BIO309: Computational Biology, 2025

Final

Zhengyu LIANG, PhD (梁征宇)

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

BIO309: Computational Biology, 2025

Lecture 1: Introduction

Zhengyu LIANG, PhD (梁征宇)

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Grading System

- Attendance and Class Performance（10% + 10%）
- Class Practice/Programming/Assignments（30%）
- Research Project/Final Presentation（20%）
- Final Examination, Letter Grading（30%）
- Quiz & Mid-Term Test（No）

签到 + 课堂

上机 + 作业

实践 + 报告

闭卷考试

Computational Biology

Contents hide

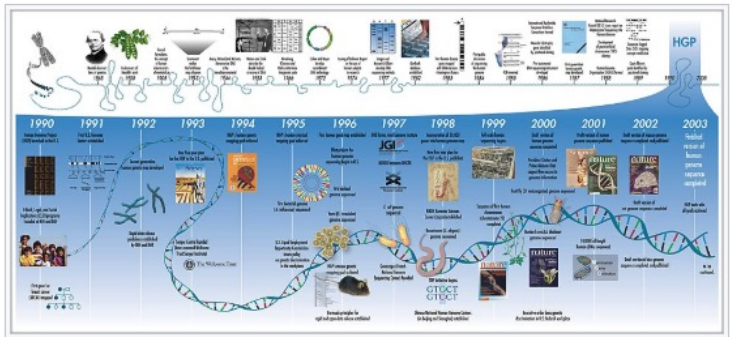
Computational biology

26 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Not to be confused with Bioinformatics or Biological computing.



This timeline displays the year-by-year progress of the Human Genome Project in the context of genetics since 1865. Starting in 1990, by 1999, Chromosome 22 became the first human chromosome to be completely sequenced.

Computational biology refers to the use of techniques in computer science, data analysis, mathematical modeling and computational simulations to understand biological systems and relationships.[1] An intersection of computer science, biology, and data science, the field also has foundations in applied mathematics, molecular biology, cell biology, chemistry, and genetics.[2]

History [edit]

Bioinformatics, the analysis of informatics processes in biological systems, began in the early 1970s. At this time, research in artificial intelligence was using network models of the human brain in order to generate new algorithms. This use of biological data pushed biological researchers to use computers to evaluate and compare large data sets in their own field.[3]

目录 隐藏

计算生物学 [编辑]

26种语言

条目 讨论 大陆简体 阅读 编辑 查看历史 工具

维基百科，自由的百科全书

计算生物学（Computational Biology）是生物学的一个分支。根据美国国家卫生研究所（NIH）的定义，它是指开发和应用数据分析及理论的方法、数学建模和计算机仿真技术，用于生物学、行为学和社会群体系统的研究的一门学科。该领域被广泛定义，包括计算机科学，应用数学，动画，统计学，生物化学，化学，生物物理学，分子生物学，遗传学，基因组学，生态学，进化，解剖学，神经科学和科学可视化的基础。

计算生物学与生物计算学不同，生物计算是计算机科学和计算机工程的子领域，使用生物工程和生物学建造计算机，但是类似于生物信息学，这是一个跨学科的科学，使用计算机存储和处理生物数据。

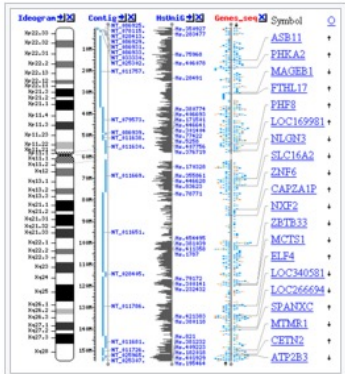
学科历史 [编辑]

主要研究内容 [编辑]

生物序列的片断拼接 [编辑]

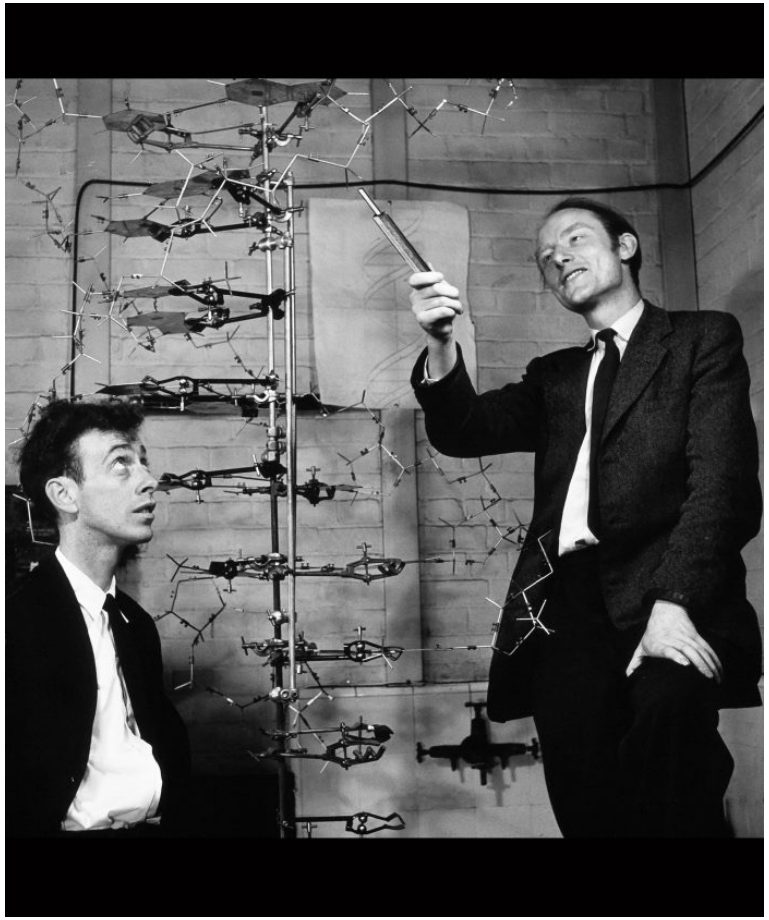
序列比对 [编辑]

