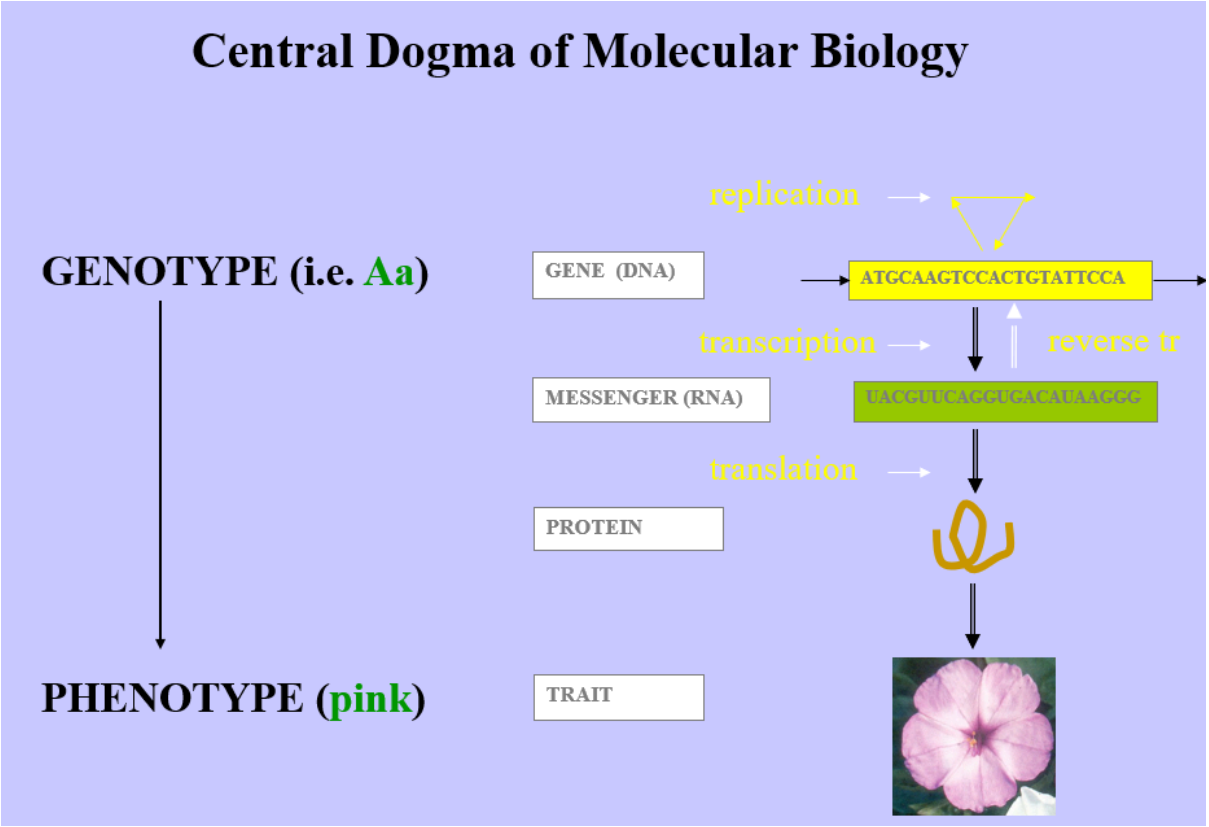


Computational Goals of Bioinformatics

Learn & Generalize: Discover conserved patterns (models) of sequences, structures, metabolism & chemistries from well-studied examples.

Prediction: Infer function or structure of newly sequenced genes, genomes, proteomes or proteins from these generalizations.

