BSc Biotechnology II year
(BBT-201)

Unit I
Fundamentals of Computers & Bioinformatics

Introduction to Computers

A **computer** is an electronic device that processes data to perform various tasks. It operates by following a set of instructions, known as programs, to execute calculations, manage data, and perform complex functions efficiently. Computers have transformed multiple industries, including education, healthcare, business, and entertainment.

Characteristics of Computers

Computers possess several key characteristics that make them powerful tools for various applications:

1. **Speed** – Computers can process millions of instructions per second, making them much faster than human calculations.
2. **Accuracy** – Unlike humans, computers perform tasks with a high degree of accuracy and minimal errors.
3. **Automation** – Once programmed, a computer can execute tasks automatically without requiring manual intervention.
4. **Versatility** – Computers can handle a wide range of applications, from word processing to complex simulations.
5. **Storage** – They have vast storage capacities to store data permanently or temporarily.
6. **Connectivity** – Computers can connect to networks (such as the internet), allowing communication and data sharing globally.
7. **Multitasking** – A computer can run multiple applications simultaneously without significant performance loss.

Capabilities of Computers

Computers are used in diverse fields due to their numerous capabilities:

1. **Data Processing** – Computers process large volumes of data quickly and efficiently.
2. **Communication** – They enable communication through emails, video conferencing, and social media.
3. **Automation of Tasks** – Businesses use computers for automating repetitive tasks like payroll processing.
4. **Entertainment** – They support gaming, music, movies, and digital art creation.
5. **Scientific Research** – Computers assist in simulations, modeling, and complex calculations in science and engineering.
6. **Artificial Intelligence** – Modern computers can learn, analyze, and make decisions using AI technologies.

Generations of Computers

Computers have evolved through five generations, each marked by technological advancements:

1. **First Generation (1940-1956) – Vacuum Tubes**
   - Used vacuum tubes for circuitry and magnetic drums for memory.
   - Large, slow, and generated a lot of heat.
   - Example: ENIAC, UNIVAC.
2. **Second Generation (1956-1963) – Transistors**
   - Used transistors instead of vacuum tubes, making them smaller and more efficient.
   - Faster processing speeds and reduced heat generation.
   - Example: IBM 1401, CDC 1604.
3. **Third Generation (1964-1971) – Integrated Circuits (ICs)**
   - Used ICs instead of transistors, increasing processing power and reducing size.
   - Allowed for the development of operating systems and programming languages.
   - Example: IBM System/360, PDP-8.
4. **Fourth Generation (1971-Present) – Microprocessors**
   - Introduction of microprocessors (single-chip processors).
   - Marked the rise of personal computers (PCs) and the internet.
   - Example: Intel 4004, Apple Macintosh, IBM PC.

5. **Fifth Generation (Present & Beyond) – Artificial Intelligence (AI)**
    - Focuses on AI, quantum computing, and advanced machine learning.
    - Uses high-speed processors, cloud computing, and neural networks.
    - Example: IBM Watson, AI-powered devices like smartphones and robots.

Software, Hardware, Memory, Control Unit, ALU

Computers function through a combination of **hardware** and **software**, working together to process and store data. Key components like the **Control Unit (CU)** and **Arithmetic Logic Unit (ALU)** play crucial roles in executing instructions.

1. Hardware

Hardware refers to the physical components of a computer that you can touch and see. These components work together to execute tasks efficiently.

Types of Hardware:

- **Input Devices** – Used to enter data into a computer. *(e.g., Keyboard, Mouse, Scanner, Microphone)*
- **Output Devices** – Display or present processed data. *(e.g., Monitor, Printer, Speaker)*
- **Storage Devices** – Store data permanently or temporarily. *(e.g., Hard Drive, SSD, USB, DVD)*
- **Processing Unit** – The **Central Processing Unit (CPU)** is the "brain" of the computer that processes instructions.
- **Motherboard** – Connects all hardware components and allows communication between them.
- **Power Supply Unit (PSU)** – Provides electrical power to all computer components.

2. Software

Software refers to a set of instructions that tell the computer how to perform tasks. Unlike hardware, software is intangible and cannot be physically touched.

Types of Software:

- **System Software** – Controls hardware and manages basic functions. *(e.g., Operating Systems like Windows, macOS, Linux)*
- **Application Software** – Performs specific tasks for users. *(e.g., MS Office, Web Browsers, Games, Media Players)*

- **Utility Software** – Helps maintain and optimize the system. *(e.g., Antivirus, Disk Cleanup, File Compression)*

3. Memory

Memory is essential for storing data and instructions while the computer is running. It determines the speed and efficiency of computing tasks.

Types of Memory:

- **Primary Memory (Volatile)** – Temporary storage that is lost when power is off.
    - **Random Access Memory (RAM)** – Stores currently active programs and data.
    - **Cache Memory** – High-speed memory that speeds up CPU operations.
- **Secondary Memory (Non-Volatile)** – Permanent storage for data and applications.
    - **Hard Disk Drive (HDD)**, **Solid State Drive (SSD)**, **USB Drives**.
- **Read-Only Memory (ROM)** – Stores firmware and system boot instructions. It cannot be modified easily.
- **Virtual Memory** – Uses hard disk space to act as additional RAM when needed.

4. Control Unit (CU)

The **Control Unit (CU)** is a crucial part of the **CPU**. It directs and coordinates all activities in the computer system.

Functions of the Control Unit:

- Fetches instructions from memory.
- Decodes instructions to understand what needs to be done.
- Directs the flow of data between components.
- Controls input and output operations.

5. Arithmetic Logic Unit (ALU)

The **Arithmetic Logic Unit (ALU)** is another important part of the **CPU** that performs all mathematical and logical operations.

Functions of the ALU:

- **Arithmetic Operations** – Performs basic calculations such as addition, subtraction, multiplication, and division.
- **Logical Operations** – Compares values and determines conditions like greater than, less than, or equal to.
- **Bitwise Operations** – Manipulates data at the binary level (e.g., AND, OR, NOT, XOR operations).

Input & Output Devices

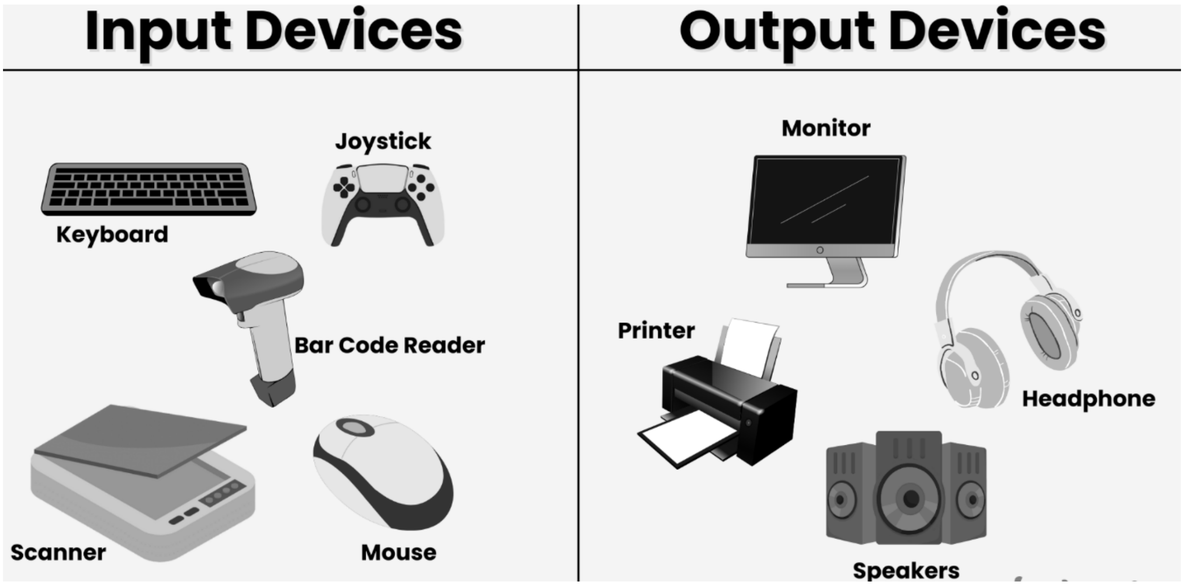
- Input and output devices are essential components of a computer system. They allow users to interact with the computer by providing input (data entry) and receiving output (results of processing).

1. Input Devices

Input devices are used to enter data and instructions into a computer. They convert user actions or data into a format that the computer can process.

Types of Input Devices:

1. **Keyboard** – The most common input device used to enter text, numbers, and commands. *(e.g., QWERTY keyboard, mechanical keyboard)*
2. **Mouse** – A pointing device used to interact with graphical elements on the screen. *(e.g., Optical mouse, Wireless mouse)*
3. **Touchscreen** – Allows users to interact with a device by touching the screen. *(e.g., Smartphones, Tablets)*
4. **Scanner** – Converts physical documents or images into digital form. *(e.g., Flatbed scanner, Barcode scanner)*
5. **Microphone** – Captures sound and converts it into a digital signal for voice recognition, communication, or recording.
6. **Joystick/Game Controller** – Used for gaming to control movement and actions in a virtual environment.
7. **Webcam** – Captures real-time images and videos for video conferencing and streaming.
8. **Biometric Devices** – Capture unique biological features for security authentication. *(e.g., Fingerprint scanner, Facial recognition sensor)*



2. Output Devices

Output devices display or present the processed data in a human-readable form. They convert digital data into visual, audio, or physical output.

Types of Output Devices:

1. **Monitor (Screen/Display)** – Displays visual output from the computer. *(e.g., LED, LCD, OLED monitors)*
2. **Printer** – Produces a physical copy of digital documents. *(e.g., Inkjet printer, Laser printer, 3D printer)*
3. **Speaker** – Outputs sound, music, or voice from the computer.
4. **Headphones/Earphones** – Personal audio output devices for private listening.
5. **Projector** – Displays images and videos on a large screen for presentations or entertainment.
6. **Plotter** – Used for printing large-scale graphics, such as engineering drawings and maps.

3. Input/Output (I/O) Devices

Some devices function as both input and output devices, meaning they can receive and send data.

Examples of I/O Devices:

1. **Touchscreen** – Acts as both an input (when touched) and an output (when displaying information).
2. **External Storage Devices (USB Drive, Hard Drive, SSD)** – Allow data to be read (output) and written (input).
3. **CD/DVD Drive** – Reads (output) and writes (input) data on CDs or DVDs.
4. **Modem** – Sends (input) and receives (output) data for internet access.
5. **Multifunction Printer (MFP)** – Combines printing (output) and scanning (input) functions.

Types of Computers (Analog, Digital & Hybrid)

Computers are classified based on how they process data. The three main types are **Analog Computers, Digital Computers, and Hybrid Computers**. Each type serves different purposes and is used in specific fields.

1. Analog Computers

Analog computers process **continuous data** (real-world physical quantities) such as temperature, speed, pressure, and voltage. They do not use binary digits (0s and 1s) but instead represent data using electrical, mechanical, or hydraulic signals.

Characteristics of Analog Computers:

- Work with continuous data.
- Provide approximate results rather than precise values.
- Used for real-time simulations and control systems.
- Faster in processing certain types of calculations (e.g., physics-based modeling).

Examples of Analog Computers:

- **Thermometer** – Measures temperature using mercury or digital sensors.
- **Speedometer** – Displays vehicle speed based on continuous input from wheels.
- **Seismograph** – Records earthquake waves and vibrations.
- **Analog Voltmeter** – Measures voltage variations in electrical circuits.

2. Digital Computers

Digital computers process **discrete data** in binary form (0s and 1s). They are the most common type of computers used today. Digital computers perform complex calculations, store large amounts of data, and support a wide range of applications.

Characteristics of Digital Computers:

- Use binary numbers (0s and 1s) to process data.
- Provide accurate and precise results.
- Store and retrieve data efficiently.
- Can perform multitasking and run multiple applications.

Types of Digital Computers:

1. **Supercomputers** – Extremely powerful computers used for complex simulations, weather forecasting, and scientific research. *(e.g., IBM Summit, Fugaku)*
2. **Mainframe Computers** – Large-scale computers used by organizations for bulk data processing. *(e.g., IBM Z-series)*
3. **Minicomputers** – Mid-range computers used for business and industrial applications.
   (e.g., PDP-11, VAX 750)
4. **Microcomputers (Personal Computers - PCs)** – Common computers used at home and in offices. *(e.g., Laptops, Desktops, Tablets, Smartphones)*

Examples of Digital Computers:

- Desktop Computers
- Laptops
- Smartphones
- ATMs
- Digital Watches

3. Hybrid Computers

Hybrid computers combine the features of both **analog and digital computers**. They process both continuous (analog) and discrete (digital) data. Hybrid computers are mainly used in specialized fields where real-time processing and precise calculations are required.

Characteristics of Hybrid Computers:

- Convert analog signals into digital form for processing.
- Provide real-time data with high accuracy.
- Used in medical, industrial, and scientific applications.

Examples of Hybrid Computers:

- **Medical Equipment (ECG & MRI Machines)** – Convert body signals (analog) into digital data for diagnosis.
- **Weather Forecasting Systems** – Use analog sensors to collect environmental data and process it digitally.
- **Missile Control Systems** – Combine real-time analog data with digital computing for targeting and control.
- **Industrial Process Control** – Monitor and control temperature, pressure, and speed in manufacturing.