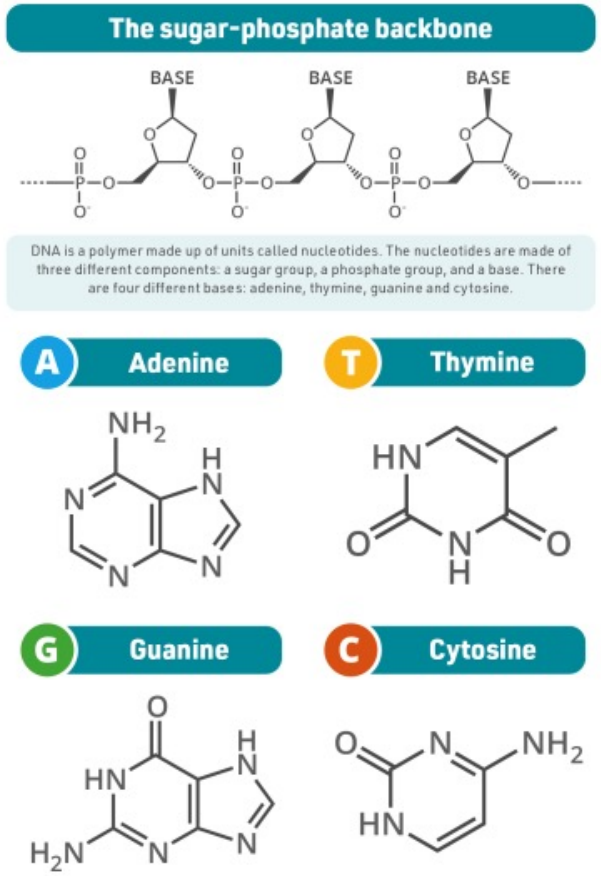
序列比对所研究的基本问题是两个或多个序列间的相似性。序列比对是计算生物学的基本问题之一。对序列的两两比对，已经有了基于动态规划的较成熟的算法以及建立在此基础上的软件包BLAST。对于两个序列的局部比对问题，可以用史密斯–沃特曼算法来解决。多重序列的比对目前还缺乏快速并且十分有效的算法。



部分测序的基因组。

The origins of CB/Bioinformatics

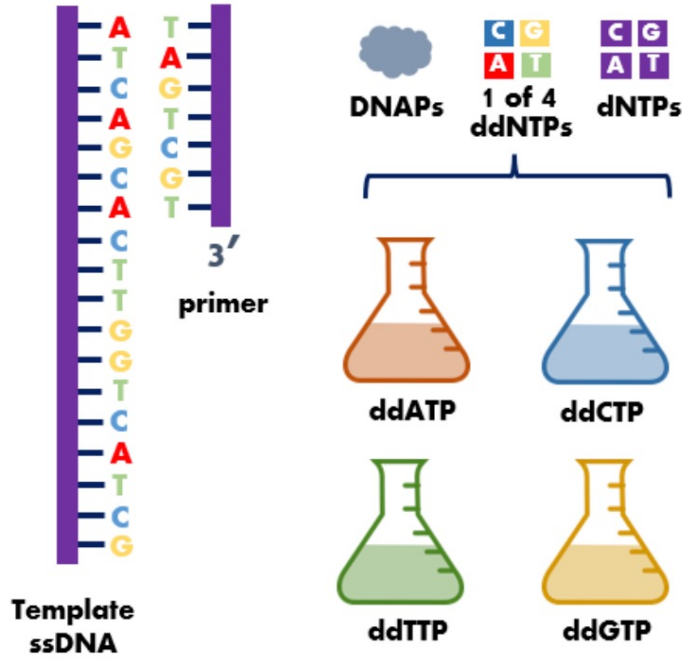




On February 28, 1953, James Watson and Francis Crick, **with help from Rosalind Franklin's X-ray diffraction data**, announced the discovery of double-helical structure of DNA.

Sanger Sequencing

- In 1977, Frederick Sanger developed Sanger Sequencing via chain-termination method, and sequenced the first complete genome: one of bacteriophage ϕ174.



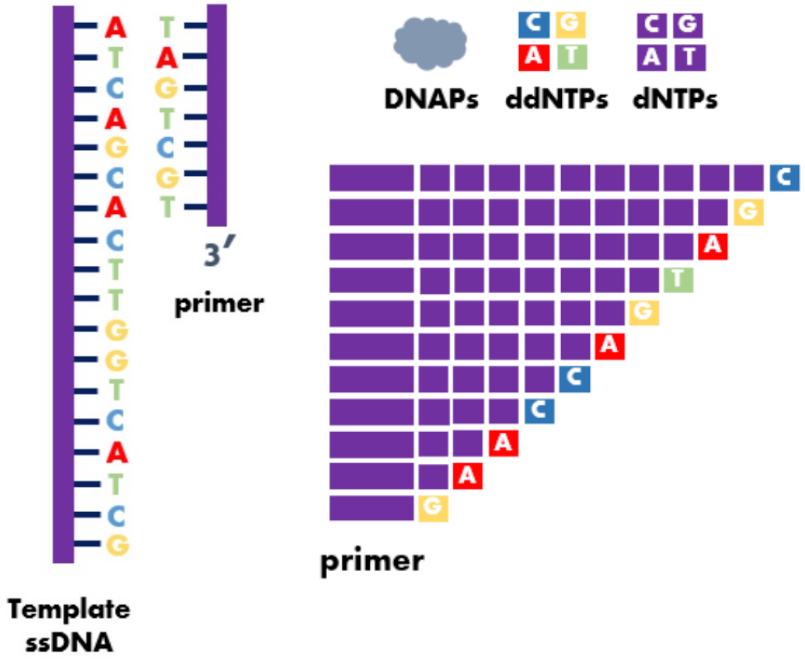
DNA Replication and Chain Termination:

- Sanger sequencing is based on the principle of **DNA replication**. A DNA strand serves as a template, and nucleotides (A, T, C, G) are added one by one to a growing complementary strand by the enzyme **DNA polymerase**.
- The method uses special nucleotides called **dideoxynucleotides (ddNTPs)**, which lack a hydroxyl group (-OH) at the 3' carbon of the sugar ring. Because of this, once a ddNTP is incorporated into the growing DNA strand, the chain cannot be extended further, resulting in **termination** of replication.

1.DNA polymerase extension with dNTP

Sanger Sequencing

- In 1977, Frederick Sanger developed Sanger Sequencing via chain-termination method, and sequenced the first complete genome: one of bacteriophage ϕ174.