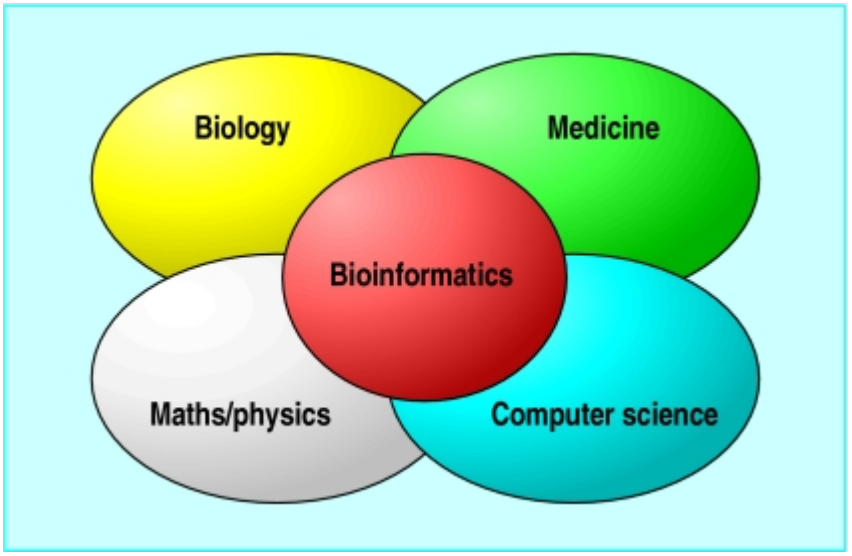
Organize & Integrate: Develop a systematic and genomic approach to molecular interactions, metabolism, cell signaling, gene expression…

Basis of systems biology • Simulate: Model gene expression, gene regulation, protein folding, protein-protein interaction, protein-ligand binding, catalytic function, metabolism…

Goal of systems biology. • Engineer: Construct novel organisms or novel functions or novel regulation of gene

Target: Mutations, RNAi to specific genes and transcripts or drugs to specific protein targets. Practical biological and medical use of bioinformatics.

Bioinformatics, an interdisciplinary field combining computer science, mathematics, physics, and biology, helps analyze and interpret this data. It plays a crucial role in managing biological and medical information.



- Bioinformatics applies computational tools to interpret biological data.
- It is essential for modern biology and medicine.
- Tools like **BLAST** and Ensembl rely on internet access for data analysis.
- A major achievement is genome sequence analysis, especially of the human genome.
- Future developments could improve understanding of the genome, aiding drug discovery and personalized treatments.

With the **massive influx of genome data, computer databases are crucial for organizing and analyzing biological information.** Public and private databases store genetic data, some available for free while others require subscriptions. Bioinformatics is a rapidly growing field, essential for modern biology, medicine, and drug discovery.

🌱 Unlocking Biology's Big Data Revolution

📈 From High-Tech to High-Impact

Biology has entered the **big data era**, thanks to advances in technology that let researchers generate vast amounts of data—quickly and affordably. Whether it's DNA sequences, gene expression, or cellular images, we're now collecting more biological data than ever before.

But this explosion of data raises a critical challenge:
How do we make sense of it all?

🧠 Enter: Machine Learning

Machine learning (ML) is a game-changer in biology. These smart algorithms can sift through complex datasets, discover patterns, and make predictions—without being explicitly told what to look for.

🔍 Two Key Types of ML:

- **Supervised Learning**
  Learns from labeled data (like training a model to recognize tumors based on known examples).

- **Unsupervised Learning**
  Finds hidden patterns in unlabeled data (like clustering cells based on gene activity).

Both types play crucial roles in turning big data into **biological insight**.

🧬 The Power of Integration

Today's biology isn't just about measuring one thing—it's about connecting **many layers of information**.

📊 Deep Integration

Combines multiple measurements from the same sample (e.g., mRNA, proteins, transcription factors) to understand internal systems.

🌍 Broad Integration

Looks across many datasets and conditions to uncover general principles that apply across diseases, cell types, or species.

Together, these strategies help answer complex questions, like:

- How do genes work together in networks?
- What drives disease in different individuals?
- Are there universal rules that govern biology?

🧱 Case Study: Big Data in Action

📚 Landmark Datasets:

🔬 The Cancer Genome Atlas (TCGA)

- 7,000+ tumor samples
- Tracks mutations, gene expression, proteins, and more
- A goldmine for cancer researchers

🧬 The ENCODE Project

- 2,600+ datasets
- Focus on DNA-binding proteins and regulatory elements
- 1,200+ ChIP-seq datasets for transcription factors

🌐 Data Storage Giants

- **European Bioinformatics Institute (EBI):** Stores 20+ petabytes of biological data
- Genomic data doubles in size **every year**

🧪 ArrayExpress

- 1.3 million+ genome-wide assays
- Over 45,000 experiments
- Made public by individual researchers

🖥️ Tools for Everyone

You don’t need to be a programmer to analyze big data.