Comparison of Analog, Digital, and Hybrid Computers

Feature	Analog Computers	Digital Computers	Hybrid Computers
Data Type	Continuous (Real-world signals)	Discrete (Binary: 0s & 1s)	Both Continuous & Discrete
Accuracy	Approximate	Highly Accurate	High Accuracy
Speed	Fast for real-time calculations	Fast for logical and complex processing	Combines both fast real-time and accurate processing
Usage	Scientific simulations, measuring devices	Business, education, entertainment, general computing	Medical, industrial, military applications
Example	Thermometer, Speedometer	Laptop, Smartphone, ATM	MRI Machine, Weather Systems

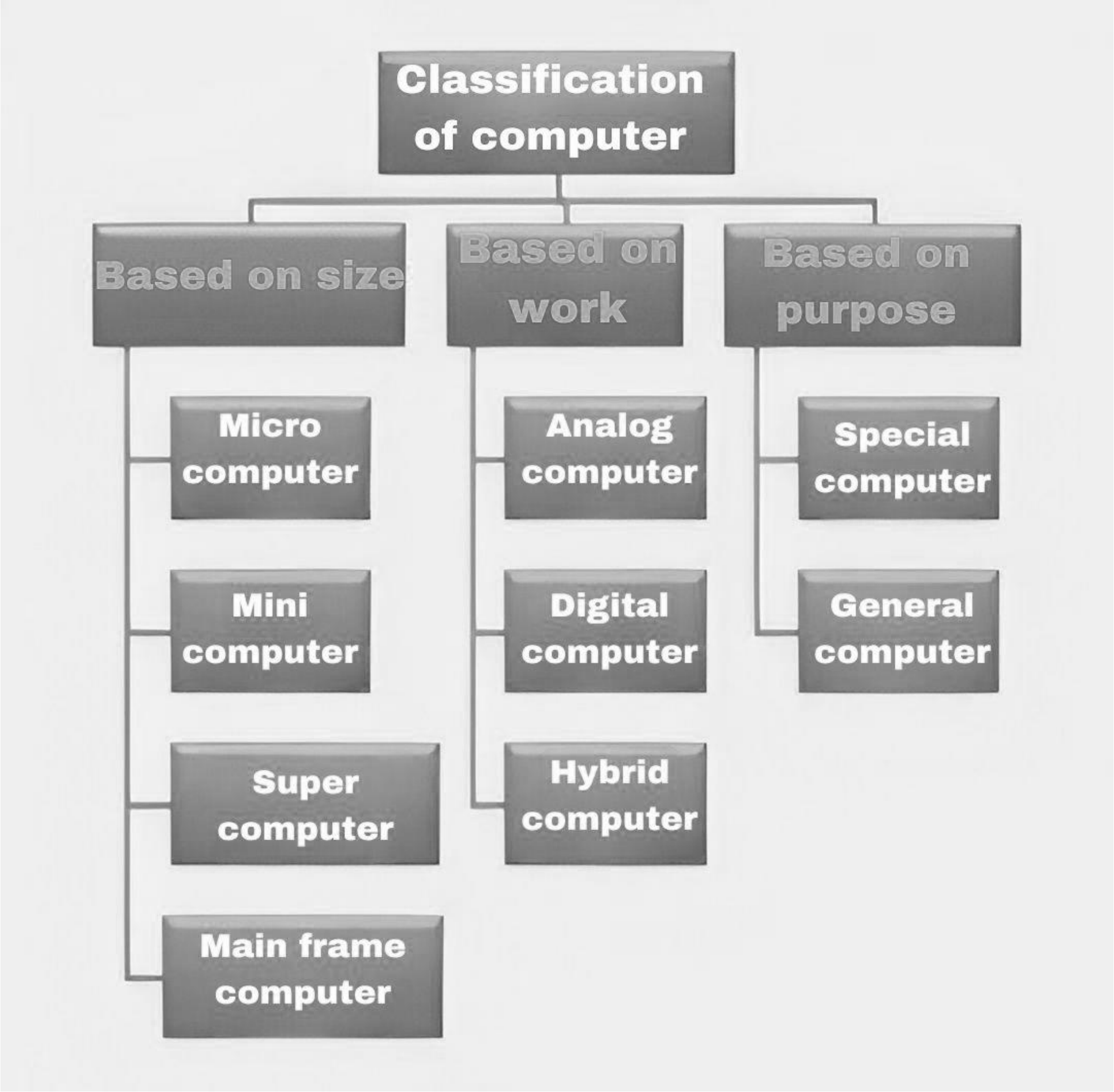
Classification of Computers

Computers can be classified based on **size, purpose, and data handling capabilities**. These classifications help distinguish computers based on their processing power, functionality, and applications.

1. Classification Based on Size

a) Supercomputers

- The most powerful and fastest computers.
- Used for complex scientific calculations and simulations.
- Capable of processing massive amounts of data at extremely high speeds.
- Used in weather forecasting, space research, nuclear simulations, and AI research.
- **Examples:** IBM Summit, Fugaku, Titan.



b) Mainframe Computers

- Large, powerful computers that handle massive amounts of data.
- Used by governments, banks, and large organizations for bulk processing.
- Can support thousands of users simultaneously.

- **Examples:** IBM Z-Series, Unisys ClearPath.

c) Minicomputers (Mid-Range Computers)

- Smaller than mainframes but still powerful.
- Used by medium-sized businesses for data processing.
- Can support multiple users at once.
- **Examples:** DEC PDP-11, IBM AS/400.

d) Microcomputers (Personal Computers - PCs)

- The most commonly used computers today.
- Designed for personal or business use.
- Includes desktops, laptops, tablets, and smartphones.
- **Examples:** Dell Inspiron (desktop), MacBook (laptop), iPad (tablet).

2. Classification Based on Purpose

a) General-Purpose Computers

- Designed for a variety of tasks, such as browsing, gaming, office work, and communication.
- Can run multiple applications.
- **Examples:** Laptops, Desktops, Smartphones.

b) Special-Purpose Computers

- Designed for a specific task or application.
- Used in scientific research, industrial control, medical diagnosis, and military systems.
- **Examples:** ATM machines, MRI scanners, Traffic control systems.

3. Classification Based on Data Handling Capabilities

a) Analog Computers

- Process **continuous data** such as temperature, pressure, and speed.
- Used in scientific and engineering applications.
- **Examples:** Speedometer, Seismograph, Analog voltmeter.

b) Digital Computers

- Process **discrete data** in binary (0s and 1s).
- Most modern computers fall into this category.
- **Examples:** Laptops, Desktops, ATMs, Smartphones.

c) Hybrid Computers

- Combine the features of analog and digital computers.
- Used in real-time applications where both continuous and discrete data are needed.
- **Examples:** MRI machines, Weather forecasting systems, Missile control systems.

System Software & Application Software

- Software is a set of programs that enable a computer to perform specific tasks. It is broadly classified into **System Software** and **Application Software** based on its function.

1. System Software

System software is designed to manage and control the hardware components of a computer. It acts as an interface between the user and hardware, ensuring that applications function smoothly.

Types of System Software:

a) Operating System (OS)

- Manages hardware resources and provides a user interface.
- Handles file management, memory management, and process control.
- **Examples:** Windows, macOS, Linux, Android, iOS.

b) Utility Software

- Helps in maintaining and optimizing the system.
- Performs tasks like antivirus protection, disk cleanup, and file compression.
- **Examples:** CCleaner, WinRAR, Norton Antivirus.

c) Device Drivers

- Enable communication between hardware devices and the operating system.
- Required for printers, keyboards, graphics cards, etc.
- **Examples:** Printer drivers, Graphics drivers (NVIDIA, AMD).

d) Firmware

- Embedded software that controls specific hardware functions.
- Stored in ROM and works without user interaction.

- **Examples:** BIOS, Router firmware, Smart TV firmware.

2. Application Software

Application software is designed for end users to perform specific tasks such as writing documents, playing games, or browsing the internet.

Types of Application Software:

a) General-Purpose Software

- Designed for a wide range of users and tasks.
- **Examples:**
  - Microsoft Office (Word, Excel, PowerPoint) – For document creation and analysis.
  - Google Chrome, Mozilla Firefox – Web browsers.
  - VLC Media Player – For playing audio and video files.

b) Specialized Software

- Designed for specific industries or tasks.
- **Examples:**
  - AutoCAD – Used for engineering and architectural design.
  - Photoshop – Used for image editing.
  - Tally – Used for accounting and financial management.

c) Custom Software

- Developed for a particular organization or individual needs.
- **Examples:**
  - Banking software for financial institutions.
  - Hospital management systems for healthcare facilities.

d) Mobile Applications

- Designed for smartphones and tablets.
- **Examples:**
  - WhatsApp – Messaging app.
  - Instagram – Social media platform.
  - Uber – Ride-booking application.

Comparison of System Software & Application Software

Feature	System Software	Application Software
Function	Manages system resources and hardware.	Performs specific user-oriented tasks.
User Interaction	Works in the background.	Directly used by users.

Dependency	Required for system functionality.	Depends on system software to function.
Examples	Windows OS, Antivirus, Drivers.	MS Word, Photoshop, Chrome.

Number System Conversions (Binary, Decimal, Octal, Hexadecimal)

A **number system** is a way of representing numbers using a specific base. The most commonly used number systems in computing are:

1. **Binary (Base-2)** – Uses digits **0** and **1**.
2. **Decimal (Base-10)** – Uses digits **0 to 9** (our everyday number system).
3. **Octal (Base-8)** – Uses digits **0 to 7**.
4. **Hexadecimal (Base-16)** – Uses digits **0 to 9** and letters **A to F** (A=10, B=11, ..., F=15).

1. Converting Between Number Systems

a) Decimal to Other Bases

1. **Decimal to Binary**

- Divide the decimal number by **2**.
- Write down the remainder (0 or 1).
- Continue dividing until the quotient is 0.
- The binary number is the remainders read **from bottom to top**.

Example: Convert 25 (Decimal) to Binary

25 ÷ 2 = 12, remainder 1

12 ÷ 2 = 6, remainder 0

6 ÷ 2 = 3, remainder 0

3 ÷ 2 = 1, remainder 1

1 ÷ 2 = 0, remainder 1

- **Binary Equivalent: 11001_2**

- **Decimal to Octal**

  - Divide the decimal number by **8** and note the remainder.
  - Read the remainders from **bottom to top**.

Example: Convert 45 (Decimal) to Octal

45 ÷ 8 = 5, remainder 5

5 ÷ 8 = 0, remainder 5

- **Octal Equivalent: 55_8**

- **Decimal to Hexadecimal**

  - Divide the decimal number by **16**.
  - Convert remainders above 9 to letters (A-F).
  - Read from **bottom to top**.

Example: Convert 254 (Decimal) to Hexadecimal

254 ÷ 16 = 15, remainder 14 (E)

15 ÷ 16 = 0, remainder 15 (F)

b) Binary to Other Bases

1. **Binary to Decimal**
   - Multiply each bit by **2 raised to its position value**, starting from **0** (right to left).
   - Sum the results.

   Example: Convert 1011_2 to Decimal

$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= (8) + (0) + (2) + (1)$

$= 11_{10}$

Binary to Octal

- Group the binary digits into **sets of three** (from right to left).
- Convert each group to its octal equivalent.

Example: Convert 101110_2 to Octal

Grouping: 101 110

$101_2 = 5_8$, $110_2 = 6_8$

- **Octal Equivalent: 56_8**

- **Binary to Hexadecimal**

  - Group the binary digits into **sets of four** (from right to left).
  - Convert each group to its hexadecimal equivalent.

Example: Convert 10110110_2 to Hexadecimal

Grouping: 1011 0110

$1011_2 = B_{16}$, $0110_2 = 6_{16}$

c) Octal & Hexadecimal Conversions

1. **Octal to Binary**

- Convert each octal digit to its **3-bit binary** equivalent.
- **Example:** Convert 56_8 to Binary

$5_8 = 101_2$, $6_8 = 110_2$

Hexadecimal to Binary

- Convert each hexadecimal digit to its **4-bit binary** equivalent.
- **Example:** Convert $2F_{16}$ to Binary

$2_{16} = 0010_2$, $F_{16} = 1111_2$

Octal to Decimal

- Multiply each digit by **8 raised to its position value** and sum the results.
- **Example:** Convert 237_8 to Decimal

$(2 \times 8^2) + (3 \times 8^1) + (7 \times 8^0)$

$= (2 \times 64) + (3 \times 8) + (7 \times 1)$

$= 128 + 24 + 7$

$= 159_{10}$

Hexadecimal to Decimal

- Multiply each digit by **16 raised to its position value** and sum the results.
- **Example:** Convert $3F_{16}$ to Decimal

$(3 \times 16^1) + (F \times 16^0)$

$= (3 \times 16) + (15 \times 1)$

$= 48 + 15$

$= 63_{10}$

Summary Table for Quick Conversion

From	To Binary	To Decimal	To Octal	To Hexadecimal
Binary	—	Multiply by 2^n	Group in 3s	Group in 4s
Decimal	Divide by 2	—	Divide by 8	Divide by 16
Octal	Convert each digit to 3-bit binary	Multiply by 8^n	—	Convert to binary, then to hex
Hexadecimal	Convert each digit to 4-bit binary	Multiply by 16^n	Convert to binary, then to octal	—

Operating System and Its Functions

An **Operating System (OS)** is system software that manages computer hardware and software resources, providing an interface between the user and the computer. It ensures smooth operation and enables users to interact with the system efficiently.

Functions of an Operating System

The primary functions of an operating system include:

1. Process Management

- A process is a program in execution.
- The OS **creates, schedules, and terminates processes**.
- It ensures efficient CPU utilization by managing multiple processes (multitasking).

Key tasks in process management:

- Process scheduling (assigning CPU time to processes).
- Handling process synchronization and inter-process communication (IPC).
- Managing process states (New, Ready, Running, Waiting, Terminated).

Example: Windows Task Manager allows users to monitor and manage running processes.

2. Memory Management

- The OS **allocates and deallocates memory** to processes.
- It keeps track of memory usage and prevents conflicts.
- Implements **virtual memory**, allowing programs to use more memory than physically available using disk storage.

Example: When multiple applications run on a PC, the OS manages RAM allocation dynamically.

3. File Management

- The OS organizes and controls the storage and retrieval of data on storage devices.
- It provides functions like creating, editing, deleting, and organizing files and directories.
- Implements **file access permissions** for security.

Example: Windows File Explorer allows users to manage files efficiently.

4. Device Management

- The OS manages input and output (I/O) devices such as keyboards, printers, USB drives, and monitors.
- Uses **device drivers** to communicate with hardware.
- Handles device allocation and deallocation.

Example: When a printer is connected, the OS loads the necessary drivers and queues print jobs.

