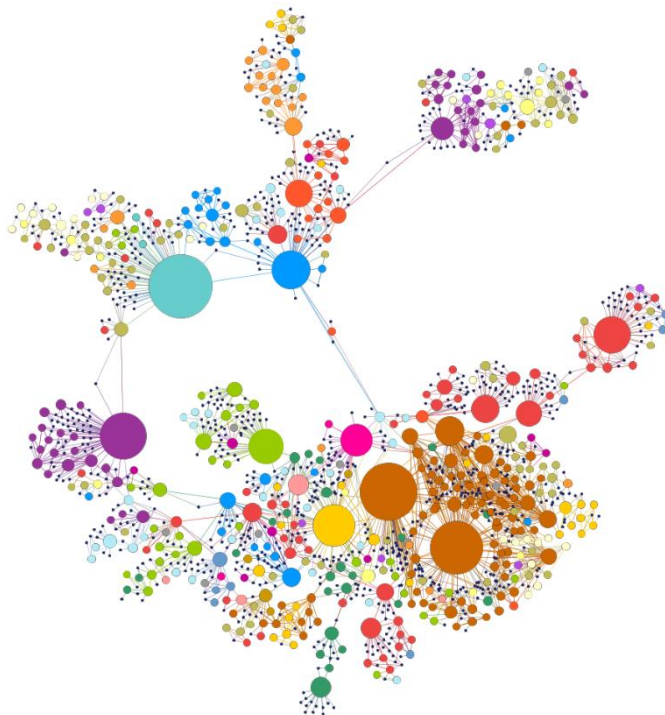
Topic 5. Algorithms in Bioinformatics

Lesson 3. Network Properties

Network Modelling

- Network modeling is the study of complex systems composed of interconnected elements, represented as nodes and edges.
- An important tool in bioinformatics because biological systems are inherently complex and interconnected.
- By modeling biological systems as networks, researchers can gain insights into their structure, function, and evolution.
- In this lesson, we will introduce basic concepts in network modeling

- A network consists of two basic components: nodes and edges.
- Nodes represent individual entities in the network, while edges represent the relationships between nodes.

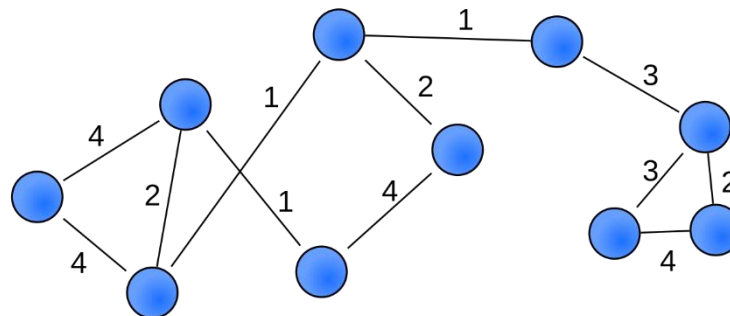


ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices
Action Type KA226 - Partnerships for Digital Education Readiness

Weighted networks

- In a weighted network, each edge is assigned a weight or value that represents the strength or importance of the connection between two nodes.
- Weighted networks are used to capture the strength of interactions between nodes in various biological systems.
- More accurate representation of biological systems than unweighted networks, as they take into account the strength of interactions between nodes.
- Example
 - Protein-Protein Interaction Networks (PPINs): In PPINs, the weight of an edge can represent the confidence score of the interaction between two proteins.

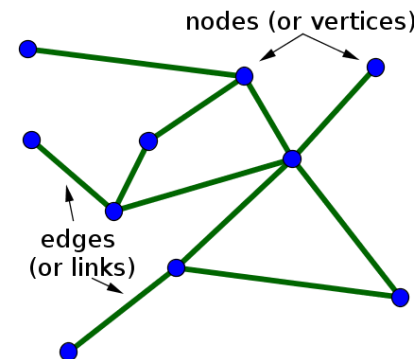


Unweighted networks

- Unweighted networks refer to networks where all edges have equal importance and are not assigned any weight.
- In contrast to weighted networks, unweighted networks only capture the presence or absence of connections between nodes.
- Examples
 - Metabolic networks: These networks represent the metabolic pathways within a cell and are often modeled as unweighted networks.

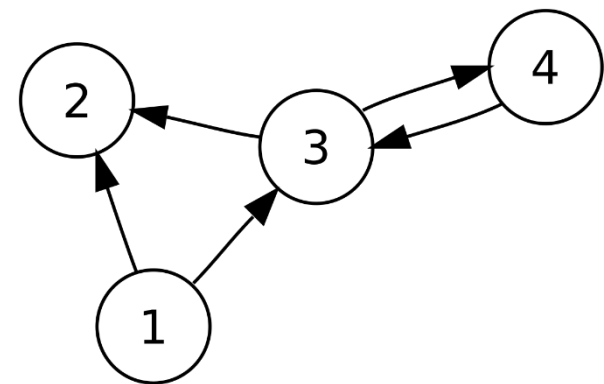
Undirected Networks

- In an undirected network, the edges do not have a direction or a specific orientation. They simply represent a connection between two nodes.
- This means that if node A is connected to node B, then node B is also connected to node A.
- Examples
 - Social networks: Friendship networks, where two individuals who consider themselves friends are connected by an undirected edge.
 - Biological networks: Metabolic networks, where nodes represent metabolites and edges represent reactions between them.



Directed Networks

- A directed network is a type of network in which the edges have a direction and represent a one-way relationship between nodes.
- Examples:
 - Social networks: Twitter, where users can follow others, but those others may not necessarily follow them back
 - Gene regulatory networks: where a gene can regulate the expression of another gene, but not vice versa

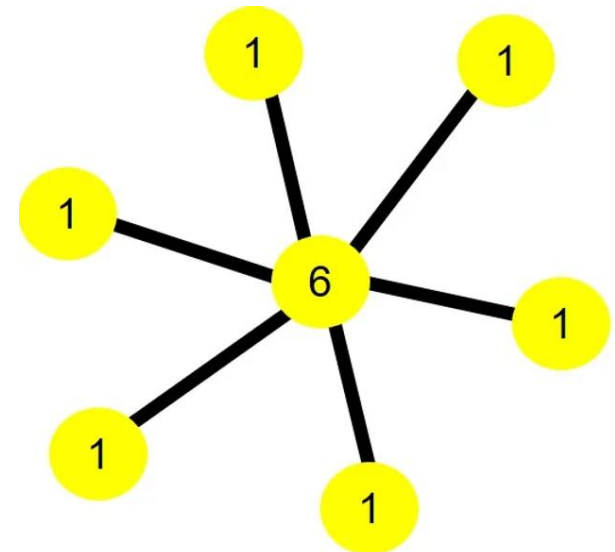


Degree Centrality

- Degree centrality is a measure of the number of connections a node has in a network.
- It is calculated as the number of edges connected to a node, also known as its degree.

Calculating the degree centrality of this network:

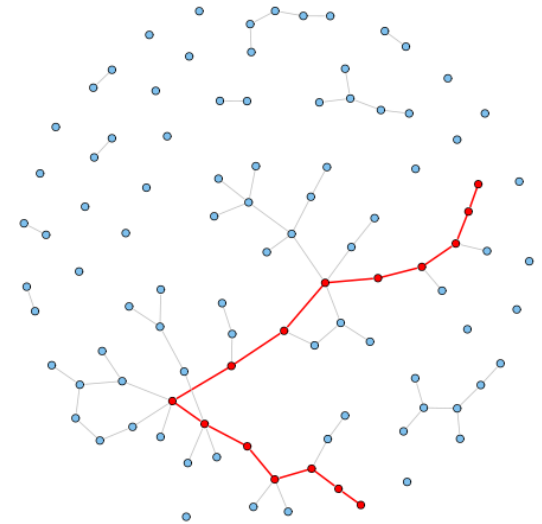
- The peripheral nodes have degree 1
- The central 6



- Degree centrality is a widely used metric in network analysis that measures the number of connections that a node has with other nodes in the network.
- It is important because it provides insights into the relative importance of nodes in a network.
- Nodes with high degree centrality are considered to be more important in the network, as they have a large number of connections with other nodes

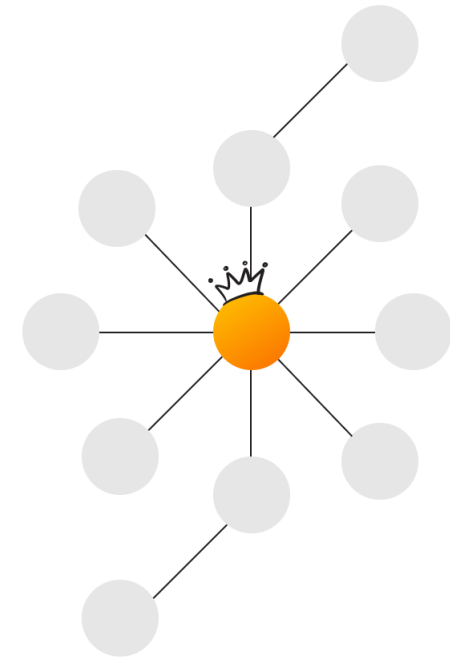
Shortest path length

- The shortest path length between two nodes in a network is the minimum number of edges that must be traversed to go from one node to the other.
- Shortest path length is an important measure of network efficiency and robustness.
- It can help identify key nodes or edges that are important for the overall connectivity of the network.



Betweenness centrality

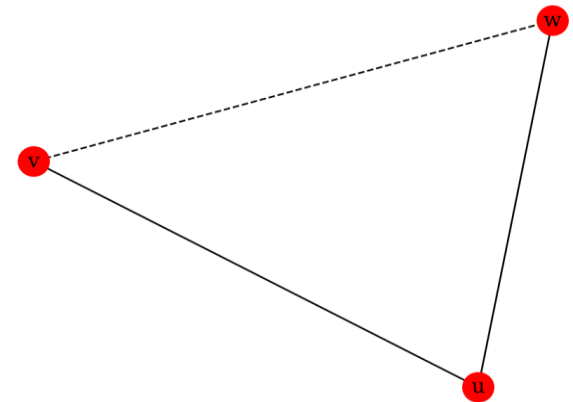
- Betweenness centrality measures how often a node lies on the shortest path between other nodes in the network.
- A node with high betweenness centrality is important in maintaining the flow of information in the network.
- It can also help to identify key nodes that, if removed, could greatly impact the overall network structure and function.



A node with a high degree centrality score

Clustering coefficient

- The clustering coefficient measures the degree to which nodes in a network tend to cluster together. It is defined as the fraction of pairs of a node's neighbors that are also neighbors of each other.
- The clustering coefficient of a node is calculated by dividing the number of edges between the node's neighbors by the total number of possible edges between them.
- High clustering coefficients can indicate the presence of tightly-knit groups or communities within a network, while low clustering coefficients can indicate a more random or disorganized structure



Network theoretical modelling

- While many networks in bioinformatics are based on real-world data
- However, theoretical models provide a useful framework for understanding network behavior and properties.
- They mimic real networks' properties
- There are many different types of theoretical network models, each with their own set of assumptions and characteristics.

Importance

- **Understanding Biological Systems:** Networks provide a powerful tool for understanding complex biological systems, including genetic, metabolic, and disease networks.
- **Predicting Network Behavior:** By analyzing network properties and simulating theoretical network models, researchers can predict how biological networks will behave under different conditions.
- **Identifying Key Components:** Network properties such as degree centrality, betweenness centrality, and clustering coefficient can be used to identify key components within biological networks, such as important genes or proteins.
- **Advancing Precision Medicine:** Network modeling can help to identify individual-specific disease networks, which can be used to develop personalized treatment plans for patients based on their unique genetic and environmental factors.

Exercise

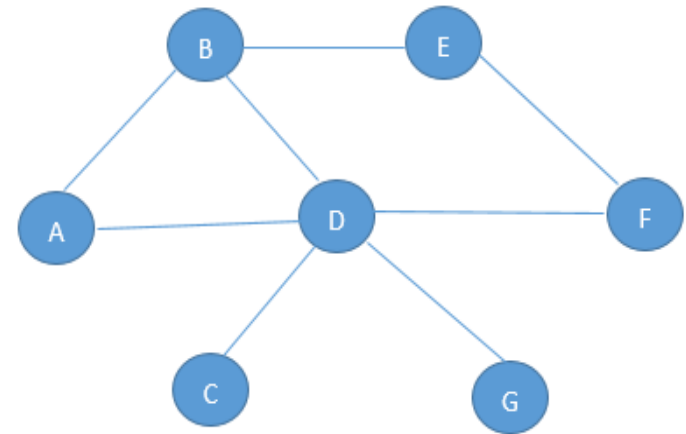
- Exercise 1

Define the network properties mentioned in this lesson. What do high/low values of these properties mean for this network?

