

# Bioinformatics

Chap\_03

Sequence Alignment

## Understanding Gene and Protein Relatedness

- **Key Question:** Is a gene or protein related to others?
- **Significance:** Relatedness at the sequence level suggests:
  - **Homology:** Shared evolutionary origin.
  - **Common Functions:** Similar roles in biology.

## Sequence Analysis and Domains

- **Approach:** Analyze DNA/protein sequences to find shared domains or motifs.
- **Purpose:** Identify patterns that indicate relatedness among molecules.

## Importance of Sequence Alignment

- **Context:** Essential as genomes of many organisms are sequenced.
- **Goal:** Understand protein relationships within and across organisms.
- **Impact:** Fundamental to decoding the biology of life.

## Pairwise Sequence Alignment

- **Focus:** Comparing two sequences (DNA or protein).
- **Perspective:** Evolutionary view of amino acid/nucleotide alignment.
- **Tools:** Algorithms and programs for aligning sequences.

## Protein vs. DNA Alignment

- **Preference:** Protein alignment is often more informative than DNA.
- **Reasons:**
  - DNA changes (e.g., third codon position) may not alter amino acids.
  - Amino acids share biophysical properties (e.g., lysine/arginine).
  - Protein scoring systems capture related amino acid relationships.
- **Outcome:** Protein comparisons identify homology better than DNA (Pearson, 1996).
- **Tool Example:** TBLASTN translates DNA to proteins for searches.

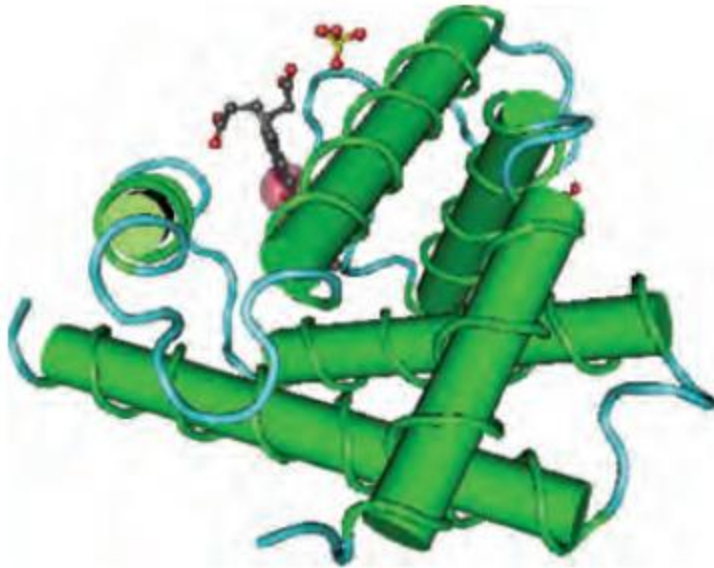
## When to Use Nucleotide Alignment

- **Applications:**
  - Confirm DNA sequence identity in database searches.
  - Identify polymorphisms.
  - Analyze cloned cDNA or regulatory regions.

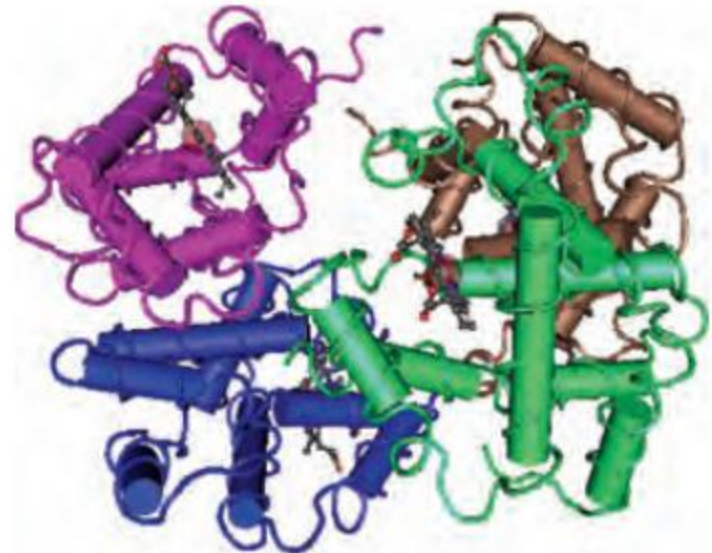
## Definitions – Homology, Similarity, Identity

- **Homology:** Sequences share common evolutionary ancestry (no degrees; homologous or not).
- **Identity:** Quantitative measure of invariant residues.
- **Similarity:** Includes identical + similar residues (e.g., conservative substitutions).
- **Example:** Human myoglobin (NP\_005359.1) and beta globin (NP\_000509.1):
  - Diverged ~450 MYA.
  - Share similar 3D structures despite low sequence identity (e.g., 26% for myoglobin/alpha globin).

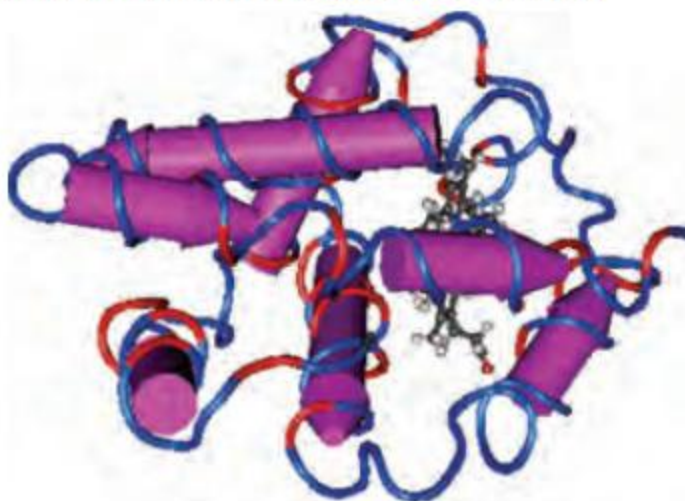
(a) Human myoglobin (3RGK)



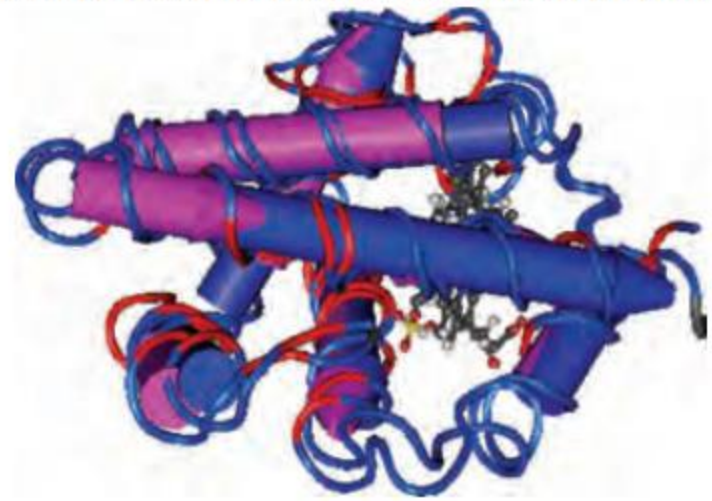
(b) Human hemoglobin tetramer (2H35)



(c) Human beta globin (subunit of 2H35)



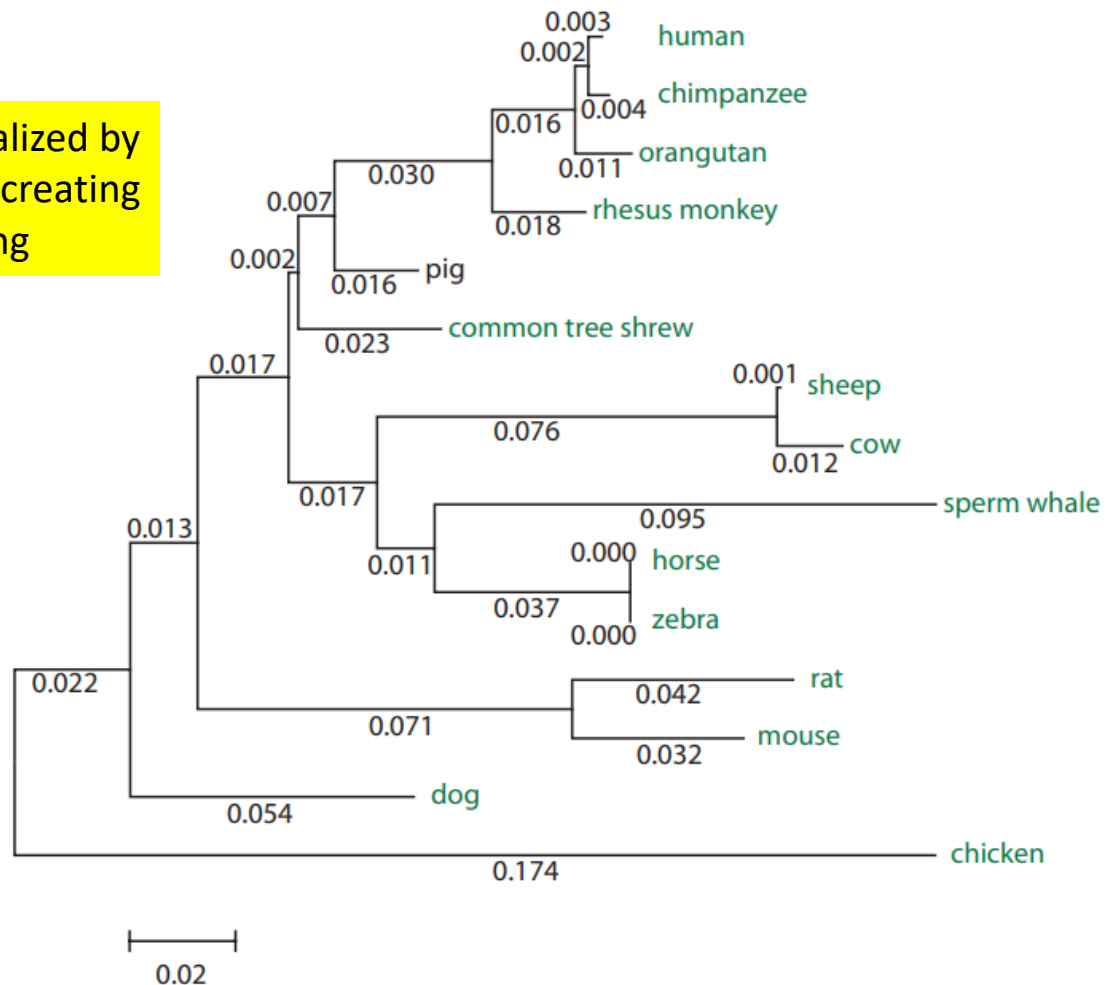
(d) Pairwise alignment of beta globin and myoglobin



## Orthologs vs. Paralogs

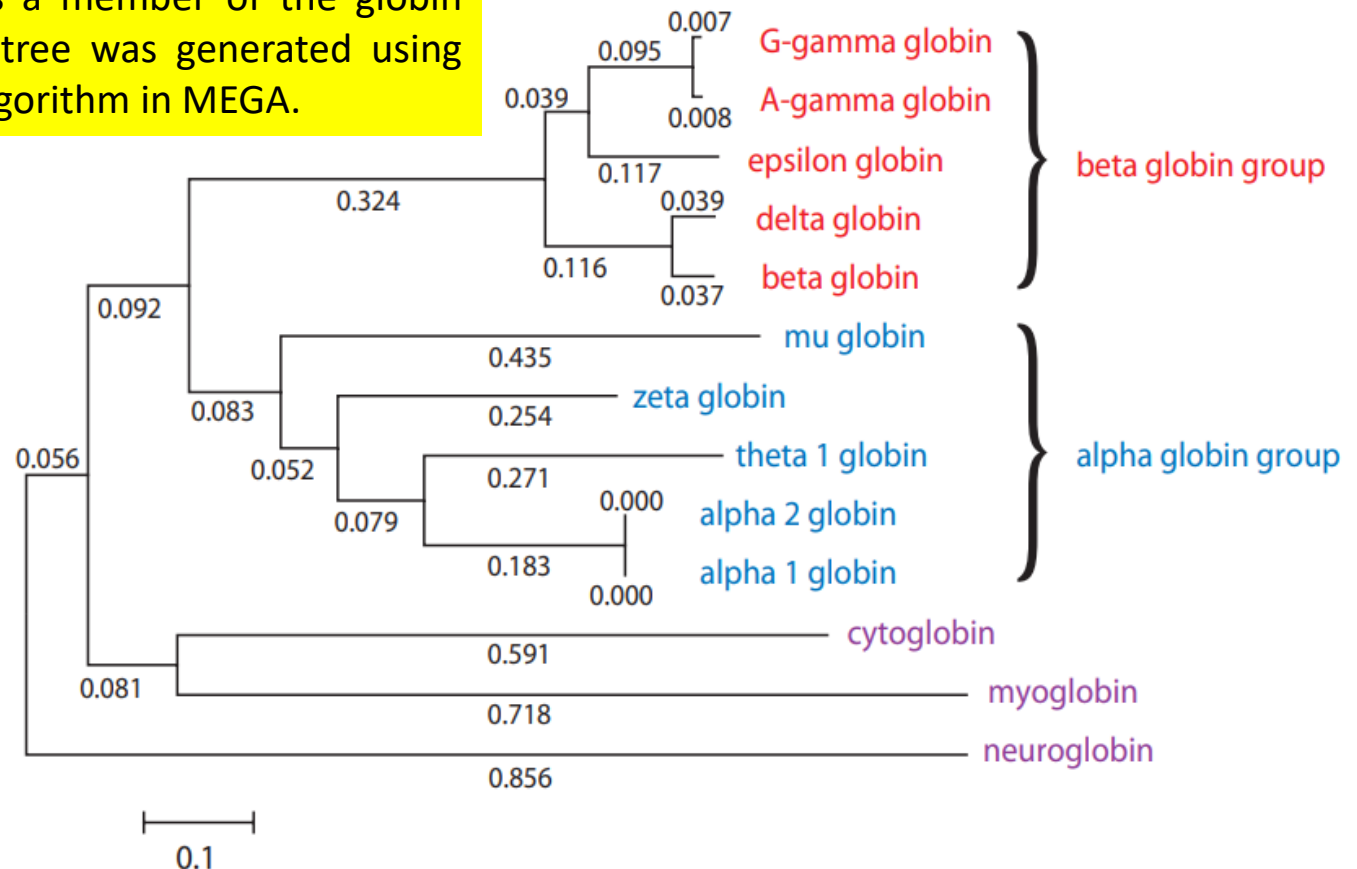
- **Orthologs:** Homologous sequences in different species from speciation (e.g., human/rat myoglobin, similar function).