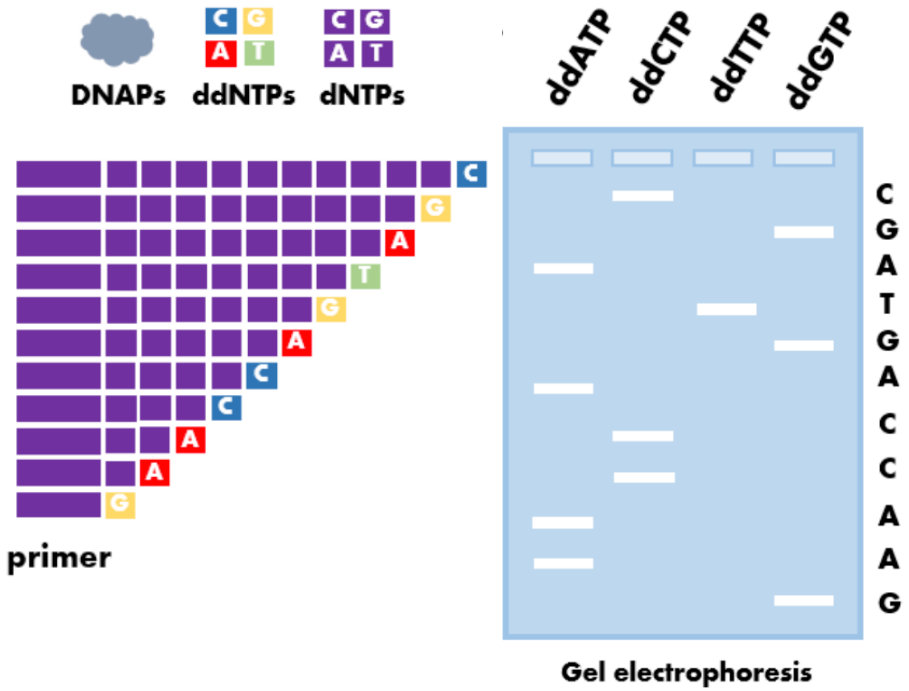
Random Incorporation of ddNTPs:

- During the sequencing process, a mixture of regular nucleotides (**deoxynucleotides, dNTPs**) and modified chain-terminating **dideoxynucleotides** (ddNTPs) is used.
- The ddNTPs are randomly incorporated at different positions along the DNA sequence. Since each ddNTP corresponds to a specific base (ddATP, ddTTP, ddCTP, ddGTP), the DNA fragments are terminated at various points, producing a mixture of DNA fragments of varying lengths, each ending with a ddNTP.

2.Extension terminated when ddNTP incorporated

Sanger Sequencing

- In 1977, Frederick Sanger developed Sanger Sequencing via chain-termination method, and sequenced the first complete genome: one of bacteriophage ϕ174.



Four Separate Reactions (in the early method):

- Initially, Sanger's method required four separate reactions, each containing one type of ddNTP (ddATP, ddTTP, ddCTP, or ddGTP) along with the other three normal dNTPs. Each reaction produced DNA fragments ending with a specific base (A, T, C, or G).
- Over time, these fragments would be separated by **size** using **gel electrophoresis**, with shorter fragments migrating faster than longer ones.

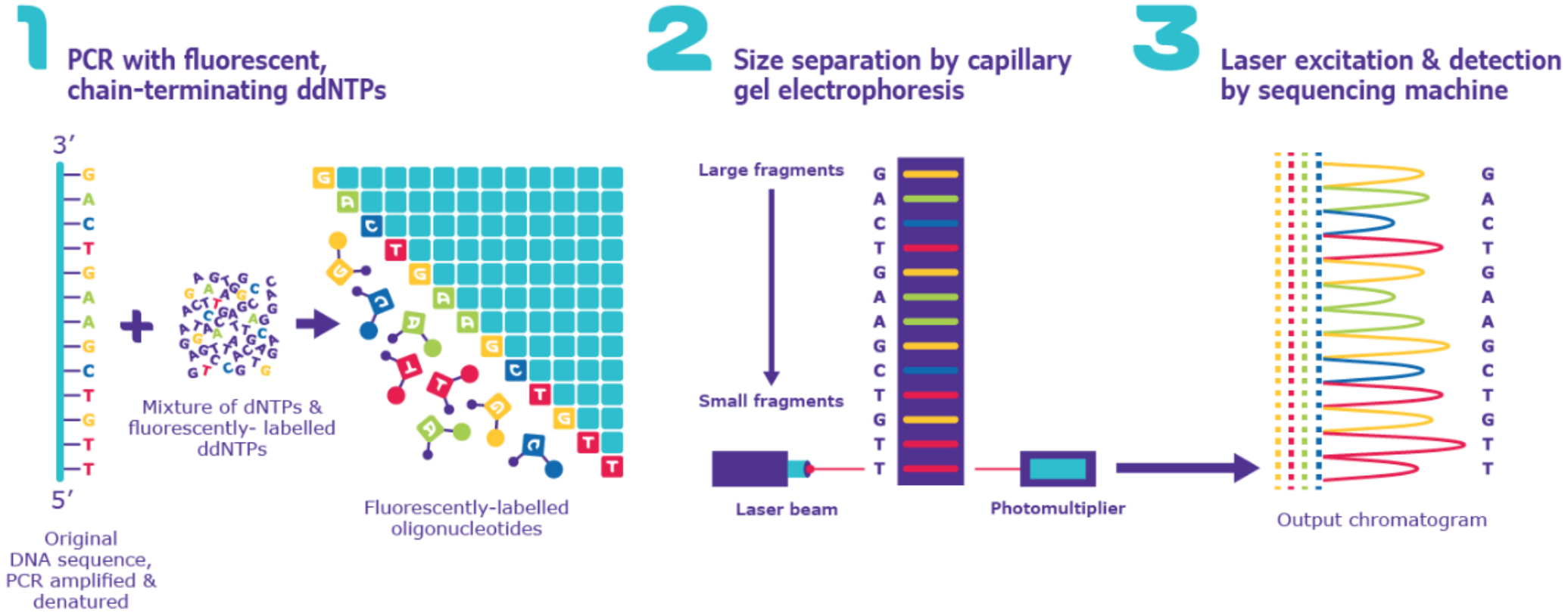
Gel Electrophoresis and Sequence Reading:

- The resulting DNA fragments from each reaction were run on a **polyacrylamide gel** to separate them by size.
- The sequence of the DNA could be determined by reading the gel from bottom (shortest fragment) to top (longest fragment), deducing the sequence of nucleotides based on the size of the fragments and the terminating base.

3.Separate DNA by Gel electrophoresis and read the DNA bands

Modern Sanger Sequencing

- In modern automated Sanger sequencing, ddNTPs are labeled with **different fluorescent dyes**, each color corresponding to one of the four bases (A,C,G,T).