- Exercise 2

Given the network in the picture, compute:

- The degree centrality of the nodes A, E, D.
- The shortest path length between The nodes (A,G), (D,E)



Summary

- In this lesson we have introduced basic network concepts
 - Directed Networks
 - Undirected Networks
 - Weighted Networks
 - Unweighted Networks
- We have defined network properties
 - Degree Centrality
 - Betweenness Centrality
 - Shortest Paths Length
 - Clustering Coefficient
- Theoretical Network modelling
- Importance of Network modelling in Bioinformatics



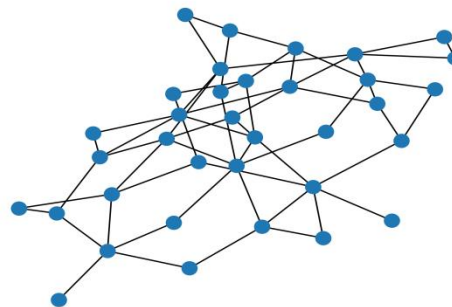
Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 4. Random Networks

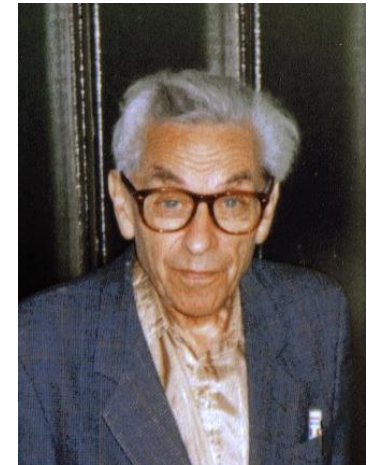
Network Models

- Network science: Reproduces properties of real networks.
- The simplest network topology is the random network.
- Links between the nodes are randomly chosen.
- Each interaction has the same probability to exist.



Erdős - Rényi Model

- $G(N, L)$ Model
 - N nodes with L randomly put links (used by Erdős and Rényi)
- $G(N, p)$ Model
 - N nodes and every edge has the same probability p to exist
- Random graphs are named Erdős - Rényi Networks for their contribution
- To construct the network
 - Start with N nodes
 - Create a random number for a node pair.
 - If the number is smaller than p then connect the links.
 - Repeat for every node pair in the network



Pál Erdős



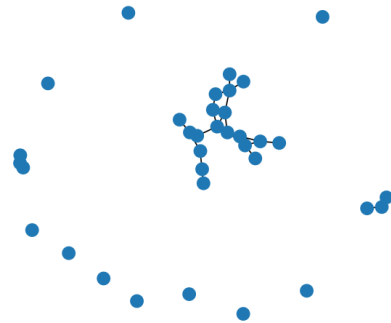
Alfréd Rényi

G(N, p) Model

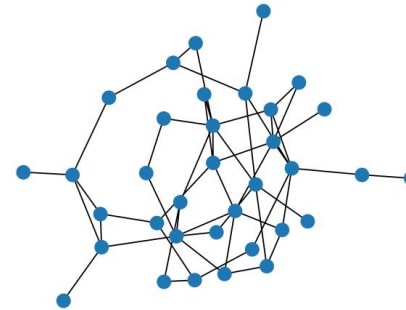
- All the node pairs are $L_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$
- Expected number of links: $\langle L \rangle = p \frac{N(N-1)}{2}$
- Mean degree: $\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1)$

Where k_i is the number of nodes, node i is connected with

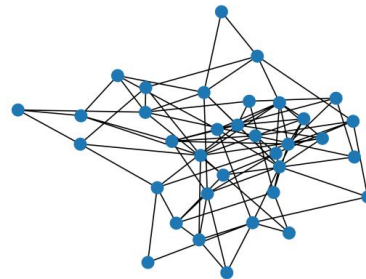
Probability p



$p = 0.05$



$p = 0.1$



$p = 0.15$

ERASMUS+

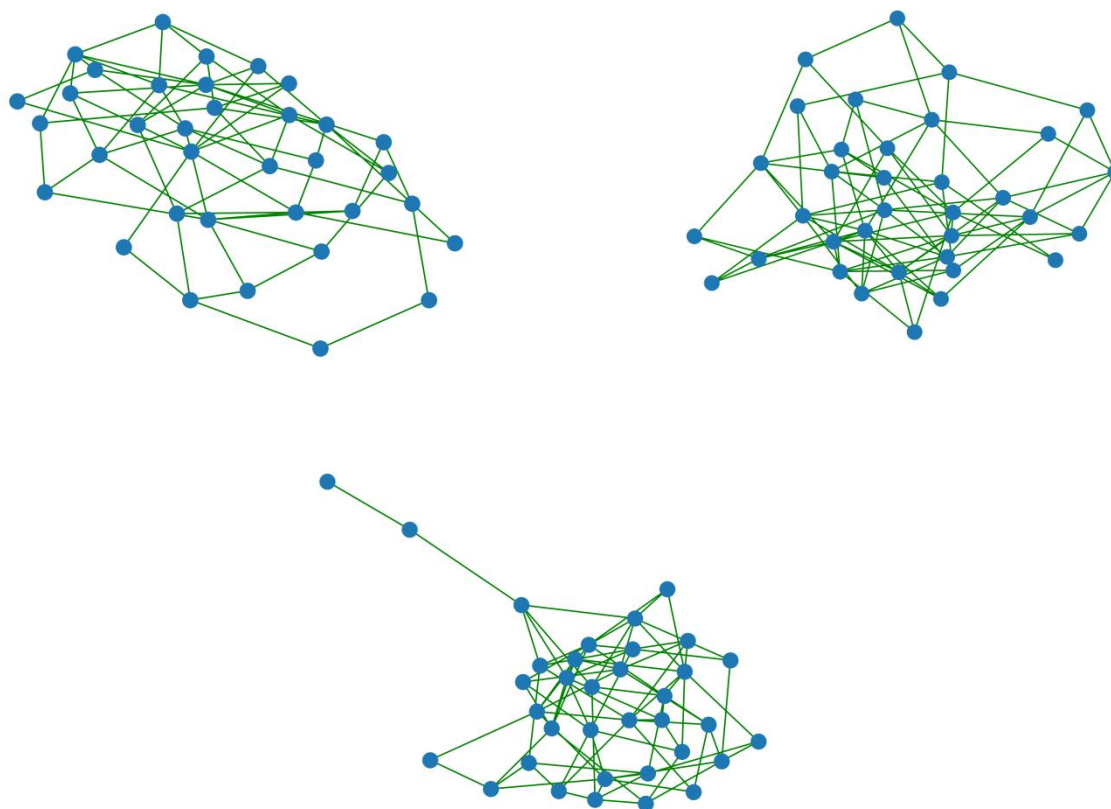
Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Stochastic model

- Probability p is the parameter that controls the density of the network
- However, $G(N, p)$ is a stochastic model, created by pseudorandom numbers
- Seed plays an important part
- Below we present 3 networks with $N=35$, $p = 0.15$ and different seeds

$N=35$, $p=0.15$



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Randomness

- p and N are the same
- However, these graphs have different number of total links
- Similar but not exactly the same structure
- If the first pseudorandom number (seed) and the random number generator was the same, we would have exactly the same network in every realization

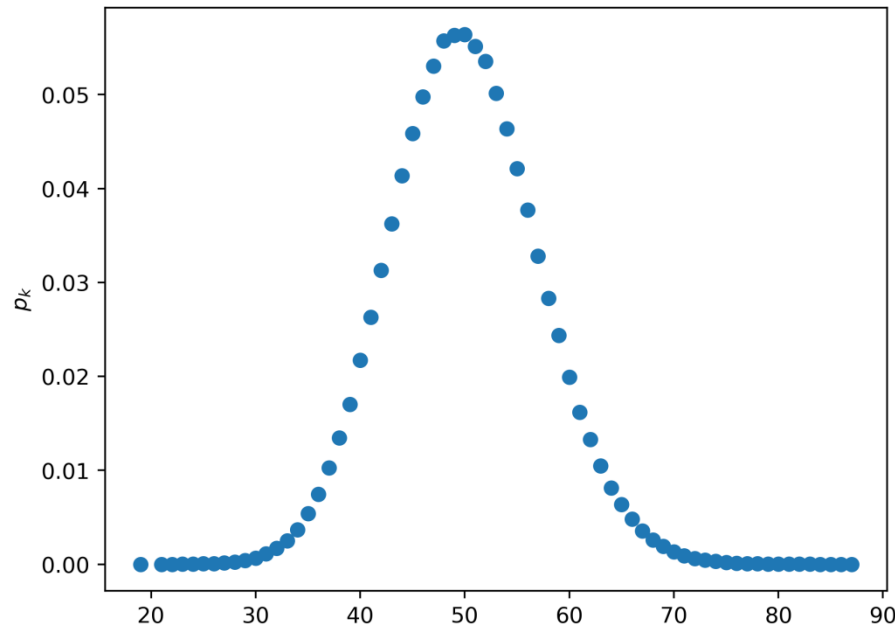
Degree Distribution

- Probability that k links are present: p^k
- Probability that the rest of the links are missing: $(1-p)^{N-1-k}$
- $\binom{N-1}{k}$ ways to choose k other vertices.
- Probability of being connected to k other nodes:

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

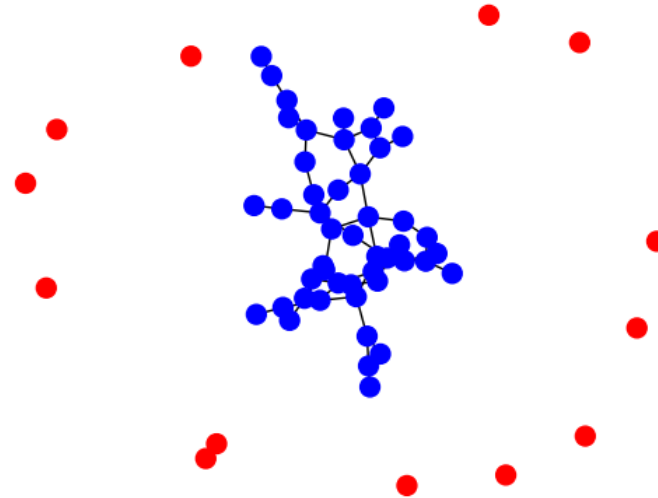
- Binomial distribution

Degree Distribution



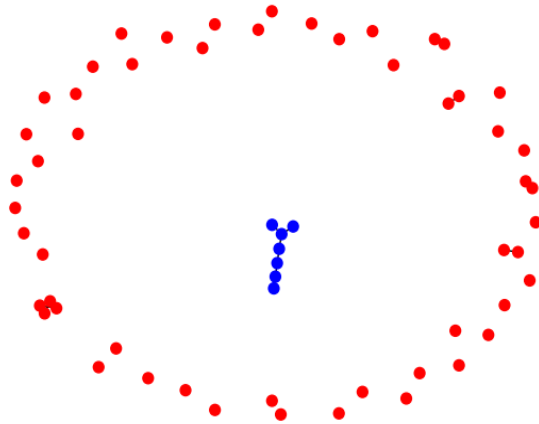
Degree distribution for a random network with $N = 10000$, $p = 0.005$, averaged for 100 realizations. The format of the distribution is binomial

Giant component



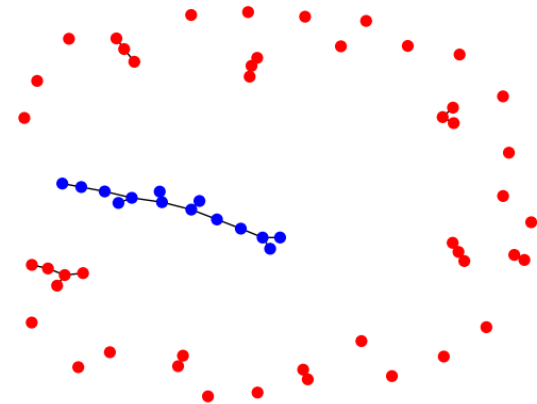
- When p is large enough a big fraction of the network is connected
- That fraction is named Giant Component (The blue nodes in the picture)