A group of myoglobin orthologs, visualized by multiply aligning the sequences then creating a phylogenetic tree by neighbor-joining



- **Paralogs:** Homologous sequences from gene duplication (e.g., human alpha 1/alpha 2 globin, 100% identical).
- **Globin Family:** All paralogs with distinct roles (distribution, expression timing, abundance).

Paralogous human globins: Each of these proteins is human, and each is a member of the globin family. This unrooted tree was generated using the neighbor-joining algorithm in MEGA.





## Pairwise Alignment Process

- **Tool:** NCBI BLASTP for proteins (or BLASTN for nucleotides).
- **Steps:**
  - Select BLASTP, enable “Align two or more sequences.”
  - Input sequences (e.g., beta globin FASTA, myoglobin accession).
  - Set parameters (e.g., PAM250 matrix, gap penalties).
  - Run alignment.
- **Output:** Shows identical (e.g., WGKV) and similar residues (e.g., T/S).

Connect to

[Protein BLAST: Align two or more sequences using BLAST](#)

**Query:** NP\_000509.1

**Subject:** NP\_005953

## Alignment Details

- **Identity:** 25% for beta globin/myoglobin (37/145 residues).
- **Similarity:** 39% (57 similar residues, e.g., leucine/valine).
- **Types:**
  - **Local:** Aligns subsets of sequences.
  - **Global:** Includes all residues.
- **Scoring:** Matches score high, mismatches often negative; uses matrices like BLOSUM/PAM.

Pairwise alignment of human beta globin (the “query”) and myoglobin (the “subject”).

(b) Illustration of how raw scores are calculated, using the result of a separate search with just amino acids 12–33 of HBB (corresponding to the region with green shaded letters between the arrowheads in (a)). The raw score is 35, rounded up to 36; this represents the sum of the match scores (from a BLOSUM62 matrix in this case), the mismatch scores, the gap opening penalty (set to  $-11$  for this search), and the gap extension penalty (set to  $-1$ ). Raw scores are subsequently converted to bit scores.

## Importance and Interpretation

- **Goal:** Maximize identity/conservation to assess similarity and homology.
- **Caution:**
  - Avoid saying “percent homology” (homology is binary).
  - Use “high similarity” instead of “highly homologous.”
- **Evidence:** Structural studies + evolutionary analyses confirm homology best.
- **Statistical Significance:** Expect values assess if alignments are random.

## Gaps in Pairwise Alignment

- **Purpose:** Identify mutations (substitutions, insertions, deletions) causing sequence divergence.
- **Substitutions:** Nonidentical amino acids aligned (e.g., serine/threonine).
- **Gaps:** Represent insertions/deletions with dashes.
  - **Example:** One gap in human beta globin/myoglobin alignment (Following Fig., between arrowheads).
- **Effect:** Equalizes alignment length, models evolutionary changes.

Score = 43.9 bits (102), Expect = 1e-09, Method: Composition-based stats.  
Identities = 37/145 (25%), Positives = 57/145 (39%), Gaps = 2/145 (1%)

```

Query   4      LTPEEKSAVTALWGKVNVD--EVGGEALGRLLVVYPWTQRRFFESFGDLSTPDVAVMGNPKV   61
      → L+  E   V  +WGKV  D    G E L RL   +P T   F+ F   L + D +   + +
Sbjct   3      LSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDL   62

Query   62     KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK   121
      → K HG  VL A    L    + +      L++ H  K  +   +   +   ++ VL
Sbjct   63     KKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPG   122

Query   122    EFTPPVQAAYQKVVAGVANALAHKY   146
      → +F    Q A  K +      +A  Y
Sbjct   123    DFGADAQGAMNKALELFRKDMASNY   147
```

# Gap Penalties

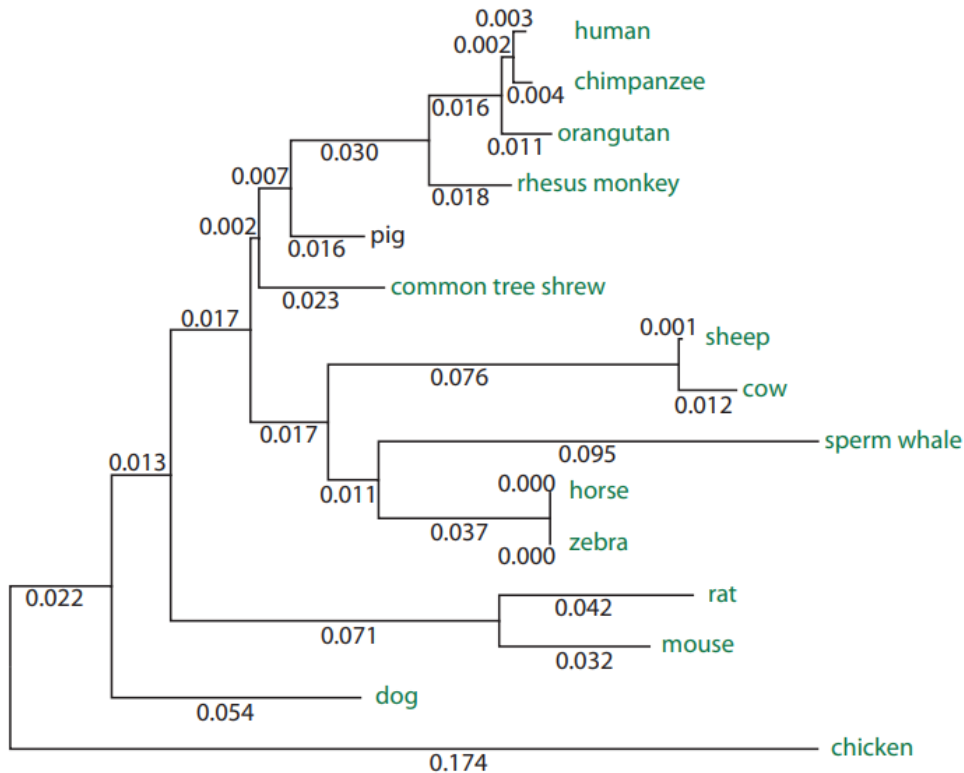
- **Scoring Scheme: Affine gap costs.**
  - **Gap Creation:** Penalty of  $-a$  (e.g.,  $-11$ , Following Fig.).
  - **Gap Extension:** Penalty of  $-b$  per residue.
  - **Formula:** For a gap of  $k$  residues,  $\text{score} = -(a + bk)$ .
  - **Single Residue Gap:**  $\text{Score} = -(a + b)$ .

Score = 18.1 bits (35), Expect = 0.015, Method: Composition-based stats.  
Identities = 11/24 (45%), Positives = 12/24 (50%), Gaps = 2/24 (8%)

Query	12	V T A L W G K V N V D -- E V G G E A L G R L L	33
Sbjct	11	V +W G K V D G E L R L V L N V W G K V E A D I P G H G Q E V L I R L F	34
match		4 11 5 6 6 5 4 5 6 4 4	sum of matches: +60 (round up to +61)
mismatch		-1 1 0 -2 -2 -4 0 -2 0 -3 0	sum of mismatches: -13
gap open		-11	sum of gap penalties: -13
gap extend		-2	
total raw score: 61 - 13 - 13 = 35			

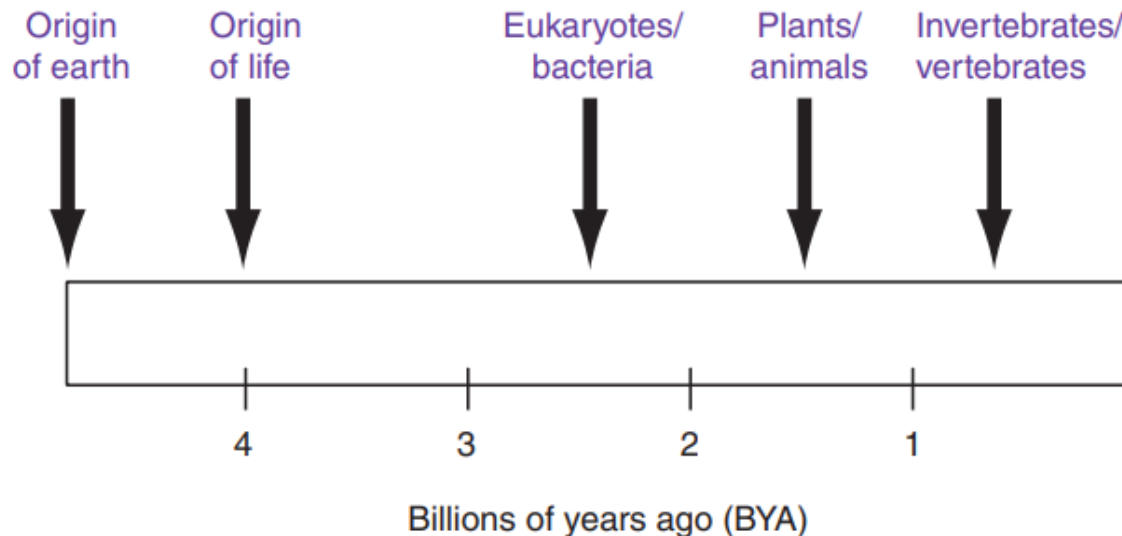
## Pairwise Alignment and Homology

- **Homology:** Implies a shared common ancestor.
- **Example:** Myoglobin sequences from human, horse, chicken (Following Fig.).
  - Diverged ~310 MYA (before human/chicken split, Chapter 19).
- **Process:** Aligning homologous sequences explores evolutionary history.



## Evolution and Globin Family

- **Timeline of Life** (Following Fig., detailed in Chapter 15):
  - Bacteria fossils: ~3.5 billion years old.
  - Archaea fossils: >3 billion years old.
  - Eukaryotes: Emerged similarly early.
- **Globins:**
  - Vertebrate and plant globins diverged ~1.5 billion years ago (See above slide).
  - Bacterial/archaeal globins suggest origin >2 billion years ago.





## Scoring Matrices Overview

- **Purpose:** Assign scores to aligned amino acids in protein pairwise alignments.
- **Example:** Beta globin/myoglobin alignment scores (Fig. 3.5a).
- **Models:**
  - **Dayhoff Model (1966, 1978):** Basis for scoring via PAM matrices.
  - **BLOSUM Matrices:** Alternative by Henikoff & Henikoff, used in BLAST/HMMER (Chapters 4, 5).

Score = 43.9 bits (102), Expect = 1e-09, Method: Composition-based stats.  
Identities = 37/145 (25%), Positives = 57/145 (39%), Gaps = 2/145 (1%)

```

      ▼                               ▼
Query  4    LTPEEKSAVTALWGKVNVD--EVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV  61
      → L+  E   V  +WGKV  D    G E L RL  +P T   F+ F   L + D +   +   +
Sbjct  3    LSDGEWQLVLNVWGKVEADIPGHGQEVLIIRLFKGH PETLEKFDKFKHLKSEDEMKASEDL  62

Query  62    KAHGKKVLGAFSDGLAHL DNLKGT FATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK  121
      → K HG  VL A    L    + +      L++ H  K  +   +   +   ++ VL
Sbjct  63    KKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHHPG  122

Query  122   EFTPPVQAAYQKVVAGVANALAHKY  146
      → +F    Q A  K  +      +A  Y
Sbjct  123   DFGADAQGAMNKALELFRKDMASNY  147
```

## Dayhoff Model Step 1 – Accepted Point Mutations (PAM)

- **Approach:** Cataloged 1572 substitutions in 71 protein families (Box 3.4).
- **Definition:** PAM = Amino acid replacement accepted by natural selection.
  - Requires DNA mutation and species-wide adoption.
- **Analysis:** Compared sequences to inferred ancestral sequences (Fig. 3.7, Box 3.5).



- **Method:** Partial multiple sequence alignment of alpha 1 globin, beta globin, delta globin, myoglobin (Fig. a).
  - Four columns show amino acid differences (e.g., A vs. G, marked in red).
- **Phylogenetic Tree (Fig. b):**
  - Shows extant sequences (1–4) and ancestral nodes (5–6).
  - Inferred ancestors via maximum parsimony (PAUP, Chapter 7).
- **Insight:** No direct amino acid swaps (e.g., alanine to glycine); changes evolved from ancestral residues (e.g., glutamate → alanine/glycine).

**Maximum Parsimony (MP)** is used in **phylogenetic analysis** (not directly in generating multiple sequence alignments, but rather in interpreting them). Here's a breakdown of how MP relates to **multiple sequence alignment (MSA)**:

### **What is Maximum Parsimony?**

Maximum Parsimony is a criterion for selecting among possible phylogenetic trees. The best tree under MP is the one that **minimizes the total number of evolutionary changes** (mutations, insertions, deletions) required to explain the observed sequences.