BIO309: Computational Biology, 2025

Lecture 2: Linux & Shell Command

Zhengyu LIANG, PhD (梁征宇)

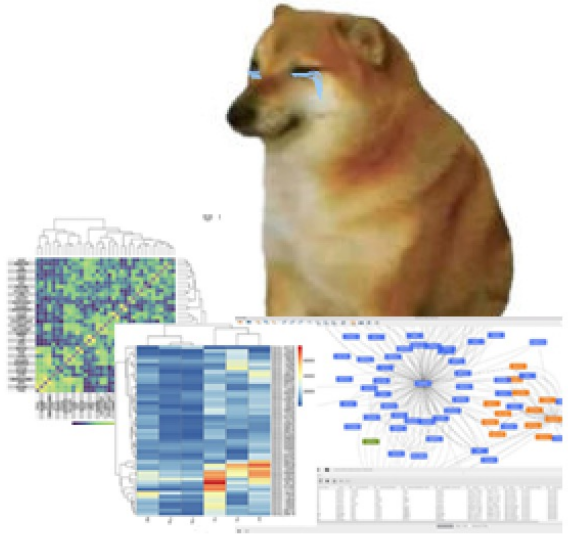
南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Biologist meets Big-data

Molecular biologists
THEN

Discovered how genes work with some bacteria and a glass pipette

Molecular biologists
NOW



Nobody understands my multi-omics single-cell data
Need mechanistic insight

Biologist meets Big-data

ChatGPT Auto

why does molecular biologist need big data in scientific research in current and near future?

1. High-throughput Technologies

Advances like **next-generation sequencing (NGS)**, **single-cell RNA sequencing**, **mass spectrometry**, and **imaging techniques** generate vast amounts of data. For example, sequencing a human genome now produces hundreds of gigabytes of raw data, requiring advanced tools for storage, processing, and interpretation.

2. Systems Biology and Omics Studies

Fields like **genomics**, **proteomics**, **metabolomics**, and **transcriptomics** are highly data-driven. Researchers study complex biological systems by integrating data from multiple sources (e.g., gene expression, protein interactions), necessitating large-scale data analytics to unravel these systems.

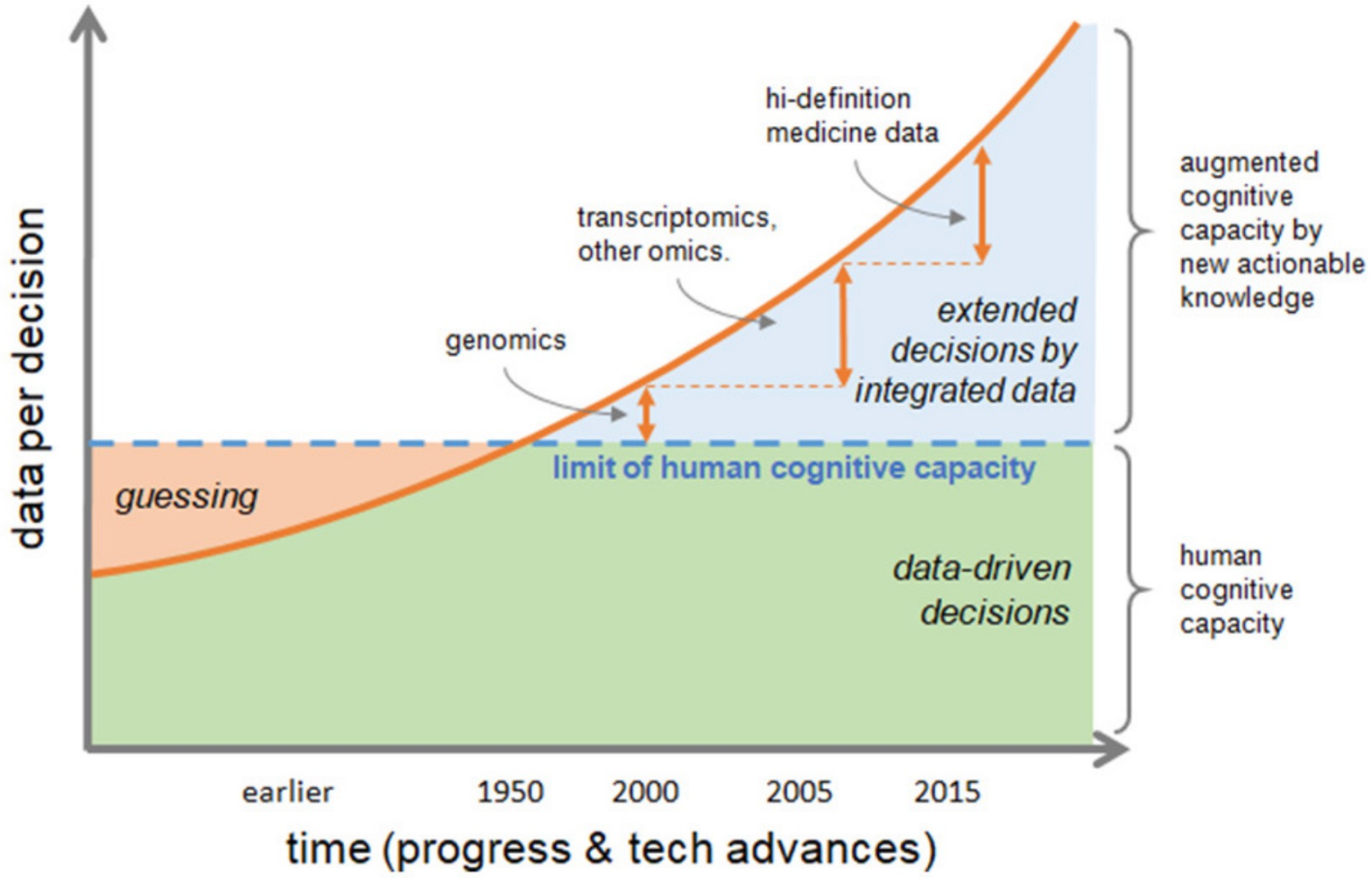
5. Machine Learning and AI Applications

Big data supports the application of **artificial intelligence (AI)** and **machine learning (ML)** in molecular biology. These techniques are increasingly used for **pattern recognition**, **predicting protein structures**, analyzing complex datasets, and discovering new drugs. Large datasets are critical for training and improving these models.

8. Data-Driven Hypothesis Generation

The availability of vast datasets allows researchers to take a **data-driven approach** to research, generating hypotheses from patterns observed in the data rather than traditional experimentation. This accelerates discoveries by focusing efforts on the most promising avenues revealed by big data analytics.