🧰 What's Available:

- **R packages** for supervised and unsupervised ML (e.g., `caret`, `randomForest`, `cluster`)
- **Web-based platforms** that offer point-and-click ML tools—great for wet lab biologists

These resources make it easier than ever to extract meaning from data, even with minimal coding skills.

🚀 Looking Ahead

The biological sciences are being reshaped by big data. The tools now exist to:

- Ask smarter scientific questions
- Integrate information across many levels
- Reveal insights that were previously invisible

Whether you're a biologist, data scientist, or curious learner—**the future of discovery is in your hands.**

Bioinformatics and Genomics

The sequencing of the entire human genome, along with many other organisms, is a major achievement in bioinformatics. The first complete genome of a free-living organism, *Haemophilus influenzae*, was sequenced in 1995 using the "shotgun" technique. Since then, genomes of bacteria (*Mycoplasma genitalium, Mycobacterium tuberculosis*) and eukaryotic species (*yeast, worms, fruit flies, mustard weed*) have been sequenced. Ongoing projects include zebrafish, pufferfish, mice, rats, and primates. This growing database of genetic information will greatly enhance our understanding of biology, disease, and medicine.

Useful Bioinformatics Websites

- **NCBI (www.ncbi.nlm.nih.gov)** – Provides bioinformatics tools and databases
- **GenBank (www.ncbi.nlm.nih.gov/Genbank)** – Stores DNA sequences
- **Ensembl (www.ensembl.org)** – Genome annotation database
- **SWISS-PROT (www.expasy.org/sprot/)** – **Protein** sequence database
- **EBI (www.ebi.ac.uk)** – Research center for bioinformatics
- **Unigene (www.ncbi.nlm.nih.gov/UniGene)** – Collects gene sequence data

- **ISCB (www.iscb.org/)** – Advances computational biology

Bioinformatics Tools

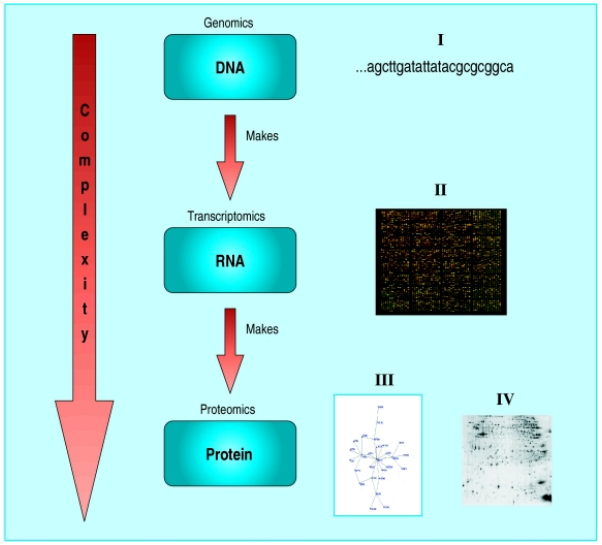
Bioinformaticians use specialized software and the internet to analyze DNA and protein sequences. Tools like **BLAST** allow scientists to compare genetic sequences and identify similar genes in different species. Pharmaceutical companies and biomedical labs increasingly rely on bioinformatics for large-scale data analysis. Tools like **BLAST** help researchers compare unknown sequences with existing data to predict functions and relationships between genes.

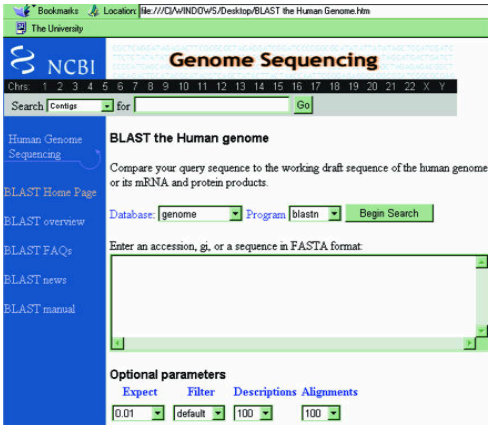
Simplified Explanation of Functional Genomics

Since the human genome was first mapped, the focus has shifted from studying genes to understanding their functions. Functional genomics explores how genes work, how they produce proteins, and what roles those proteins play in the body.