## Relation to Multiple Sequence Alignment (MSA)

Before performing a maximum parsimony analysis, you usually need to:

1. **Align your sequences** using MSA tools (e.g., Clustal Omega, MUSCLE, MAFFT).
2. The resulting alignment is then used to infer evolutionary relationships using MP.

The **MSA** serves as **input** for the **MP algorithm**, which treats each column in the alignment as a character and tries **to find** the **tree** that explains the **fewest number of changes across all columns**.

## Steps in Maximum Parsimony Analysis

**1.Input:** Aligned DNA, RNA, or protein sequences.

**2.Tree Construction:**

- Generate possible phylogenetic trees (usually heuristic, as the number grows exponentially).

**3.Score Trees:**

- For each tree, calculate how many changes (mutations) are needed to explain the observed sequences.

**4.Choose Tree:**

- Select the tree(s) with the **lowest total number of changes**.

## Tools Supporting MP Analysis

- **PAUP\*** (Phylogenetic Analysis Using Parsimony)
- **MEGA** (Molecular Evolutionary Genetics Analysis)
- **PHYLIP**
- **TNT** (Tree analysis using New Technology)
- **Mesquite**

## Example Use Case

Say you have an MSA of 5 DNA sequences. Using MP, you can:

- Generate several candidate trees.
- Use Fitch's algorithm (or similar) to count the minimal number of changes for each site.
- Add the total changes across all sites.
- Choose the tree with the least total change.

## Dayhoff Model Step 1 – Accepted Point Mutations (PAM)

- Findings: Substitution frequencies shown in Fig. 3.8.
  - Common:** Asparagine/serine.
  - Rare:** Cysteine/tryptophan.

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A																				
R	30																			
N	109	17																		
D	154	0	532																	
C	33	10	0	0																
Q	93	120	50	76	0															
E	266	0	94	831	0	422														
G	579	10	156	162	10	30	112													
H	21	103	226	43	10	243	23	10												
I	66	30	36	13	17	8	35	0	3											
L	95	17	37	0	y	75	15	17	40	253										
K	57	477	322	85	0	147	104	60	23	43	39									
M	29	17	0	0	0	20	7	7	0	57	207	90								
F	20	7	7	0	0	0	0	17	20	90	167	0	17							
P	345	67	27	10	10	93	40	49	50	7	43	43	4	7						
S	772	137	432	98	117	47	86	450	26	20	32	168	20	40	269					
T	590	20	169	57	10	37	31	50	14	129	52	200	28	10	73	696				
W	0	27	3	0	0	0	0	0	3	0	13	0	0	10	0	17	0			
Y	20	3	36	0	30	0	10	0	40	13	23	10	0	260	0	22	23	6		
V	365	20	13	17	33	27	37	97	30	661	303	17	77	10	50	43	186	0	17	
	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val

## Dayhoff Model Step 2 – Amino Acid Frequencies

- **Purpose:** Model mutation probabilities using amino acid occurrence rates.
- **Data:** Frequency of each amino acid ( $f_i$ ) listed in Table 3.1.
- **Importance:** Basis for calculating substitution likelihoods.

**Table 3.1** Normalized frequencies of amino acids. These values sum to 1. If the 20 amino acids were equally represented in proteins, these values would all be 0.05 (i.e., 5%); instead, amino acids vary in their frequency of occurrence.

Gly	0.089	Arg	0.041
Ala	0.087	Asn	0.040
Leu	0.085	Phe	0.040
Lys	0.081	Gln	0.038
Ser	0.070	Ile	0.037
Val	0.065	His	0.034
Thr	0.058	Cys	0.033
Pro	0.051	Tyr	0.030
Glu	0.050	Met	0.015
Asp	0.047	Trp	0.010

*Source:* Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.



## Dayhoff Model Step 3 – Relative Mutability

- **Calculation:** Relative Mutability = Mutations observed ( $m_i$ ) / Frequency ( $f_i$ ) (Table 3.2).
- RM simply describes how often each amino acid is likely to change over a short evolutionary period.
- **Insights:**
  - **Less Mutable:** Critical residues (e.g., tryptophan, cysteine) → Harmful substitutions.
  - **Highly Mutable:** Asparagine, serine, aspartic/glutamic acid → Easily replaced.

**Table 3.2** Relative mutabilities of amino acids. The value of alanine is arbitrarily set to 100.

Asn	134	His	66
Ser	120	Arg	65
Asp	106	Lys	56
Glu	102	Pro	56
Ala	100	Gly	49
Thr	97	Tyr	41
Ile	96	Phe	41
Met	94	Leu	40
Gln	93	Cys	20
Val	74	Trp	18

## Dayhoff Model Step 3 – Relative Mutability

- **Common Substitutions (Fig. 3.8):**
  - Glutamic/aspartic acid, serine/alanine, serine/threonine, isoleucine/valine.
- **Genetic Code (Box 3.6):** Single-nucleotide changes drive common substitutions.

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A																				
R	30																			
N	109	17																		
D	154	0	532																	
C	33	10	0	0																
Q	93	120	50	76	0															
E	266	0	94	831	0	422														
G	579	10	156	162	10	30	112													
H	21	103	226	43	10	243	23	10												
I	66	30	36	13	17	8	35	0	3											
L	95	17	37	0	y	75	15	17	40	253										
K	57	477	322	85	0	147	104	60	23	43	39									
M	29	17	0	0	0	20	7	7	0	57	207	90								
F	20	7	7	0	0	0	0	17	20	90	167	0	17							
P	345	67	27	10	10	93	40	49	50	7	43	43	4	7						
S	772	137	432	98	117	47	86	450	26	20	32	168	20	40	269					
T	590	20	169	57	10	37	31	50	14	129	52	200	28	10	73	696				
W	0	27	3	0	0	0	0	0	3	0	13	0	0	10	0	17	0			
Y	20	3	36	0	30	0	10	0	40	13	23	10	0	260	0	22	23	6		
V	365	20	13	17	33	27	37	97	30	661	303	17	77	10	50	43	186	0	17	
	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val

The substitutions that occur in proteins can also be understood with reference to the genetic code.

		Second nucleotide					
		T	C	A	G		
First nucleotide	T	TTT Phe 171 TTC Phe 203 TTA Leu 73 TTG Leu 125	TCT Ser 147 TCC Ser 172 TCA Ser 118 TCG Ser 45	TAT Tyr 124 TAC Tyr 158 TAA Ter 0 TAG Ter 0	TGT Cys 99 TGC Cys 119 TGA Ter 0 TGG Trp 122	T C A G	Third nucleotide
	C	CTT Leu 127 CTC Leu 187 CTA Leu 69 CTG Leu 392	CCT Pro 175 CCC Pro 197 CCA Pro 170 CCG Pro 69	CAT His 104 CAC His 147 CAA Gln 121 CAG Gln 343	CGT Arg 47 CGC Arg 107 CGA Arg 63 CGG Arg 115	T C A G	
	A	ATT Ile 165 ATC Ile 218 ATA Ile 71 ATG Met 221	ACT Thr 131 ACC Thr 192 ACA Thr 150 ACG Thr 63	AAT Asn 174 AAC Asn 199 AAA Lys 248 AAG Lys 331	AGT Ser 121 AGC Ser 191 AGA Arg 113 AGG Arg 110	T C A G	
	G	GTT Val 111 GTC Val 146 GTA Val 72 GTG Val 288	GCT Ala 185 GCC Ala 282 GCA Ala 160 GCG Ala 74	GAT Asp 230 GAC Asp 262 GAA Glu 301 GAG Glu 404	GGT Gly 112 GGC Gly 230 GGA Gly 168 GGG Gly 160	T C A G	

## Amino Acid Substitutions & Genetic Code

- Most common substitutions involve single-nucleotide changes.
- Less mutable amino acids (e.g., Trp, Cys, Phe, Tyr) often have only 1–2 codons, making mutations more disruptive.
- Low mutability suggests strong natural selection against substitutions.
- Some potential single-nucleotide changes (e.g., Gly  $\leftrightarrow$  Trp) are not observed, likely due to negative selection.
- About 20% of observed substitutions require two nucleotide changes.

## Dayhoff Model Step 4 – Mutation Probability Matrix (1 PAM)

- **Matrix M:** Shows probability of amino acid  $j \rightarrow i$  over 1 PAM (Fig. 3.9).

Each element of the matrix  $M_{ij}$  shows the probability that an **original amino acid  $j$**  (see the columns) will be **replaced by another amino acid  $i$**  (see the rows) over a defined **evolutionary interval**.

The **interval** is **one PAM**, which is defined as the **unit** of **evolutionary divergence** in which **1%** of the **amino acids** have been **changed** between the **two protein sequences**.

- **PAM Definition:** 1% amino acid divergence (not time-based).
- **Features:**
  - **Diagonal:** Highest scores (e.g., alanine  $\rightarrow$  alanine: 98.7%).
  - **Column sums:** 100%.
  - **Mutability variation:** Asparagine (98.22% unchanged) vs. tryptophan (99.76% unchanged).
- **Context:** Divergence rates vary across proteins (Fig. 7.5, molecular clock).

**1% divergence** of protein sequence **may occur over vastly different time frames** for protein families that undergo substitutions at different rates

## Dayhoff Model Step 4 – Mutation Probability Matrix (1 PAM)

		Original amino acid																			
		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
Replacement amino acid	A	98.7	0.0	0.1	0.1	0.0	0.1	0.2	0.2	0.0	0.1	0.0	0.0	0.1	0.0	0.2	0.4	0.3	0.0	0.0	0.2
	R	0.0	99.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0
	N	0.0	0.0	98.2	0.4	0.0	0.0	0.1	0.1	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0
	D	0.1	0.0	0.4	98.6	0.0	0.1	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	C	0.0	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	Q	0.0	0.1	0.0	0.1	0.0	98.8	0.3	0.0	0.2	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
	E	0.1	0.0	0.1	0.6	0.0	0.4	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	G	0.2	0.0	0.1	0.1	0.0	0.0	0.1	99.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.1
	H	0.0	0.1	0.2	0.0	0.0	0.2	0.0	0.0	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.1	0.0	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.3
	L	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.2	99.5	0.0	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.2
	K	0.0	0.4	0.3	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	99.3	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0
	M	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	99.5	0.0	0.0	0.0	0.0	0.3	0.0
	P	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	99.3	0.1	0.0	0.0	0.0	0.0
	S	0.3	0.1	0.3	0.1	0.1	0.0	0.1	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.2	98.4	0.4	0.1	0.0	0.0
	T	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.0	0.1	0.3	98.7	0.0	0.0	0.1
	W	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
	Y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	99.5	0.0
	V	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.0	99.0

The PAM1 mutation probability matrix. The **original amino acid *j*** is arranged in columns (across the top), while the **replacement amino acid *i*** is arranged in rows.

Dayhoff et al. multiplied values by 10,000 (offering added precision) while here we multiply by 100 so that, for example, the first cell's value of 98.7 corresponds to 98.7% occurrence of ala remaining ala over this evolutionary interval.

## Dayhoff Model Step 4 – Mutation Probability Matrix (1 PAM)

### Mutation Probability Matrix – Nondiagonal Elements

- **Formula:**  $M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}}$  (Fig. 3.9)
  - $M_{ij}$ : Probability amino acid j  $\rightarrow$  i.
  - $A_{ij}$ : Accepted mutation from Fig. 3.8 (e.g., alanine  $\rightarrow$  arginine).
  - $\lambda$ : Proportionality constant.
  - $m_j$ : Mutability of amino acid j (Table 3.2).
- **Purpose:** Quantifies substitution likelihood for pairwise alignment scoring.

### Mutation Probability Matrix – Diagonal Elements

- **Formula:**  $M_{jj} = 1 - \lambda m_j$  (Fig. 3.9)
  - $M_{jj}$ : Probability amino acid j remains unchanged.
- **Example:** Alanine column (Fig. 3.9):
  - Sum of probabilities = 100%.
  - Non-diagonal sum proportional to alanine's mutability (Table 3.2).

## Dayhoff Model Step 4 – Mutation Probability Matrix (1 PAM)

### Substitution Patterns

- **Observation:** Each amino acid has likely replacements if mutated.
- **Relevance:** Form the scoring system's basis (Dayhoff Steps 5–7).
  - Rewards likely substitutions.
  - Penalizes unlikely ones.

### Assumptions and Ancestral Sequences

- **Data Source:** Mostly from extant organisms; ancestral sequences inferred (Box 3.5, Chapter 7).
- **Assumption:** Mutations are undirected (equally likely in both directions).
- **PAM1 Context:** Close protein relationships suggest ancestral residue resembles observed residues (Fig. 3.9).



## Dayhoff Model Step 5 – PAM Matrices Overview

- **Purpose:** Model amino acid substitutions for proteins with varying identity levels.
- **PAM1 Basis:** Alignments of closely related proteins ( $\geq 85\%$  identical, 1% change, Fig. 3.9).
- **Goal:** Extend to distantly related proteins using higher PAM matrices (e.g., PAM250).

		Original amino acid																			
		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
Replacement amino acid	A	98.7	0.0	0.1	0.1	0.0	0.1	0.2	0.2	0.0	0.1	0.0	0.0	0.1	0.0	0.2	0.4	0.3	0.0	0.0	0.2
	R	0.0	99.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0
	N	0.0	0.0	98.2	0.4	0.0	0.0	0.1	0.1	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0
	D	0.1	0.0	0.4	98.6	0.0	0.1	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	C	0.0	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	Q	0.0	0.1	0.0	0.1	0.0	98.8	0.3	0.0	0.2	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
	E	0.1	0.0	0.1	0.6	0.0	0.4	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	G	0.2	0.0	0.1	0.1	0.0	0.0	0.1	99.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.1
	H	0.0	0.1	0.2	0.0	0.0	0.2	0.0	0.0	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.1	0.0	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.3
	L	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.2	99.5	0.0	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.2
	K	0.0	0.4	0.3	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	99.3	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0
	M	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	99.5	0.0	0.0	0.0	0.0	0.3	0.0
	P	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	99.3	0.1	0.0	0.0	0.0	0.0
	S	0.3	0.1	0.3	0.1	0.1	0.0	0.1	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.2	98.4	0.4	0.1	0.0	0.0
	T	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.0	0.1	0.3	98.7	0.0	0.0	0.1
	W	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
	Y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	99.5	0.0
	V	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.0	99.0

**Fig. 3.9 The PAM1 mutation probability matrix.** The original amino acid  $j$  is arranged in columns (across the top), while the replacement amino acid  $i$  is arranged in rows.