Decision Making Process



Data-self decisions

Data-driven decisions

Decisions by phenotype

BIO309: Computational Biology, 2025

Lecture 9: Pairwise Sequence Alignment

Zhengyu LIANG, PhD (梁征宇)

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Homology, Similarity

Cell, Vol. 50, 667, August 28, 1987, Copyright

"Homology" in Proteins and Nucleic Acids: A Terminology Muddle and a Way out of It

Gerald R. Reeck,[1] Christoph de Haën,[2] David C. Teller,[3] Russell F. Doolittle,[4] Walter M. Fitch,[5] Richard E. Dickerson,[6] Pierre Chambon,[7] Andrew D. McLachlan,[8] Emanuel Margoliash,[9] Thomas H. Jukes,[10] and Emile Zuckerkandl[11]

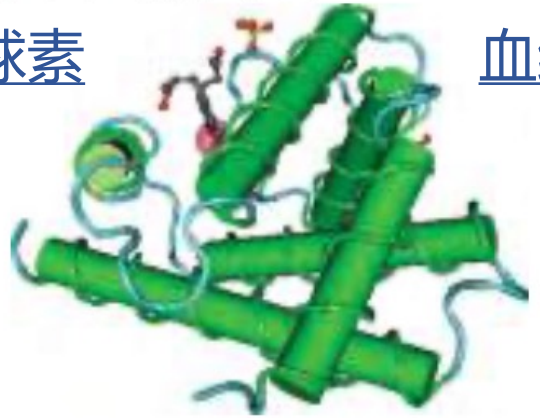
- "Homology" has the precise meaning in biology of "having" a common evolutionary origin.
- "Homology" is a concept of quality: amino acid or nucleotide sequences are either homologous or they are not. They cannot exhibit a particular "level of homology" or "percent".
- "Similarity" is a quantitative property: two sequences possess a certain level of similarity.
- Homologous proteins or nucleic acid segments can range from highly similar to not recognizably similar.

Hazards arise by using "homology" to mean similarity?

Homology, Similarity

(a) Human myoglobin (3RGK)

肌血球素



(b) Human hemoglobin tetramer (2H35)

血红素

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

β珠蛋白

(d) Pairwise alignment of beta globin and myoglobin

- Myoglobin and beta globin have very **similar structure**, indicated by X-ray.
- When two sequences are **homologous**, their sequences usually share significant **identity**.
- But some have sequence **diverged**, sharing no recognizable identity (β- vs neuro-globin, 22%)
- In general, **3D structure** diverge much more slowly than **amino acid sequence** identity between two proteins .
- **Recognizing homology** is challenging bioinformatics problem.

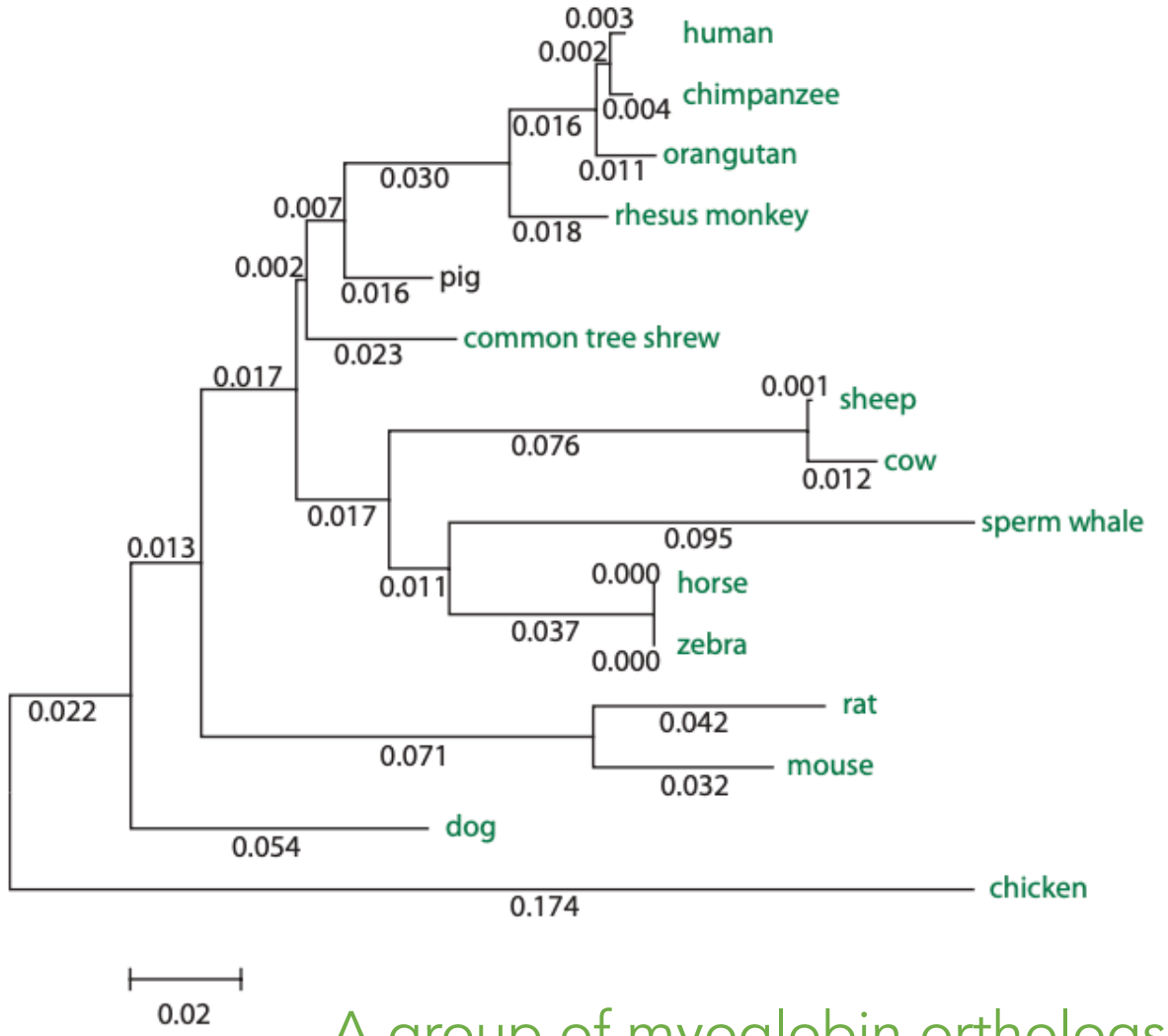
subunit of hemoglobin

Homologous proteins almost share a significantly related 3D structure

Two types of homology

- *Orthologs*（*直系同源*） are homologous sequences **in different species** that arose from a common ancestral gene during speciation.
- *Orthologs* are presumed to have **similar biological functions**: human and rat myoglobin both transport oxygen in muscle cells.