5. User Interface (UI) Management

- Provides a user-friendly interface to interact with the computer.
- Two main types of interfaces:
    - **Graphical User Interface (GUI):** Windows, macOS, Android.
    - **Command Line Interface (CLI):** Linux Terminal, Command Prompt (CMD).

Example: Windows OS provides a GUI, while Linux also offers a CLI.

6. Security & Access Control

- Protects system data from unauthorized access and cyber threats.
- Implements user authentication (login credentials, biometrics).
- Provides **firewalls, encryption, and antivirus support**.

Example: Windows Defender protects against malware and unauthorized access.

7. Network Management

- Enables communication between computers in a network.
- Supports protocols like TCP/IP for internet connectivity.
- Manages network sharing, remote access, and security.

Example: Windows and macOS support Wi-Fi, Bluetooth, and Ethernet connections.

8. Error Handling & System Monitoring

- Detects and reports hardware or software errors.
- Provides diagnostic tools to troubleshoot issues.

Example: The Blue Screen of Death (BSOD) in Windows indicates a system error.

Types of Operating Systems

1. **Single-User OS** – Supports one user at a time (e.g., Windows, macOS).
2. **Multi-User OS** – Allows multiple users simultaneously (e.g., UNIX, Linux).
3. **Real-Time OS (RTOS)** – Used in critical systems like aircraft controls and medical devices (e.g., VxWorks).
4. **Embedded OS** – Used in embedded systems like smart TVs and IoT devices (e.g., Android in smart TVs).
5. **Distributed OS** – Manages a group of networked computers as one system (e.g., Apache Hadoop).

Operating Systems

Operating Systems (OS) serve as a bridge between computer hardware and users, enabling smooth operation and resource management. Some of the most popular operating systems include **MS-DOS, Windows, UNIX, and Linux**.

1. MS-DOS (Microsoft Disk Operating System)

MS-DOS (Microsoft Disk Operating System) is a command-line operating system developed by **Microsoft** in the early 1980s. It was widely used before the rise of graphical operating systems like **Windows** and played a crucial role in the early days of personal computing.

Common MS-DOS Commands

Command	Function
`DIR`	Lists files and directories
`CD`	Changes the current directory
`COPY`	Copies files
`DEL`	Deletes files
`FORMAT`	Formats a disk
`CHKDSK`	Checks disk integrity
`EDIT`	Opens a simple text editor
`CLS`	Clears the screen
`EXIT`	Exits the command shell

- **Developer:** Microsoft
- **Release Year:** 1981
- **Type:** Command-Line Interface (CLI) OS
- **Platform:** x86

Key Features:

- Single-user, single-tasking OS.
- Uses a **command-line interface (CLI)** with no graphical user interface (GUI).
- Limited memory management and multitasking capabilities.
- Popular in the early 1980s for personal computers before Windows.

Current Usage:

- **Obsolete** but still used in embedded systems and legacy applications.

2. Windows OS

Windows OS is a family of **graphical operating systems** developed by **Microsoft**. It is the most widely used desktop operating system globally, offering a **user-friendly interface, multitasking capabilities, and extensive software support**.

- **Developer:** Microsoft
- **Release Year:** 1985 (Windows 1.0)
- **Type:** GUI-based OS
- **Platform:** x86, x64

Key Features:

- **Graphical User Interface (GUI)** with icons, menus, and windows.
- Supports multitasking, networking, and multimedia applications.
- Regular updates and new versions with improved security and performance.
- Supports a vast range of software and hardware.

Popular Windows Versions:

- **Windows 95/98** – Early GUI versions.
- **Windows XP** – One of the most stable and popular releases.
- **Windows 7** – Improved security and performance.

- **Windows 10** – Universal apps, Cortana, and frequent updates.
- **Windows 11** – Latest version with a redesigned UI, improved multitasking, and gaming features.

Current Usage:

- **Most widely used OS** for personal computers, gaming, business applications, and enterprise use.

3. UNIX

- **Developer:** AT&T Bell Labs (Dennis Ritchie & Ken Thompson)
- **Release Year:** 1969
- **Type:** Multi-user, multitasking OS
- **Platform:** Various (RISC, x86, ARM)

Key Features:

- Command-line and GUI-based system.
- Multi-user and multitasking capabilities.
- Highly **secure and stable** (preferred for servers).
- Uses a hierarchical file system.
- Written in **C language**, making it portable across devices.

Common UNIX Commands:

- `ls` – Lists files and directories.
- `pwd` – Displays the current directory path.
- `cp` – Copies files.
- `rm` – Deletes files.

Popular UNIX Variants:

- **IBM AIX** – Used in enterprise environments.
- **HP-UX** – Hewlett-Packard's UNIX version.
- **Solaris** – Developed by Sun Microsystems (now owned by Oracle).

Current Usage:

- Used in **servers, mainframes, and high-performance computing** environments.

4. Linux

- **Developer:** Linus Torvalds
- **Release Year:** 1991
- **Type:** Open-source, multi-user OS
- **Platform:** x86, x64, ARM

Key Features:

- **Open-source and free** – Anyone can modify and distribute it.
- **Highly secure and stable**, making it ideal for servers.
- **Customizable** – Available in different distributions (distros).
- Can run on various devices, from **PCs to supercomputers and embedded systems**.

Popular Linux Distributions (Distros):

- **Ubuntu** – User-friendly, best for beginners.
- **Fedora** – Cutting-edge technology, used by developers.
- **Debian** – Stable and secure, used in enterprise environments.
- **Red Hat Enterprise Linux (RHEL)** – Used in businesses and servers.
- **Kali Linux** – Designed for ethical hacking and cybersecurity.

Common Linux Commands:

- `ls` – Lists files and directories.
- `mkdir` – Creates a new directory.
- `sudo` – Runs commands as an administrator.
- `grep` – Searches for text within files.

Current Usage:

- **Widely used in servers, supercomputers, IoT devices, and cybersecurity**.
- Android OS (used in smartphones) is based on Linux.

Comparison of MS-DOS, Windows, UNIX, and Linux

Feature	MS-DOS	Windows	UNIX	Linux

User Interface	CLI	GUI & CLI	CLI & GUI	CLI & GUI
Multitasking	No	Yes	Yes	Yes
Security	Low	Moderate	High	High
Customization	No	Limited	Limited	High (open-source)
Usage	Legacy systems	Personal & business computers	Servers & workstations	Servers, desktops, and embedded systems
Cost	Free	Paid	Paid (some versions)	Free & Paid (enterprise versions)

Basic Commands for DOS & Linux

Both **DOS (Disk Operating System)** and **Linux** use a command-line interface (CLI) for managing files, directories, and system tasks. Below are some essential commands for both operating systems.

1. DOS (Disk Operating System) Commands

- DOS commands are mainly used in **Windows Command Prompt (CMD)** and **MS-DOS environments**.
- Commands are typically **not case-sensitive**.

Basic DOS Commands

Command	Description	Example
`DIR`	Lists files and directories	`DIR`
`CD`	Changes the directory	`CD Documents`
`MD` / **`MKDIR`**	Creates a new directory	`MD NewFolder`
`RD` / **`RMDIR`**	Removes a directory	`RD OldFolder`
`DEL`	Deletes a file	`DEL file.txt`
`COPY`	Copies a file	`COPY file.txt D:\Backup`
`MOVE`	Moves a file or directory	`MOVE file.txt D:\Documents`
`REN`	Renames a file	`REN oldname.txt newname.txt`
`CLS`	Clears the screen	`CLS`
`EXIT`	Exits the DOS command prompt	`EXIT`
`TYPE`	Displays the content of a text file	`TYPE file.txt`
`ECHO`	Prints a message or enables/disables command display	`ECHO Hello World`
`DATE`	Displays or sets the system date	`DATE`

TIME	Displays or sets the system time	`TIME`
CHKDSK	Checks disk errors and displays status	`CHKDSK C:`
FORMAT	Formats a disk drive	`FORMAT D:`
ATTRIB	Changes file attributes	`ATTRIB +R file.txt`
PATH	Displays or sets the system path	`PATH C:\Windows`
VER	Displays the OS version	`VER`

2. Linux Commands

- Linux commands are **case-sensitive**.
- They are used in **Linux Terminal** or **Shell (Bash, Zsh, etc.)**.

Basic Linux Commands

Command	Description	Example
ls	Lists files and directories	`ls -l`
pwd	Shows current directory path	`pwd`
cd	Changes the directory	`cd /home/user/Documents`
mkdir	Creates a new directory	`mkdir NewFolder`
rmdir	Removes an empty directory	`rmdir OldFolder`
rm	Deletes a file or directory	`rm file.txt`
cp	Copies a file	`cp file.txt /backup/`
mv	Moves or renames a file	`mv old.txt new.txt`
cat	Displays file contents	`cat file.txt`
echo	Prints a message or writes to a file	`echo "Hello" > file.txt`
clear	Clears the terminal screen	`clear`
exit	Exits the terminal	`exit`
touch	Creates an empty file	`touch newfile.txt`
whoami	Displays the current user	`whoami`
date	Shows the system date and time	`date`
df	Displays disk usage	`df -h`
du	Shows disk space usage of files and directories	`du -sh *`
uname	Displays system information	`uname -a`
chmod	Changes file permissions	`chmod 755 file.sh`
chown	Changes file ownership	`chown user:group file.txt`
ps	Shows active processes	`ps aux`
kill	Kills a process by ID	`kill 1234`
top	Displays running processes dynamically	`top`
shutdown	Shuts down the system	`shutdown -h now`
reboot	Restarts the system	`reboot`

Comparison of DOS & Linux Commands

What is Networking?

Networking refers to the practice of connecting multiple computers and devices to share resources, communicate, and exchange data. A network can be **wired** (using cables) or **wireless** (using Wi-Fi, Bluetooth, etc.).

Basic Components of a Network:

- **Nodes:** Devices connected in a network (e.g., computers, printers, routers).
- **Switches & Hubs:** Devices that connect multiple computers in a LAN.
- **Routers:** Connects different networks and directs data traffic.
- **Network Cables/Wi-Fi:** Mediums used to transmit data.
- **IP Address:** A unique identifier for devices in a network.

Types of Networks

Action	DOS Command	Linux Command
List files	`DIR`	`ls`
Change directory	`CD folder`	`cd folder`
Create directory	`MD folder`	`mkdir folder`
Remove directory	`RD folder`	`rmdir folder`
Delete file	`DEL file.txt`	`rm file.txt`
Copy file	`COPY file.txt D:\`	`cp file.txt /backup/`
Move file	`MOVE file.txt D:\`	`mv file.txt /backup/`
Rename file	`REN old.txt new.txt`	`mv old.txt new.txt`
Show file content	`TYPE file.txt`	`cat file.txt`
Clear screen	`CLS`	`clear`
Show current directory	`CD`	`pwd`
Show active processes	`TASKLIST`	`ps aux`
Kill process	`TASKKILL /PID 1234`	`kill 1234`
Display OS version	`VER`	`uname -a`

Networking is categorized into different types based on the geographical area it covers.

1. Local Area Network (LAN)

- **Definition:** A network covering a small geographical area, such as an office, home, or school.
- **Devices connected:** Computers, printers, servers, and other peripherals.
- **Communication Speed: Fast** (up to 10 Gbps in modern LANs).

- **Technology Used:** Ethernet (wired), Wi-Fi (wireless).

Examples:

- A home Wi-Fi network.
- Office network connecting employees' computers.
- A school computer lab.

Advantages of LAN:

- High speed due to short distances.
- Secure and easy to manage.
- Allows resource sharing (e.g., printers, files).

Disadvantages of LAN:

- Limited coverage (only works in a small area).
- Installation and maintenance require technical expertise.

2. Metropolitan Area Network (MAN)

- **Definition:** A network that spans a **city or large campus**.
- **Communication Speed: Moderate to High** (100 Mbps to 1 Gbps).
- **Technology Used:** Fiber optics, Ethernet, microwave links.

Examples:

- A city's public Wi-Fi network.
- University-wide networks connecting different campus buildings.
- Cable TV and broadband services.

Advantages of MAN:

- Covers a **larger area** than LAN.
- High-speed connectivity between LANs.
- Cost-effective for organizations operating in a city.

Disadvantages of MAN:

- Requires **more infrastructure** (fiber optics, routers).

- Expensive to maintain.

3. Wide Area Network (WAN)

- A network that covers a **large geographical area**, often across countries or continents.
- **Communication Speed: Varies** (1 Mbps to several Gbps).
- **Technology Used:** Satellite links, fiber optics, leased telephone lines.

  Examples:

- The **Internet** (the world's largest WAN).
- A multinational company's private network connecting offices worldwide.
- A government's secure communication system.

Advantages of WAN:

- Covers **long distances** (global connectivity).
- Allows remote access to resources.
- Essential for **businesses, governments, and cloud services**.

Disadvantages of WAN:

- **Slower** than LAN/MAN due to long-distance transmission.
- More **expensive** due to infrastructure needs.
- **Security risks** (e.g., cyber-attacks, data breaches).

Comparison of LAN, MAN, and WAN

Feature	LAN (Local Area Network)	MAN (Metropolitan Area Network)	WAN (Wide Area Network)
Coverage	Small area (home, office)	Medium area (city, university)	Large area (countries, global)
Speed	High (1-10 Gbps)	Moderate (100 Mbps - 1 Gbps)	Varies (1 Mbps - Gbps)
Technology	Ethernet, Wi-Fi	Fiber optics, microwave	Satellite, leased lines
Cost	Low	Moderate	High
Examples	Office Wi-Fi, Home Network	City Wi-Fi, Cable TV	Internet, Global corporate network

What is Network Topology?

Network topology refers to the **arrangement of devices (nodes), cables, and network elements** in a communication network. It defines how data flows between devices.

There are two types of network topologies:

1. **Physical Topology** – The actual layout of devices and cables.
2. **Logical Topology** – The way data flows within the network.

Types of Network Topologies

1. Bus Topology

- **Definition:** All devices are connected to a **single central cable** (backbone).
- **Data Transmission:** Data travels in both directions along the backbone.