## Examples of Protein Conservation

- **Conserved Proteins:** GAPDH, high identity, rare substitutions (Fig. 3.10).

<a href="#">NP_002037.2</a>	164	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGALQNII	207
<a href="#">XP_001162057.1</a>	164	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGALQNII	207
<a href="#">NP_001003142.1</a>	162	IHDHFGIVEGLMTTVHAIITATQRTVDGPGSGKMWRDGRGAAQNII	205
<a href="#">XP_893121.1</a>	168	IHDNFGIMEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	211
<a href="#">XP_576394.1</a>	162	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	205
<a href="#">NP_058704.1</a>	162	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	205
<a href="#">XP_001070653.1</a>	162	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	205
<a href="#">XP_001062726.1</a>	162	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	205
<a href="#">NP_989636.1</a>	162	IHDNFGIVEGLMTTVHAIITATQRTVDGPGSGKLWRDGRGAAQNII	205
<a href="#">NP_525091.1</a>	161	INDNFEIVEGLMTTVHATTATQRTVDGPGSGKLWRDGRGAAQNII	204
<a href="#">XP_318655.2</a>	161	INDNFGILEGLMTTVHATTATQRTVDGPGSGKLWRDGRGAAQNII	204
<a href="#">NP_508535.1</a>	170	INDNFGIIEGLMTTVHAVTATQRTVDGPGSGKLWRDGRGAGQNII	213
<a href="#">NP_595236.1</a>	164	INDTFGIEEGLMTTVHATTATQRTVDGPGSKDWRGGRGASANII	207
<a href="#">NP_011708.1</a>	162	INDAFGIEEGLMTTVHSLTATQRTVDGPGSHKDWRGGRTASGNII	205
<a href="#">XP_456022.1</a>	161	INDEFGIDEALMTTVHSITATQRTVDGPGSHKDWRGGRTASGNII	204
<a href="#">NP_001060897.1</a>	166	IHDNFGIIEGLMTTVHAIITATQRTVDGPGSSKDWRGGRAASFNII	209

Multiple sequence alignment of a portion of the glyceraldehyde 3-phosphate dehydrogenase (GAPDH) protein from 13 organisms.

Columns in the alignment that even have a single amino acid change are indicated with arrowheads.

## Examples of Protein Conservation

- **Less Conserved:** Kappa caseins, frequent substitutions, many gaps, diverse residues in columns (Fig. 3.11).
- **Need:** Higher PAM matrices (e.g., PAM100, PAM250) for distant relationships.

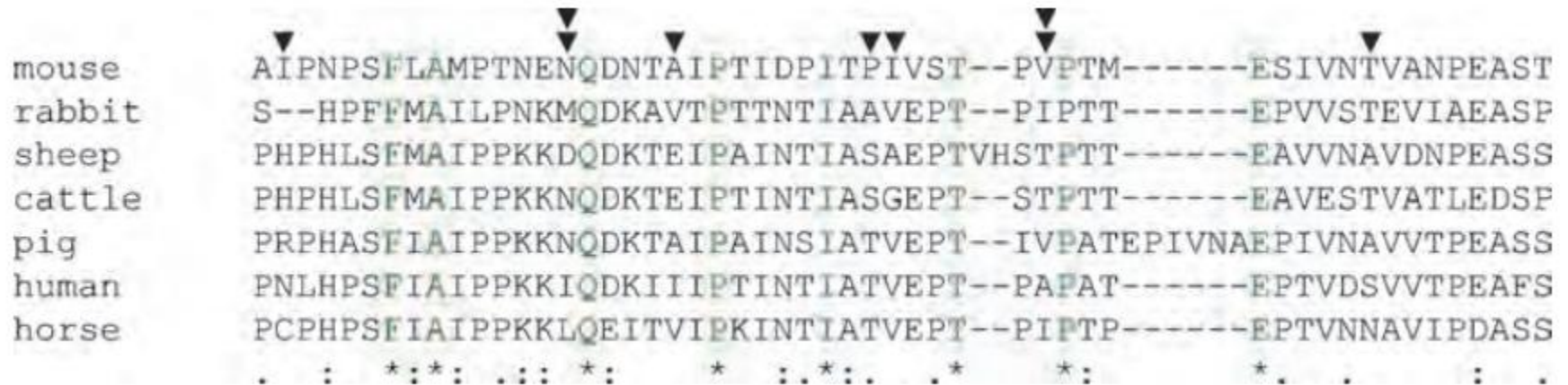


Fig. 3.11 Multiple sequence alignment of seven kappa caseins, representing a protein family that is relatively poorly conserved. Only a portion of the entire alignment is shown. Note that just eight columns of residues are perfectly conserved (indicated with asterisks), and gaps of varying length form part of the alignment. In several columns, there are four different aligned amino acids (arrowheads); in two instances, there are five different residues (double arrowheads).

## Deriving Higher PAM Matrices

- **Method:** Multiply PAM1 matrix by itself (Box 3.7).
  - Avoids errors from scaling  $\lambda$  alone (Equations 3.1, 3.2).
  - Models multiple substitutions over greater evolutionary distances.
- **Extremes:**
  - **PAM0:** Unit diagonal, no changes (Fig. 3.12, upper panel).
  - **PAM $\infty$ :** Equal probabilities, matches background frequencies (Fig. 3.12, lower panel; Table 3.1).

replacement amino acid replacement amino acid

PAM0	A	R	N	D	C	Q	E	G
A	100	0	0	0	0	0	0	0
R	0	100	0	0	0	0	0	0
N	0	0	100	0	0	0	0	0
D	0	0	0	100	0	0	0	0
C	0	0	0	0	100	0	0	0
Q	0	0	0	0	0	100	0	0
E	0	0	0	0	0	0	100	0
G	0	0	0	0	0	0	0	100

original amino acid

PAM $\infty$	A	R	N	D	C	Q	E	G
A	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7
R	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1
N	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
D	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
C	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Q	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
E	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
G	8.9	8.9	8.9	8.9	8.9	8.9	8.9	8.9

## PAM250 Matrix

- **Definition:** PAM1 matrix multiplied 250 times (Fig. 3.13).
- **Context:** ~20% amino acid identity, used in BLAST (Chapter 4).

		Original amino acid																			
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Replacement amino acid	A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
	R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
	N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
	D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
	C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
	Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
	E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
	G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
	H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
	I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
	L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
	K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
	M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
	F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
	P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
	S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
	T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
	W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
	Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
	V	7	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	7	2	4	17

The PAM250 mutation probability matrix.

### • Features:

- Less distinct diagonal vs. PAM1 (Fig. 3.9).
- Example: Alanine → Alanine (13%), Alanine → Glycine (12%).
- Tryptophan/Cysteine: >50% chance unchanged.



## Dayhoff Model Step 6 – Relatedness Odds Matrix

- **Purpose:** Convert the mutation probability matrix ( $M_{ij}$ ) to odds of amino acid substitutions in homologous sequences.
- **Formula:**  $R_{ij} = \frac{M_{ij}}{f_i}$ 
  - $M_{ij}$ : Probability amino acid  $j \rightarrow i$ .
  - $f_i$ : Normalized frequency of amino acid  $i$  by chance.

## Interpreting the Odds Ratio

- $R_{ij} = 1$ : Substitution (e.g., alanine  $\rightarrow$  asparagine) occurs at chance level.
- $R_{ij} > 1$ : Substitution more frequent than chance (e.g., serine  $\rightarrow$  threonine, conservative).
- $R_{ij} < 1$ : Substitution less likely than chance (not favored).

## Application to Alignment

- **Process:**
  - Calculate  $R_{ij}$  for each aligned position.
  - Multiply probabilities to score the entire alignment.
- **Goal:** Quantify likelihood of observed substitutions in homologous proteins.

## Dayhoff Model Step 7 – Log-Odds Scoring Matrix

- **Definition:** Logarithmic form of relatedness odds matrix for scoring pairwise alignments.
- **Formula:**  $s_{ij} = 10 \times \log_{10}(M_{ij} / f_i)$  (Equation 3.4)
  - $M_{ij} (q_{ij})$ : Substitution frequency from mutation probability matrix (Figs. 3.9, 3.13).
  - $f_i$ : Background frequency of amino acid i (Table 3.1).
- **Purpose:** Sum scores for aligned residues (logarithms simplify multiplication).

**Table 3.1**

Gly	0.089	Arg	0.041
Ala	0.087	Asn	0.040
Leu	0.085	Phe	0.040
Lys	0.081	Gln	0.038
Ser	0.070	Ile	0.037
Val	0.065	His	0.034
Thr	0.058	Cys	0.033
Pro	0.051	Tyr	0.030
Glu	0.050	Met	0.015
Asp	0.047	Trp	0.010

## PAM250 Log-Odds Matrix

- **Matrix:** Scores for PAM250 (Fig. 3.14), rounded to integers.
- **Examples:**
  - Cysteine → Leucine:  $M_{ij} = 0.02$ ,  $f_i = 0.085 \rightarrow s_{ij} = -6.3$  (Equation 3.5).
  - Lysine → Arginine:  $M_{ij} = 0.09$ ,  $f_i = 4.1\% \rightarrow s_{ij} = 3$  (Fig. 3.14).
- **Symmetry:** Scores independent of sequence order (unlike Fig. 3.13).

$$s_{(\text{cysteine, leucine})} = 10 \times \log_{10} \left( \frac{0.02}{0.085} \right) = -6.3.$$



A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	-2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

**Log-odds matrix for PAM250.** High PAM values (e.g., PAM250) are useful for aligning very divergent sequences. s algorithms for pairwise alignment, multiple sequence alignment, and database searching (e.g., BLAST) allow you to select an assortment of PAM matrices such as PAM250, PAM70, and PAM30.

## Interpreting Scores

- **Positive Scores:** More frequent than chance.
  - +17 (Tryptophan): 50x more likely ( $10^{1.7} = 50$ ).
  - +2: 1.6x more likely ( $10^{0.2} = 1.6$ ).
- **Negative Scores:** Less frequent than chance.
  - -10: One-tenth as likely.
- **Neutral Score (0):** Chance-level alignment.
- **Highest/Penalized:** Tryptophan (+17), cysteine (+12); severe penalties for their substitutions.

## PAM250 vs. PAM10

- **Comparison** (Figs. 3.14, 3.15):
  - **PAM10:** Higher scores for matches (e.g., alanine → alanine: +7 vs. +2).
  - **PAM10:** Greater penalties for mismatches (e.g., aspartate → arginine: -17 vs. -1).
  - **PAM10:** Negative for some positive PAM250 substitutions (e.g., glutamate → asparagine: -5 vs. +1).
- **Context:** PAM10 for closer relationships, PAM250 for distant (~20% identity).

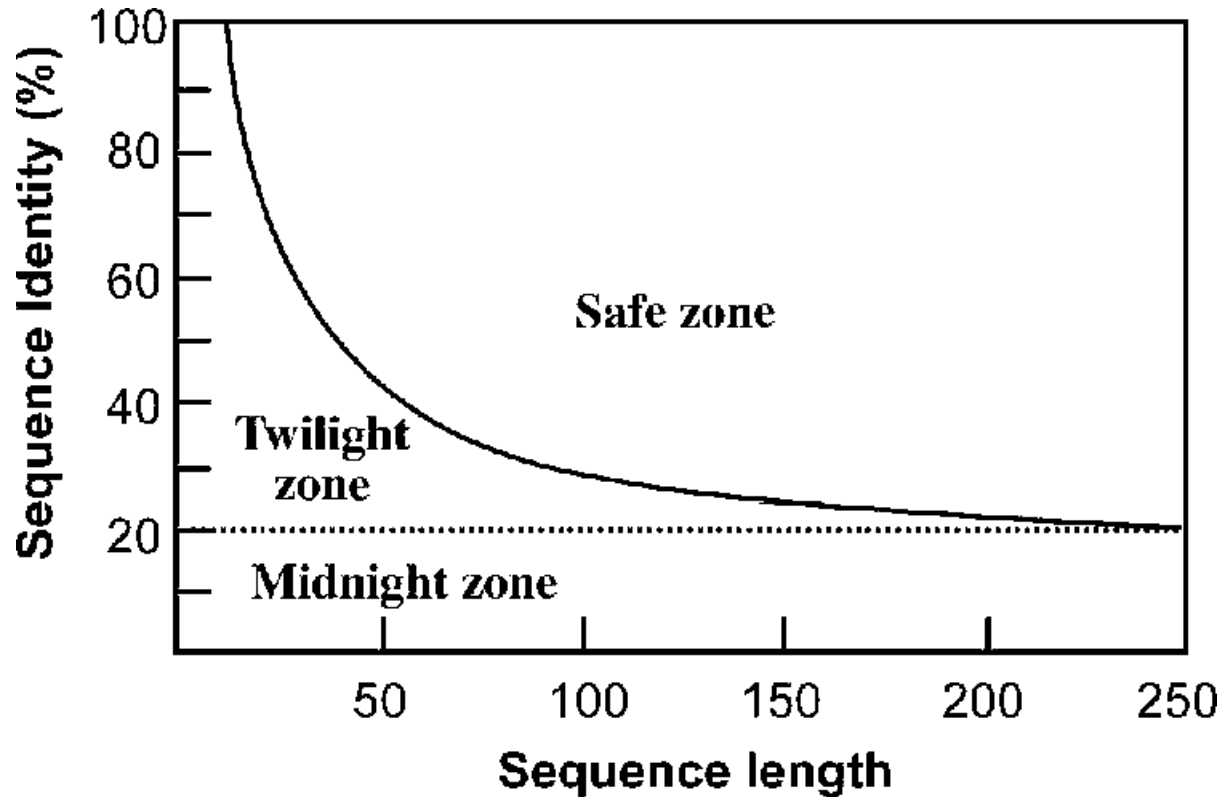
## Target Frequencies

- **Definition:**  $q_{ij}$  reflects substitution likelihood for specific evolutionary distances.
- **Examples:**
  - Human vs. chimp beta globin: High match likelihood (e.g., serine → threonine,  $q_{S,T} = 0.05$ ).
  - Human vs. bacterial globin: Lower match likelihood (e.g.,  $q_{S,T} = 0.4$ ).