Clustering coefficient



- $\langle k \rangle < 1$

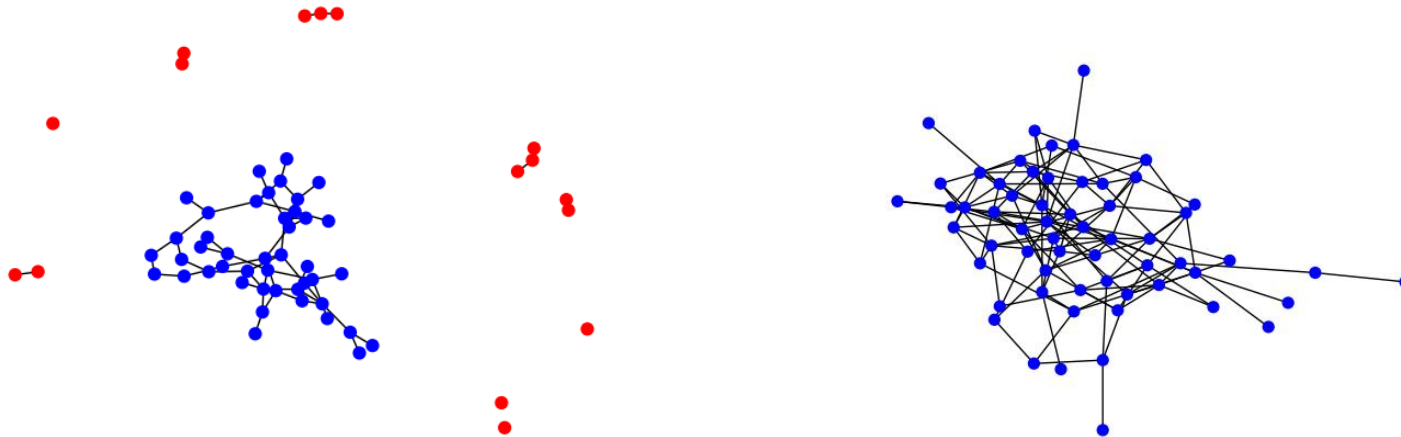
- No Giant Component



- $\langle k \rangle = 1$

- No Giant Component

Network theoretical modelling



- $\langle k \rangle > 2$

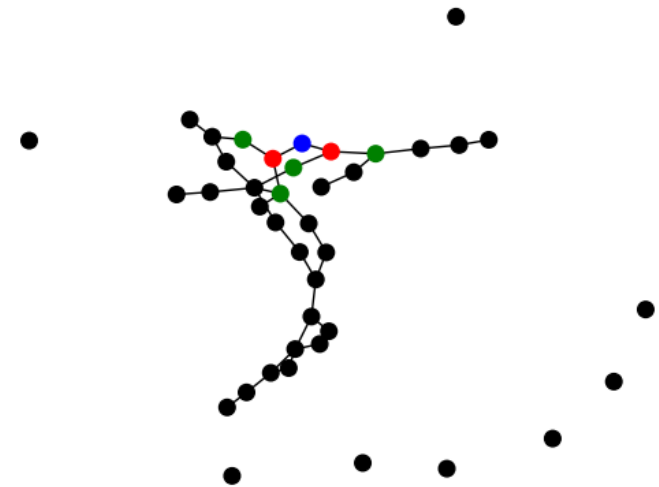
- Single Giant Component

- $\langle k \rangle \gg \ln(N)$

- Single Giant Component

Path lengths

- A random node (blue node) has on average
 - $\langle k \rangle$ neighbors at distance 1 (red nodes)
 - $\langle k \rangle^2$ neighbors at distance 2 (green nodes)
 - $\langle k \rangle^d$ neighbors at distance d
- Shortest path length
A path between two nodes with the minimum number of edges



$N = 40, p = 0.051$

Small World properties

- Average distance $\langle d \rangle$
Average distance between all node pairs
- Network diameter d_{\max}
Shortest distance between the most distant nodes
- $\langle d \rangle$ and d_{\max} are small on Erdős - Rényi networks
 - Small world phenomena

Exercise 1

- Create an Erdős–Rényi network with $N = 10000$ nodes. Apply the rule that between two nodes there is a probability $p = 1/6$ that there is a connection.
- Find the number of k connections of each node. It will be a table with $N = 1000$ values of integers.
- Make a plot of the distribution of k , $P(k)$, as a function of k and calculate the mean value of k . The results should be an average of 1000 simulations.
- Do the same for a network with $N = 100000$ nodes.

Exercise 2

- Create Erdős–Rényi networks with $N=10000$ nodes following the same process as in exercise 1, for $\langle k \rangle = 1, 3, 15$.
- Probability p will be computed as a function of $\langle k \rangle$ using the relations presented in the powerpoint.
- For each $\langle k \rangle$ create 1000 realizations. For each realization calculate the number of nodes that belong to the giant component.
- Plot the distribution of the relative size of the largest cluster (nodes that belong to the giant component as a fraction to the total nodes) for each $\langle k \rangle$.

Summary

- In this lesson we have introduced the random network model and focused on its properties
- The analyzed properties were:
 - Degree distribution
 - Clustering coefficient
 - Giant component
 - Path lengths
- Small world properties



Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 5. Small World Networks

Milgram's Experiment

- Real social networks have small average path lengths.
- To confirm it, psychologist Stanley Milgram conducted an experiment.
- He sent 296 packages to randomly chosen people from Omaha



Stanley Milgram

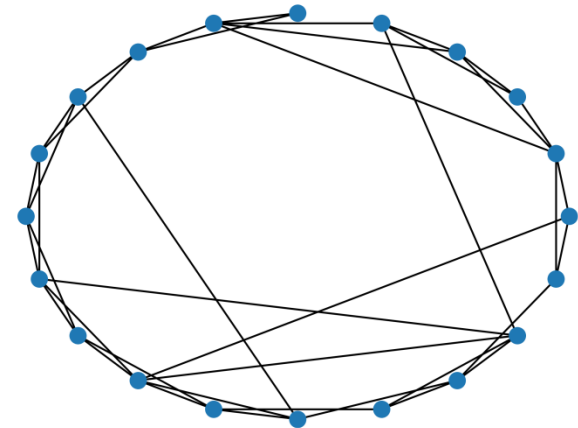
(photo by Wikipedia)

Milgram's Experiment

- Each person had information of a target (a Milgram's friend)
- Every person had to send the packages to someone that:
 - They knew
 - They thought he/she was close to the target
- Target received 64/296 packages
- Average path length for the received packages: 5.5

Small World Networks

- Networks with the following properties
 - Highly clustered
 - Short average path lengths
- They are applied to
 - Sociology
 - Earth Sciences
 - Computing
 - Brain networks



Clustering Coefficient

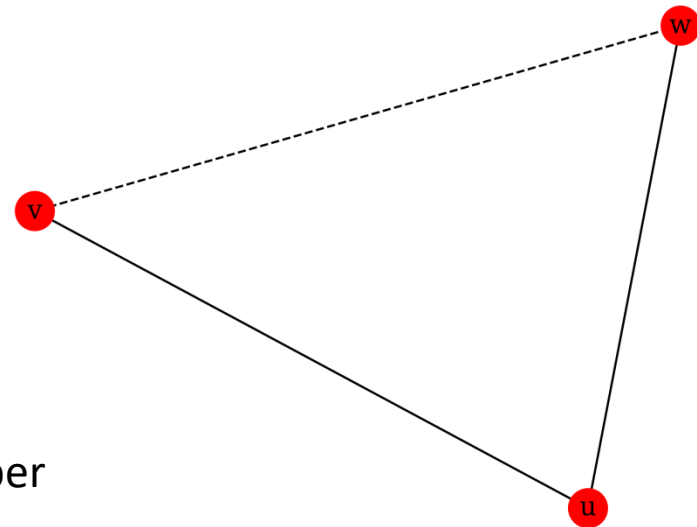
- u is a friend of v and w is a friend of u. Is v a friend of w as well?

- If v is a friend of w the triad vuw is closed

- **Clustering coefficient**

$C = (\text{number of closed path lengths of 2}) / (\text{number of path lengths of 2})$

- Quantifies the closed triads

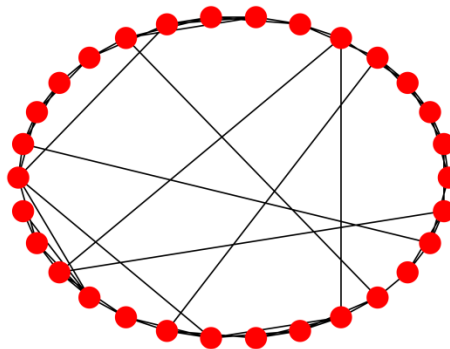


Clustering Coefficient

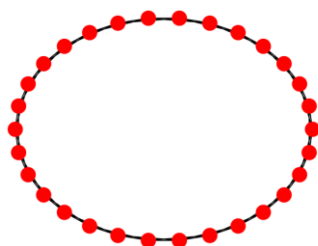
- Let's suppose we have a random network with every node having the same number of connections c .
- Assume u is a friend of v and w is a friend of w .
- The probability that w is a friend of u is c/N .
- A typical value for c/N would be $3 \cdot 10^{-4}$ (Film actors, Newman: Networks, an introduction).
- However much larger values have been observed (in social or technological networks as an example).
- A friend of a friend is more likely to be also my friend.

Watts Strogatz graphs

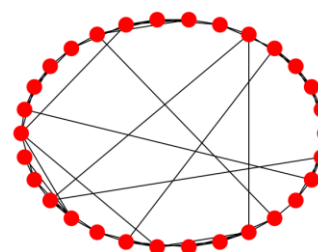
- Best known model that creates small world graphs
- Presented by Duncan Watts and Steve Strogatz (Watts and Strogatz 1998)
- **Model:**
 - Start with N nodes, each one connected with k neighbors
 - Rewire the second node of each edge with probability p



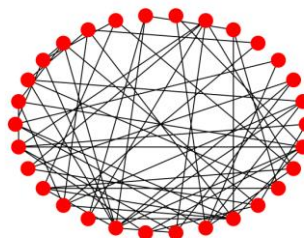
Variation of p



$p=0$



$p=0.2$

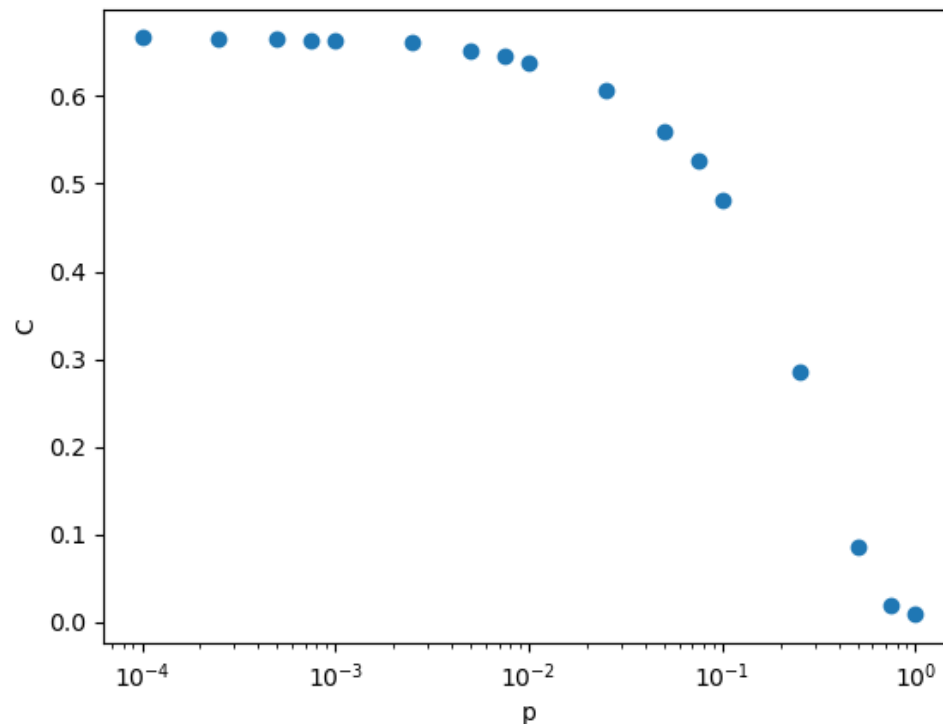


$p=1$

Randomness

- p defines randomness.
- At $p=0$, we see a perfectly symmetrical graph. Each node is connected with k closest neighbors.
- When p increases, randomness increases
- At $p = 1$ we have a random graph