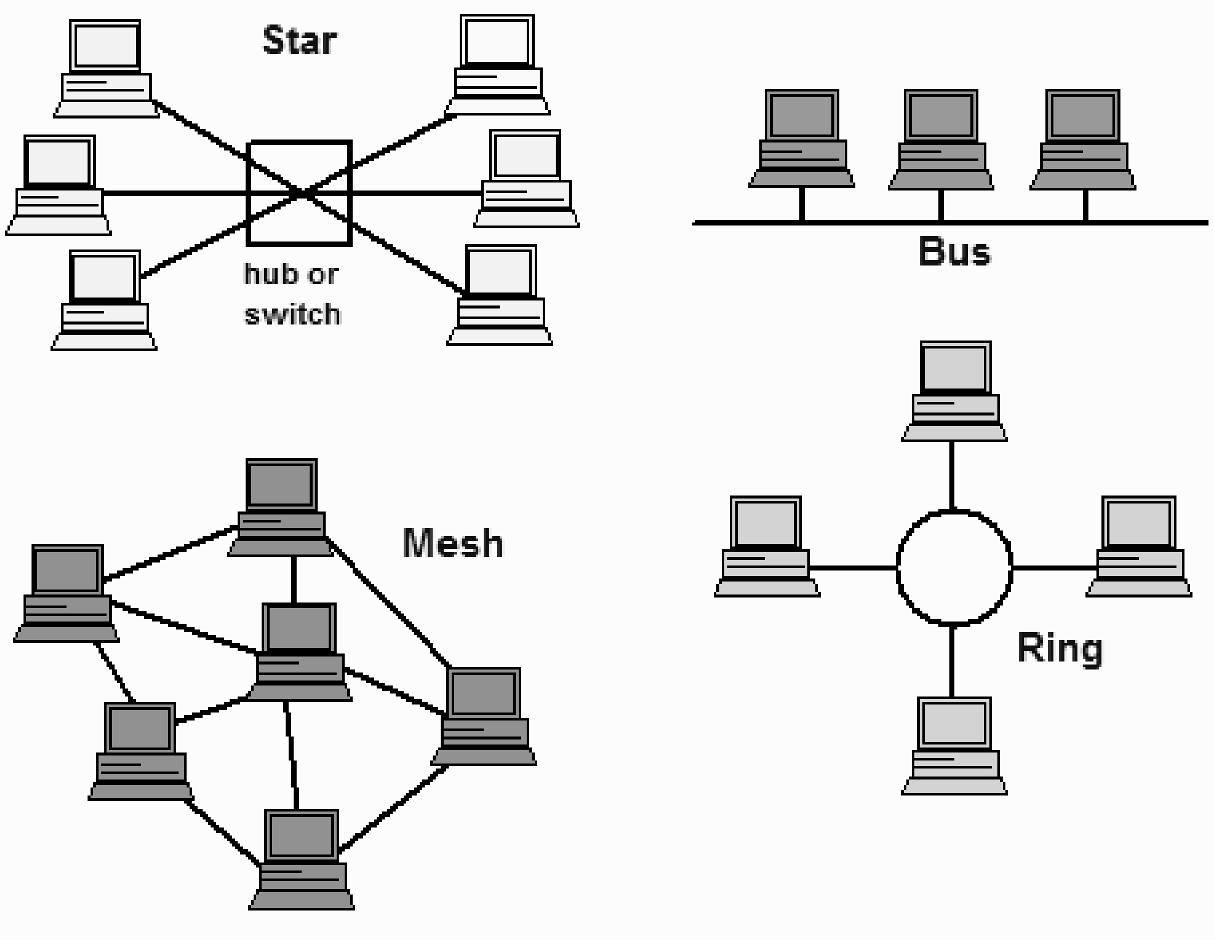
Advantages:

- **Easy to set up** (requires less cable).
- **Cost-effective** for small networks.

Disadvantages:

- **If the main cable fails, the entire network fails**.
- **Slow performance** when multiple devices transmit data.

Example: Used in early Ethernet networks.



2. Star Topology

- **Definition:** All devices are connected to a **central switch or hub**.
- **Data Transmission:** Data passes through the central hub before reaching the destination.

Advantages:

- If one device fails, the network remains operational**.**
- Easy to troubleshoot and expand**.**

Disadvantages:

- **If the central hub/switch fails, the whole network goes down**.
- **More expensive** due to additional hardware (switch or hub).

Example: Common in modern home and office networks.

3. Ring Topology

- **Definition:** Devices are connected in a **circular loop**.
- **Data Transmission:** Data travels in **one direction (unidirectional)** or both directions (bidirectional).

Advantages:

- **No need for a central hub**.
- **Data collisions are minimized** because data flows in a set direction.

Disadvantages:

- **If one device fails, the entire network is affected**.
- **Difficult to troubleshoot and expand**.

Example: Used in fiber-optic networks and some office environments.

4. Mesh Topology

- **Definition:** Every device is connected to every other device.
- **Data Transmission:** Multiple paths exist between devices, ensuring redundancy.

Types:

- **Full Mesh:** Every device is connected to every other device.
- **Partial Mesh:** Some devices are connected to multiple other devices, but not all.

Advantages:

- **Highly reliable** – even if one connection fails, data can take another route.
- **Fast and efficient data transfer** due to multiple paths.

Disadvantages:

- **Very expensive and complex** due to the large number of connections.
- **Requires a lot of cables and maintenance**.

Example: Used in **military communications and large-scale networks**.

5. Tree Topology (Hierarchical Topology)

- **Definition:** A combination of **star and bus topologies** in a hierarchical manner.
- **Data Transmission:** Data flows from parent nodes to child nodes.

Advantages:

- **Scalable** – easy to expand by adding new nodes.
- **Efficient for large organizations** with multiple departments.

Disadvantages:

- If the main backbone cable fails, the whole network is affected**.**
- More cabling is required than bus or star topology**.**

◆ **Example:** Used in **corporate networks and universities**.

6. Hybrid Topology

- **Definition:** A combination of two or more network topologies (e.g., star + bus, mesh + tree).
- **Data Transmission:** Varies based on the combination used.

Advantages:

- **Highly flexible and scalable**.
- **Reliable** – failure in one section does not affect the entire network.

Disadvantages:

- **Complex setup and maintenance**.
- **Expensive** due to multiple topologies and devices.

Example: Used in **large enterprises and data centers**.

Comparison of Network Topologies

Topology	Structure	Advantages	Disadvantages	Use Cases
Bus	Single central cable	Cost-effective, easy setup	Failure of backbone cable	Small offices, early Ethernet

			affects the whole network	
Star	Devices connected to a central hub/switch	Easy troubleshooting, scalable	If the hub fails, the network fails	Home & office networks
Ring	Devices connected in a circle	No data collisions, predictable performance	Failure of one node disrupts the entire network	Fiber-optic networks
Mesh	Every device is connected to every other device	Highly reliable, no single point of failure	Expensive, complex to manage	Military, large-scale networks
Tree	Hierarchical (star + bus)	Scalable, well-structured	If the backbone fails, the network goes down	Large organizations, universities
Hybrid	Mix of different topologies	Flexible, reliable	Complex, costly	Large enterprises, data centers

Evolution of the Internet

The **Internet** is a vast global network that connects millions of computers worldwide, enabling communication, data sharing, and access to information. Its development has gone through several stages:

1. ARPANET (1969) – The Beginning

- The **Advanced Research Projects Agency Network (ARPANET)** was the first network, created by the **U.S. Department of Defense**.
- It used **packet switching** technology to allow multiple computers to communicate.
- Initially, only four universities were connected.

2. Expansion of Networks (1970s - 1980s)

- **TCP/IP (Transmission Control Protocol/Internet Protocol)** was developed in the **1970s**, forming the foundation of modern networking.
- The first **email** system was created in 1971.
- The **Domain Name System (DNS)** was introduced in 1983, allowing the use of readable addresses like **google.com** instead of IP numbers.

3. World Wide Web (1990s)

- **Tim Berners-Lee** developed the **World Wide Web (WWW)** in 1989, introducing web browsers and hyperlinks.
- The first **web browser (Mosaic)** was created in 1993, followed by **Netscape and Internet Explorer**.
- The rise of **websites, online services, and search engines** (like Yahoo! and Google) revolutionized the Internet.

4. Modern Internet (2000s - Present)

- The rise of **social media, e-commerce, and cloud computing** (Facebook, Amazon, Google).
- The development of **mobile internet (3G, 4G, 5G)**.
- The emergence of **Artificial Intelligence (AI)**, **Internet of Things (IoT)**, and **blockchain technology**.

Internet Services

The Internet provides various services that allow communication, data transfer, and information sharing. Some of the most important services include:

1. Email (Electronic Mail)

- Allows users to send and receive digital messages over the Internet.
- Works using **SMTP (Simple Mail Transfer Protocol)** for sending emails and **IMAP/POP3** for receiving.

  Examples: Gmail, Yahoo Mail, Outlook.

Advantages:

- Fast and cost-effective communication.
- Supports attachments (documents, images, videos).
- Can be accessed from anywhere.

Disadvantages:

- Spam and phishing emails.
- Requires an internet connection.

2. FTP (File Transfer Protocol)

- Used for **uploading and downloading files** between computers over the Internet.
- Works using **FTP clients** like **FileZilla** or command-line FTP commands.

Examples: Uploading website files to a server.

Advantages:

- Fast and reliable file transfers.
- Supports large file sizes.

Disadvantages:

- Requires a separate FTP client or command-line knowledge.
- Less secure unless encrypted (SFTP).

3. Telnet (Remote Login)

- A command-line tool that allows users to access **remote computers** over a network.
- Uses **TCP/IP protocol** to establish a connection.

Examples: System administrators using Telnet to manage remote servers.

Advantages:

- Provides remote access to a computer from anywhere.

Disadvantages:

- **Not secure** (data is transmitted in plain text).
- Mostly replaced by **SSH (Secure Shell)** for security.

4. Usenet (User Network)

- A **decentralized online discussion system** where users can post and read messages in "newsgroups."

- Works on the **NNTP (Network News Transfer Protocol)**.

Examples: Academic and technical discussions in the 1980s & 1990s.

Advantages:

- Supports threaded discussions.
- Allows file sharing (though limited).

Disadvantages:

- Declining usage due to social media and modern forums.
- Not user-friendly for beginners.

5. News (Online News Services)

- Websites that provide up-to-date **news articles, blogs, and information**.
- Includes both traditional media and independent online platforms.

Examples: BBC, CNN, Google News.

Advantages:

- Instant access to worldwide news.
- Variety of sources available.

Disadvantages:

- Fake news and misinformation.
- Requires internet access.

6. WWW (World Wide Web)

- A system of **interlinked web pages and multimedia** that can be accessed via a web browser.

- Uses **HTTP (HyperText Transfer Protocol)** and **HTTPS (secure version)** for communication.

Service	Purpose	Protocol Used	Example Uses
Email	Sending messages	SMTP, IMAP, POP3	Gmail, Outlook
FTP	File transfer	FTP, SFTP	Uploading website files
Telnet	Remote access	Telnet	Server management
Usenet	Online discussion	NNTP	Newsgroups
News	Information updates	HTTP, RSS	BBC, Google News
WWW	Browsing the internet	HTTP, HTTPS	Google, YouTube

Examples: Google, YouTube, Wikipedia.

Advantages:

- Provides unlimited information and services.
- Easy to access and use via browsers.
- Supports multimedia (text, images, videos).

Disadvantages:

- Security risks (malware, hacking).
- Overload of information.

Comparison of Internet Services

Uses of the Internet in Biotechnology

The **Internet** has transformed **biotechnology** by enabling global collaboration, advanced research, and faster data analysis. Here are some key ways the Internet is used in biotechnology:

Bioinformatics & Online Databases

- **Bioinformatics** is a field that uses the Internet for analyzing biological data, such as DNA sequences and protein structures.
- Scientists use **online databases** to store and access genetic and molecular information.

Examples of Online Databases:

- **NCBI (National Center for Biotechnology Information)** – Provides access to genomic data.
- **GenBank** – Stores DNA sequence information.
- **Protein Data Bank (PDB)** – Stores 3D structures of proteins and molecules.

Benefits:

- Easy access to vast biological data.
- Helps in **drug discovery** and genetic research.

2. Genomic Research & DNA Sequencing

- The Internet allows researchers to share **genomic data** globally.
- **Cloud computing** helps in analyzing large DNA sequences quickly.
- Platforms like **BLAST (Basic Local Alignment Search Tool)** help compare genetic sequences.

Benefits:

- Faster identification of disease-causing genes.
- Helps in **personalized medicine** and gene therapy.

3. Online Collaboration & Research

- Scientists across the world collaborate using the Internet.
- Platforms like **ResearchGate, PubMed, and Google Scholar** provide access to research papers.
- Virtual conferences and online forums allow real-time discussions on biotechnology advancements.

Benefits:

- Faster sharing of discoveries.
- Reduces research duplication.

4. Drug Discovery & Development

- **Pharmaceutical companies** use the Internet to analyze molecular structures and find new drug compounds.
- **AI and cloud computing** help in **drug simulations and virtual screening**.
- **Online clinical trial databases** track drug testing and patient outcomes.

Example Platforms:

- **ClinicalTrials.gov** – Tracks global drug trials.
- **PubChem** – Provides chemical and drug information.

Benefits:

- Speeds up drug discovery.
- Reduces costs through online simulations.

5. Agricultural Biotechnology & GMOs

- The Internet helps farmers and scientists **track genetically modified crops (GMOs)**.
- **Satellite imaging & IoT sensors** provide real-time monitoring of soil health and crop growth.
- **Online genetic databases** help modify crops for higher yield and disease resistance.

Benefits:

- Improves food security.
- Helps in sustainable agriculture.

6. Medical Biotechnology & Telemedicine

- **Internet-based diagnostics** allow for **remote genetic testing** and personalized medicine.
- **Telemedicine platforms** help biotech companies provide genetic counseling and healthcare services online.