Key areas include:

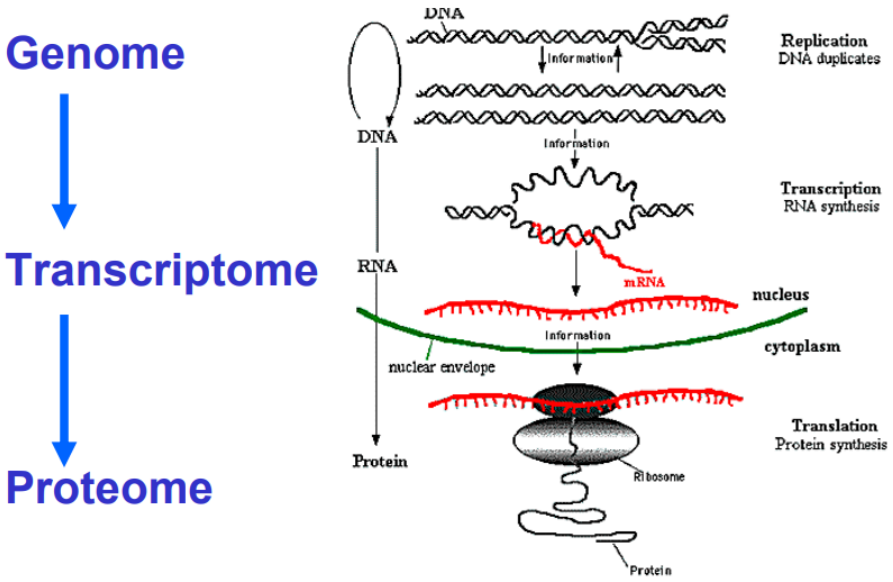
- Proteomics: Studying all proteins (proteome) a cell produces.
- Transcriptomics: Analyzing messenger RNA (transcriptome), which helps create proteins.
- DNA Microarrays: Technology that tracks gene activity and classifies diseases like cancer for better treatments.





Bioinformatics helps analyze vast amounts of genetic data. It aids in:

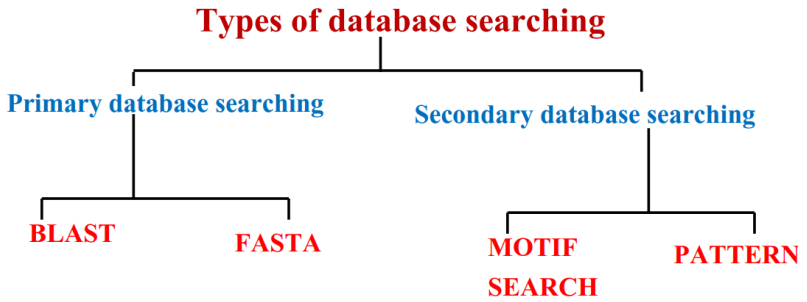
- Predicting gene and protein functions.
- Understanding gene-disease links.
- Designing drugs based on genetic markers (e.g., targeted cancer therapies like Imatinib).
- Personalizing medicine through pharmacogenomics, where treatments are tailored based on a person's genetic profile.



Types of Biological Databases

Biological databases are categorized into three types based on their content:

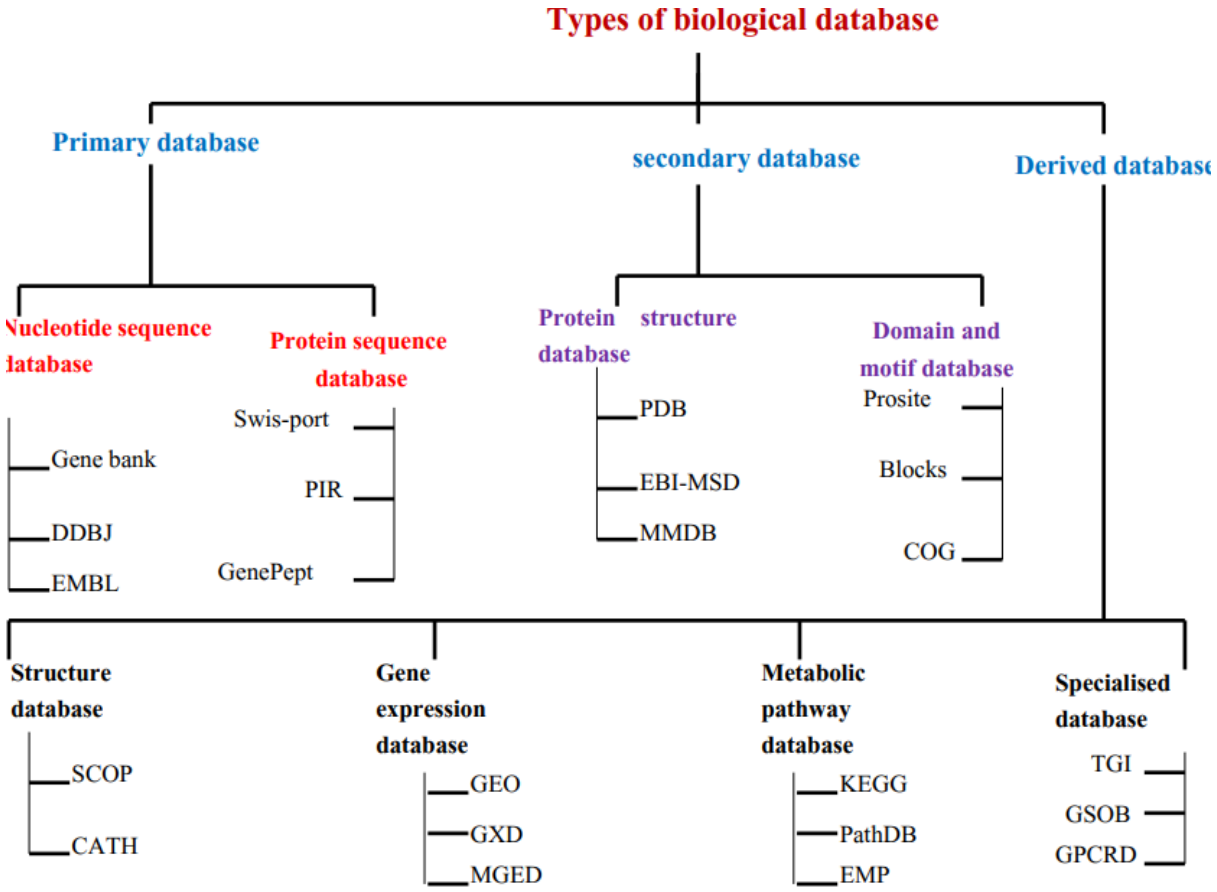
1. **Primary Databases** – Store raw biological data like DNA sequences or protein structures. Examples: GenBank, PDB, DDBJ.
2. **Secondary Databases** – Contain curated or processed information derived from primary databases. Examples: PIR, SWISS-PROT, Pfam.
3. **Specialized Databases** – Focus on specific organisms or data types. Examples: Flybase, HIV Sequence Database, Ribosomal Database Project.



Popular Biological Databases:

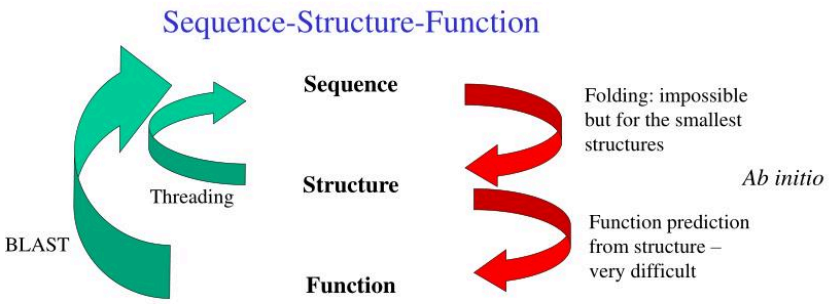
- **GenBank** – A comprehensive DNA sequence database by NCBI.
- **EMBL** – A nucleotide sequence database managed by EBI.
- **DDBJ** – Japan's nucleotide sequence repository, collaborating with GenBank and EMBL.
- **PDB** – Stores 3D structures of proteins, DNA, and RNA.

- **PIR** – A protein information database with multiple related resources.
- **PROSITE** – A database of protein patterns and functional sites.
- **Pfam** – Stores protein family and domain information.
- **KEGG** – Contains data on genomes, pathways, and diseases.
- **OMIM** – A database of human genes and genetic disorders.



Importance of Biological Databases:

- Organize vast biological data.
- Aid researchers in analysis and discovery.
- Enable the development of bioinformatics tools.
- Support collaboration and data sharing.



Bioinformatics Tools

Bioinformatics tools help analyze biological data, focusing on sequence, structure, and function analysis.