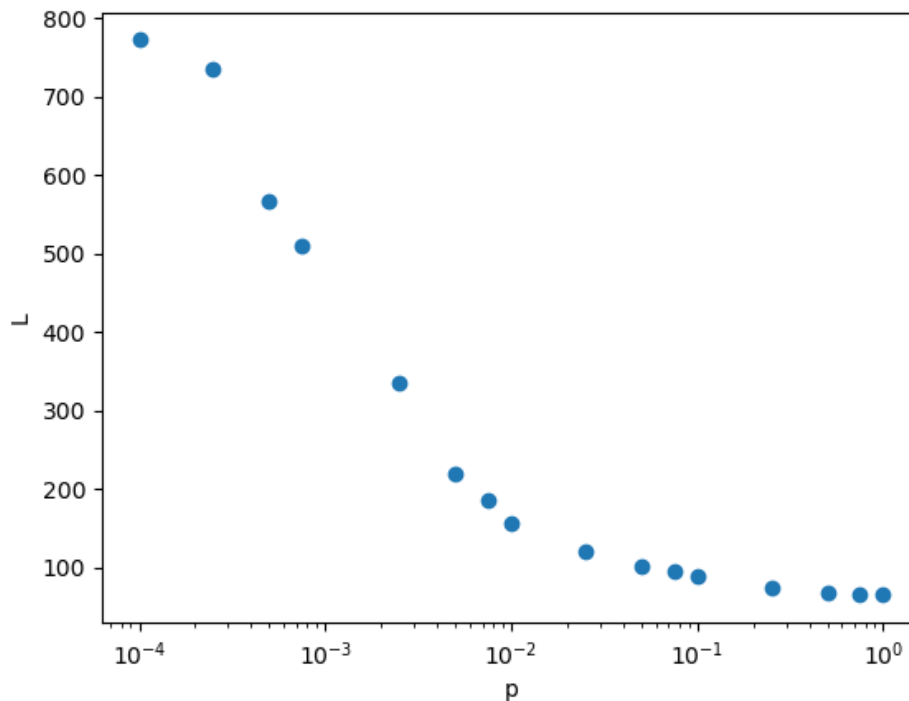
High Clustering



$N = 1000, \langle k \rangle = 10$

- We observe large values of the clustering coefficient for a wide range of values of rewiring probability

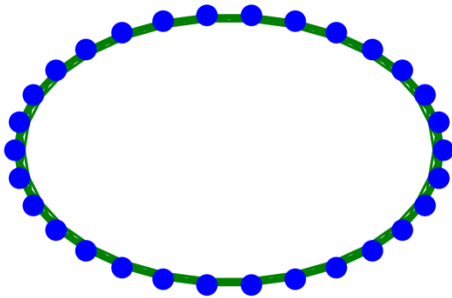
Low average shortest path length



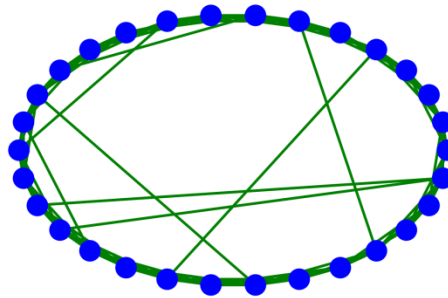
- We observe small values of the average shortest path for a wide range of values of the rewiring probability

$N = 1000, \langle k \rangle = 10$

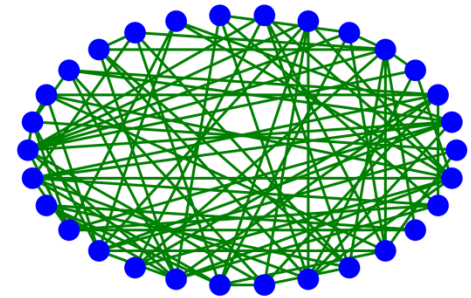
Increasing p



$p = 0$
High Clustering
High Diameter

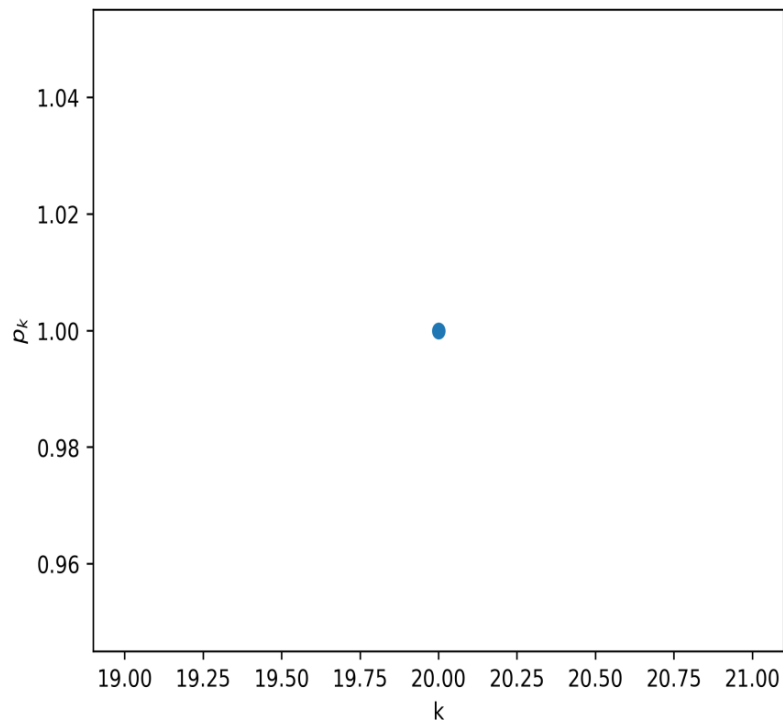


$p = 0.1$
High Clustering
Low Diameter



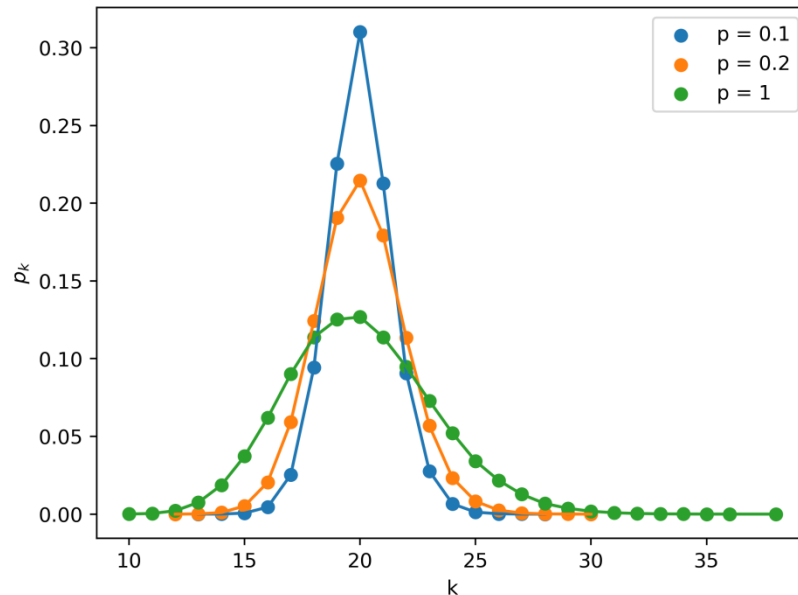
$p = 1$
Low Clustering
Low Diameter

Degree Distribution, $p=0$



- When $p=0$, all the nodes have exactly k connections. The degree distribution is a point.

Degree Distribution



- When p increases, the distribution broadens. There is larger variation in the values of the connections each node has. (Randomness)

- When $p=1$, the network is a random network

Exercise 1

- Create Small World Watts Strogatz networks with $N = 1000$.
- Initially, every node of the network will have exactly k connections and afterwards the edges will be rewired with probability equal to p .
- Prepare the following 2 plots: Plot the probability distribution function PDF, $P(k)$ as a function of k , while keeping constant rewiring probability in the first plot ($p = 0.4$) and constant $\langle k \rangle$ in the second ($\langle k \rangle = 10$).
- 3 curves in both plots are expected. In the first one, plot one curve for each $\langle k \rangle$ equal to $\langle k \rangle = 6, 10, 16$.
- In the second plot, prepare 3 curves, one for each rewiring probability equal to $p = 0.1, 0.2, 0.4$. The results should be an average of 1000 simulations.

Exercise 2

- Create Watts Strogatz networks with $N=10000$, $\langle k \rangle = 6, 10, 15$.
- For each $\langle k \rangle$ calculate the Clustering Coefficient and the Shortest Average Path as a function of the rewiring probability p ,
- The range of p will be 0 to 1 with a step of 0.1.
- The results will be averages of 1000 realizations. Compare the quantities for the different values of $\langle k \rangle$.

Summary

- Real world networks have
 - High clustering
 - Short average paths
- Like the Small World Networks. However the Small World Networks produce:
- Unrealistic Degree Distribution
 - Real world networks have many nodes with small degree but they also have hubs (nodes with very high degree compared to the average degree)



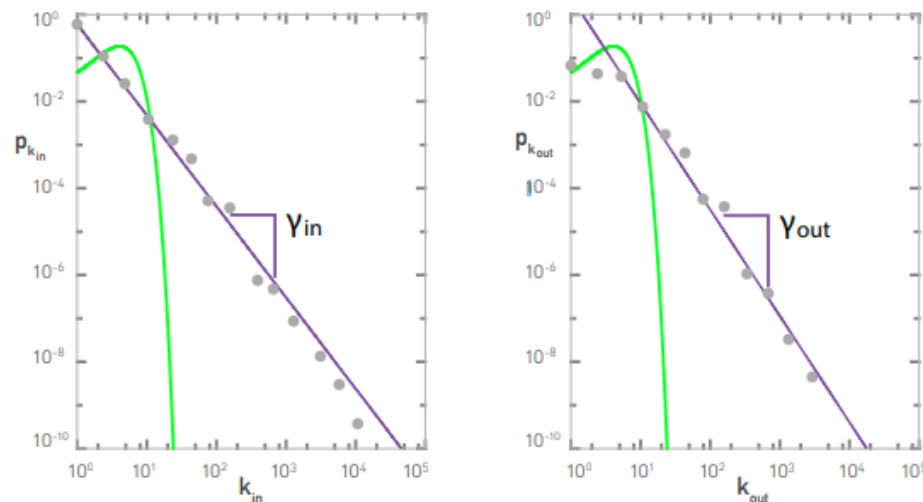
Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 6. Scale Free Networks

World wide web

- The world wide web is a directed network
- Every node has a degree k_{in} (the links towards the node) and k_{out} (the links outwards the node)
- Its degree distribution differs from the distributions of the ER and Small World networks, it has a power law form and is representative of many networks



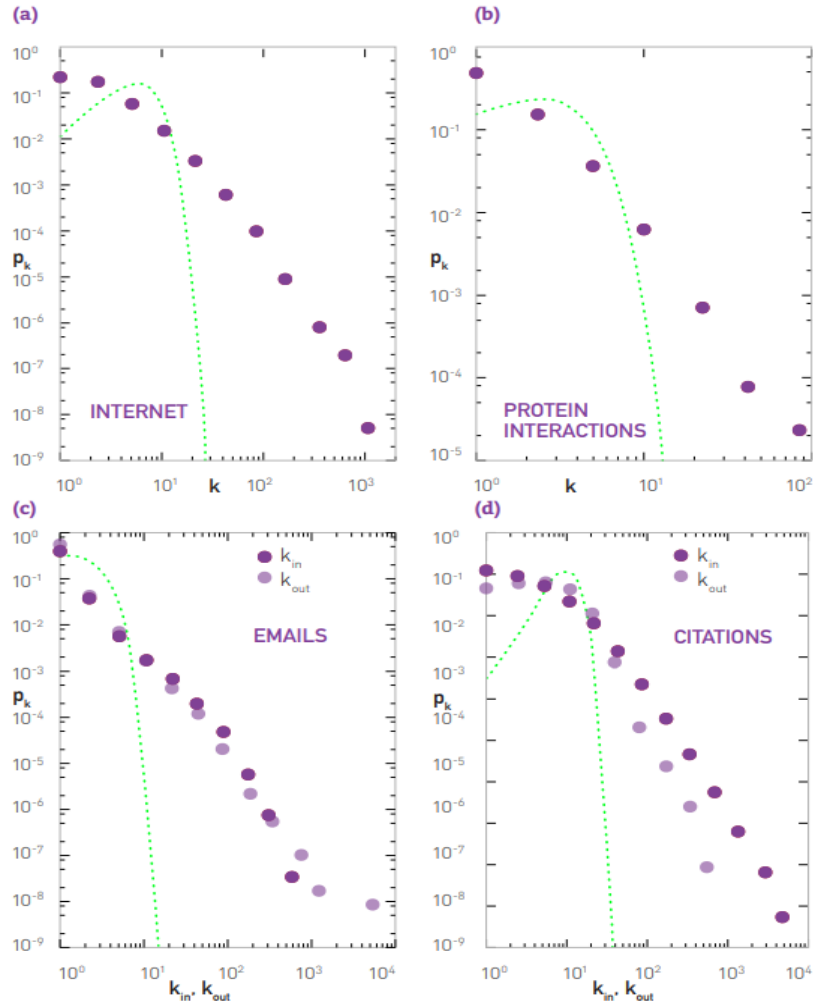
Source: Barabási, Network Science

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Real Networks



Many Real networks have
power law degree
distribution

Source: Barabási, Network Science

ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness

Discrete formalization

- A node has exactly k degree where $k = 0, 1, 2, 3, \dots$. The degree distribution is equal to

$$p_k = Ck^{-\gamma}$$

- p_k is probability density function:

$$\sum_{k=1}^{\infty} p_k = 1$$

Thus

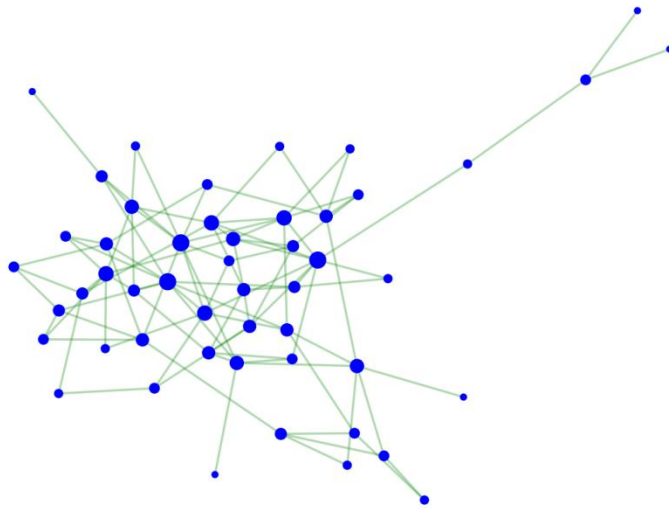
$$C = \frac{1}{\sum_{k=1}^{\infty} p_k}$$