Example Applications:

- **23andMe** – Offers at-home DNA testing for health insights.
- **CRISPR-based genetic treatments** are shared via global research networks.

Benefits:

- Provides faster and more accurate diagnoses.
- Enables global access to medical biotechnology.

7. AI & Big Data in Biotechnology

- AI-powered biotech tools use **machine learning** to analyze biological data.
- The Internet helps biotech companies process **large datasets from experiments**.

Example Applications:

- AI helps in **predicting protein structures** (e.g., AlphaFold by DeepMind).
- Big data analytics improve **vaccine development** (e.g., COVID-19 research).

Benefits:

- Accelerates medical research.
- Improves accuracy in biotech experiments.

8. Biotechnology Education & Online Learning

- **E-learning platforms** offer biotech courses, workshops, and certifications.
- Students and researchers can access **biotech simulations and virtual labs**.

Example Platforms:

- **Coursera, edX** – Offer biotechnology courses.
- **Virtual Labs** – Allow students to conduct online biology experiments.

Benefits:

- Makes biotech education more accessible.
- Helps students and professionals upskill.

Introduction to Bioinformatics

- Bioinformatics combines computer science, statistics, mathematics, chemistry, and engineering to analyse, explore, integrate, and execute biological information for research and development purposes.

- Bioinformatics is an interdisciplinary research area at the interface between computer science and biological science.
- Bioinformatics involves the technology that uses computers to store, retrieve, manipulate, and distribute information related to biological macromolecules such as DNA, RNA, and proteins.
- Bioinformatics differs from a related field known as computational biology. Bioinformatics is limited to the sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered computational molecular biology for modelling and simulations.

Brief background

❖ **The origin: 1950-1970**

In the late 1950s, in addition to major advances in the determination of protein structures through crystallography, the first sequence (i.e. amino acid) of a protein, insulin, was published.

❖ **Dayhoff: the first bioinformatician**

• Margaret O. 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'.

• The first database of protein sequences was created by Margaret O. Dayhoff between 1965 and 1978. It was released yearly in a series of volumes called "Atlas of protein sequence and structure."

❖ **Decoding the DNA language:** the genetic code specific nucleotide configurations in the DNA molecule encode the requirements for living organism, more specifically, its "proteins." The RNA sequences, which are transcribed from DNA, dictate the amino acid sequence of the proteins they encode was formalised by Francis Crick in his sequence hypothesis, which is now also known as the "Central Dogma." The three dimensional structure of the protein is then determined from the amino acid sequence.

Main Components of Bioinformatics

Bioinformatics is an interdisciplinary field that combines **biology, computer science, and data analysis** to store, analyze, and interpret biological data. It plays a crucial role in **genomics, proteomics, drug discovery, and personalized medicine**.

1. Biological Databases

Biological databases store **genetic, protein, and biochemical data**, allowing scientists to retrieve and analyze biological information.

Types of Databases:

- **Genomic Databases** – Store DNA sequences. (Example: GenBank, Ensembl)
- **Protein Databases** – Store protein structures. (Example: UniProt, PDB)
- **Metabolic Pathway Databases** – Store biochemical reactions. (Example: KEGG, BioCyc)

Importance:

- Helps in DNA & protein sequence analysis.
- Provides reference data for research.

2. Sequence Analysis

This involves comparing DNA, RNA, or protein sequences to find similarities, mutations, or evolutionary relationships.

Common Tools & Techniques:

- **BLAST (Basic Local Alignment Search Tool)** – Compares sequences with databases.
- **Clustal Omega** – Used for multiple sequence alignments.
- **FASTA** – Helps find similar sequences in different organisms.

Importance:

- Identifies disease-related genes.
- Helps in drug discovery and vaccine development.

3. Structural Bioinformatics

Focuses on predicting and analyzing **3D structures of biological molecules** (proteins, DNA, RNA).

Common Tools:

- **Protein Data Bank (PDB)** – Stores 3D structures.
- **AlphaFold** – AI-based protein structure prediction.
- **PyMOL & Chimera** – Visualization tools for molecular structures.

Importance:

- Helps understand protein functions.
- Useful in designing new drugs and therapies.

4. Functional Bioinformatics

Analyzes **gene functions, protein interactions, and metabolic pathways** to understand biological systems.

Key Techniques & Tools:

- **Gene Ontology (GO)** – Classifies gene functions.
- **STRING Database** – Predicts protein-protein interactions.
- **Pathway Analysis (KEGG, BioCyc)** – Studies metabolic and signaling pathways.

Importance:

- Helps identify drug targets.
- Aids in understanding diseases and genetic disorders.

5. Computational Biology & Algorithms

Uses **mathematical models and algorithms** to analyze biological data, simulate biological processes, and predict outcomes.

Common Algorithms:

- **Hidden Markov Models (HMMs)** – Used in sequence alignment.
- **Machine Learning (AI in Bioinformatics)** – Helps in pattern recognition (e.g., AlphaFold).
- **Phylogenetic Tree Construction (MEGA, PhyML)** – Analyzes evolutionary relationships.

Importance:

- Improves accuracy of biological predictions.
- Helps in personalized medicine and disease modeling.

6. Data Analysis & Bioinformatics Software

Involves **processing and interpreting biological data** using specialized tools and programming languages.

Popular Bioinformatics Software & Languages:

- **R & Python** – Used for statistical and machine learning applications.
- **Bioconductor (R package)** – For analyzing genomic data.
- **Biopython & BioPerl** – Libraries for bioinformatics programming.

- **Importance:**
- Helps manage large biological datasets.
- Supports research in genomics and proteomics.

7. Systems Biology & Network Analysis

Integrates biological data from multiple sources to model entire biological systems and interactions.

Common Approaches:

- **Metabolic & Signaling Pathway Analysis** – Studies how genes and proteins interact.
- **Gene Expression Analysis** – Helps understand how genes are turned on/off in diseases.
- **Network Biology (Cytoscape)** – Analyzes protein-protein and gene interactions.

Importance:

- Helps in developing new therapies and biomarker discovery.
- Provides insights into complex diseases like cancer.

Experimental Tasks in Bioinformatics

Bioinformatics involves various **experimental tasks** that help in the analysis of biological data. These tasks are essential for understanding genetic structures, protein functions, and disease mechanisms.

1. Sequence Analysis

This involves the study of DNA, RNA, and protein sequences to understand their structure, function, and evolution.

Key Tasks:

- **DNA & RNA Sequencing:** Identifying the order of nucleotides (A, T, G, C).
- **Protein Sequencing:** Determining the order of amino acids in a protein.
- **Multiple Sequence Alignment:** Comparing sequences from different organisms to find similarities.

Tools Used:

- **BLAST (Basic Local Alignment Search Tool)** – Compares sequences with databases.
- **Clustal Omega** – Aligns multiple sequences.
- **FASTA** – Finds similar sequences.

Applications:

- Identifying genetic mutations related to diseases.
- Studying evolutionary relationships.

2. Genome Annotation

Genome annotation is the process of identifying **genes, regulatory elements, and other functional regions** in a DNA sequence.

Key Tasks:

- **Gene Prediction:** Identifying coding regions in DNA sequences.
- **Functional Annotation:** Assigning biological functions to genes.
- **Comparative Genomics:** Comparing genomes of different species.

Tools Used:

- **GenBank & Ensembl** – Genome databases.
- **Gene Ontology (GO)** – Provides functional information.
- **NCBI ORF Finder** – Identifies coding sequences.

Applications:

- Helps in identifying disease-related genes.
- Supports drug discovery and genetic engineering.

3. Structural Bioinformatics

Structural bioinformatics focuses on predicting and analyzing the **3D structures of biological molecules**, such as proteins and nucleic acids.

Key Tasks:

- **Protein Structure Prediction:** Determining how proteins fold and function.
- **Molecular Docking:** Studying interactions between drugs and proteins.
- **Protein-Ligand Interaction Analysis:** Understanding drug binding sites.

Tools Used:

- **AlphaFold** – AI-based protein structure prediction.
- **PDB (Protein Data Bank)** – Stores 3D protein structures.
- **PyMOL & Chimera** – Visualization tools for molecular structures.

Applications:

- Helps in drug design and development.
- Improves understanding of protein functions.

4. Phylogenetics and Evolutionary Analysis

This involves studying the **evolutionary relationships** between different species using genetic data.

Key Tasks:

- **Phylogenetic Tree Construction:** Understanding genetic relationships.
- **Molecular Evolution Studies:** Analyzing genetic changes over time.
- **Comparative Genomics:** Comparing entire genomes to find similarities and differences.

Tools Used:

- **MEGA (Molecular Evolutionary Genetics Analysis)** – Constructs phylogenetic trees.
- **PhyML & RAxML** – Used for evolutionary studies.
- **ClustalW** – Performs multiple sequence alignments.

Applications:

- Tracing the origin and evolution of diseases.
- Understanding species evolution.

5. Systems Biology & Network Analysis

Systems biology integrates different types of biological data to understand how genes, proteins, and metabolic pathways work together.

Key Tasks:

- **Metabolic Pathway Analysis:** Studying biochemical reactions in cells.
- **Protein-Protein Interaction (PPI) Analysis:** Understanding how proteins interact.
- **Gene Regulatory Networks:** Mapping gene interactions.

Tools Used:

- **KEGG (Kyoto Encyclopedia of Genes and Genomes)** – Analyzes metabolic pathways.
- **STRING Database** – Studies protein interactions.
- **Cytoscape** – Visualizes biological networks.

Applications:

- Helps in personalized medicine.
- Identifies new drug targets.

6. Functional Genomics & Gene Expression Analysis

Functional genomics studies how genes are expressed and regulated in different biological conditions.

Key Tasks:

- **Microarray Analysis:** Measures gene expression levels.
- **RNA-Seq (Transcriptomics):** Analyzes RNA sequences to study gene expression.
- **Differential Gene Expression Analysis:** Identifies genes activated in diseases.

Tools Used:

- **GEO (Gene Expression Omnibus)** – Stores gene expression data.
- **Bioconductor (R package)** – Analyzes gene expression.
- **DESeq2 & EdgeR** – Used for RNA-Seq data analysis.

Applications:

- Helps in cancer research and personalized medicine.
- Identifies biomarkers for diseases.

7. Drug Discovery and Virtual Screening

Bioinformatics plays a key role in **computer-aided drug design (CADD)** by predicting drug-target interactions.

Key Tasks:

- **Molecular Docking:** Predicting how drugs bind to proteins.
- **Pharmacogenomics:** Studying how genes affect drug response.
- **Drug Target Identification:** Finding molecules for new drug development.

Tools Used:

- **AutoDock & DOCK** – Used for molecular docking.
- **SwissDock** – Predicts drug binding sites.
- **ZINC Database** – Provides drug-like molecules.

Applications:

- Reduces time and cost of drug discovery.
- Helps in designing targeted therapies.

8. Big Data & Artificial Intelligence in Bioinformatics

With the increasing amount of biological data, AI and big data technologies are used to analyze and predict biological outcomes.

Key Tasks:

- **Machine Learning Models for Genomics:** Predicts gene functions.
- **Deep Learning for Protein Structure Prediction:** Improves protein modelling accuracy.
- **Data Mining in Bioinformatics:** Extracts useful patterns from large datasets.

Tools Used:

- **TensorFlow & PyTorch** – Machine learning frameworks.
- **AlphaFold** – AI-based protein structure prediction.
- **Galaxy Platform** – A cloud-based bioinformatics tool.

Applications:

- Improves accuracy in genomics and proteomics research.
- Enhances precision medicine approaches.

Biological databases

- Since biology is becoming a more data-rich field, there is a greater need than ever for robust and consistent massive databases (such as protein and nucleotide sequences, three-dimensional structures from X-ray crystallography, and NMR).
- A biological database is a collection of data that is organized so that it contents can easily be accessed, managed and updated.
- Biological databases play a fundamental role in bioscience particularly in bioinformatics.
- They offer scientists the opportunity to access sequence and structure data for tens of thousands of sequences from a broad range of organisms. Biological databases represent an invaluable resource in support of biological research.
- Biological databases use all three types of database structures: flat files, relational, and object-oriented.

- Despite the obvious drawbacks of using flat files in database management, many biological databases still use this format.
- The justification for this is that this system involves the minimum amount of database design and the search output can be easily understood by working biologists.
- Based on their contents, biological databases can be roughly divided into three categories: primary databases, secondary databases, and specialized databases.
- Primary databases contain original biological data. They are archives of raw sequence or structural data submitted by the scientific community. GenBank and Protein Data Bank (PDB) are examples of primary databases.
- Secondary databases contain computationally processed or manually curated information, based on original information from primary databases. Translated protein sequence databases containing functional annotation belong to this category.
- Examples are SWISS-Prot and Protein Information Resources (PIR) (successor of Margaret Dayhoff's Atlas of Protein Sequence and Structure. Specialized databases are those that cater to a particular research interest. For example, FlyBase, HIV sequence database, and Ribosomal Database Project are databases that specialize in a particular organism or a particular type of data. A list of some frequently used databases is provided in Table.

Primary database

- There are three major public sequence databases that store raw nucleic acids sequence data produced and submitted by researchers worldwide: GenBank, the European Molecular Biology Laboratory (EMBL) database and the DNA Data Bank of Japan (DDBJ), which are all freely available on the Internet.
- Most of the data in the databases are contributed directly by authors with a minimal level of annotation. A small number of sequences, especially those published in the 1980s, were entered manually from published literature by database management.
- Presently, sequence submission to either GenBank, EMBL, or DDBJ is a precondition for publication in most scientific journals to ensure the fundamental molecular data to be made freely available. These three public databases closely collaborate and exchange new data daily. They together constitute the International Nucleotide Sequence Database Collaboration. This means that by connecting to any one of the three databases, one should have access to the same nucleotide sequence data.

- Although the three databases all contain the same sets of raw data, each of the individual databases has a slightly different kind of format to represent the data. Fortunately, for the three-dimensional structures of biological macromolecules, there is only one centralized database, the PDB.
- This database archives macromolecules' atomic coordinates (proteins and nucleic acids) determined by X-ray crystallography and NMR. It uses a flat-file format to represent protein names, authors, experimental details, secondary structure, cofactors, and atomic coordinates. The web interface of PDB also provides viewing tools for simple image manipulation.