<b>A</b>	7																				
<b>R</b>	-10	9																			
<b>N</b>	-7	-9	9																		
<b>D</b>	-6	-17	-1	8																	
<b>C</b>	-10	-11	-17	-21	10																
<b>Q</b>	-7	-4	-7	-6	-20	9															
<b>E</b>	-5	-15	-5	0	-20	-1	8														
<b>G</b>	-4	-13	-6	-6	-13	-10	-7	7													
<b>H</b>	-11	-4	-2	-7	-10	-2	-9	-13	10												
<b>I</b>	-8	-8	-8	-11	-9	-11	-8	-17	-13	9											
<b>L</b>	-9	-12	-10	-19	-21	-8	-13	-14	-9	-4	7										
<b>K</b>	-10	-2	-4	-8	-20	-6	-7	-10	-10	-9	-11	7									
<b>M</b>	-8	-7	-15	-17	-20	-7	-10	-12	-17	-3	-2	-4	12								
<b>F</b>	-12	-12	-12	-21	-19	-19	-20	-12	-9	-5	-5	-20	-7	9							
<b>P</b>	-4	-7	-9	-12	-11	-6	-9	-10	-7	-12	-10	-10	-11	-13	8						
<b>S</b>	-3	-6	-2	-7	-6	-8	-7	-4	-9	-10	-12	-7	-8	-9	-4	7					
<b>T</b>	-3	-10	-5	-8	-11	-9	-9	-10	-11	-5	-10	-6	-7	-12	-7	-2	8				
<b>W</b>	-2	-5	-11	-21	-22	-19	-23	-21	-10	-20	-9	-18	-19	-7	-20	-8	-19	13			
<b>Y</b>	-11	-14	-7	-17	-7	-18	-11	-20	-6	-9	-10	-12	-17	-1	-20	-10	-9	-8	10		
<b>V</b>	-5	-11	-12	-11	-9	-10	-10	-9	-9	-1	-5	-13	-4	-12	-9	-10	-6	-22	-10	8	
	<b>A</b>	<b>R</b>	<b>N</b>	<b>D</b>	<b>C</b>	<b>Q</b>	<b>E</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>L</b>	<b>K</b>	<b>M</b>	<b>F</b>	<b>P</b>	<b>S</b>	<b>T</b>	<b>W</b>	<b>Y</b>	<b>V</b>	

**Figure 3.15 Log-odds matrix for PAM10.** Low PAM values such as this are useful for aligning very closely related sequences. Compare this with the PAM250 matrix (Fig. 3.14) and note that there are larger positive scores for identical matches in this PAM10 matrix and larger penalties for mismatches

# SEQUENCE HOMOLOGY VERSUS SEQUENCE SIMILARITY



The **three zones** of **protein sequence alignments**.

- **Two protein sequences** can be regarded as **homologous** if the **percentage sequence identity** falls in the **safe zone**.
- **Sequence identity** values **below** the **zone boundary**, but **above 20%**, are considered to be in the **twilight zone**, where **homologous relationships** are **less certain**.
- **The region below 20%** is the **midnight zone**, where **homologous relationships** cannot be reliably determined.

# SEQUENCE SIMILARITY VERSUS SEQUENCE IDENTITY

There are **two ways** to **calculate** the **sequence similarity/identity**:

- The **use** of the **overall sequence lengths** of **both sequences**

✓ The **sequence similarity**

$$S = [(L_s \times 2) / (L_a + L_b)] \times 100$$

where **S** is the **percentage sequence similarity**, **L<sub>s</sub>** is the **number** of **aligned residues with similar characteristics**, and **L<sub>a</sub>** and **L<sub>b</sub>** are the **total lengths** of **each individual sequence**.

✓ The **sequence identity** (I%) can be **calculated** in a **similar fashion**:

$$I = [(L_i \times 2) / (L_a + L_b)] \times 100$$

where **L<sub>i</sub>** is the **number** of **aligned identical residues**

# SEQUENCE SIMILARITY VERSUS SEQUENCE IDENTITY

The **second method** of calculation is to **derive** the **percentage of identical/similar residues** over the **full length** of the **smaller sequence** using the formula:

$$I(S)\% = L_{i(s)} / L_a \%$$

where  $L_a$  is the **length** of the **shorter** of the **two sequences**

# METHODS

The **overall goal** of **pairwise sequence alignment**

To **find** the **best pairing** of **two sequences**, such that there is **maximum correspondence** among residues.

To **achieve** this **goal**, **one sequence** needs to be **shifted relative to** the **other** to **find** the **position** where **maximum matches** are found.

**There are two different alignment strategies:**

- **Global alignment**
- **Local alignment**



# Global Alignment and Local Alignment

## Global alignment (beginning to end alignment)

- ✓ In **global alignment**, **two sequences to be aligned** are assumed to be generally **similar over their entire length**.
- ✓ This **method** is more **applicable** for **aligning** two closely related sequences of roughly the **same length**.

For **divergent sequences** and **sequences** of **variable lengths**, this **method** may not be able to generate **optimal results** because **it** fails to recognize **highly similar local regions** between the **two sequences**.

# Global Alignment and Local Alignment

## Local alignment

It only **finds** local regions with the **highest level of similarity** between the **two sequences** and **aligns** these **regions** **without** regard for the **alignment of the rest of the sequence regions**.

- The **two sequences** to be aligned can be of **different lengths**.
- This **approach** is more appropriate for **aligning divergent biological sequences** containing only modules that are **similar**, which are referred to as **domains** or **motifs**.

“.” indicates **identical residue matches**

“.” indicates **similar residue matches**

```
seq1  EARDF-NQYYSSIKRSGSIQ
      .  :  . . . . . . . . . .
seq2  LPKLFIDQYYSSIKRTMG-H
```

## global sequence alignment

```
seq1  NQYYSSIKRS
      . . . . . . . . . .
seq2  DQYYSSIKRT
```

## local sequence alignment

# Alignment Algorithms

Both types of **algorithms** (global and local) can be based on one of the **three methods**:

- The dot matrix method
- The dynamic programming method
- The word method

## Dot Matrix Method

The **most basic sequence alignment method** is the **dot matrix method**, also known as the **dot plot method**.

It is a **graphical way** of comparing two sequences in a **two-dimensional matrix**.

# Dot Matrix Method

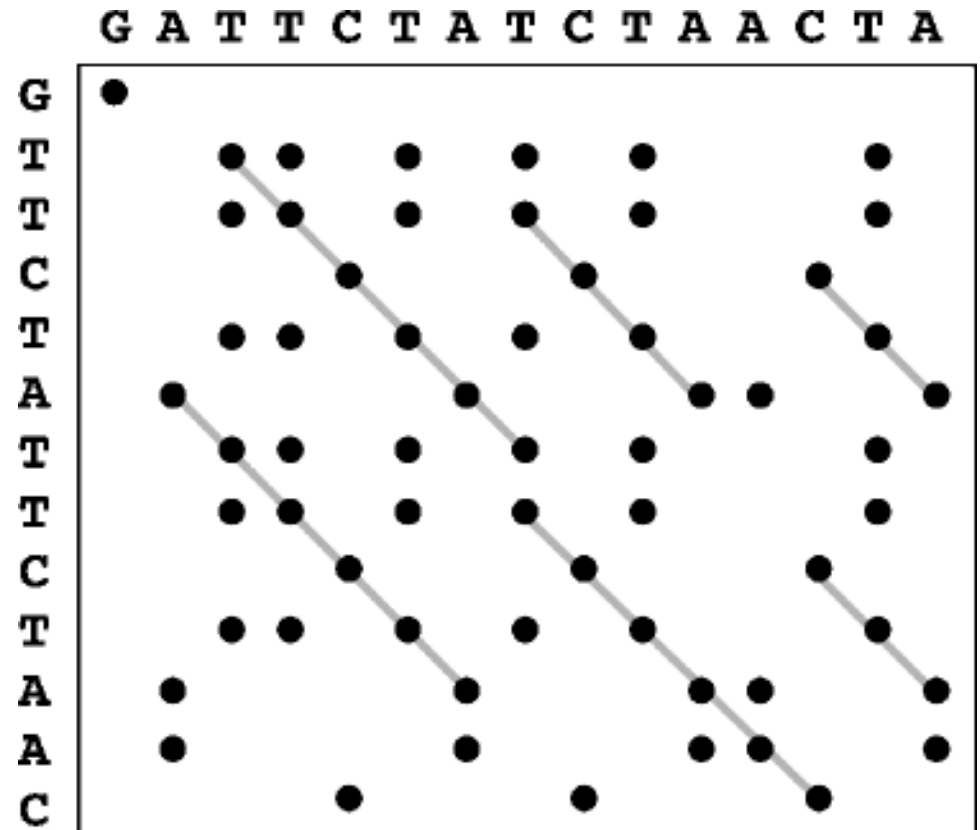
How to work:

Scanning each residue of one sequence for similarity with all residues in the other sequence.

In the matrix:

Matching: ●

Otherwise: Blank



If there are interruptions in the middle of a diagonal line, they indicate insertions or deletions. Parallel diagonal lines within the matrix represent repetitive regions of the sequences.

# Dot Matrix Method

## Problems

### ✓ High noise level

When comparing **large sequences**, **dots** are plotted **all over the graph**, obscuring identification of the **true alignment**.

For **DNA sequences**, the **problem** is particularly **acute** because **there are only four possible characters in DNA** and **each residue** therefore has a **one-in-four chance** of **matching** a **residue** in **another sequence**.

### ✓ A filtering technique to reduce noise level:

#### Window or Tuple strategy

**Instead** of using a **single residue** to scan for similarity, a “**window**” of fixed length covering a stretch of residue pairs is used. **HOW TO WORK?**

**Windows** **slide** across the **two sequences** to compare all possible stretches. **Dots** are **only** placed when a stretch of residues equal to the window size from one sequence **matches completely** with a stretch of another sequence.

# Dot Matrix Method

## ✓ Note:

If the **selected window** size is **too long**, **sensitivity** of the **alignment** is **lost**.

- For **comparing protein sequences**, a **weighting scheme** has to be used to account for similarities of **physicochemical properties** of amino acid residues.

The **dot matrix method** gives a **direct visual statement** of the **relationship** between **two sequences** and helps **easy identification** of the **regions** of **greatest similarities**.

## Applications:

- It is **useful** in **identifying** chromosomal repeats and in **comparing gene order conservation** between **two closely related genomes**.
- It can also be used in **identifying nucleic acid secondary structures** through **detecting self-complementarity** of a **sequence**.

# Dynamic Programming Method

**Dynamic programming** is a **method** that **determines** optimal alignment by matching **two sequences** for all possible pairs of characters between the **two sequences**.

By **searching** for the **set of highest scores** in this **matrix**, the **best alignment** can be **accurately obtained**.

## How to work:

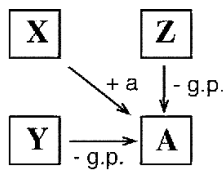
Dynamic programming works by first **constructing** a **two-dimensional matrix** whose **axes** are the **two sequences** to be compared.

The **residue matching** is according to a **particular scoring matrix**.

The **scores** are **calculated** one **row** at a **time**.

This **starts** with the **first row** of **one sequence**, which is used to **scan** through the **entire length** of the **other sequence**, followed by **scanning of the second row**. The matching scores are calculated.

The **scanning** of the **second row** **takes into account** the **scores already obtained** in the **first round**. The **best score** is **put** into the **bottom right corner** of an **intermediate matrix**.



A is the maximum score from one of the three directions plus matching score at the current position

	A	T	T	G	C
A	1	0	0	0	0
G					
G					
C					



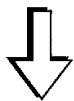
	A	T	T	G	C
A	1	0	0	0	0
G	0	1			
G					
C					



	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	
G					
C					



	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1		
G					
C					



	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	2
G	0	1	1	3	3
C	0	1	1	3	4



	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	2
G	0	1	1	3	3
C	0	1	1	3	4

This process is **iterated** until values for **all** the **cells** are **filled**. Thus, the **scores** are **accumulated** along the **diagonal** going from the **upper left corner** to the **lower right corner**.

**Next step:** To **find** the **path** that **represents** the **optimal alignment**. (tracing back through the matrix in reverse order).

The **best matching path** is the **one** that has the **maximum total score**. If **two** or **more paths** reach the same highest score, **one** is chosen **arbitrarily** to represent the **best alignment**.

**Final alignment**  
**Total score = 9**

A	T	T	G	C
A	-	G	G	C



## Note:

The **path** can also **move** horizontally or vertically at a certain point, which corresponds to introduction of a **gap** or an **insertion** or **deletion** for one of the two sequences.