And

$$p_k = \frac{1}{\sum_{k=1}^{\infty} p_k} k^{-\gamma}$$

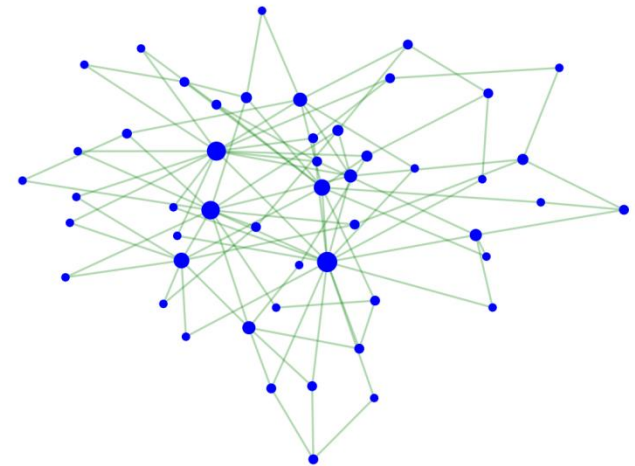
ERASMUS+

Random vs Scale Free



Random Network

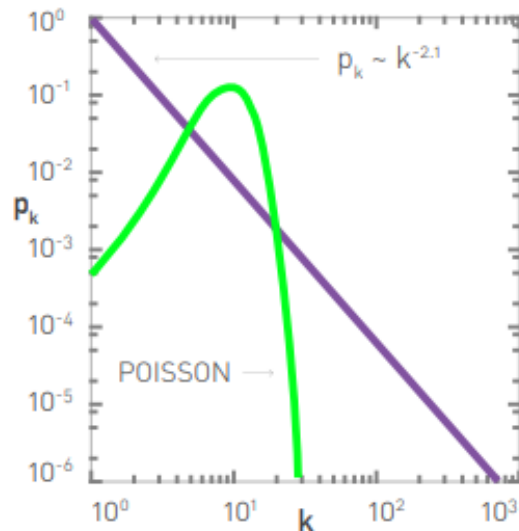
- No large difference in k between nodes



Scale Free Network

- Few nodes with many connections (hubs)
- Many nodes with small degree

Poisson vs Power Law



- For both distributions $\langle k \rangle$ is equal to 11
- For the power law distribution (Scale Free Networks) most nodes have small k and few nodes have high k
- For the Poisson distribution (Random Networks) most nodes have similar k .

P_k as a function of k for Poisson and Power Law distribution. $\langle k \rangle = 11$ for both. (source Barabási: Network Science)

Barabási-Albert Model



Albert-László Barabási



Réka Albert

- Preferential Attachment, has been introduced in many science domains to explain various distributions since 1923.
- In 1999 in network scientist László Barabási & Réka Albert have introduced a network model
- This model is based on 2 properties, Preferential Attachment and Growth
- It creates Scale Free Networks, and is one of the most well known network models.

Growth & Preferential Attachment

Initially a network of m_0 nodes is created

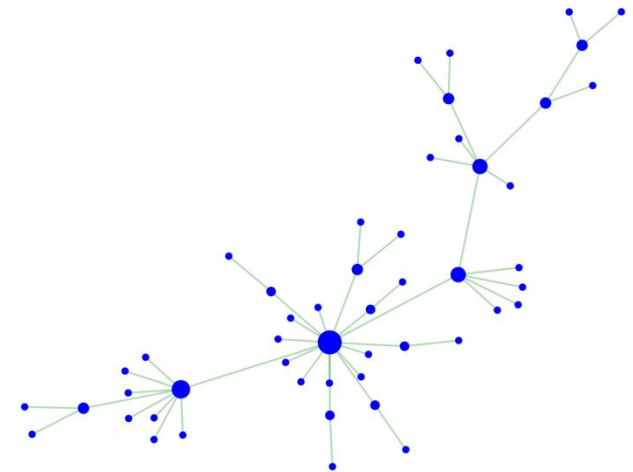
Growth

- New nodes are inserted into the network. Every new node creates m connections.

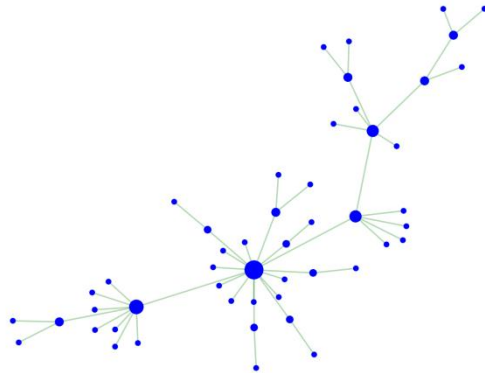
Preferential Attachment

- Every new node has a larger probability of connecting with a node with high degree. The probability of node j connecting with a node i is equal to

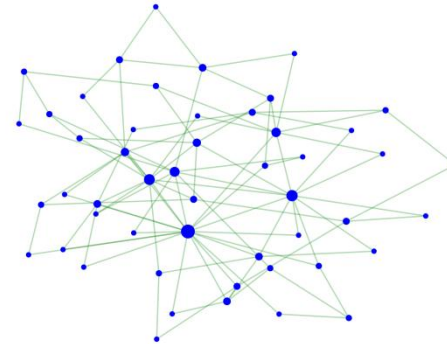
$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$



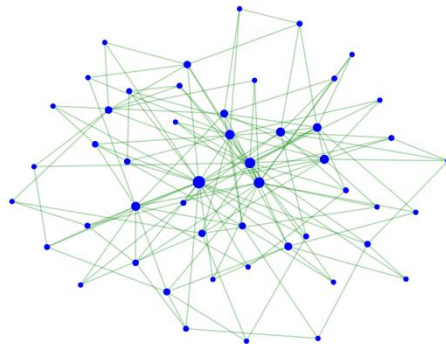
BA Networks with N=50



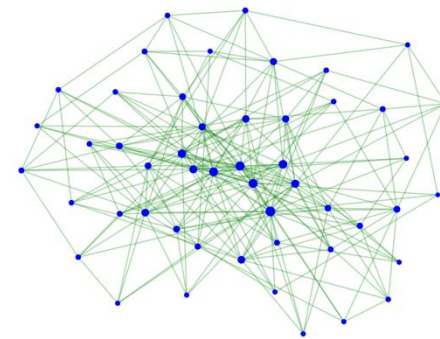
m=1



m=2

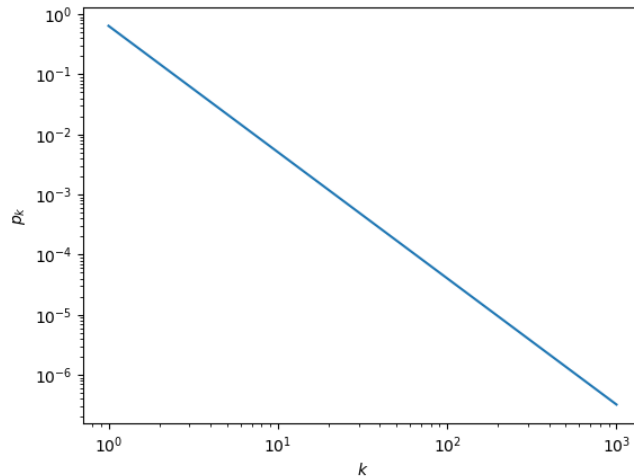


m=3



m=10

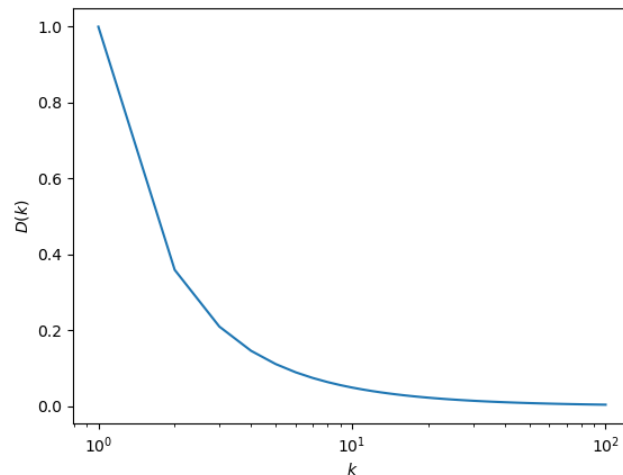
Generate k sequence



- Calculate p_k as a function of k

- Compute $D(k)$, where

$$D(k) \sim \sum_{k' \geq k} p_{k'}$$



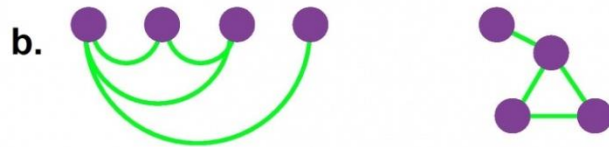
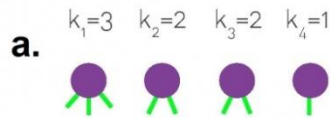
- Create N random numbers between 0 and 1

- Use the plot of $D(k)$ to assign each random number to a k .

$$k = D^{-1}(r)$$

where r is the random number

Configuration Model



- Knowing the degree distribution of the network, we assign the k of each node

- From the degree sequence edges are created randomly with probability

$$p_{ij} = \frac{k_i k_j}{2L - 1}$$

- Multilinks and self loops are allowed.

Configuration Model. Source:
Barabási, Network Science

Scale Free Property

- n-th moment of the degree distribution

$$\langle k^n \rangle = \sum_{k_{\min}}^{\infty} k^n p_k = \int_{k_{\min}}^{\infty} k^n p(k) dk$$

For scale free networks

$$\langle k^n \rangle = C \frac{k_{\max}^{n-\gamma+1} - k_{\min}^{n-\gamma+1}}{n - \gamma + 1}$$

- n=1, $\langle k \rangle$ is the average degree
- n=2, with $\langle k^2 \rangle$ we calculate the variance $\sigma^2 = \langle k^2 \rangle - \langle k \rangle^2$
- In the limit
 - If $n-1+\gamma \leq 0$
 - All moments are finite

Scale Free Property

- If $n-1+\gamma > 0$
 - Moments larger than $\gamma - 1$ diverge
- Scale free networks' degree is around 2-3
 - First moment is finite
 - Second moment diverges

Conclusion

- Scale free networks lack a scale.
- If we pick randomly a node, we don't know what to expect. It could have an extremely large degree or an extremely low degree.