Secondary Database

- Sequence annotation information in the primary database is often minimal. To turn the raw sequence information into more sophisticated biological knowledge, much postprocessing of the sequence information is needed. This requires the need for secondary databases, which contain computationally processed sequence information derived from the primary databases.
- The amount of computational processing work varies greatly among the secondary databases; some are simple archives of translated sequence data from identified open reading frames in DNA, whereas others provide additional annotation and information related to a higher level of information regarding structure and functions.
- A prominent example of a secondary database is SWISS-PROT, which provides detailed sequence annotation that includes structure, function, and protein family assignment. The sequence data are mainly derived from TrEMBL, a database of translated nucleic acid sequences stored in the EMBL database.
- The annotation of each entry is carefully curated by human expert sand thus is of good quality. The protein annotation includes function, domain structure, catalytic sites, cofactor binding, posttranslational modification, metabolic pathway information, disease association, and similarity with other sequences.
- Much of this information is obtained from scientific literature and entered by database curators. The annotation provides significant added value to each original sequence record.
- The data record also provides cross-referencing links to other online resources of interest. Other features such as very low redundancy and high level of integration with

other primary and secondary databases make SWISS-PROT very popular among biologists.

- A recent effort to combine SWISS-PROT, TrEMBL, and PIR led to the creation of the UniProt database, which has larger coverage than any of the three databases while at the same time maintaining the original SWISS-PROT feature of low redundancy, cross-references, and a high quality of annotation.
- There are also secondary databases that relate to protein family classification according to functions or structures. The Pfam and Blocks databases contain aligned protein sequence information as well as derived motifs and patterns, which can be used for the classification of protein families and inference of protein functions.
- The DALI database is a protein secondary structure database that is vital for protein structure classification and threading analysis to identify distant evolutionary relationships among proteins.
- Major Biological Databases: AceDB, DDBJ, EMBL, Entrez, ExPASY, FlyBase, FSSP, GenBank, HIVdatabases, Microarray gene expression database, OMIM, PIR, PubMed Ribosomal database project, SRS, SWISS-Prot, TAIR and DNA microarray data.
- Secondary databases, which contain computationally processed sequence information derived from the primary database.

  Example: Swiss-Prot (provides detailed sequence annotation that includes structure, function, and protein family assignment. **SwissProt (**sequence data derived from TrEMBL database, provide detailed information protein function, post translational modifications, secondary, quaternary structures)**, Pfam** (Protein Families)**, ProSite** (protein families, domain and functional sites)**, PRINTS** (uses fingerprints method) and **BLOCKS** (Ungapped multiple alignment of segment of related protein sequences)

Composite database

- It contains the combined information of multiple primary databases
- Example: NrDB (nonredundant database), GenBank, PDB, PIR

Specialized databases

- Specialized databases normally serve as specific research community or focus on a particular organism. The content of these databases may be sequences or other types of information. These sequences in the databases may overlap with a primary database, but may also have new data submitted directly by authors.

- Because they are often curated by experts in the field, they may have unique organizations and additional annotations associated with the sequences. Many genome databases that are taxonomic specific fall within this category.
- Examples include Flybase, WormBase, AceDB, and TAIR. In addition, there are also specialized databases that contain original data derived from functional analysis. For example, GenBank, EST database and Microarray Gene Expression Database at the European Bioinformatics Institute (EBI) are some of the gene expression databases available.

Information retrieval from biological database

- There are several retrieval systems for biological data. The most popular retrieval systems for biological databases are Entrez and Sequence Retrieval Systems (SRS) which provide access to multiple databases for retrieval of integrated search results.
- requires the use of Boolean operators. This is to join a series of keywords using logical terms such as AND, OR, and NOT to indicate relationships between the keywords used in a search.
- AND=means search must contain both words
- OR= means search containing either word or both
- NOT=excludes results containing either one of the words
- Parentheses () =defines a concept if there are multiple words

Entrez

- The NCBI developed and maintains Entrez, a biological database retrieval system.
- It is a gateway that allows text-based searches for a wide variety of data, including annotated genetic sequence information, structural information, as well as citations and abstracts, full papers, and taxonomic data.
- The key feature of Entrez is its ability to integrate information, which comes from cross-referencing between NCBI databases based on preexisting and logical relationships between individual entries.
- This is highly convenient for users do not have to visit multiple databases located in disparate places. For example, in a nucleotide sequence page, one may find cross-referencing links to the translated protein sequence, genome mapping data, or to the related PubMed literature information, and protein structures if available.

- Effective use of Entrez requires an understanding of the main features of the search engine. There are several options common to all NCBI databases that help to narrow the search.
- One option is “Limits,” which helps to restrict the search to a subset of a particular database. It can also be set to restrict a search to a particular database (e.g., the field for author or publication date) or a particular type of data (e.g., chloroplast DNA/RNA).
- Another option is “Preview/Index,” which connects different searches with the Boolean operators and uses a string of logically connected keywords to perform a new search.
- These searches can also be limited to a particular search field (e.g., gene name or accession number). The “History” option provides a record of the previous searches so that the user can review, revise, or combine the results of earlier searches. There is also a “Clipboard” that stores search results for later viewing for a limited time.
- To store information in the Clipboard, the “Send to Clipboard” function should be used. One of the databases accessible from Entrez is a biomedical literature database known as PubMed, which contains abstracts and, in some cases, full text articles from nearly 4,000 journals.
- An important feature of PubMed is the retrieval of information based on medical subject headings (MeSH) terms. The MeSH system consists of a collection of more than 20,000 controlled and standardized vocabulary terms used for indexing articles. In other words, it is a thesaurus that helps convert search keywords into standardized terms to describe a concept. By doing so, it allows “smart” searches in which a group of accepted synonyms are employed so that the user not only gets exact matches but also related matches on the same topic that otherwise might have been missed. Another way to broaden the retrieval is by using the “RelatedArticles” option. PubMed uses a word weight algorithm to identify related articles with similar words in the titles, abstracts, and MeSH. By using this feature, articles on the same topic that were missed in the original search can be retrieved.

GenBank

- GenBank is the most complete collection of annotated nucleic acid sequence data for almost every organism.
- The content includes genomic DNA, mRNA, cDNA, ESTs, high throughput raw sequence data, and sequence polymorphisms. There is also a GenPept database for protein sequences, the majority of which are conceptual translations from DNA

sequences, although a small number of the amino acid sequences are derived using peptide sequencing techniques.

- There are two ways to search for sequences in GenBank. One is using text-based keywords similar to a PubMed search.

GenBank Sequence Format

- GenBank text-based method
- It is a relational database but outcome of file is in flat files
- It contains header, feature and sequence entry.

Abstract Syntax Notation One.

Abstract Syntax Notation One (ASN.1) is a data markup language with a structure specifically designed for accessing relational databases. It describes sequences with each item of information in a sequence record separated by tags so that each sub-portion of the sequence record can be easily added to relational tables and later extracted. Though more difficult for people to read, this format makes it easy for computers to filter and parse the data. This format also facilitates the transmission and integration of data between databases.

Conversion of Sequence Formats

- In sequence analysis and phylogenetic analysis, there is a frequent need to convert between sequence formats. One of the most popular computer programs for sequence format conversion is Readseq, written by Don Gilbert at Indiana University.

It recognizes sequences in almost any format and writes a new file in an alternative format.

SRS (Sequence retrieval system) is a retrieval system maintained by the EBI, which is comparable to NCBI Entrez.

- It is not as integrated as Entrez, but allows the user to query multiple databases simultaneously, another good example of database integration.
- It also offers direct access to certain sequence analysis applications such as sequence similarity searching and Clustal sequence alignment.
- Queries can be launched using “Quick Text Search” with only one query box in which to enter information. There are also more elaborate submission forms, the “Standard Query Form” and the “Extended Query Form.”
- The standard form allows four criteria (fields) to be used, which are linked by Boolean operators. The extended form allows many more diversified criteria and fields to be used.

These search results contain the query sequence and sequence annotation as well as links to literature, metabolic pathways, and other biological databases.

BIOLOGICAL SEQUENCE ANALYSIS

Since the sequencing of bacteriophage ØX-174 in 1977, the nucleotide sequences of thousands of organisms have constantly been decoded and thereby getting stored in different biological databases. The nucleotide sequences of the organisms are being analyzed by various scientific groups to decode the underlying information. This is called genome annotations. The most common annotations involve:

- identification of protein-coding genes
- identifications and analyses of genes that lead to RNA biogenesis
- detection of the presence of regulatory sequences, structural motifs and repetitive sequences in the genes

The nucleotide information of genes within a species or between different species can be compared to identify the similarity between the gene products, viz., the proteins, RNAs or the nucleotides themselves. The databases contain large amounts of data. The genome information of an organism is also very huge. With such large chunks of data, it is simply impractical to analyze biological sequences manually, Thus, Bioinformatics tools are routinely used to analyze the sequence information. The most widely used tool for genome analyses is BLAST. The method used for this purpose is called pair-wise sequence alignment. However, there are other tools available that provide different sets of results. A common practice in genome analyses is to compare the characteristics of different genomes. For such purposes, the technique that is used is called the multiple sequence alignment.

Pair-wise Sequence Alignment

Dayhoff *et al.*, in 1978 studied the sequence alignments among 34 superfamilies of protein sequences and listed the accepted Point mutations i.e. replacement of amino acid by another residue by natural selection. Based on accepted mutation and probabilities of occurrence of each amino acid they generated a mutation probability matrix known as PAM. PAM1 matrix was generated for closely related protein sequences having at least 85% sequence identities. Later the PAM1 matrix is multiplied by itself several times to generate several other PAM matrices. One of such derived matrices. the PAM250 matrix is used for sequence alignment of proteins with 20% amino acid sequence identities. In other words, the PAM250 matrix is used for sequences comparison of evolutionarily distantly related proteins. PAM mutation probability matrix is then converted into a scoring matrix called log-odds matrix or relatedness odds matrix. Another log-odds matrix developed by Henikoff and Henikoff (1992, 1996) is called BLOcks SUbstitution Matrix (BLOSUM) using the BLOCKS database which consists of more than 500 groups of local alignments of distantly related protein sequences. High value of BLOSUM (BLOSUM90) and low-value PAM matrices (PAM30) are used for conserved protein sequences whereas low BLOSUM and high PAM numbers are used for distantly related proteins BLOSUM62 matrix merged 62% or more identical sequences into one alignment; thus, proteins having less than 62% sequence identities are analysed by this BLOSUM62 matrix and it is the default scoring matrix used by most BLAST algorithms for searching sequence similarities between sequences. Two types of pair-wise sequence alignments are:

(i) Global alignment technique: It is derived from the Needleman- Wunsch algorithm (1970), which is based on dynamic programming which attempts to align every residue in every query sequence. It is most useful when the sequences in the query set are similar and of roughly equal length.

(ii) Local alignment technique: It is more useful for dissimilar sequences that are suspected to contain regions of similarity or similar sequence motifs within their larger sequence contexts- The Smith-Waterman algorithm (1981) is a general local alignment method which is also based on dynamic programming algorithm, for scoring the matrix and it uses trace-back procedure to obtain the optimal alignment.

Dynamic programming

Alignments of amino acid sequences of proteins use a substitution matrix to assign scores to amino acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other. The nucleotide sequences of DNA and RNA may be used scoring the matrix. In practices, a simple scoring scheme is generally used to score the sequence alignments, a positive score is used for a match, a less positive or a negative score for a mismatch and a negative score for a gap penalty. Though the local alignment is accurate but it is relatively slow when pairwise local alignment is used for a query sequence against the database. The computational running space and time then become a significant issue. Two other popular rapid heuristic versions of local alignment algorithms have been developed based on identifying words or k-tuples. Such processes would involve 1/2 residues for prior and up to 6 bases for DNA searching. The locally aligned regions a called high-scoring segment pairs or HSPs. The expect (E) value is the measure of statistical significance which represents the number of hits one can expect to obtain by chance; that means the lower the E-value is the measure more significant is the alignment. The E-value decreases exponentially with a high score (S) value corresponding to better alignments extension value.

Word methods: The heuristic methods are significantly more efficient than dynamic programming. These methods are especially useful in large-scale database searches and are best known for their implementation in the database search tools in a number of web portals, such as EMBL FASTA and NCBI BLAST. NCBI BLAST program is a collection of different tools like Blast and PSI-BLAST.

BLASTp: This program compares a protein query sequence against a protein sequence database. In this software program the default matrix is BLOSUM62 with a word size of 6, gap existence of 11 and extension value 1.

PSI-BLAST: This BLAST program is a specialized, advanced and more sensitive position-specific BLAST search which is used for distantly related proteins sharing the same conserved three-dimensional structures but low sequence identities. An initial BLAST search is used to perform a multiple-sequence alignment. By analysing this alignment, it then creates a specialized position-specific scoring matrix which again is used as query iterative. The original query is used as the template for generating the PSSM profile. The original query is used as the template for generating the PSSM profile.

Multiple Sequence Alignment

The alignments of more than two protein or DNA sequences are very useful for studying the homologous group and they provide structure-function and evolution of the protein or gene families. It is more sensitive than pairwise alignment to detect homologs, to find conserved domains or motifs or consensus regions The most commonly used algorithm for multiple sequence alignment is derived from Feng and Doolittle's (1987 1990) progressive alignment method. The basic strategy of this method is calculating pairwise alignment scores between all sequences and then align two closely related sequences followed by addition of sequences progressively to the alignment. The most popular web version of the multiple sequence alignment tool is Clustal W program. To generate a guide tree, this program uses mostly distance matrix instead of similarity matrix The two main features of a guide tree are its branching order or topology and branching length which is proportional to evolutionary distance. The two main methods of tree construction are: (a) Unweighted pair group method of arithmetic averages (UPGMA) and (b) Neighbour-joining method. The multiple sequence alignment output arranges the sequences as presented in the guide tree, *i.e.*, two closely related sequences create a pairwise alignment and then sequences are added one by one. The newer web version of multiple sequence alignment is Clustal Omega which uses seeded guide trees and HMM profile-profile techniques to generate alignments.