## Gap Penalties

- Performing **optimal alignment** between sequences often involves **applying gaps** that represent **insertions** and **deletions**.
- In natural evolutionary processes:
  - ✓ **Insertions** and **deletions** are relatively rare in comparison to substitutions,
  - ✓ So, introducing gaps should be made more difficult computationally, reflecting the rarity of insertional and deletional events in evolution.
- There is **no evolutionary theory** to determine **a precise cost** for introducing **insertions** and **deletions**:
  - ✓ So, assigning penalty values can be more or less **arbitrary**.

## Note:

- ✓ If the **penalty values** are set **too low**, gaps can become too numerous to **allow** even **nonrelated sequences** to be **matched up** with **high similarity scores**.
- ✓ If the **penalty values** are set **too high**, gaps may become too difficult to appear, and **reasonable alignment** cannot be achieved, which is also **unrealistic**.

Based on difference between opening and extending gap cost:

- **Affine gap penalties:** **gap opening** should have a **much higher penalty** than **gap extension**.
- ✓ The **total gap penalty** ( $W$ ) is a **linear function** of **gap length**, which is calculated using the formula:

$$W = \gamma + \delta \times (k - 1)$$

where  $\gamma$  is the **gap opening penalty**,  $\delta$  is the **gap extension penalty**, and  $k$  is the **length** of the **gap**.

- **Constant gap penalty:** assigns the **same score** for **each gap position** regardless whether it is **opening** or **extending**. However, **this penalty scheme** has been found to be **less realistic** than the affine penalty.

### **Note:**

**Gaps** at the **terminal regions** are **often** treated with **no penalty** because **in reality** many true homologous sequences **are** of **different lengths**. Consequently, **end gaps** can be **allowed** to **be free** to avoid getting unrealistic alignments.

# Global Sequence Alignment: Algorithm of Needleman and Wunsch

One of the **first** and **most important** algorithms for aligning **two protein sequences** was described by **Needleman and Wunsch** (1970).

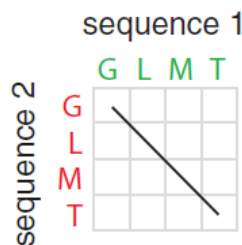
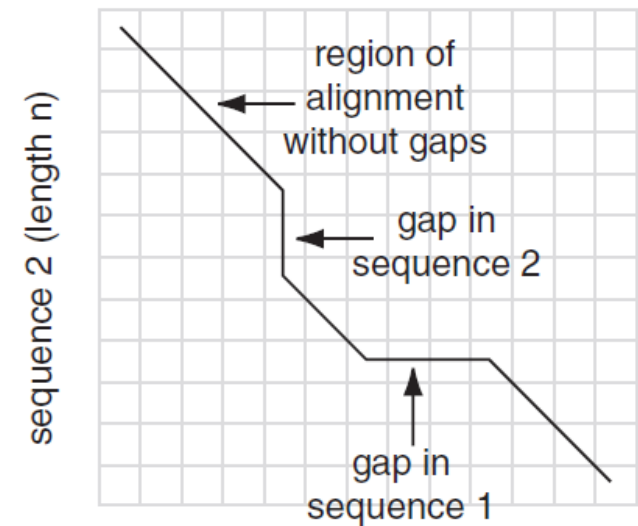
- The **result** is **optimal**, **but not all possible alignments need to be evaluated**.
- An **exhaustive pairwise comparison** would be **too computationally expensive** to perform.

Describing the Needleman–Wunsch approach to **global sequence alignment** in three steps:

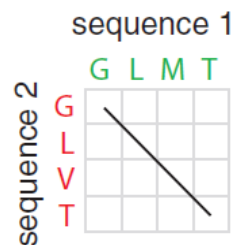
(1) **setting up a matrix**; (2) **scoring the matrix**; and (3) **identifying the optimal alignment**.

## Step 1: Setting Up a Matrix

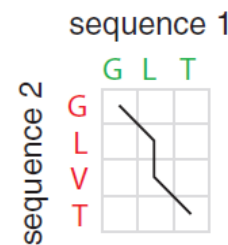
sequence 1 (length m)



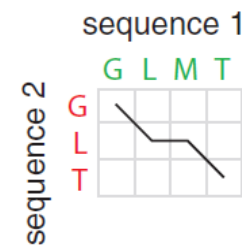
1 GLMT  
2 GLMT



1 GLMT  
2 GLVT



1 GL-T  
2 GLVT



1 GLMT  
2 GL-T

## Step 2: Scoring the Matrix

We set up two matrices:

an **amino acid identity matrix** and then a **scoring matrix**.

- ❑ We create a **matrix** of dimensions  $m + 1$  by  $n + 1$  (for the first and second sequences on the  $x$ - and  $y$ -axes respectively).

- ❑ **Gap penalties** (here having a value of **-2** for **each gap position**) are placed along the first row and column.

- This will allow us to introduce a terminal gap of any length.

- ❑ We **fill** in positions of identity (gray-filled cells); this is called an **identity matrix**.

(a)

		Sequence 2								
		F	M	D	T	P	L	N	E	
Sequence 1		0	-2	-4	-6	-8	-10	-12	-14	-16
	F	-2								
	K	-4								
	H	-6								
	M	-8								
	E	-10								
	D	-12								
	P	-14								
	L	-16								
	E	-18								

## Step 2: Scoring the Matrix

### □ Defining a scoring system

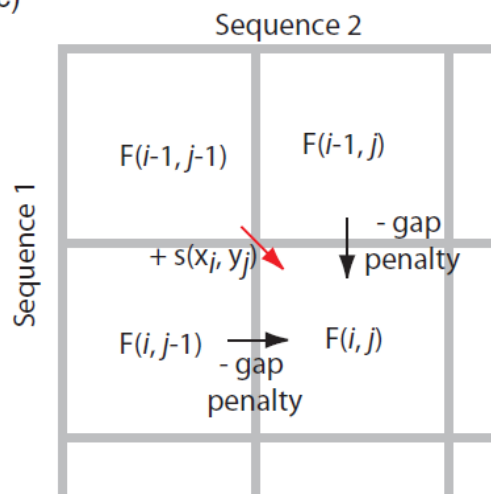
(a)

		Sequence 2								
		F	M	D	T	P	L	N	E	
Sequence 1		0	-2	-4	-6	-8	-10	-12	-14	-16
	F	-2								
	K	-4								
	H	-6								
	M	-8								
	E	-10								
	D	-12								
	P	-14								
	L	-16								
	E	-18								

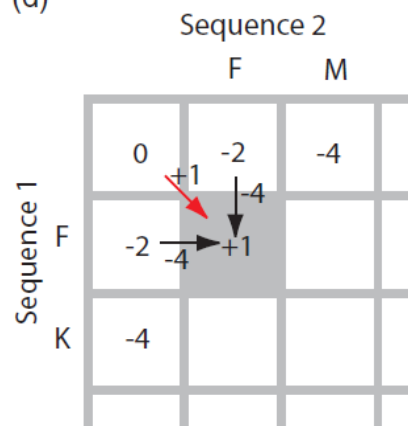
$$\text{Score} = \text{Max} \begin{cases} F(i-1, j-1) + s(x_i, y_i) \\ F(i-1, j) - \text{gap penalty} \\ F(i, j-1) - \text{gap penalty} \end{cases}$$

Score (this example) = +1 (match)  
-2 (mismatch)  
-2 (gap penalty)

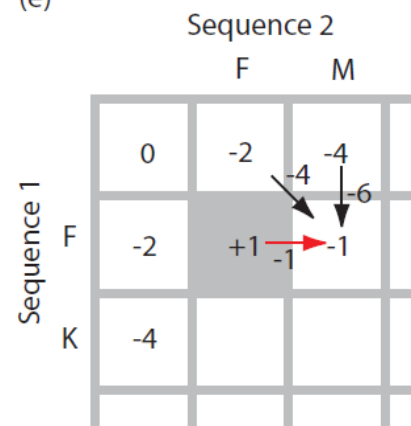
(c)



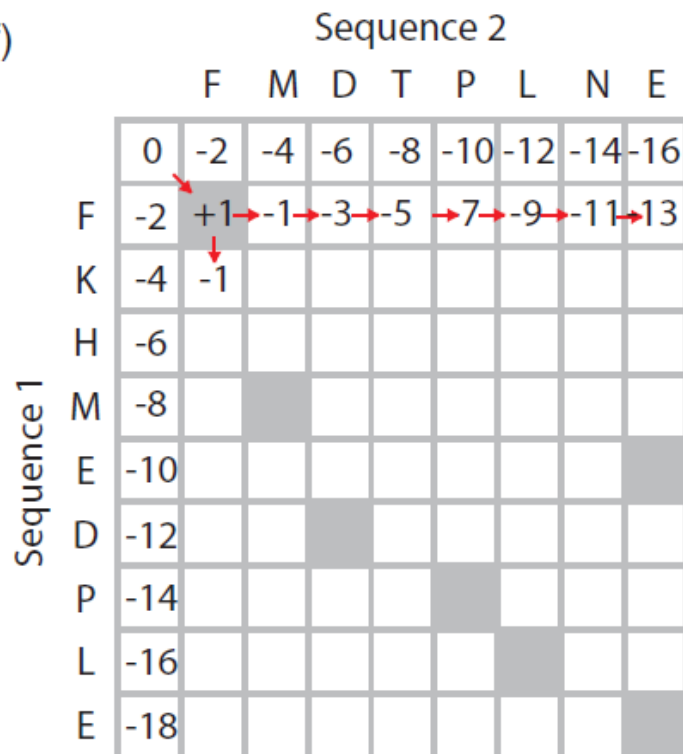
(d)



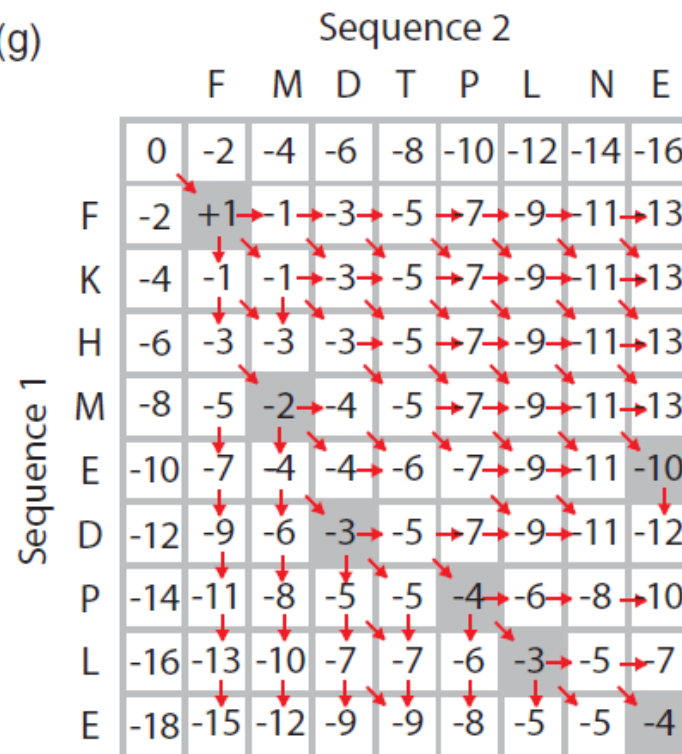
(e)



(f)



(g)



## Identifying the Optimal Alignment

(a)

a)

Sequence 2

Sequence 1

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1	-1	-3	-5	-7	-9	-11	-13
H	-6	-3	-3	-3	-5	-7	-9	-11	-13
M	-8	-5	-2	-4	-5	-7	-9	-11	-13
E	-10	-7	-4	-4	-6	-7	-9	-11	-10
D	-12	-9	-6	-3	-5	-7	-9	-11	-12
P	-14	-11	-8	-5	-5	-4	-6	-8	-10
L	-16	-13	-10	-7	-7	-6	-3	-5	-7
E	-18	-15	-12	-9	-9	-8	-5	-5	-4

(b)

b)

Sequence 2

Sequence 1

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1	-1	-3	-5	-7	-9	-11	-13
H	-6	-3	-3	-3	-5	-7	-9	-11	-13
M	-8	-5	-2	-4	-5	-7	-9	-11	-13
E	-10	-7	-4	-4	-6	-7	-9	-11	-10
D	-12	-9	-6	-3	-5	-7	-9	-11	-12
P	-14	-11	-8	-5	-5	-4	-6	-8	-10
L	-16	-13	-10	-7	-7	-6	-3	-5	-7
E	-18	-15	-12	-9	-9	-8	-5	-5	-4

(c)

		+1	-1	-3	-2	-4	-3	-5	-4	-3	-5	-4
Sequence 1		F	K	H	M	E	D	-	P	L	-	E
Sequence 2		F	-	-	M	-	D	T	P	L	N	E

## Local Sequence Alignment – Smith-Waterman Algorithm

- **Purpose:** Aligns subsets of protein/DNA sequences rigorously (Smith & Waterman, 1981).
- **Applications:** Database searches (e.g., aligning protein domains, BLAST, Chapter 4).
- **Key Feature:** No penalties for starting/ending alignment internally (unlike global alignment).