System behavior as a function of γ

- An interesting question is what would be the behavior of the system as a function of γ

- $\gamma \leq 2$ (Anomalous Regime)

- First moment diverges
- 2nd moment diverges
- k_{\max} increases faster than the network
(Odd predictions)

- $2 < \gamma < 3$

- First moment finite
- Second moment diverges

- $k_{\max} \sim N^{\frac{1}{\gamma-1}}$

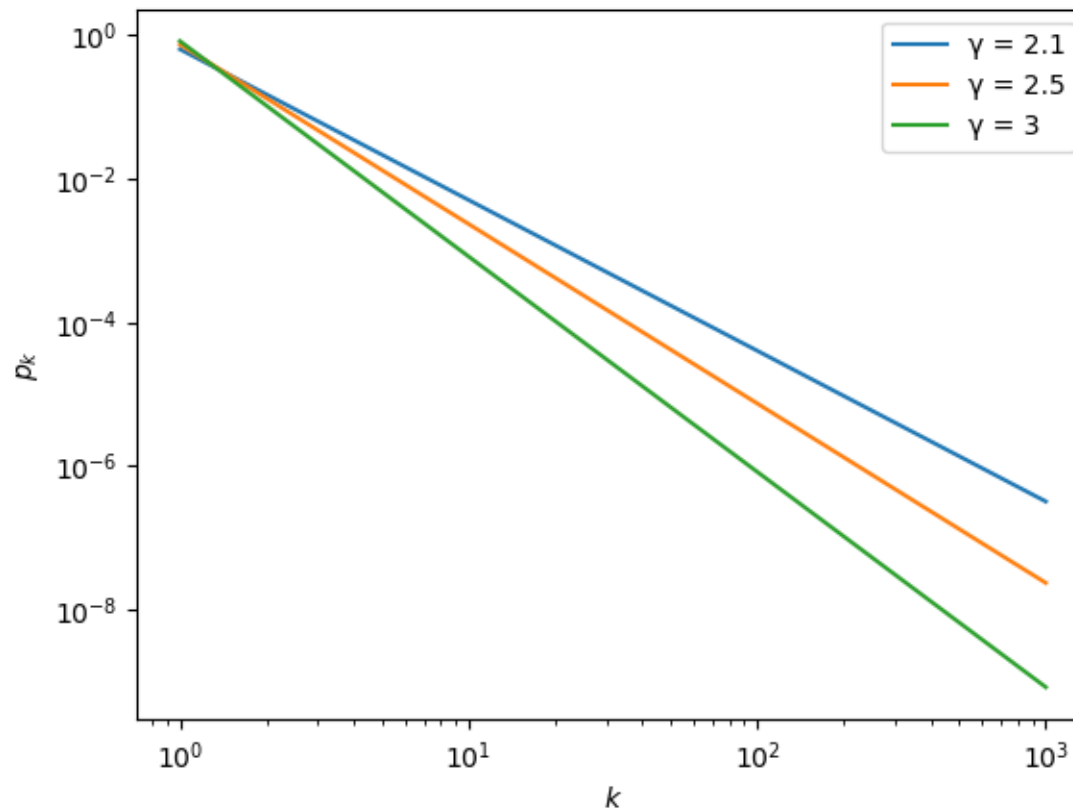
Ultra small world property

- $\gamma > 3$ (random regime)
 - 1st moment finite
 - 2nd moment finite

Same behaviour as a random network



Variation of γ



ERASMUS+

Key Action KA2 - Cooperation for innovation and the exchange of good practices

Action Type KA226 - Partnerships for Digital Education Readiness



Exercise

- Create a Barabási-Albert network with $N = 1000$ nodes and $m=2,4,10$.
- In the same figure plot the three degree distributions, one for each value of m .
- Find the connectivity of the biggest hub of the network. What are your conclusions?
- Calculate the 1st and the 2nd moment using the relations mentioned in powerpoint
- The results should be average of 1000 simulations.

Summary

- In this lesson we have introduced scale free networks
- We examined:
- Barabasi albert model.
- A model which is based upon the following ideas
 - Growth
 - Preferential attachment
- Configuration Model
 - A random network, given a degree distribution (could be scale free)
- Scale free networks lack a scale



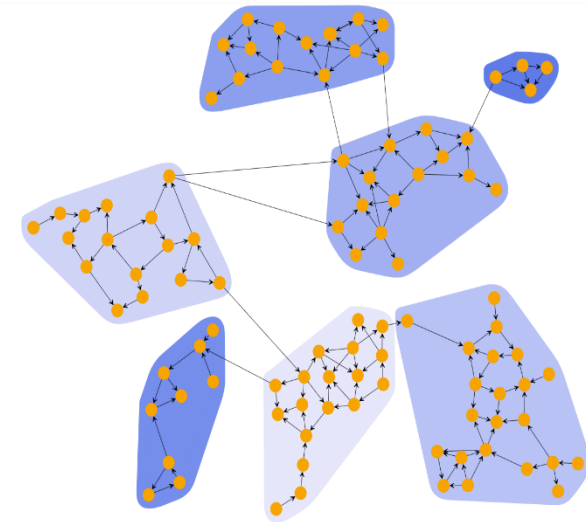
Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 7. Clustering

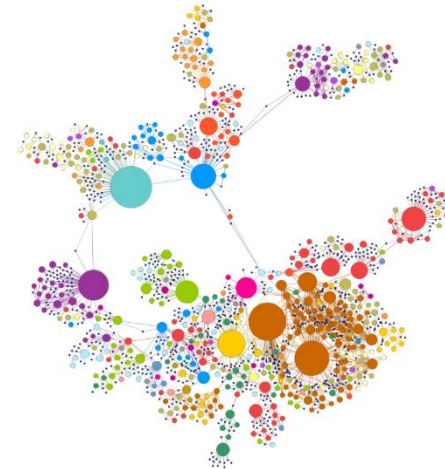
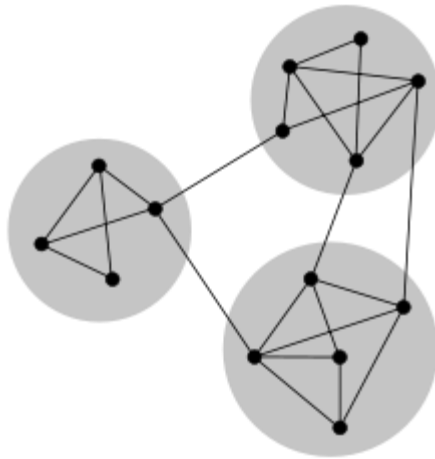
Clustering

- Clustering is the process of grouping together similar objects based on certain criteria.
- Clustering is a fundamental technique in network analysis and has applications in various fields including biology, social science, and computer science.
- In this presentation, we will discuss different types of clustering algorithms and their applications in network analysis.



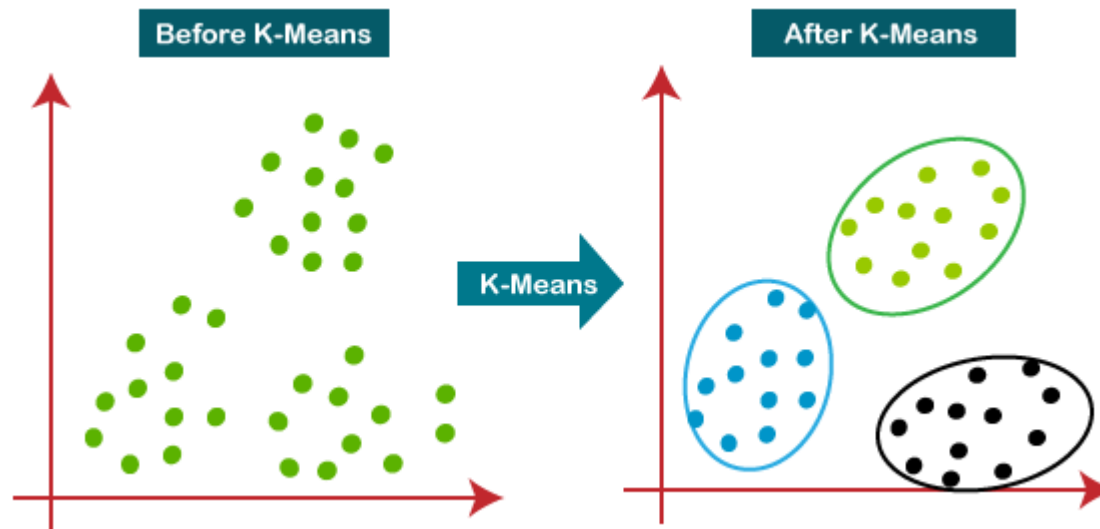
Goal of clustering

- Clustering can be used to understand the structure and function of a network, and can reveal important information about the relationships between nodes
- To identify communities or modules within the network, where nodes within a community are more densely connected to each other than to nodes in other communities



Types of Clustering

- A. Cluster together nodes with similar properties (such as degree)
 - Hierarchical Clustering:
 - A clustering method that builds a tree-like structure by iteratively merging smaller clusters into larger ones.
 - K-Means Clustering:
 - A partition-based clustering method that partitions nodes into a pre-defined number of clusters based on their distance to the cluster centroids.



Communities

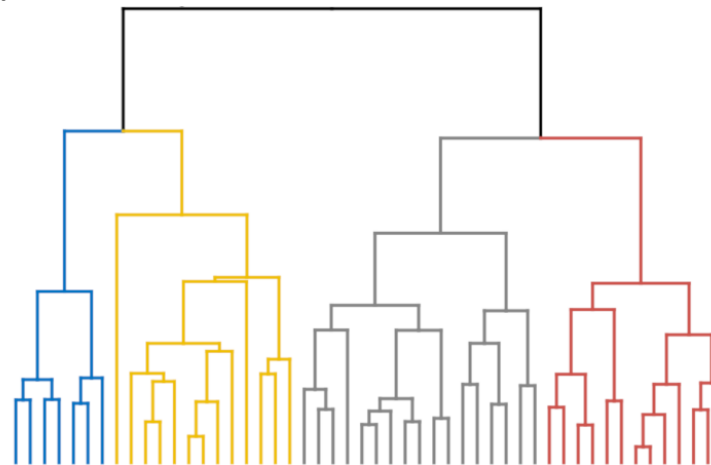
- B. To identify communities or modules within the network, where nodes within a community are more densely connected to each other than to nodes in other communities
 - Modularity-Based Clustering:
 - A community detection algorithm that aims to maximize the modularity score of a network by partitioning nodes into non-overlapping communities.
- There are several types of clustering algorithms, each with its own strengths and weaknesses.

Hierarchical Clustering

- Hierarchical clustering is a method of clustering that creates a tree-like hierarchy of clusters.
- Two main types of hierarchical clustering:
- Agglomerative clustering
 - starts with each data point as its own cluster and merges them iteratively.
- Divisive clustering
 - starts with all data points as one cluster and splits them recursively.

Example

- Consider a network of social interactions between people.
- The hierarchical clustering algorithm would group people together based on their degree of interaction, with the most tightly connected groups forming clusters at higher levels of the hierarchy.
- The resulting hierarchy could be visualized as a dendrogram, with the leaves representing individual nodes and the branches representing the merging of clusters.



ERASMUS+

Algorithm

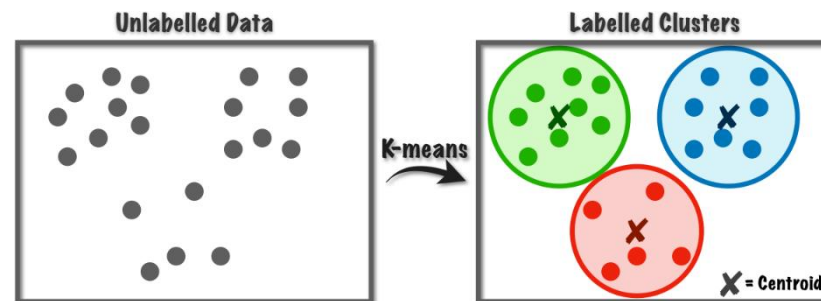
- Step 1: Start by assigning each node to its own cluster.
- Step 2: Calculate the distance between each pair of clusters using a distance metric such as Euclidean distance or correlation distance.
- Step 3: Merge the two closest clusters into a single cluster.
- Step 4: Recalculate the distances between this new cluster and all other clusters.
- Step 5: Repeat steps 3 and 4 until all nodes are in a single cluster or until a stopping criterion is met.
- Step 6: Create a dendrogram to visualize the hierarchy of clusters.

K-means clustering

- An unsupervised machine learning technique used for clustering analysis
- Divides the data into K clusters based on their similarity
- Each cluster has its own centroid, which represents the center of the cluster

Algorithm

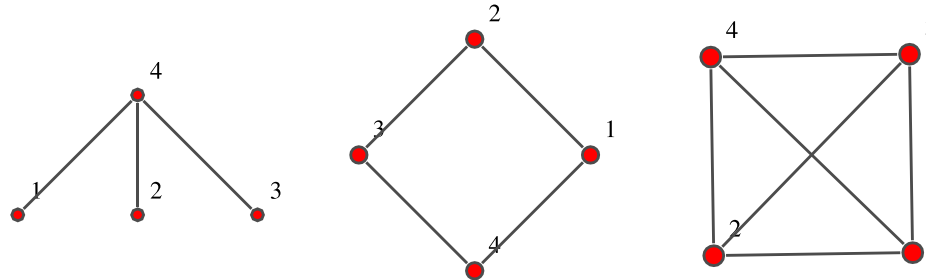
- Choose the number of clusters (k) to group the data into.
- Randomly select k data points to serve as the initial centroids for each cluster.
- Assign each data point to the nearest centroid.



Algorithm

- Recalculate the centroids as the mean of all data points assigned to it.
- Repeat steps 3 and 4 until the centroids no longer move or a maximum number of iterations is reached.
- The resulting clusters are based on the final centroid positions.