Types of BLAST

The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between protein or nucleotide sequences. The program compares nucleotide or protein sequences to a database sequence and calculates the matches' statistical significance. BLAST refers to a suite of programs used to generate alignments between a nucleotide or protein sequence, referred to as a “query” and nucleotide or protein sequences within a database, referred to as “subject” sequences. The original BLAST program used a protein “query” sequence to scan a protein sequence database. A version operating on nucleotide query” sequences and a nucleotide sequence database soon followed.

The introduction of an intermediate layer in which nucleotide sequences are translated into their corresponding protein sequences according to a specified genetic code allows cross-comparisons between nucleotide and protein sequences. Specialized variants of BLAST allow fast searches of nucleotide databases with very large query sequences, or the generation of alignments between a single pair of sequences. Both the standalone and web version of BLAST are available from the National Center for Biotechnology Information. The web version provides searches of the complete genomes of *Homo sapiens* as well as those of many model organisms, including mouse, rat, fruit fly, and *Arabidopsis thaliana*, allowing BLAST alignments to be seen in a full genomic context.

Query and Database Sequence Formats

BLAST “query” sequences are given as character strings of single-letter nucleotide or amino acid codes, preceded by a definition line, beginning with a “>” symbol and containing identifiers and descriptive information. This format is known as FASTA. BLAST databases are constructed from concatenated FASTA formatted sequences using a program called “format” that produces a mixture of binary- and ASCII-encoded files containing the sequences and indexing information used during the BLAST search.

Scoring of Alignments and Substitution Matrices

A BLAST alignment consists of a pair of sequences, in which every letter in one sequence is paired with, or “aligned to,” exactly one letter or a gap in the other. The alignment score is computed by assigning a value to each aligned pair of letters and then summing these values over the length of the alignment. For protein sequence alignments, scores for every possible amino acid letter pair are given in a “substitution matrix” where likely substitutions have positive values and unlikely substitutions have negative values. By default, BLAST uses the “BLOSUM62” matrix, a member of the most commonly used series of substitution matrix, however, several members of the PAM series are also available. For nucleotide alignments, BLAST uses a reward of +2 for aligned pairs of identical letters and a penalty of −3 for each non-identical aligned pair. The creation of a gap in an alignment results in a negative “gap-creation” penalty, with each extension of a preexisting gap incurring a lesser penalty. For a detailed treatment of the theory of alignment scoring.

Overview of the Algorithm

BLAST begins a search by indexing all character strings of a certain length within the “query” by their starting position in the query. The length of the string to index, called the “word size” is configurable by the user. The allowable range for the “word size” varies according to the BLAST program used; typical values are 3 for protein-to-protein sequence searches and 11 for nucleotide-to-nucleotide searches. BLAST then scans the database looking for matches between the “words” indexed in the “query” and strings found within the database sequences. For nucleotide-to-nucleotide searches, these matches must be exact; for protein-to-protein searches, the score of the match as determined using a substitution matrix, must exceed a specified threshold. When a word match is found, two nearby words in the case of protein searches, BLAST attempts to extend both forward and backward from the match to produce an alignment. BLAST will continue this extension as long as the alignment score continues to increase or until it drops by a critical amount owing to the negative scores given by mismatches. This critical amount is known as the "drop off."

Statistical Significance

The alignments found by BLAST during a search are scored, as previously described, and assigned a statistical value, called the "Expect Value." The "Expect Value" is the number of times that an alignment as good or better than that found by BLAST would be expected to occur by chance, given the size of the database searched. An "Expect Value" threshold, set by the user, determines which alignments will be reported. A higher "Expect Value" threshold is less stringent and the BLAST default of "10" is designed to ensure that no biologically significant alignment is missed. However, "Expect Values" in the range of 0.001 to 0.0000001 are commonly used to restrict the alignments shown to those of high quality.

Nucleotide BLAST

- Nucleotide BLAST refers to the use of a member of the BLAST suite of programs, such as "blastn" to search with a nucleotide "query" against a database of nucleotide "subject" sequences.

Protein BLAST

- Protein-to-protein sequence searches are performed using the original member of the BLAST suite of programs, known as "blastp."

BLASTx

- Compares a nucleotide query sequence that is translated to a protein with a protein database.

Translated BLAST

- Translated BLAST (tBLASTx) searches use a genetic code to translate either the “query,” database “subjects,” or both, into protein sequences, which are then aligned as in “blastp.” The translations are performed in the three forward as well as the three reverse reading frames so that no possible translation is missed.

Genome BLAST

- Genome BLAST refers to the application of any of the BLAST search programs to the complete genomic sequence of an organism or the transcript and protein sequences derived from its annotation.

BLAST2Sequences

- BLAST2Sequences is used to compare two sequences, protein or nucleotide, using any one of the principal BLAST variants, “blastp,” “blastn,” “tblastn,” “blastx,” “tblastx,” or MegaBLAST.

Drug Designing

Drug designing is the process of discovering and developing new medications using **bioinformatics, computational tools, and molecular modeling techniques**. It aims to create **effective, safe, and targeted drugs** for treating diseases.

There are two main approaches:

1. **Structure-Based Drug Design (SBDD)** – Uses the 3D structure of a biological target (e.g., proteins, enzymes) to design drugs.
2. **Ligand-Based Drug Design (LBDD)** – Uses the structure of known active compounds to develop new drugs.

Stages in Drug Designing

1. Target Identification

The first step is identifying a **biological target**, such as a **protein, enzyme, or receptor**, that plays a role in disease.

Key Tasks:

- Identifying disease-related proteins (e.g., cancer, viruses).
- Studying **protein-protein interactions**.
- Analyzing genes associated with diseases.

Tools Used:

- **NCBI & UniProt** – Find protein and gene sequences.
- **PDB (Protein Data Bank)** – Provides 3D structures of proteins.
- **STRING Database** – Analyzes protein interactions.

Importance:

- Helps in selecting the right drug target.
- Reduces unwanted side effects.

2. Lead Identification & Screening

A **lead compound** is a molecule that interacts with the biological target and has the potential to become a drug.

Key Tasks:

- Identifying **small molecules** that bind to the target.
- Screening chemical libraries for potential drugs.
- Using **virtual screening** to test millions of molecules.

Tools Used:

- **ZINC Database** – Provides drug-like molecules.
- **PubChem** – Contains chemical and bioactivity data.
- **SwissADME** – Predicts drug properties (solubility, absorption, etc.).

Importance:

- Helps in selecting promising drug candidates.
- Speeds up drug discovery using computational methods.

3. Molecular Docking

Molecular docking simulates how a **drug molecule (ligand)** binds to a **target protein (receptor)** to form a stable interaction.

Key Tasks:

- Predicting the **binding affinity** of drug molecules.
- Finding the best **drug-receptor interaction**.
- Ranking molecules based on docking scores.

Tools Used:

- **AutoDock & AutoDock Vina** – Perform molecular docking.
- **SwissDock** – Predicts drug binding sites.
- **PyMOL & Chimera** – Visualizes docking results.

Importance:

- Helps design effective drugs.
- Reduces experimental costs by predicting interactions before lab testing.

4. ADMET Prediction (Absorption, Distribution, Metabolism, Excretion, Toxicity)

Before testing in humans, drugs must be evaluated for **safety and effectiveness**.

Key Tasks:

- Predicting if the drug is **absorbed** in the body.
- Checking if it reaches the **target organ**.
- Analyzing **metabolism and toxicity risks**.

Tools Used:

- **SwissADME & ADMET Predictor** – Evaluate drug properties.
- **pkCSM & PreADMET** – Predict drug toxicity.

Importance:

- Reduces failure rates in clinical trials.
- Ensures drugs are safe for human use.

5. Molecular Dynamics Simulation

Molecular Dynamics (MD) simulation is a computational technique used to study the movement and interactions of atoms and molecules over time. It is widely used in **drug discovery, biomolecular research, and materials science** to understand molecular behavior, predict drug-target interactions, and optimize drug candidates. Simulates how a drug interacts with its target **over time** to check its **stability** in a biological environment.

How MD Simulation Works

1. **Initialization** – Define the molecular system (e.g., protein-ligand complex, lipid membrane).
2. **Force Field Application** – Apply classical physics-based force fields (e.g., AMBER, CHARMM, GROMOS) to model atomic interactions.
3. **Energy Minimization** – Optimize the structure to remove steric clashes.
4. **Equilibration** – Simulate at physiological conditions (temperature, pressure, solvent).
5. **Production Run** – Simulate the molecular movements over nanoseconds to microseconds.
6. **Trajectory Analysis** – Extract information on **binding affinity, conformational changes, and stability**.

Key Applications of MD Simulation

1. Drug Discovery & Development

Protein-Ligand Binding Studies – Predict how drugs interact with target proteins. **Binding Free Energy Calculations** – Estimate drug affinity using MM-PBSA/MM-GBSA methods.

Virtual Screening – Filter drug candidates by simulating their binding stability.

2. Protein Dynamics & Stability

Understanding Protein Folding – Study conformational changes in disease-related proteins. **Mutational Effects** – Predict how genetic mutations alter protein function.

3. Biomolecular Interactions

Membrane Permeability Studies – Simulate how drugs cross lipid bilayers. **Enzyme Mechanisms** – Model substrate binding and reaction pathways.

Key Tasks:

- Checking if the drug remains bound to the protein.
- Analyzing how the complex behaves under different conditions.

Tools Used:

- **GROMACS** – Performs molecular dynamics simulations.
- **AMBER & CHARMM** – Simulate biomolecular interactions.

Importance:

- Ensures drug effectiveness in real-world conditions.
- Helps refine drug designs.

6. In Vitro & In Vivo Testing (Laboratory & Animal Studies)

In drug discovery, **in vitro** (laboratory-based) and **in vivo** (animal-based) testing play critical roles in evaluating drug safety, efficacy, and pharmacokinetics before clinical trials in humans. These preclinical studies help determine whether a drug candidate is suitable for further development. Before human trials, drugs are tested in **cells (in vitro) and animals (in vivo)** to confirm their biological effects.

1. In Vitro Testing (Laboratory Studies)

In vitro testing refers to **experiments conducted outside a living organism**, typically in test tubes, petri dishes, or cell culture models.

Key Applications of In Vitro Studies

Drug Screening – Identifies potential drug candidates based on biological activity. **Mechanism of Action Studies** – Understands how a drug interacts with its target (e.g., enzymes, receptors). **Toxicity Testing** – Assesses cytotoxicity, genotoxicity, and hepatotoxicity. **ADME Profiling** – Evaluates Absorption, Distribution, Metabolism, and Excretion properties.

Common In Vitro Models

- **Cell Culture Models** – Human and animal cells grown in controlled environments.
- **Organoids & 3D Cultures** – Miniaturized organ-like structures for better biological relevance.
- **High-Throughput Screening (HTS)** – Automated screening of large compound libraries.
- **Microfluidic "Organ-on-a-Chip" Systems** – Mimic organ functions for more accurate drug testing.

Key Tasks:

- Testing drug **activity and toxicity** in lab settings.
- Studying the drug's **effectiveness in animal models**.

Importance:

- Helps in selecting safe and effective drugs.
- Determines the right dosage for human trials.

7. Clinical Trials & FDA Approval

If a drug passes preclinical testing, it moves to **clinical trials** (human testing) in three phases:

- **Phase 1:** Tests safety on a small group.
- **Phase 2:** Tests effectiveness on patients.
- **Phase 3:** Large-scale testing before FDA approval.

Importance:

- Ensures the drug is safe and effective for humans.

- Determines the correct dosage for treatment.

Applications of Drug Designing

- **Cancer Treatment** – Designing targeted therapies for tumors.
- **COVID-19 & Viral Diseases** – Identifying antiviral drugs (e.g., Remdesivir).
- **Neurological Disorders** – Developing drugs for Alzheimer’s and Parkinson’s.
- **Personalized Medicine** – Tailoring treatments based on genetic profiles.

Drug Delivery & Targeting in Bioinformatics

What is Drug Delivery?

Drug delivery refers to the process of transporting a pharmaceutical compound to its **intended site of action** in the body. It aims to enhance **efficacy, minimize side effects, and optimize absorption**.

There are two main types:

1. **Conventional Drug Delivery** – Pills, injections, and topical applications.
2. **Targeted Drug Delivery** – Directs drugs to specific tissues or cells, reducing toxicity.

Types of Drug Delivery Systems

1. Conventional Drug Delivery

Drugs are administered **without targeting** a specific site, which can lead to **side effects and low efficiency**.

Common Methods:

- **Oral Delivery** – Tablets, capsules (e.g., painkillers, antibiotics).
- **Injection (Parenteral Delivery)** – Intravenous (IV), intramuscular (IM).
- **Topical & Transdermal** – Creams, patches.

2. Targeted Drug Delivery

Drugs are **delivered directly to diseased cells** to increase effectiveness and reduce side effects.

Key Features:

- **Specificity** – Drug only affects the target area.
- **Controlled Release** – Drug is released gradually for prolonged effect.
- **Reduced Toxicity** – Minimal impact on healthy tissues.

Advantages:

- **Higher efficacy** – More drug reaches the affected area.
- **Lower dosage needed** – Reduces side effects.
- **Extended drug lifespan** – Slow and controlled release.

Types of Targeted Drug Delivery Systems

1. Nanoparticle-Based Drug Delivery

Nanoparticle-based drug delivery is a cutting-edge approach that enhances the precision, efficiency, and safety of therapeutic agents. By using nanoparticles (NPs) as carriers, drugs can be delivered directly to diseased cells, reducing side effects and improving therapeutic outcomes. This method is widely applied in cancer therapy, infectious diseases, neurodegenerative disorders, and gene therapy. Uses **nanoparticles (NPs)** to transport drugs **directly to cells or tissues**.