ortho = exact

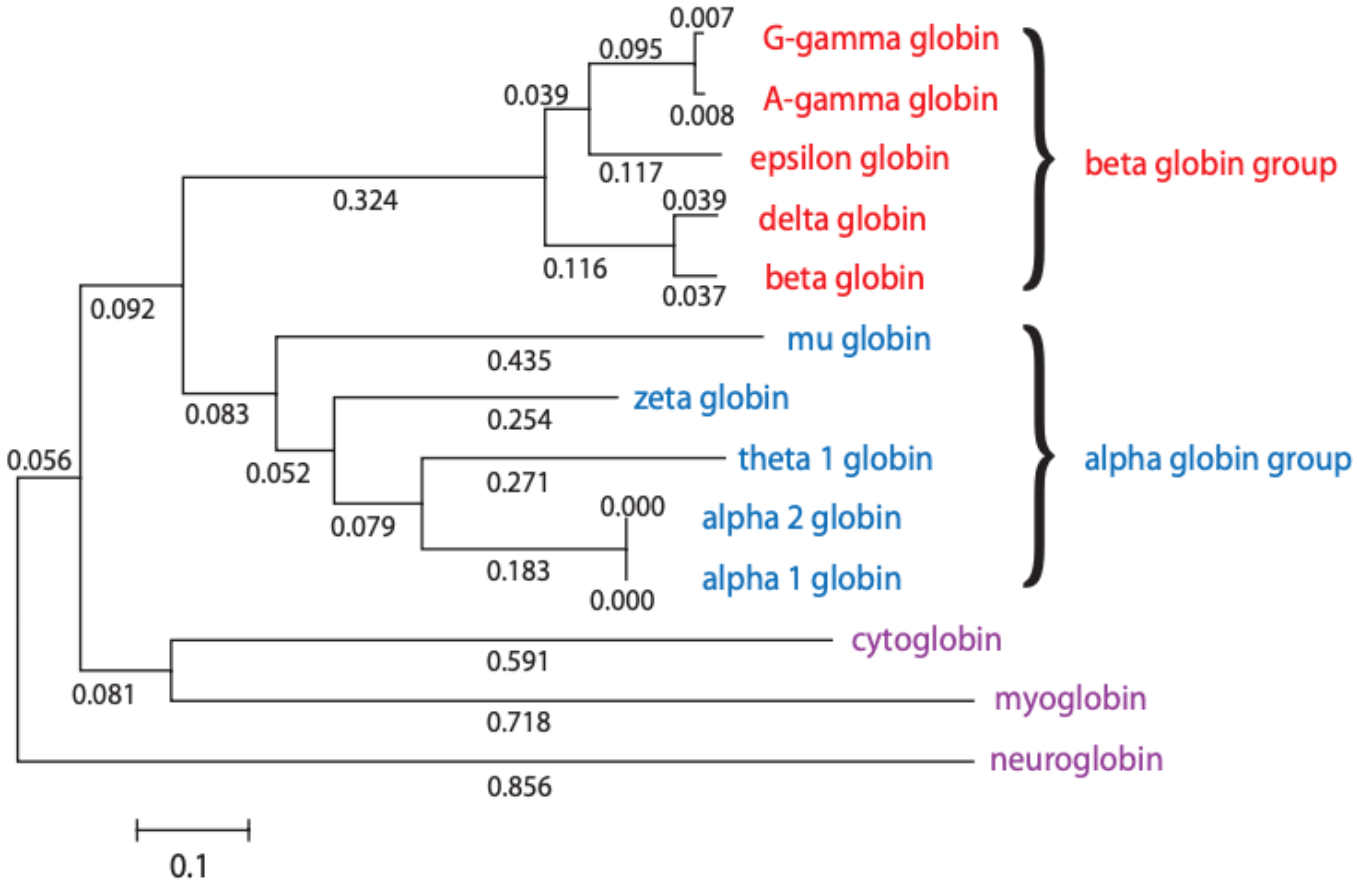


A group of myoglobin orthologs

Two types of homology

- *Paralogs（旁系同源）* are homologous sequences that arose by a **mechanism** such as gene duplication.
- Human α-1 globin is **paralogous** to α-1 globin (sharing 100% identity)

para = in parallel



Paralogous human globins

Two types of homology

- Orthologs and paralogs **do not** necessarily have the same function.
- Two sequences (DNA or protein) are defined as **homologous** based on achieving significant alignment score.

```
Query   12  VTALWGKVNVD--EVGGEALGRLL  33
            V  +WGKV  D    G E L RL
Sbjct   11  VLNVWGKVEADIPGHGQEVLIRLF  34
```

- **Pairwise alignment** is the process of lining up two sequences to achieve maximal levels of identity.
- The purpose of a pairwise alignment is to assess the degree of **similarity and the possibility** of homology between two molecules.

Dayhoff Model / Scoring Matrices 打分矩阵

- **Margaret Dayhoff** provided a model of the rules by which evolutionary change occurs in proteins.
- **Purpose**: quantify the probability of one amino acid being substituted by another over evolutionary time.
- **Hypothesis**: some amino acid substitution are more likely to occur than other due to similarities in biochemical properties.
- **Approach**: create a method to reflect the probabilities based on observed substitution.

Step 1: Accepted Point Mutations

- Dayhoff examined 1572 changes in **71 groups** of closely related proteins from different species, assuming that these proteins diverged from a common ancestor relatively recently.
- Their definition of **"accepted" mutations** was based on empirically observed amino acid **substitution**.
- The table reveals which substitution are unlikely to occur, while others are commonly.

PROTEIN	PAMS PER 100 MILLION YEARS
Immunoglobulin (Ig) kappa chain C region	37
Kappa casein	33
Epidermal growth factor	26
Serum albumin	19
Hemoglobin alpha chain	12
Myoglobin	8.9
Nerve growth factor	8.5
Trypsin	5.9
Insulin	4.4
Cytochrome c	2.2
Glutamate dehydrogenase	0.9
Histone H3	0.14
Histone H4	0.10

	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

Step 2: Frequency of Amino Acid

- To model the probability that one aligned amino acid in a protein changes to another, we also need to know the frequencies of occurrence of each amino acid.

TABLE 3.1 Normalized frequencies of amino acid. 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

Step 3: Relative Mutability of Amino Acids

- Dayhoff calculated the **relative mutability** of the amino acid (m_i / f_i)
- Number of times each amino acid was observed to mutates (m_i)
- Overall frequency of occurrence of that amino acid (f_i)
- This simply describes **how often** each amino acid is likely to change over a short evolution.
- **Biology**: the less mutable residues probably have important structural or functional roles in proteins.
- **Thinking**: relationship to genetic code?

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

Step 4: Mutation Probability Matrix

- Dayhoff generated a *mutation probability matrix M*, using **accepted mutations** and **probabilities of occurrence** of each amino acid.
- Matrix M_{ij} shows the probability that an **original amino acid j** (column) will be replaced by **another amino acid i** (rows) over a defined evolutionary interval (**one PAM**).
- **A PAM** is defined as the unit of evolutionary divergence in which **1% of the amino acids have been changed** between the two protein sequences.
- The **PAM matrix** is defined in terms of percent **amino acid divergence** and not in unit of **years**.