NP_824492.1	1	MCGDMTVHTVEYIRYRIPEQQSAEFLAAYTRAAQQLAAAPQCVDYELARC	50
NP_337032.1	1		0
NP_824492.1	51	EEDFEHFVLRITWTSTEDHIEGFRKSELFPDFLAEIRPYISSIEEMRHYK	100
NP_337032.1	1		0
NP_824492.1	101	PTTVRGTTGAAVPTLYAWAGGAEAFARLTEVFYEKVLKDDVLAPVFEGMAP	150
NP_337032.1	1	MEGMDQMPKSFYDAVGGAKTFDAIVSRFYAQVAEDEVLRVY----P	43
NP_824492.1	151	EH-----AAHVALWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRR	195
NP_337032.1	44	EDDLAGAEERLRMFLEQYWGGPRTYSE-QRGHPRLMRHAPFRISLIERD	92
NP_824492.1	196	RWVNLLQDAADDAGLPT-DAEFRSAFLAYAEWGTRLAVYFSGPDVAPPAE	244
NP_337032.1	93	AWLRCHMTAVASIDSETLDDEHRRELLDYLEMAAHSV--NSPF	134
NP_824492.1	245	QPVPQWSWGAMPPYQP	260
NP_337032.1	135		134



## Matrix Construction

- **Structure:** Matrix of size  $(m+1) \times (n+1)$  for sequences of lengths  $m$  and  $n$ .
  - Extra row (top) and column (left) filled with zeros.
- **Scoring Rules** (Fig. 3.24):
  - *Maximum of four values:*
    - Diagonal score  $(i-1, j-1)$  + match/mismatch score  $(s[i,j])$ .
    - Left score  $(i, j-1)$  – gap penalty.
    - Above score  $(i-1, j)$  – gap penalty.
    - Zero (ensures no negative scores).

NP_824492.1	113	TLYAWAGGAEAFARLTEVFYKVLKDDVLAPVFEGMAPEH-----AAHVA	157
		:. ...   : .:.:. . .: .: : .. :   . ....	
NP_337032.1	10	SFYDAVGGAKTFDAIVSRFYAQVAEDEVLRVY----PEDDLAGAEERLR	55
NP_824492.1	158	LWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRRRWVNLLQDAADD	207
		: .:.:   .     . .:.:.: .... :..: .. :..:.. ...	
NP_337032.1	56	MFLEQYWGGPRTYSE-QRGHPRLMRHAPFRISLIERDAWLRCMHTAVAS	104
NP_824492.1	208	AGLPT-DAEFRSAFLAYAE	225
		....   . . ... . .	
NP_337032.1	105	IDSETLDDEHRRELLDYLE	123

## Alignment Process

- **Steps:**
  - Identify highest matrix value (e.g., **3.3** in the Fig.) as alignment end.
  - Trace back diagonally to a zero value to define alignment start.
- **Outcome:** Aligns only optimal regions, not necessarily entire sequences.

(a)

		Sequence 1												
		C	A	G	C	C	U	C	G	C	U	U	A	G
Sequence 2		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	A	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
	A	0.0	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.7
	U	0.0	0.0	0.0	0.7	0.3	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.7
	G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.7	0.7	1.0
	C	0.0	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3
	C	0.0	1.0	0.7	0.0	1.0	3.0	1.7	1.3	1.0	1.3	1.7	0.3	0.0
	A	0.0	0.0	2.0	0.7	0.3	1.7	2.7	1.3	1.0	0.7	1.0	1.3	1.3
	U	0.0	0.0	0.7	1.7	0.3	1.3	2.7	2.3	1.0	0.7	1.7	2.0	1.0
	U	0.0	0.0	0.3	0.3	1.3	1.0	2.3	2.3	2.0	0.7	1.7	2.7	1.7
	G	0.0	0.0	0.0	1.3	0.0	1.0	1.0	2.0	3.3	2.0	1.7	1.3	2.3
	A	0.0	0.0	1.0	0.0	1.0	0.3	0.7	0.7	2.0	3.0	1.7	1.3	2.3
	C	0.0	1.0	0.0	0.7	1.0	2.0	0.7	1.7	1.7	3.0	2.7	1.3	1.0
	G	0.0	0.0	0.7	1.0	0.3	0.7	1.7	0.3	2.7	1.7	2.7	2.3	1.0
	G	0.0	0.0	0.0	1.7	0.7	0.3	0.3	1.3	1.3	2.3	1.3	2.3	2.0

## Example – Nucleic Acid Alignment

- **Sequences:** RNA (CAGCCUCGCUUAG, AAUGCCAUUGACGG)
- **Scoring:** +1 (match), -1/3 (mismatch), -1.3 (gap).
- **Result:** Local alignment with identities, mismatch, and gap (Fig. b).
- **Comparison:** Global alignment includes entire sequences (Fig. c).

(a)

		Sequence 1													
		C	A	G	C	C	U	C	G	C	U	U	A	G	
Sequence 2	A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	A	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	
	U	0.0	0.0	0.0	0.7	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	
	G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	0.7	0.7	
	C	0.0	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3	
	C	0.0	1.0	0.7	0.0	1.0	3.0	1.7	1.3	1.0	1.3	1.7	0.3	0.0	
	A	0.0	0.0	2.0	0.7	0.3	1.7	2.7	1.3	1.0	0.7	1.0	1.3	1.3	
	U	0.0	0.0	0.7	1.7	0.3	1.3	2.7	2.3	1.0	0.7	1.7	2.0	1.0	
	U	0.0	0.0	0.3	0.3	1.3	1.0	2.3	2.3	2.0	0.7	1.7	2.7	1.7	
	G	0.0	0.0	0.0	1.3	0.0	1.0	1.0	2.0	3.3	2.0	1.7	1.3	2.3	
	A	0.0	0.0	1.0	0.0	1.0	0.3	0.7	0.7	2.0	3.0	1.7	1.3	2.3	
	C	0.0	1.0	0.0	0.7	1.0	2.0	0.7	1.7	1.7	3.0	2.7	1.3	2.0	
	G	0.0	0.0	0.7	1.0	0.3	0.7	1.7	0.3	2.7	1.7	2.7	2.3	1.0	
	G	0.0	0.0	0.0	1.7	0.7	0.3	0.3	1.3	1.3	2.3	1.3	2.3	2.0	

(b)

sequence 1    GCC-UCG  
sequence 2    GCCAUUG

(c)

sequence 1    CA-GCC-UCGCUUAG  
sequence 2    AAUGCCAUUGACG-G

## Example – Protein Alignment

- **Comparison:** Local vs. global alignment of two proteins (Fig. 3.23).
  - **Local** (Fig. 3.23b): Shorter, higher % identity/similarity.
  - **Global** (Fig. 3.23a): Includes all residues, may miss conserved regions (arrowheads).
- **Note:** BLAST may exclude some matching residues based on parameters.

# Example – Protein Alignment

NP_824492.1	1	MCGDMTVHTVEYIRYRIPEQQSAEFLAAYTRAAQLAAAPQCVDYELARC	50
NP_337032.1	1		0
NP_824492.1	51	EEDFEHFVLRITWTSTEDHIEGFRKSELFPDFLAEIRPYISSIEEMRHYK	100
NP_337032.1	1		0
NP_824492.1	101	PTTVRGTTGAAVPTLYAWAGGAEAFARLTEVFYKVLKDDVLAPVFEGMAP	150
NP_337032.1	1	MEGMDQMPKSFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----P	43
NP_824492.1	151	EH-----AAHVALWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRR	195
NP_337032.1	44	EDDLAGAEERLRMFLEQYWGGPRTYSE-QRGHPRLMRHAPFRISLIERD	92
NP_824492.1	196	RWVNLLQDAADDAGLPT-DAEFRSAFLAYAEWGTRLAVYFSGPDAVPPAE	244
NP_337032.1	93	AWLRCMHTAVASIDSETLDDEHRRELLDYLEMAAHSLV--NSPF	134
NP_824492.1	245	QPVPQWSWGAMPPYQP	260
NP_337032.1	135		134
NP_824492.1	113	TLYAWAGGAEAFARLTEVFYKVLKDDVLAPVFEGMAPEH-----AAHVA	157
NP_337032.1	10	SFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----PEDDLAGAEERLR	55
NP_824492.1	158	LWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRRRWVNLLQDAADD	207
NP_337032.1	56	MFLEQYWGGPRTYSE-QRGHPRLMRHAPFRISLIERDAWLRMHTAVAS	104
NP_824492.1	208	AGLPT-DAEFRSAFLAYAE	225
NP_337032.1	105	IDSETLDDEHRRELLDYLE	123

## Box:

### Algorithms and Programs Overview

- **Algorithm:** Structured procedure in a computer program (Sedgewick, 1988).
  - **Example:** Pairwise alignment algorithms.
- **Program:** Instructions using algorithms to solve tasks.
  - **Example:** BLAST for sequence alignments (Chapters 3–5).
  - **Other:** Phylogenetic tree programs (Chapter 7).

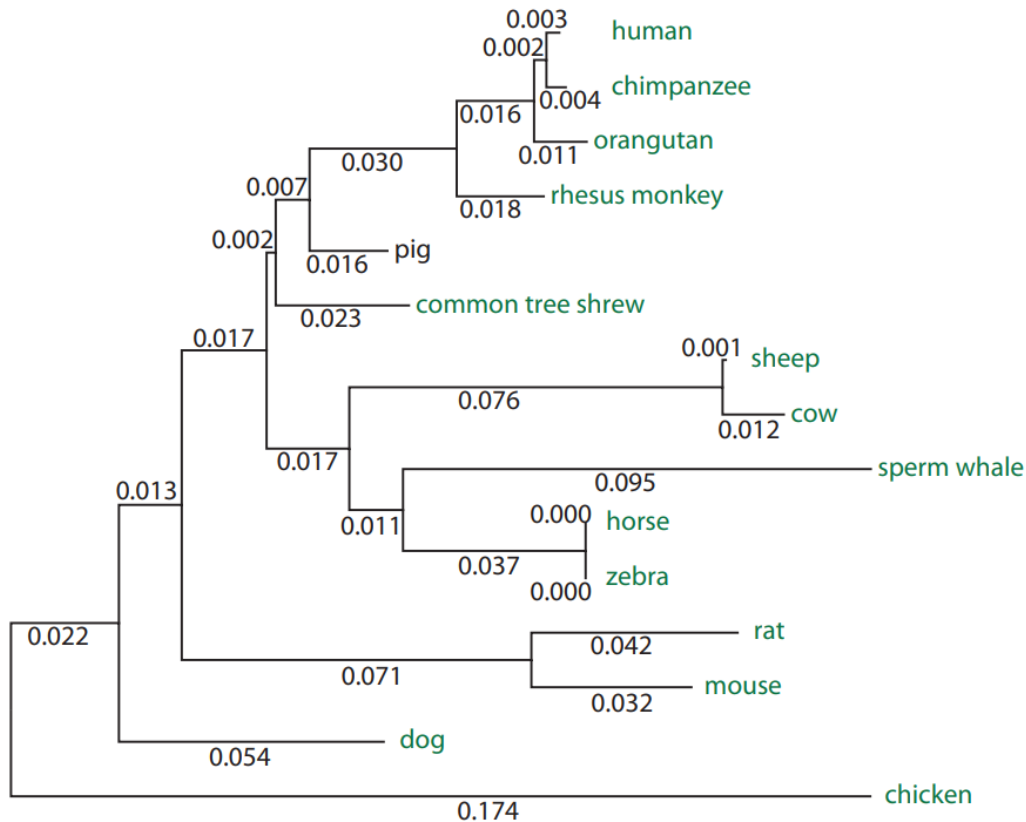
### Importance in Bioinformatics

- **Need:** Programs automate millions of operations for bioinformatics tasks.
- **Diversity:** Hundreds of programs/algorithms, each for specific tasks.
- **Limitation:** Algorithms for pairwise alignment may not scale to large databases (e.g., 10M sequences).

## Box:

### Challenges and Heuristic Algorithms

- **Complexity:** Exhaustive analysis of complex problems (e.g., arranging 13 proteins in Fig. 3.2 into a tree) requires vast memory/time.
- **Solution:** Heuristic algorithms approximate optimal solutions quickly.
  - **Example:** Finding optimal phylogenetic tree in seconds (Chapter 7).



## Speed Challenges of Smith-Waterman

- **Issue:** Smith-Waterman algorithm (optimal local alignment) is slow for large-scale database searches.
- **Time Complexity:**
  - **Needleman-Wunsch:**  $O(mn)$  steps.
  - **Smith-Waterman:**  $O(m^2n)$  steps (Fig. 3.24).
- **Scaling Problem:** For database size  $N$  and query length  $m$ , time  $\propto m \times N$ .

**N-notation:** Most algorithms have a parameter  $N$  that refers to the **number of data items to be processed** (see Sedgewick, 1988). This parameter can **greatly affect** the **time required** for the **algorithm** to perform a task.

**O-notation:** Another useful descriptor is O-notation (called “**big-Oh notation**”), which approximates the **upper bounds** of the **running time** of an algorithm.



## Algorithm Performance

- **Parameter N:** Number of data items impacts running time (Sedgewick, 1988).
  - **Linear (N):** Doubling N doubles time.
  - **Quadratic (N<sup>2</sup>):** N = 1000 → 1M operations.
- **Improvements:** Gotoh (1982), Myers & Miller (1988) reduced time/space requirements.

## Rapid Alternatives – FASTA and BLAST

- **Algorithms:** FASTA (Pearson & Lipman, 1988) and BLAST (Altschul et al., 1990).
- **Type:** Heuristic (Box 3.3), sacrifice some sensitivity for speed.
- **Advantage:** Faster than Smith-Waterman by scanning for likely matches first.

## FASTA Algorithm Steps

- **Process** (Pearson & Lipman, 1988):
  - **Lookup Table:** Scan database for short matches (ktup, e.g., 3 amino acids).
  - **Rescore:** Top 10 segments scored with matrix (e.g., PAM250).
  - **Join Regions:** Combine high-scoring regions from same proteins.
  - **Optimize:** Perform global (Needleman-Wunsch) or local (Smith-Waterman) alignment on best matches.
- **Efficiency:** Uses dynamic programming selectively for rapid results.