Types of Bioinformatics Tools:

1. **Sequence Analysis Tools** – Compare and analyze DNA/protein sequences.
    - **BLAST** – Identifies similar sequences and evolutionary relationships.
    - **ClustalW, T-Coffee** – Multiple sequence alignment tools.
    - **MEME** – Discovers sequence motifs.
    - **MEGA, PHYLIP** – Phylogenetic analysis tools.
2. **Structure Analysis Tools** – Analyze 3D structures of biomolecules.
    - **CN3D, PyMOL, RasMol** – Molecular visualization tools.
    - **MODELLER** – Predicts protein structures.
3. **Function Analysis Tools** – Understand gene/protein interactions and pathways.
    - **GEO** – Repository of gene expression data.
    - **InterProScan** – Identifies protein domains.
    - **COBRA Toolbox, Pathway Tools** – Analyze metabolic pathways.
4. **R - statistics program**
   Link to the most recent downloadable version of R - a statistics package/ programming language.

Applications of Bioinformatics Tools:

- Study genetic and evolutionary relationships.
- Identify and classify genes, proteins, and metabolic pathways.
- Predict protein structures and functions.
- Assist in drug discovery and disease research.

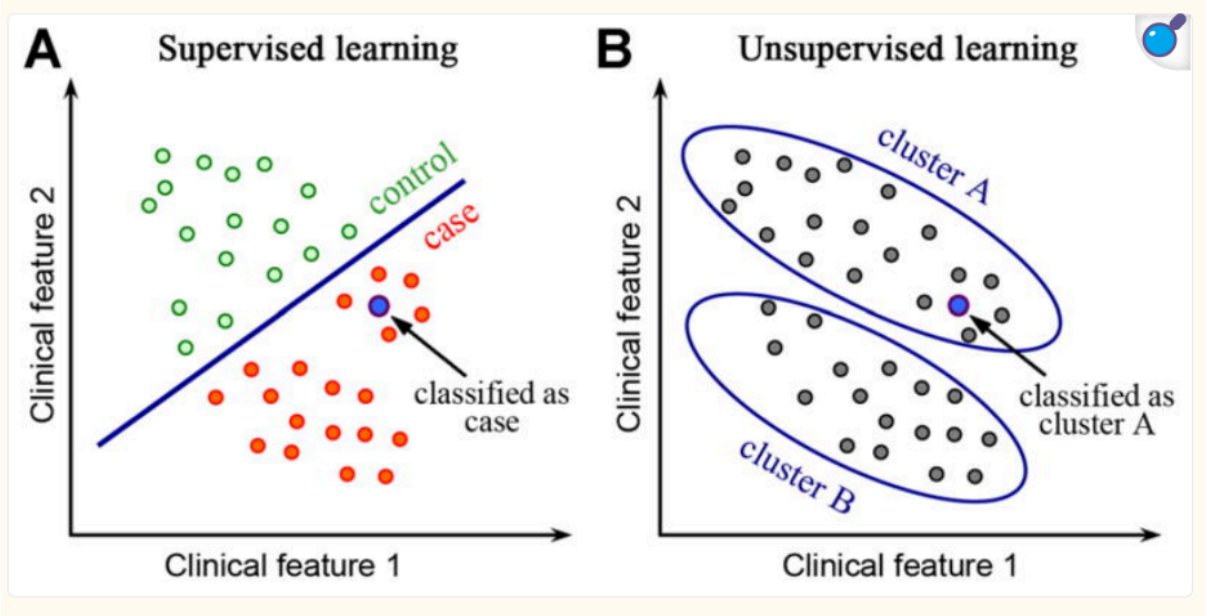
Supervised vs. Unsupervised Machine Learning

Machine learning algorithms are generally classified into **supervised** and **unsupervised** categories, depending on whether the training data includes labeled outcomes.

- **Supervised learning** uses labeled input data (e.g., "case" and "control") to train models that learn to classify new samples by mapping inputs to known outputs (see Fig. 1A).

- **Unsupervised learning**, by contrast, works on unlabeled data to identify hidden structures or patterns, such as clustering similar data points together (see Fig. 1B).

Some hybrid methods, such as **semi-supervised learning**, incorporate both labeled and unlabeled data. Although promising, these are not yet widely adopted in biological research. This discussion focuses on supervised and unsupervised methods with relevant examples.