Replacement amino acid	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
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

FIGURE 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. 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.

$$M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}} \qquad M_{jj} = 1 - \lambda m_j$$

Step 5: PAM250 and other PAM

- **PAM1 matrix** was based upon the alignment of **closely related** protein sequence (an average of 1% change).
- **Reality**: we often explore the relationships of proteins that share far **less than 99%** amino acid **identity**.
- In PAM1, λ is chosen to correspond to an evolutionary distance of **1 PAM**. $M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}}$
- **Solution**: as we make λ larger, we model a **greater evolutionary distance** by multiplying λ (PAM2, PAM3, …)
- **Problem**: this approach will fail for **greater evolutionary distances** (PAM250: 250 changes occur in two aligned sequences of length 100, λ does not account for multiple substitutions)

Python program

- **Finally**: Dayhoff instead used **matrix multiplication**: multiplied the PAM1 by itself.

Step 5: PAM250 and other PAM

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

replacement amino acid replacement amino acid

original amino acid

PAM∞	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

FIGURE 3.12 Portion of the matrices for a zero PAM value (PAM0; upper panel) or for an infinite PAM∞ value (lower panel). At PAM∞ (i.e., if the PAM1 matrix is multiplied by itself an infinite number of times), all the entries in each row converge on the normalized frequency of the replacement amino acid (see **Table 3.1**). A PAM2000 matrix has similar values that tend to converge on these same limits. In a PAM2000 matrix, the proteins being compared are at an extreme of unrelatedness. In constrast, at PAM0 no mutations are tolerated and the residues of the proteins are perfectly conserved.

- PAM0: no amino acids have changed.
- PAM∞: an equal likelihood of any amino acid being present that approximate the background probability for the frequency occurrence of each amino acid.

Step 5: PAM250 and other PAM

Replacement amino acid	Original amino acid: A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
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

FIGURE 3.13 The PAM250 mutation probability matrix. At this evolutionary distance, only one in five amino acid residues remains unchanged from an original amino acid sequence (columns) to a replacement amino acid (rows). Note that the scale has changed relative to **Figure 3.11**, and the columns sum to 100.

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

- **PAM250**: This matrix applies to an evolutionary distance where proteins share about 20% amino acid identity.
- The original amino acid is an alanine there is just a 13% chance that the second sequence will also have an alanine.

Step 6: Relatedness Odds Matrix

- **Problem**: for any **given Mutation probability matrix**, what is the probability that amino acid will change (j to i) in a homologous sequence?
- **Solution**: Dayhoff considered an **entire spectrum of models** for evolutionary change in determining target frequencies.
- Normalized frequency f_i is the probability of amino acid **i** occurring **by change**.
- **Practice**: for a comparison of two proteins, it is necessary to determine the values of Rij at each position and then multiply the probabilities.

$$R_{ij} = \frac{M_{ij}}{f_i}.$$

BOX 3.8. STATISTICAL CONCEPT: THE ODDS RATIO

Dayhoff *et al.* (1972) developed their scoring matrix by using odds ratios. The mutation probability matrix has elements M_{ij} that give the probability that amino acid j changes to amino acid i in a given evolutionary interval. The normalized frequency f_i gives the probability that amino acid i will occur at that given amino acid position by chance. The relatedness odds matrix in Equation (3.3) may also be expressed $R_{ij} = M_{ij}/f_i$, where R_{ij} is the relatedness odds ratio.

Equation (3.3) may also be written:

$$\text{Probability of an authentic alignment} = \frac{\text{P(aligned} \mid \text{authentic)}}{\text{P(aligned} \mid \text{random)}}.$$

Step 7: Log-Odds Scoring Matrix

- The logarithmic form of the relatedness odds matrix is called a **log-odds matrix**.
- The values for M_{ij} (q_{ij}, called the "**target frequencies**") are derived from **PAM** (250).
- It is convenient as it allows us **to sum the scores of the aligned residues** when we perform an overall alignment of two sequences.

$$s_{ij} = 10 \times \log_{10}\left(\frac{M_{ij}}{f_i}\right).$$

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**

FIGURE 3.14 Log-odds matrix for PAM250. High PAM values (e.g., PAM250) are useful for aligning very divergent sequences. A variety of 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. Adapted from NCBI, ftp://ftp.ncbi.nlm.nih.gov/blast/matrices/.

to determine the score assigned to a substitution from cysteine to leucine, the PAM250 mutation probability matrix value is 0.02 (**Fig. 3.13**) and the normalized frequency of leucine is 0.085 (**Table 3.1**). We therefore have:

$$s_{(\text{cysteine, leucine})} = 10 \times \log_{10}\left(\frac{0.02}{0.085}\right) = -6.3. \qquad (3.5)$$

Flow

Notation

- 20 amino acids indexed by i, j.
- C_{ij}: accepted substitution **counts** "from j to i" (off-diagonal only).
- f_i: **background frequency** of amino acid i.
- m_i: number of observed **mutations away from** amino acid i (row/column choice consistent with C).
- $M^{(1)}$: PAM1 **mutation probability matrix** (columns sum to 1; 1% expected change).
- $M^{(n)} = \left(M^{(1)}\right)^n$: PAM$n$ mutation probability matrix.
- $R_{ij}^{(n)} = M_{ij}^{(n)} / f_i$: **relatedness odds**.
- $S_{ij}^{(n)} = \text{round}\left(\beta \log_b R_{ij}^{(n)}\right)$: **log-odds scores** (Dayhoff uses $\beta = 10, b = 10$).

Step	Uses which data	Produces	Depends on
1	Observed accepted substitutions	Raw counts C_{ij}	–
2	All amino acids in dataset	Frequencies f_i	–
3	C_{ij}, f_i	Relative mutabilities M_i^{rel}	Steps 1–2
4	C_{ij}	Unscaled conditional matrix $\hat{M}$	Step 1
5	$\hat{M}$, f_i, M_i^{rel}	Scaled $M^{(1)}$	Steps 1–3
6	$M^{(1)}$	$M^{(n)}$	Step 5
7	$M^{(n)}$, f_i	Log-odds scores $S^{(n)}$	Steps 2, 5, 6

Step 1 — Accepted Point Mutations (APM)

Empirical substitutions across closely related proteins:

- Counts: C_{ij} for $i \neq j$ denote "from j to i".
- Set $C_{jj} = 0$.
  (Columns are "from j", rows are "to i".)

Step 2 — Amino-Acid Frequencies

Let n_i be the total occurrences of residue i in the dataset.

$$f_i = \frac{n_i}{\sum_k n_k} \quad \text{with} \quad \sum_i f_i = 1.$$

f_i defines the **background (random) model**.

Step 3 — Relative Mutability (diagnostic)

Total accepted mutations **away from** residue j:

$$m_j = \sum_{i \neq j} C_{ij}, \qquad M_j^{\text{rel}} = \frac{m_j}{f_j}.$$

(High $M_j^{\text{rel}} \Rightarrow j$ tends to change more often, per opportunity.)