Appendix: Adjacency matrix



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

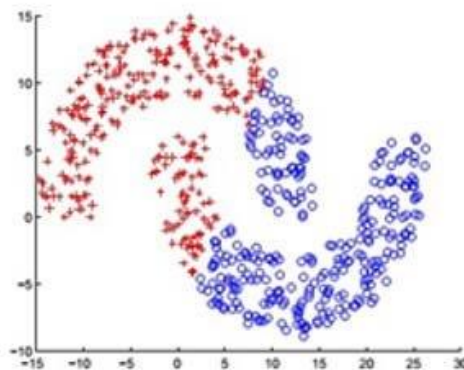
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

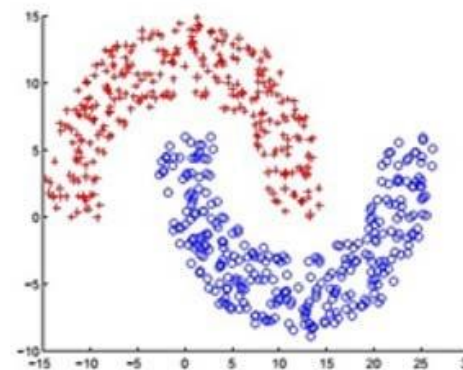
- Matrix, which represents the edges of the network
- $A_{ij} = 1$ if there is a link between the nodes i, j
- $A_{ij} = 0$ if i, j nodes aren't linked

Spectral Clustering

- Spectral clustering is a graph partitioning technique that involves using the eigenvalues and eigenvectors of a graph's Laplacian matrix to partition the graph into clusters.
- The Laplacian matrix of a graph is defined as the difference between the degree matrix and the adjacency matrix.
- Spectral clustering involves embedding the graph into a high-dimensional space and then clustering the points in this space.



(a) K-means



(b) Spectral Clustering

Steps

- Construct the adjacency matrix or the Laplacian matrix of the graph/network.
- Calculate the eigenvalues and eigenvectors of the matrix.
- Select a subset of the eigenvectors that correspond to the k largest eigenvalues.
- Form a matrix with the selected eigenvectors as columns.
- Apply clustering algorithm (e.g., k-means) to the rows of the new matrix.
- Assign each node to the cluster that it belongs to based on the clustering result.

Exercise

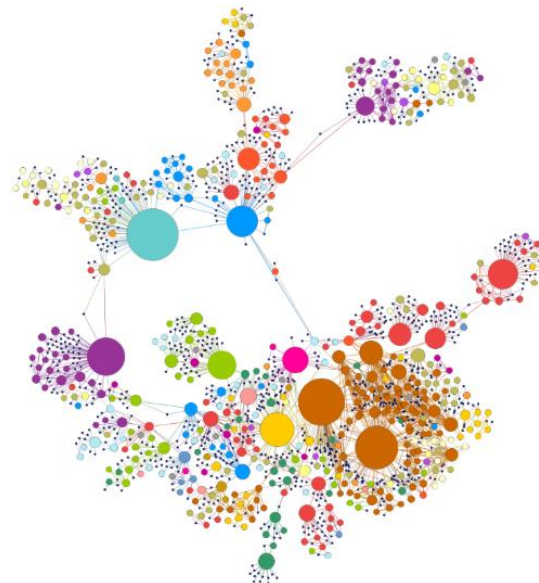
1. Obtain a dataset containing biological data that is suitable for clustering analysis, such as gene expression data, protein interaction data, or genomic sequence data.
(<https://biodbnet-abcc.ncifcrf.gov/>)
2. Apply hierarchical clustering algorithm to the dataset and visualize the resulting dendrogram.
3. Implement the k-means clustering algorithm on the same dataset
4. Evaluate and compare the clustering results obtained from hierarchical clustering and k-means clustering. Analyze the similarities and differences between the two approaches and discuss their strengths and limitations.

Summary

- Definition of clustering
- Different types of clustering and their algorithm
 - k-means
 - A partition-based clustering method that partitions nodes into a pre-defined number of clusters based on their distance to the cluster centroids.
 - Hierarchical
 - A clustering method that builds a tree-like structure by iteratively merging smaller clusters into larger ones.
 - Spectral Clustering
 - A graph-based clustering method that uses the eigenvalues and eigenvectors of the network's Laplacian matrix to partition nodes.

Communities

- A community is defined as a group of nodes that are densely connected within the group and sparsely connected to nodes outside the group.
- This approach has been shown to be effective in identifying functional modules in biological networks, such as protein-protein interaction networks and gene co-expression networks.



ERASMUS+

Modularity

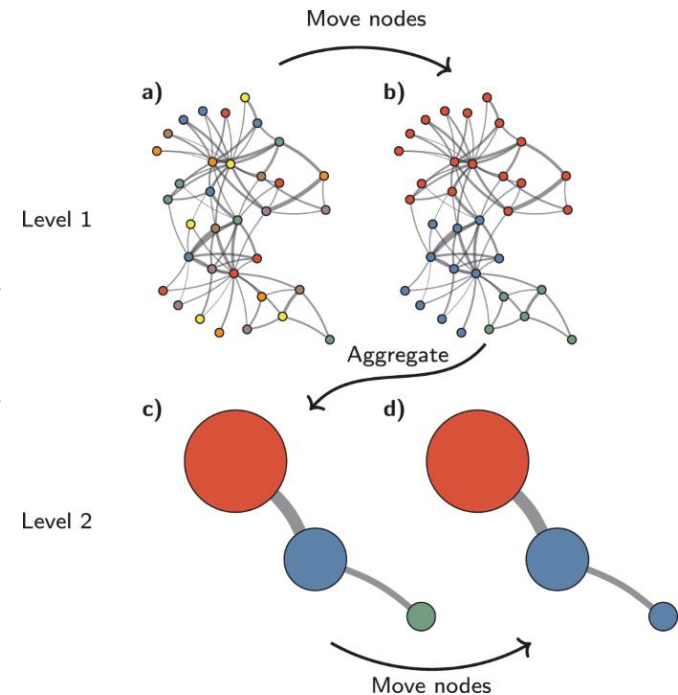
- Modularity is a measure that quantifies the degree to which a network can be divided into non-overlapping communities or modules. It is defined mathematically as:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \cdot \delta_{c_i c_j}$$

- m : total number of edges
- Weight of the edge with nodes i and j (0 or 1 in undirected networks)
- k_i is the degree centrality of node i
- δ is the Kronecker delta function (1 if the communities c_i and c_j are equal, 0 otherwise)

Louvain Algorithm

- Below we present the steps of one
- Initialization:
 - Assign each node to its own community
 - Calculate the modularity score Q of the initial partition
- Iteration:
 - For each node i , calculate the gain in modularity by moving it to each of its neighbors' communities
 - Choose the community that maximizes the modularity gain for node i and move it there
 - Repeat this process for all nodes until no more modularity gains can be achieved





- Aggregation:
 - Contract each community into a single node, preserving the total weight of edges between nodes in different communities
 - Recalculate the modularity score Q for the new partition of communities
- Repeat steps 2-3 until no more modularity gains can be achieved or a stopping criterion is met



Comparison between Algorithms

- Hierarchical clustering is flexible and can handle different types of data, but can be computationally intensive and sensitive to noise.
- K-means clustering is fast and efficient, but requires prior knowledge of the number of clusters and can be sensitive to initial conditions.
- Spectral clustering can handle non-linear relationships and is not limited by the shape of clusters, but requires careful selection of the number of clusters and can be sensitive to the choice of similarity measure.

- Modularity-based clustering is well-suited for identifying communities in networks, but it can be computationally intensive for large networks.
- Other clustering algorithms, such as density-based clustering and fuzzy clustering, also have their own strengths and weaknesses.
- Choosing the appropriate clustering algorithm depends on the research question, the type and size of data, and the computational resources available.



Clustering on Bioinformatics

- Clustering algorithms are widely used in bioinformatics to analyze large biological networks, including:
 - Protein-protein interaction networks
 - Gene co-expression networks
 - Metabolic networks
 - Regulatory networks
- Clustering can help identify functionally related groups of genes or proteins, which can provide insight into biological processes and disease mechanisms.

Prognosis

- Clustering analysis allows us to stratify patients based on their molecular profiles or disease characteristics.
- By considering similar patients within clusters, we can predict
 - disease progression
 - treatment response
 - patient outcomes.



Cancer Research

- In the context of cancer research, we can apply clustering algorithms to protein-protein interaction networks to identify potential therapeutic targets.
- By focusing on specific clusters or modules within the network, we can uncover proteins that play critical roles in cancer progression and may serve as effective targets for drug development.





Louvain on metabolic networks

- By applying the Louvain algorithm to a metabolic network, we can identify communities or modules of metabolites involved in a specific metabolic pathway.
- This enables us to identify key metabolites that play crucial roles in the pathway and may serve as potential targets for drug discovery and therapeutic interventions.

Key metabolites

- By analyzing the topology of the metabolic network, we can identify metabolites that have high degrees (number of connections) or high betweenness centrality (acting as crucial intermediates between multiple pathways).
- These key metabolites are important because they are involved in critical biochemical reactions and have the potential to influence the overall functioning of the metabolic pathway.
- Targeting these metabolites can have a significant impact on the pathway's activity, making them attractive candidates for drug discovery.



Exercise

1. For the same dataset of the exercise in lesson 9
2. Apply Louvain algorithm
3. Evaluate and compare the clustering results obtained from the clustering algorithms. Analyze the similarities and differences between the algorithms
4. Create a histogram of the number of communities created

Summary

- In this lesson we focused on clustering algorithms that partition the communities of the graph
- Louvain algorithm
 - Partitioning the network to communities, maximizing the modularity
- We compared the clustering algorithms
- Examples of clustering applications on bioinformatics
 - Prognosis
 - Cancer research
 - Finding the key components of the networks



Module 3. Bioinformatic tools

Topic 5. Algorithms in Bioinformatics

Lesson 8. Network Alignment



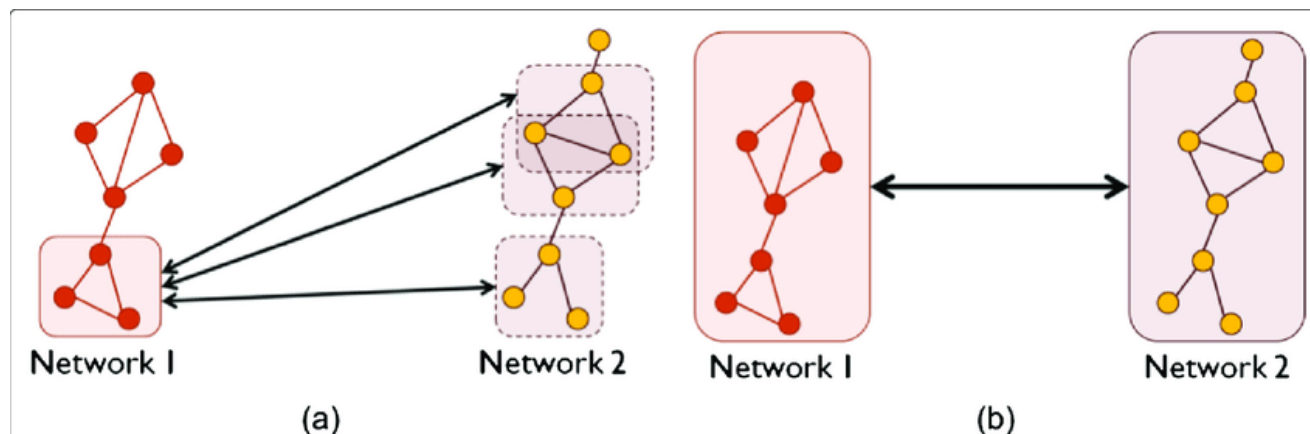
Algorithms in Bioinformatics

Lesson 11

Network alignment

Alignment

- Network alignment is the process of identifying similar structures in two or more networks.
- It involves finding a mapping between nodes in the networks that preserves structural similarity.
- In bioinformatics, network alignment is used to compare biological networks, to identify
 - functional similarities
 - and differences.