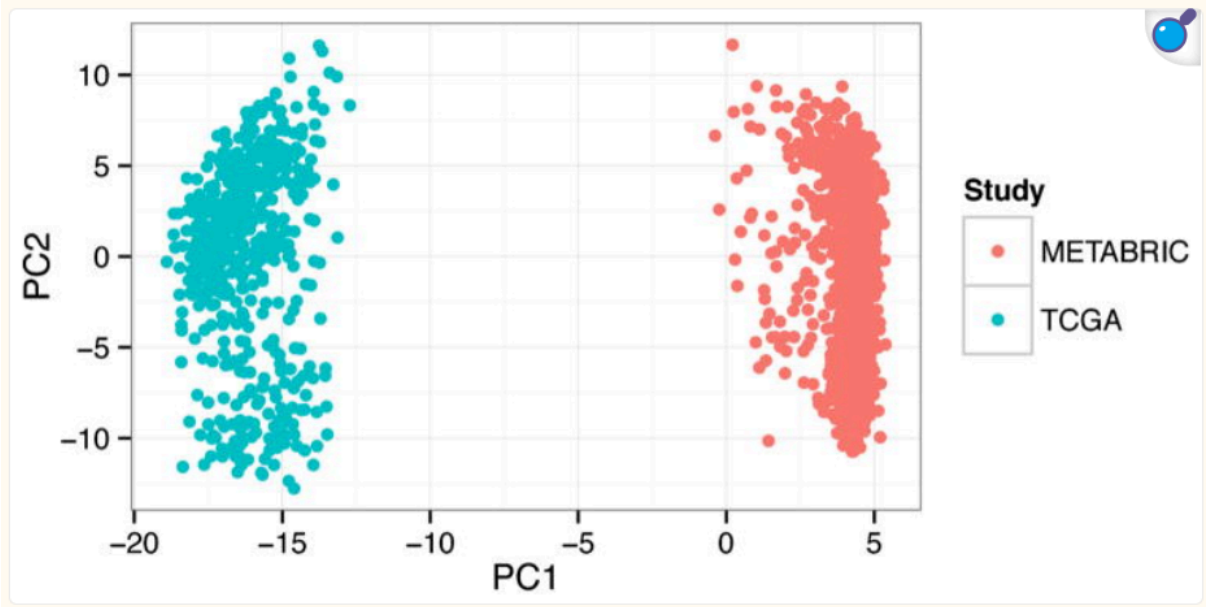


Supervised machine learning (A) uses labeled data (like "case" or "control") to train a model that can predict the labels of new samples.

Unsupervised machine learning (B) works with unlabeled data and finds patterns or groups (clusters) based on similarities between samples.

Unsupervised Learning: Discovering Hidden Patterns

Unsupervised algorithms aim to uncover intrinsic structure in datasets. In biology, this approach can reveal dominant, recurrent signals—such as gene expression trends in cancer types. However, **unsupervised methods are sensitive to confounding factors** like study batch effects. For example, **Principal Component Analysis (PCA)**—a common unsupervised technique—often identifies the source of the data (e.g., study origin) as the dominant pattern (Fig. 2). For accurate interpretation, these methods are best applied to homogeneous datasets.



PCA reveals dominant sources of variation—often confounded by batch or platform differences. **Unsupervised analysis** finds the strongest patterns in the data. For example, when analyzing breast cancer data from two major studies, the biggest difference found may simply be which study the data came from. These kinds of differences—called confounding factors—make it hard to use unsupervised methods on mixed datasets. So, these methods are usually used on more uniform data.

Applications of Unsupervised Learning

1. Identifying Molecular Subtypes of Cancer

Clustering algorithms like **k-means** are commonly used to discover molecular subtypes based on gene expression. K-means divides samples into *k* clusters defined by shared patterns. Techniques like the **GAP statistic** help estimate the optimal number of clusters.

Example: Tothill et al. (2008) used k-means to identify subtypes of ovarian cancer, which were later independently validated by TCGA (2011).

2. Genome Segmentation via Histone Modifications

Unsupervised learning also supports genome segmentation based on histone modification data. The **ChromHMM** algorithm (Ernst & Kellis, 2012) applies a multivariate Hidden Markov Model to histone ChIP-seq data to define chromatin states across the genome.

By analyzing these states, ChromHMM revealed 15 distinct chromatin patterns—including active promoters, enhancers, and repressed regions—validated across nine human cell lines.