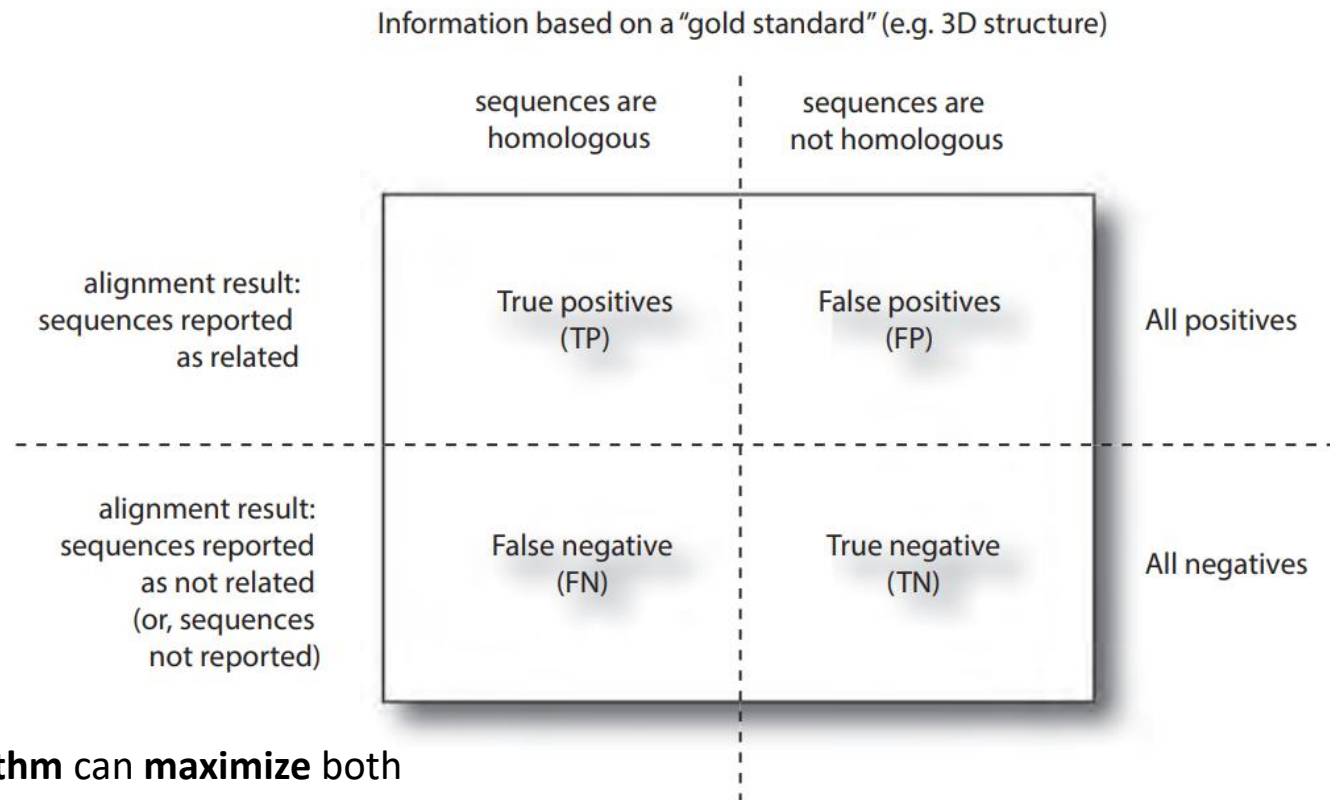
## Statistical Significance of Pairwise Alignments

- **Question:** Is an alignment statistically significant?
- **Context:** Evaluates local and global alignments.
- **Challenge:** Low identity (e.g., 20–25%) requires statistical tests to confirm significance.

## True vs. False Alignments

- **Alignment Outcomes** (Fig. 3.26):
  - **True Positives:** Homologous proteins correctly aligned.
  - **False Positives:** Non-homologous proteins aligned by chance.
  - **True Negatives:** Unrelated sequences correctly unaligned.
  - **False Negatives:** Homologous sequences incorrectly unaligned (low score).

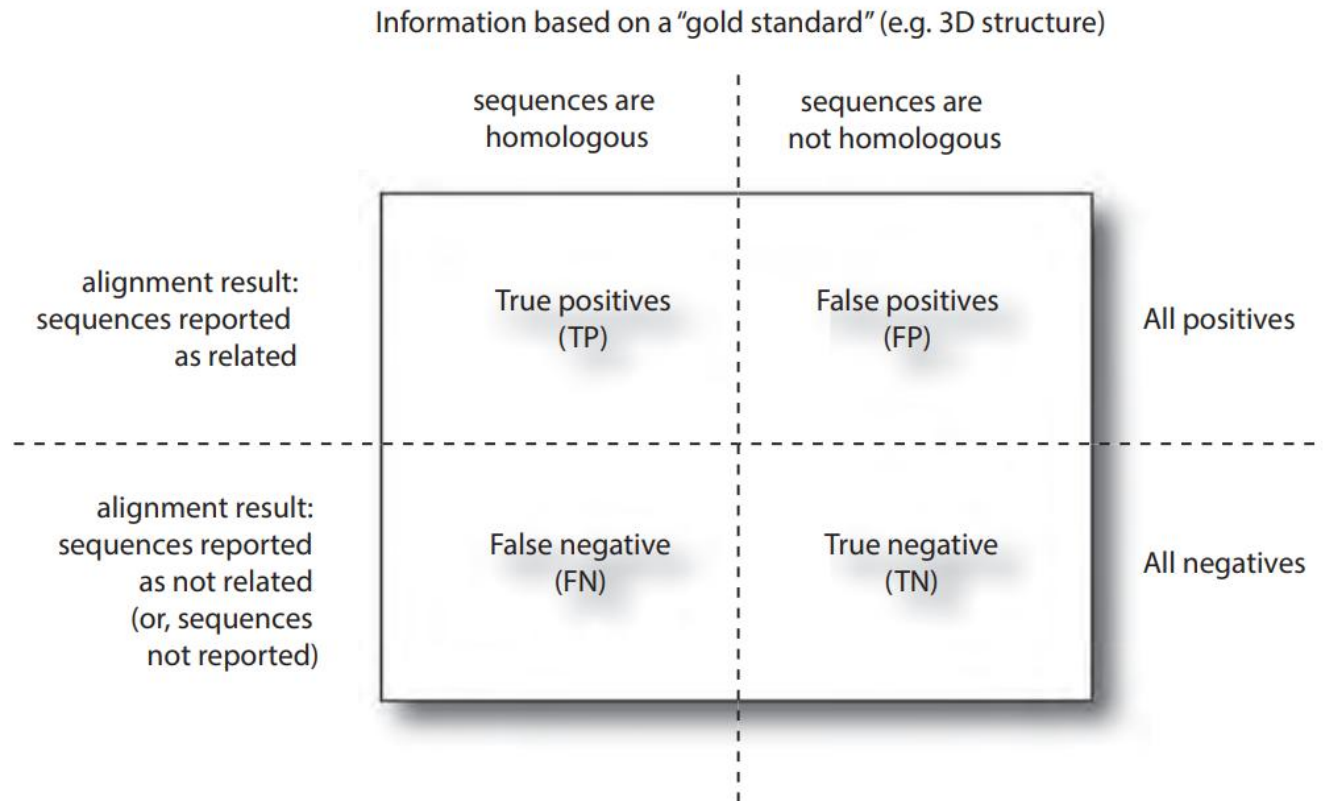
Statistical analyses of alignments provide the main method of evaluating whether an alignment represents a true positive, that is, an alignment of homologous sequences.



Ideally, an alignment algorithm can maximize both sensitivity and specificity.

## Sensitivity and Specificity

- **Goal:** Maximize sensitivity and specificity (Fig. 3.26).
- **Sensitivity:** True positives / (True positives + False negatives).
  - *Measures ability to identify homologous sequences.*
- **Specificity:** True negatives / (True negatives + False positives).
  - *Measures accuracy in identifying non-homologous sequences.*



Dayhoff Model Step 5 (of 7): PAM250 and Other PAM Matrices

The PAM1 matrix was based upon the alignment of closely related protein sequences, having an average of 1% change. To ensure that the multiple alignments were valid, proteins within a family were at least 85% identical.

replacement amino acid

original amino acid

PAM0	A	R	N	D	C	Q	E	G
A	100	0	0	0	0	0	0	0
R	0	100	0	0	0	0	0	0
N	0	0	100	0	0	0	0	0
D	0	0	0	100	0	0	0	0
C	0	0	0	0	100	0	0	0
Q	0	0	0	0	0	100	0	0
E	0	0	0	0	0	0	100	0
G	0	0	0	0	0	0	0	100

original amino acid

PAM $\infty$	A	R	N	D	C	Q	E	G
A	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7
R	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1
N	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
D	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
C	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Q	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
E	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
G	8.9	8.9	8.9	8.9	8.9	8.9	8.9	8.9

# STATISTICAL SIGNIFICANCE OF SEQUENCE ALIGNMENT

An important question in sequence alignment:

a certain degree of similarity can occur by random chance or the alignment is indeed statistically sound.

The statistical test for the relatedness of two sequences can be performed using the following procedure:

- **Optimal alignment** between **two given sequences**.
- **Generation unrelated sequences** of the **same length** through a **randomization, process** in which **one** of the **two sequences** is **randomly shuffled**.
- **Calculation** of a **new alignment score** for the **shuffled sequence pair**.
- **Repeating shuffling** and **obtaining** more **such scores**.
- **Generating parameters** for the **extreme distribution** by using the **pool of alignment scores** from the **shuffled sequences**.
- **Comparing** the **original alignment score** **against** the **distribution of random alignments** to **determine** whether the **score** is **beyond random chance**.

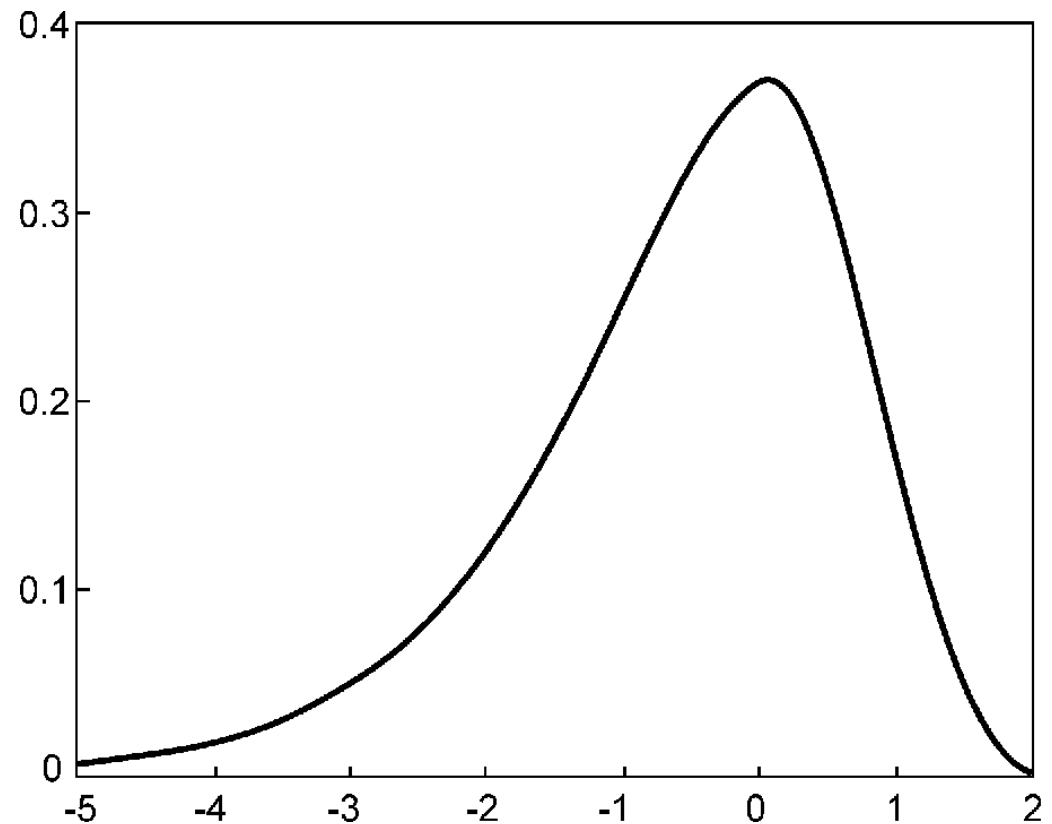
# STATISTICAL SIGNIFICANCE OF SEQUENCE ALIGNMENT

If the **score** is **located** in the **extreme margin** of the **distribution**, that means that the **alignment** between the **two sequences** is **unlikely due to random chance** and is **thus** considered **significant**.

A ***P*-value** is given to indicate the **probability** that the **original alignment** is due to **random chance**.

$$P = 1 - e^{-Kmn e^{-\lambda x}}$$

- **m** and **n**: **sequence lengths**,
- **$\lambda$** : **scaling factor** for the **scoring matrix**
- **K**: **constant** that **depends** on the **scoring matrix** and **gap penalty** combination that is used.





# STATISTICAL SIGNIFICANCE OF SEQUENCE ALIGNMENT

A  $P$ -value resulting from the test provides a much more reliable indicator of possible homologous relationships than using percent identity values.

## Interpretation of $P$ -values

- $P\text{-values} < 10^{-100}$ : an exact match between the two sequences
- $10^{-50} < P\text{-values} < 10^{-100}$ : nearly identical match
- $10^{-5} < P\text{-values} < 10^{-50}$ : sequences having clear homology
- $10^{-5} < P\text{-value} < 10^{-1}$ : possible distant homologs
- $P\text{-values} > 10^{-1}$ : two sequence may be randomly related

However, the **caveat** is that sometimes **truly related protein sequences** may **lack** the **statistical significance** at the **sequence level** owing to fast divergence rates. Their **evolutionary relationships** can **nonetheless** be revealed at the **three-dimensional structural level**.