Types of Nanoparticles Used in Drug Delivery

1. Lipid-Based Nanoparticles

- **Lipid Nanoparticles (LNPs)** – Used for mRNA delivery (e.g., COVID-19 vaccines).
- **Liposomes** – Biocompatible carriers that enhance drug solubility and prolong circulation time.
- **Solid Lipid Nanoparticles (SLNs)** – Provide controlled drug release and stability.

2. Polymer-Based Nanoparticles

- **Polymeric Micelles** – Used for delivering hydrophobic drugs.
- **Dendrimers** – Branched polymers that allow precise drug loading.
- **Polylactic-co-glycolic Acid (PLGA) Nanoparticles** – Biodegradable and FDA-approved for drug delivery.

3. Inorganic Nanoparticles

- **Gold Nanoparticles (AuNPs)** – Used in cancer imaging and photothermal therapy.
- **Silica Nanoparticles** – Serve as carriers for targeted drug delivery.
- **Iron Oxide Nanoparticles** – Used in MRI-guided drug delivery and hyperthermia treatment.

4. Carbon-Based Nanoparticles

- **Carbon Nanotubes (CNTs)** – Allow for high drug loading but require surface modifications for safety.
- **Graphene Oxide (GO)** – Offers efficient drug binding and targeted release.

Types of Nanoparticles:

- **Liposomes** – Fat-based carriers (e.g., Doxil for cancer therapy).
- **Polymeric Nanoparticles** – Biodegradable carriers.
- **Gold Nanoparticles** – Used in cancer therapy.

Applications:

- Cancer treatment (chemotherapy).
- Drug delivery for brain disorders (Alzheimer’s, Parkinson’s).

2. Liposome-Based Drug Delivery

Liposome-based drug delivery is an advanced nanotechnology approach that enhances the therapeutic efficacy and safety of drugs by encapsulating them in biocompatible lipid vesicles. Liposomes improve drug solubility, stability, targeted delivery, and controlled release, making them valuable in treating cancer, infectious diseases, neurological disorders, and gene therapy applications. Liposomes are **small, fat-based vesicles** that encapsulate drugs for targeted delivery.

What Are Liposomes?

Liposomes are spherical vesicles composed of one or more phospholipid bilayers surrounding an aqueous core. They mimic natural cell membranes, making them highly biocompatible and effective for drug delivery.

Types of Liposomes

1. **Conventional Liposomes** – Basic lipid bilayer vesicles for drug encapsulation.
2. **PEGylated Liposomes (Stealth Liposomes)** – Coated with polyethylene glycol (PEG) to extend circulation time and evade immune detection.
3. **Ligand-Targeted Liposomes** – Functionalized with antibodies or peptides for active targeting.
4. **pH-Sensitive Liposomes** – Release drugs in response to acidic tumour environments.
5. **Cationic Liposomes** – Positively charged liposomes for gene and nucleic acid delivery.

Mechanisms of Liposomal Drug Delivery

1. **Passive Targeting (EPR Effect)**
   - Liposomes accumulate in tumour tissues due to leaky blood vessels (Enhanced Permeability and Retention effect).
2. **Active Targeting**
   - Liposomes are modified with targeting ligands (e.g., antibodies, peptides) for specific cell binding.
3. **Trigger-Responsive Release**
   - Liposomes release drugs in response to stimuli like pH changes, heat, or enzymes.

◆ **Examples:**

- **Doxil (doxorubicin liposome)** – Used for cancer treatment.
- **AmBisome (amphotericin B liposome)** – Treats fungal infections.

Advantages:

- **Reduces toxicity** – Protects healthy cells.
- **Improves solubility** – Useful for water-insoluble drugs.

3. Antibody-Drug Conjugates (ADCs)

Antibody-Drug Conjugates (ADCs) are a revolutionary class of targeted biopharmaceuticals that combine the specificity of monoclonal antibodies (mAbs) with the potency of cytotoxic drugs. ADCs enable precise drug delivery to cancer cells, minimizing damage to healthy tissues and reducing systemic toxicity.

Structure of ADCs

ADCs consist of three key components:

1. **Monoclonal Antibody (mAb)**
   - Specifically binds to antigens on target cells (e.g., HER2, CD30).
   - Ensures selective delivery to diseased tissues.
2. **Linker**
   - Connects the antibody to the cytotoxic payload.
   - Can be **cleavable** (activated inside target cells) or **non-cleavable** (stable until ADC degradation).
3. **Cytotoxic Payload (Warhead)**
   - Highly potent drugs (e.g., auristatins, maytansinoids) that induce apoptosis.

ADCs use **antibodies** to **target cancer cells** while minimizing effects on healthy tissues.

Mechanism of Action

1. **Target Recognition** – The ADC binds to the specific antigen on the target cell.
2. **Internalization** – The ADC-antigen complex is internalized via endocytosis.
3. **Payload Release** – The linker is cleaved inside the lysosome, releasing the cytotoxic drug.
4. **Cell Death** – The potent drug disrupts DNA or microtubules, triggering apoptosis.

Examples:

- **Trastuzumab emtansine (Kadcyla)** – Treats breast cancer.
- **Brentuximab vedotin (Adcetris)** – Used for Hodgkin's lymphoma.

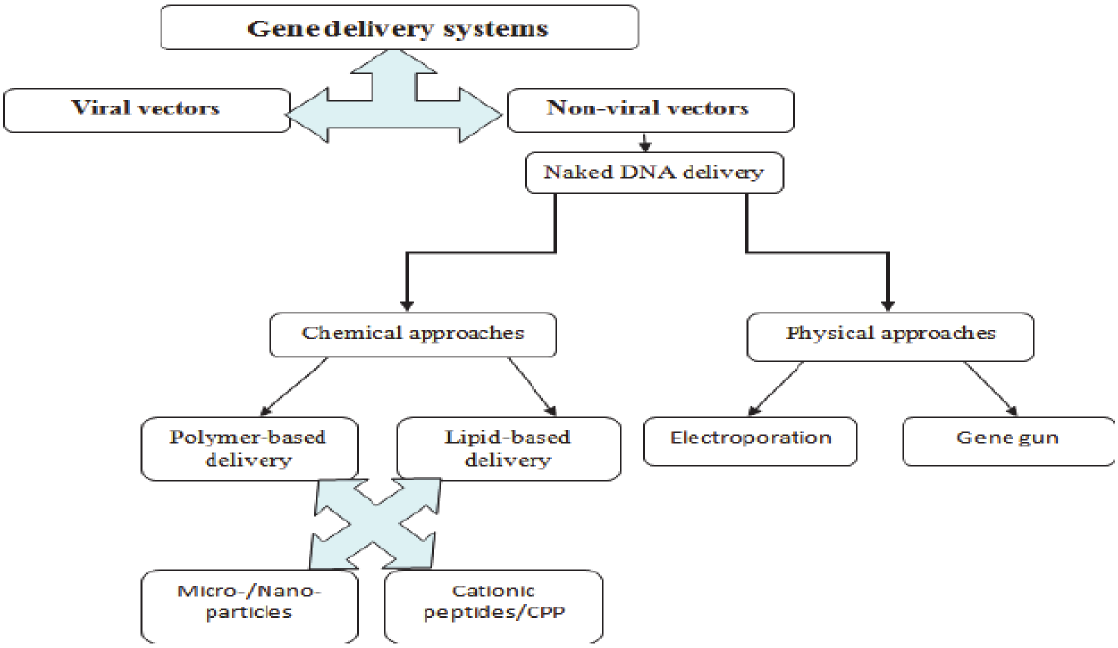
Advantages:

- **Highly specific** – Only binds to diseased cells.
- **Potent effect** – Delivers toxic drugs only to cancer cells.

4. Gene Therapy-Based Drug Delivery

Gene therapy uses **DNA or RNA-based drugs** to modify disease-causing genes. Gene therapy-based drug delivery is an advanced biomedical approach that uses genetic material to treat or prevent diseases. This method delivers therapeutic genes into a patient's cells to correct genetic defects, enhance cellular functions, or modulate disease pathways. It is a promising strategy for treating genetic disorders, cancers, and chronic diseases such as cardiovascular and neurodegenerative conditions.

Methods:



Viral Vectors:

Modified viruses deliver genetic material (e.g., CRISPR gene editing). These vectors use modified viruses to transport therapeutic genes into target cells:

- **Adeno-Associated Virus (AAV)** – High safety profile, low immunogenicity, but limited carrying capacity.
- **Lentivirus** – Can integrate into the genome for long-term expression, used in disorders like sickle cell disease.
- **Adenovirus** – High transduction efficiency but can trigger immune responses.
- **Retrovirus** – Integrates into the host genome but poses a risk of insertional mutagenesis.

Non-Viral Vectors

Liposomes or nanoparticles for safe gene delivery. These synthetic carriers offer safer alternatives:

- **Lipid Nanoparticles (LNPs)** – Used in mRNA therapies (e.g., COVID-19 vaccines).
- **Polymeric Nanoparticles** – Offer controlled drug release and targeted delivery.
- **Electroporation** – Uses electric pulses to introduce genes into cells.
- **CRISPR-Based Delivery** – Delivers CRISPR components for precise genome editing.

Applications of Gene Therapy-Based Drug Delivery

- **Cancer Treatment** – CAR-T cell therapy and oncolytic viruses.
- **Genetic Disorders** – Sickle cell disease, cystic fibrosis, muscular dystrophy.
- **Neurological Diseases** – Gene therapy for Parkinson's and Alzheimer's.
- **Ophthalmic Diseases** – Luxturna (FDA-approved for inherited retinal dystrophy).
- **Rare Diseases** – Zolgensma for spinal muscular atrophy (SMA).

Applications:

- **Cystic fibrosis** – Corrects genetic mutations.
- **Cancer immunotherapy** – Activates the immune system.

5. Smart Drug Delivery Systems

These use **biosensors** to release drugs **only when needed** (e.g., pH-sensitive, temperature-sensitive carriers).

Examples:

- **Glucose-responsive insulin** – Releases insulin only when blood sugar is high.
- **Magnetically controlled nanoparticles** – Guide drugs to the right location.

Advantages:

- **Reduces side effects** – Only activates in diseased areas.
- **Personalized medicine** – Adjusts treatment based on patient needs.

Bioinformatics in Drug Delivery & Targeting

1. Computational Drug Design for Targeting

Computational drug design is a cutting-edge approach that leverages computational methods to identify, design, and optimize drugs that specifically target disease-associated biomolecules. It accelerates drug discovery by simulating molecular interactions, predicting drug efficacy, and reducing the time and cost associated with traditional experimental methods.

Key Approaches in Computational Drug Design

1. Structure-Based Drug Design (SBDD)

This method relies on the 3D structure of a target protein or biomolecule to design small molecules that bind effectively to it.

- **Molecular Docking** – Simulates how a drug molecule fits into the binding site of a target protein.
- **Molecular Dynamics (MD) Simulations** – Studies the stability and interactions of drug-protein complexes over time.
- **Fragment-Based Drug Design (FBDD)** – Uses small molecular fragments to build potent drug candidates.
- **Virtual Screening** – Screens large libraries of compounds computationally to identify potential drug leads.

2. Ligand-Based Drug Design (LBDD)

When the 3D structure of the target is unknown, LBDD uses known active molecules to design new drugs.

- **Quantitative Structure-Activity Relationship (QSAR)** – Predicts the activity of new compounds based on chemical properties.
- **Pharmacophore Modeling** – Identifies key molecular features essential for drug-target interactions.

3. AI and Machine Learning in Drug Design

- **Deep Learning** – Predicts binding affinity, drug-likeness, and toxicity.
- **Generative Models** – Designs novel drug-like molecules using AI.
- **Predictive Analytics** – Identifies drug repurposing opportunities and adverse effects.

Targeted Drug Design for Specific Diseases

Cancer Therapy

- **Kinase Inhibitors (e.g., Imatinib, Erlotinib)** – Target cancer-driving proteins.
- **Immune Checkpoint Inhibitors (e.g., PD-1/PD-L1 inhibitors)** – Enhance immune response against tumours.

Neurological Disorders

- **β-Secretase Inhibitors for Alzheimer's Disease** – Prevent amyloid plaque formation.
- **Dopamine Agonists for Parkinson's Disease** – Improve dopamine receptor activity.

Infectious Diseases

- **Antiviral Drugs (e.g., Remdesivir for COVID-19)** – Target viral replication enzymes.
- **Antibiotic Resistance Modulation** – Computational methods identify new bacterial enzyme inhibitors.

Challenges and Future Directions

- **Drug Resistance** – Pathogens and cancers evolve resistance mechanisms.
- **Target Identification** – Not all diseases have well-defined drug targets.
- **Off-Target Effects** – Computational predictions need experimental validation to ensure safety.
- **Quantum Computing** – Future applications could enhance molecular simulations for precision drug design.

Bioinformatics Tools Used:

- **SwissDock & AutoDock** – Molecular docking for drug-target interaction.
- **ZINC Database** – Provides drug-like molecules.
- **PharmGKB** – Predicts drug response based on genetics.

Applications:

- **Identifies the best drug-target matches**.

- **Optimizes drug formulation** before lab testing.

2. AI & Machine Learning in Drug Targeting

Artificial Intelligence (AI) helps design **personalized drug delivery** based on patient genetics.

AI Tools:

- **DeepChem** – Predicts drug interactions.
- **AlphaFold** – Analyzes protein-drug binding.

Applications:

- Predicts how well a drug will work**.**
- Speeds up drug development**.**

3. Big Data & Genomics in Drug Delivery

Data Sources:

- **GenBank & Ensembl** – Provide genetic data for personalized medicine.
- **Cancer Genome Atlas (TCGA)** – Helps design cancer-targeting drugs.

Applications:

- Tailors treatments for individual patients**.**
- Identifies disease-specific drug targets**.**

Applications of Targeted Drug Delivery

- **Cancer Therapy** – Nanoparticles deliver chemotherapy to tumors.
- **Diabetes Treatment** – Smart insulin delivery for blood sugar control.
- **Brain Disorders** – Nanocarriers bypass the blood-brain barrier.
- **Personalized Medicine** – Gene-based drug delivery for rare diseases.