Supervised Learning: Learning from Known Examples

Supervised algorithms address questions that begin with, “Given what we already know…” These models are powerful for **predictive tasks**, integrating prior biological knowledge to reduce noise and enhance accuracy.

- A **"gold standard" dataset** of known positives (e.g., genes in a pathway) and negatives (e.g., genes not in the pathway) guides model training.

- Popular methods include **Support Vector Machines (SVMs)** and **penalized logistic regression**, both of which aim to build simple, generalizable models that separate classes effectively.

Supervised learning can also handle regression tasks—predicting continuous values—using methods like **regularized linear regression** or **support vector regression**.

Applications of Supervised Learning

1. Cell Lineage-Specific Gene Expression

Ju et al. (2013) applied an SVM-based method called **in silico nano-dissection** to identify podocyte-specific genes from mixed kidney tissue samples. The model outperformed both experimental and random selection methods in identifying lineage-specific expressions.

2. Predicting Gene Expression from TF and Histone Signals

Cheng and Gerstein (2012) developed a supervised regression model to predict gene expression in mouse embryonic stem cells using data from transcription factor (TF) binding and histone modification profiles (via ChIP-seq). This integrative approach links regulatory signals with expression output, offering a quantitative framework for understanding gene regulation.

Understanding Protein Structure and Design – Simplified

Proteins have complex 3D shapes made from repeating building blocks. After the first protein structures were discovered, scientists noticed common patterns—called **secondary structures**—such as **α-helices** and **β-sheets**. These are arranged and connected by loops to form the protein’s full 3D shape, known as its **tertiary structure**.

Back in 1951, Linus Pauling predicted the α-helix structure. Later, in 1974, Chou and Fasman developed a method to predict whether parts of a protein would form α-helices or β-sheets based on the amino acid sequence. But these early predictions weren’t very accurate because 3D folding affects structure too—not just the sequence.

In the early days, there were only a few known protein structures, limiting what could be learned. Still, some basic rules emerged. For example:

- **Hydrophobic (water-repelling)** amino acids are usually found inside proteins, away from water.

- **Hydrophilic (water-attracting)** or charged amino acids are found on the outside, interacting with water.

These insights led to the first attempts to **design proteins**. In 1988, researchers built a stable protein made of four α-helices using this idea—hydrophobic parts inward, hydrophilic parts outward. This "binary code" of protein design inspired many similar efforts.

Computational Protein Design

The next big leap came with **computer-based protein design**. In 1997, Dahiyat and Mayo designed a small protein (a zinc-finger motif) using only computation. They removed the zinc ions and searched for a new sequence that would keep the same shape without metal help. This approach—called **de novo design**—meant the protein had no similarity to natural sequences.

Their method fixed the protein's backbone and used algorithms (like **dead-end elimination** and **Monte Carlo simulations**) to find the best amino acid sidechains for the desired structure. Their designed protein matched the predicted structure closely, but it worked only for small proteins.

David Baker and his team designed the 93-residue protein **Top7** in 2003 to demonstrate that it was possible to create an entirely **novel protein structure and sequence** — one that does **not exist in nature** — using **automated computational methods**.

Their goals were threefold:

1. **To test the limits of de novo protein design** — creating a protein from scratch with no evolutionary history or similarity to known proteins.

2. **To validate their computational method (Rosetta)** for designing both the **backbone and sidechains** of a protein, optimizing for stability and realistic folding.

3. **To prove that such a designed protein could actually fold** into the predicted 3D structure in real life, which they confirmed through **X-ray crystallography**. Top7's successful design and folding marked a **breakthrough in computational protein design**, proving that proteins could be engineered with novel folds and functions beyond what nature has evolved.

https://www.rcsb.org/structure/1QYS

In 2003, David Baker and his team designed a completely new 93-residue protein called **Top7**, which had:

- A unique structure not seen in nature
- A sequence unlike any known protein
- A 3D shape confirmed by crystallography

Their success came from a program they created called **Rosetta**, which:

- Built protein shapes using fragments from existing structures
  Optimized both sequence and structure using energy calculations
  Used Monte Carlo methods to explore different possibilities
- Rosetta was able to design proteins by simulating how they fold, optimizing the entire shape—not just sidechains, but also the backbone.
- https://rosettacommons.org/software/