ERASMUS+

Aim of Alignment

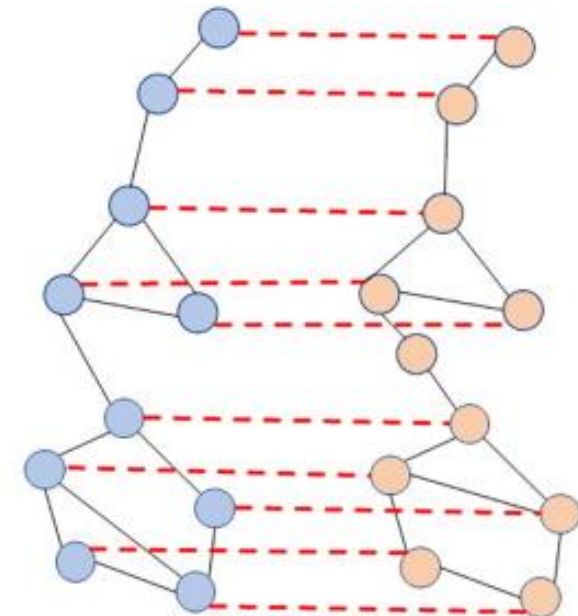
- Finding a correspondence between the nodes and edges of two or more networks.
- The aim of network alignment is to identify similar or homologous nodes and edges across different networks.
- It allows researchers to leverage the available functional information in one network to infer or predict the functions of nodes in the other network

Problem definition

- The goal of the biological network alignment problem is to cluster nodes across different networks based on
 - their biological (sequence) similarity
 - and the interaction patterns of their neighboring communities (i.e. topology similarity).
- Find a one-to-one (or many-to-many) correspondence

Global Alignment

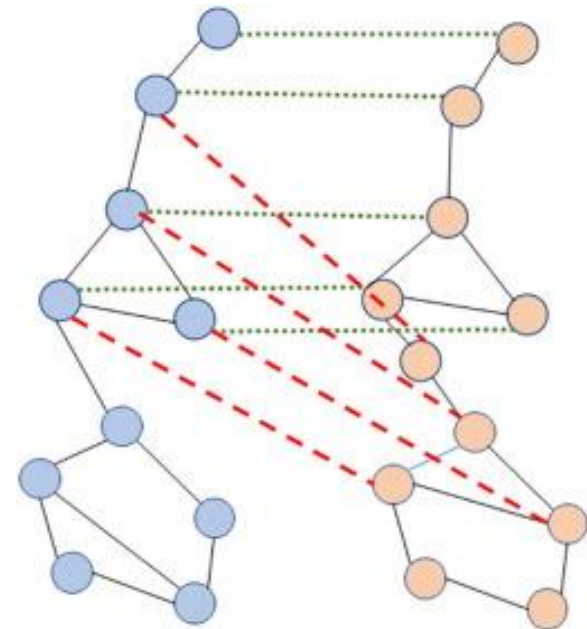
- Global network alignment aims to find a global optimal mapping of nodes between two networks.
- In global network alignment, entire networks are compared and aligned, instead of just local subgraphs.
- Global network alignment can be formulated as an optimization problem, where the goal is to find a node mapping that maximizes a predefined objective function.



(b) Global network alignment

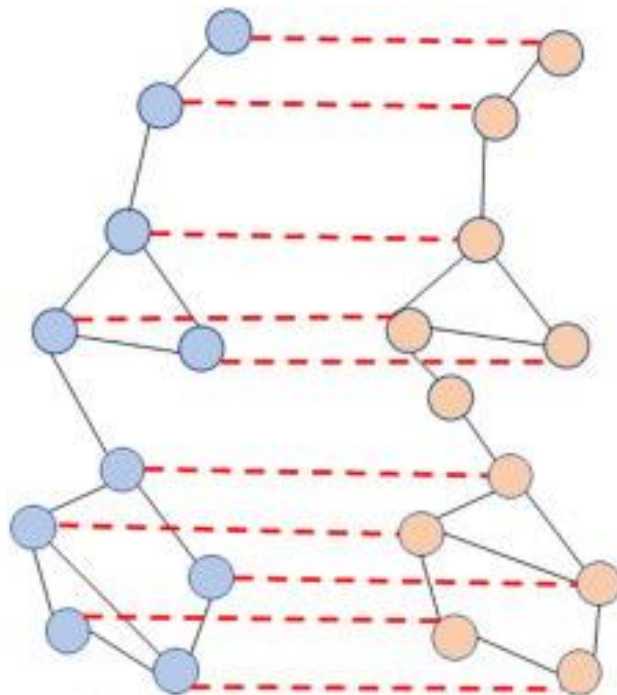
Local Alignment

- Identifying regions of high similarity between two networks rather than aligning the entire networks
- Particularly useful for comparing networks with different structures or sizes
- The goal is to identify conserved substructures between the two networks, such as motifs or functional modules
- Set of aligned regions or subnetworks, rather than a one-to-one mapping between nodes

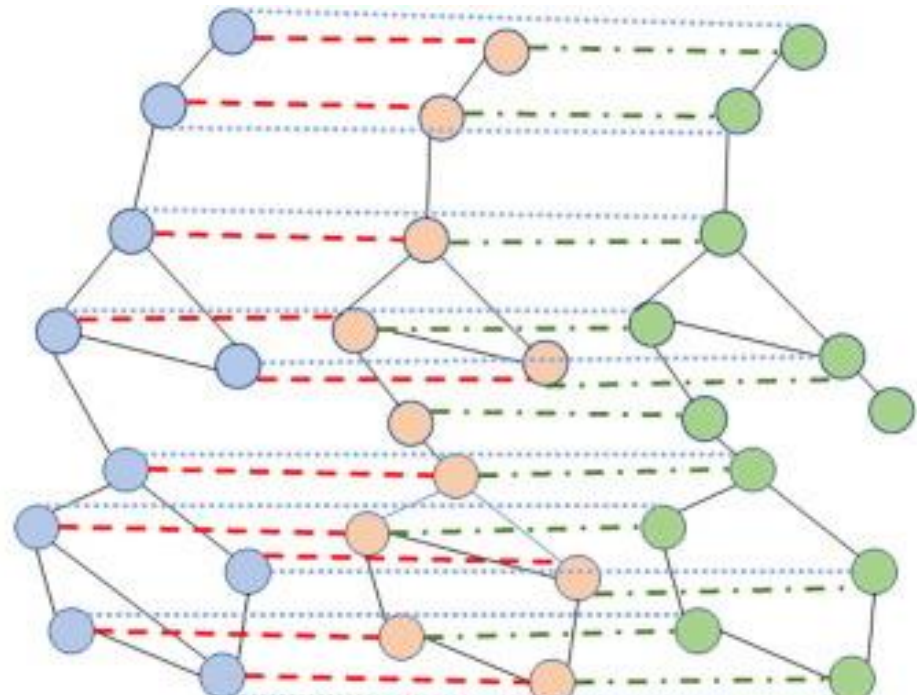


(a) Local network alignment

Pairwise vs Multiple



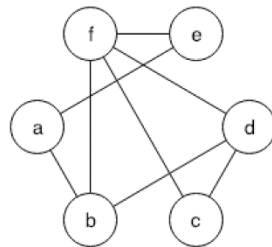
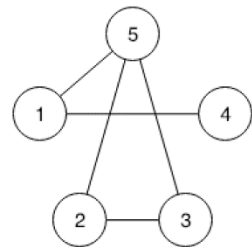
(a) Pairwise network alignment



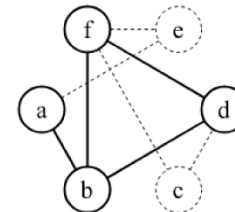
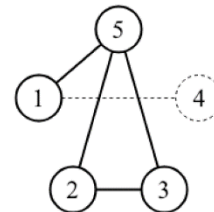
(b) Multiple network alignment

Maximum common subgraph

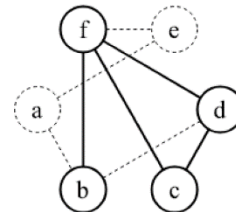
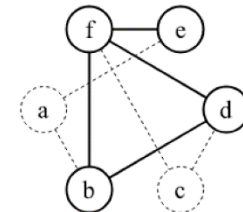
- The maximum common subgraph of two graphs is the largest possible common subgraph, i.e., the common subgraph with as many vertices as possible.



(a)



(b)



Impact on Bioinformatics

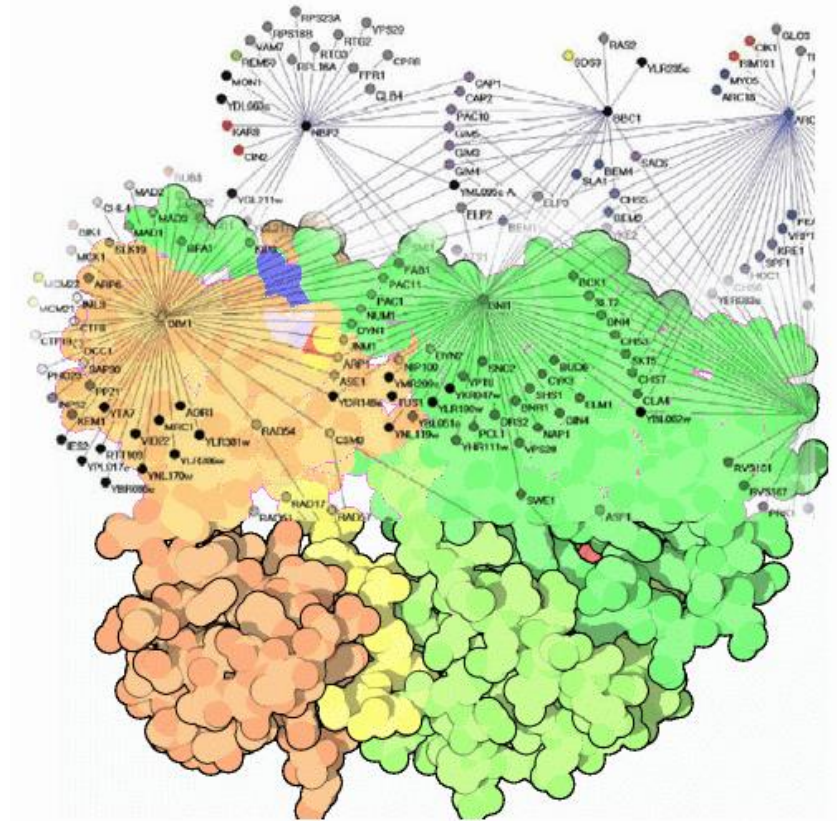
- Network alignment is a powerful tool for analyzing biological networks and comparing their structures.
- Providing insights into evolutionary relationships and mechanisms of molecular function.
- By aligning disease-specific networks, network alignment can help to identify new therapeutic targets and predict disease outcomes.
- Network alignment can also be used to integrate multi-omics data and build more comprehensive network models for understanding complex biological processes.
- Thus, network alignment is an essential technique in bioinformatics for deciphering biological networks and their functions.

Similarity measures

- Similarity measures are crucial for evaluating the quality of network alignments
- Node Based
 - Nodes with similar attributes (e.g., degree, centrality, or functional annotation) are aligned together
- Edged Based
 - Edges that connect similar nodes are aligned together
 - Edge-based measures include topological overlap, edge conservation, and graphlet degree distribution
- Network Based
 - Overall network properties are used to measure similarity between networks
 - Network-based measures include the spectral similarity, graphlet correlation, and network divergence

IsoRank

- Global network alignment algorithm proposed by Singh et al.
- The general idea in the construction of matrix R is that two nodes i and j are a good match if their neighbors also match well

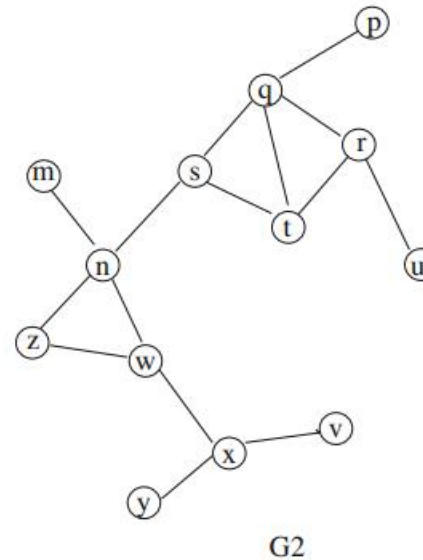
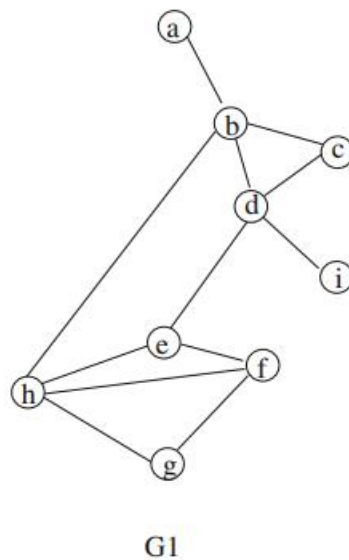


IsoRank

- First, it computes a similarity score for each pair of nodes in the two networks, and uses this to identify the best matching nodes.
- In the second step, highly matching nodes from R are extracted to find the alignment
- Can handle networks with different sizes and topologies

Exercise

- Find the Maximum common subgraph of those graphs



Summary

- Network alignment is a powerful tool for analyzing biological networks and comparing their structures.
- Process of identifying similar structures in two or more networks.
- It involves finding a mapping between nodes in the networks that preserves structural similarity.
- Types of alignment
 - Local vs Global vs Partial
 - Pairwise vs Multiple
- Example algorithm for global alignment
 - IsoRank