Step 4 — Unscaled One-Step Conditional Matrix

Column-wise conditional probabilities derived from counts:

$$\hat{M}_{ij} = \begin{cases} \dfrac{C_{ij}}{\sum_{k \neq j} C_{kj}} & (i \neq j,\ \sum_{k \neq j} C_{kj} > 0), \\ 0 & (i = j). \end{cases}$$

$\hat{M}$ encodes **which** changes are favored, not **how fast** they occur.

Step 5 — PAM1 (1% expected change)

Choose $\delta = 0.01$ so one PAM corresponds to ~1% accepted change:

$$M_{ij}^{(1)} = \delta \hat{M}_{ij} \quad (i \neq j), \qquad M_{jj}^{(1)} = 1 - \sum_{i \neq j} M_{ij}^{(1)}.$$

Each column sums to 1; expected off-diagonal probability per column is δ.

Step 6 — PAMn by Matrix Multiplication

Accumulate substitutions (allowing multiple hits at the same site):

$$M^{(n)} = \left(M^{(1)}\right)^n \quad (\text{e.g., PAM250: } M^{(250)} = (M^{(1)})^{250}).$$

(If you like linear algebra form: if diagonalizable, $M^{(n)} = S\Lambda^n S^{-1}$.)

Step 7 — Relatedness Odds and Log-Odds Scores

Compare the evolutionary model against the background:

$$R_{ij}^{(n)} = \frac{M_{ij}^{(n)}}{f_i}, \qquad S_{ij}^{(n)} = \text{round}\left(10 \log_{10} R_{ij}^{(n)}\right).$$

- $S_{ij}^{(n)} > 0$: substitution $j \to i$ is **more likely than random**.
- $S_{ij}^{(n)} < 0$: **less likely than random**.

Alignment use (ungapped portion):

For aligned residue pairs (x_k, y_k), total score

$$\text{Score} = \sum_k S_{x_k y_k}^{(n)} + \text{gap penalties}.$$

This is a **log-odds sum** comparing an evolutionary model to a random model.

Flow

Step 1 → Step 4

From accepted substitution counts C_{ij} (from $j \to i$):

$$\hat{M}_{ij} = \frac{C_{ij}}{\sum_{k \neq j} C_{kj}}$$

Each column j is normalized by its total number of observed outgoing mutations.
So yes — the numerical form of $\hat{M}$ depends only on the **empirical counts**.

However, this matrix has **relative** probabilities — not an absolute evolutionary rate.

Step 4 → Step 5 connection

The unscaled matrix $\hat{M}$ is multiplied by a scaling factor δ (derived using Step 2–3 data) to yield:

$$M_{ij}^{(1)} = \delta \, \hat{M}_{ij} \quad (i \neq j), \quad M_{jj}^{(1)} = 1 - \sum_{i \neq j} M_{ij}^{(1)}$$

Thus:

- Step 1 defines the *pattern* of substitutions.
- Steps 2–3 set the *rate* of change and ensure equilibrium amino-acid composition.

Step 2 and Step 3 — why still needed

1. **Step 2 (frequencies f_i)**
   - Tells how often each residue occurs in nature.
   - Needed later for:
     - Normalizing the evolutionary model to preserve the same amino-acid composition after many substitutions.
     - Computing log-odds ratios $R_{ij} = M_{ij}/f_i$ in Step 7.
2. **Step 3 (relative mutability $M_i^{rel} = m_i/f_i$)**
   - Diagnostic measure: amino acids that mutate more frequently per occurrence.
   - Dayhoff used it to calibrate how quickly the mutation process occurs overall, so that **on average 1% of residues change per PAM unit**.

Specifically, the scaling constant δ = 0.01 in Step 5 was chosen such that:

$$\sum_i f_i \, M_i^{rel} \, \delta = 0.01$$

i.e., the weighted average mutation probability over all residues equals 1%.

Two fi

Dayhoff/PAM model logic:

why do we divide by f_i *again* in Step 7 (when we already divided by f_i earlier to get the relative mutability $M_i^{rel} = m_i / f_i$)?

🧩 1. Two very different divisions by f_i

Step	Formula	Purpose	What it measures
Step 3	$M_i^{rel} = m_i / f_i$	**Empirical correction** for residue abundance	*Intrinsic mutability* — how often a residue tends to change when it occurs
Step 7	$R_{ij}^{(n)} = M_{ij}^{(n)} / f_i$	**Probabilistic odds ratio** between "evolutionary" and "random background" models	*Relative likelihood* of substitution $j \to i$ vs. chance appearance of i

🔬 2. Step 3 — a within-data correction

Dayhoff noticed that amino acids differ in abundance:
rare residues (like Trp, Cys) naturally appear less often and thus have fewer observed mutations, even if they are equally "mutable."

So she computed

$$M_i^{rel} = \frac{m_i}{f_i}$$

to remove this compositional bias.
👉 **Goal:** express *how mutable a residue is per occurrence*, not per dataset frequency.
This step adjusts *empirical mutation rates*, not probabilities of alignment.

At this stage we're still describing **how amino acids evolve**.

⚖️ 3. Step 7 — a between-model normalization

By Step 7, we already have a complete **Markov model** of evolution:

$$M_{ij}^{(n)} = \text{probability that } j \text{ changes to } i \text{ in } n \text{ PAM units.}$$

To score alignments, we now compare two competing models for an observed amino-acid pair (i, j):

- **Evolutionary model:** probability $P_{\text{evo}}(i|j) = M_{ij}^{(n)}$
- **Random (background) model:** probability $P_{\text{rand}}(i) = f_i$

The **log-odds ratio**

$$R_{ij}^{(n)} = \frac{M_{ij}^{(n)}}{f_i}$$

tells how much more likely that pair is to arise by homology (evolution) than by chance.
Taking logs gives additive scores used in dynamic programming alignments:

$$S_{ij}^{(n)} = \log \frac{M_{ij}^{(n)}}{f_i}.$$

👉 **Goal:** convert biological substitution probabilities into alignment weights that discriminate "true" evolutionary pairs from random coincidences.

🔁 In summary

- Step 3's division by f_i: *"How often does this residue mutate, given how common it is?"*
- Step 7's division by f_i: *"Given a substitution model, how much more likely is this observed pair than random?"*

They reuse f_i but for completely different reasons — one calibrates the **model**, the other calibrates the **score**.

Alternative to PAM: BLOSUM

- **Problem**: when comparing two sequence, it may be necessary to repeat the search using **several different scoring matrices** (i.e., PAM10 for related proteins, while PAM70 for relatively divergent proteins).
- **BLOSUM** (blocks substitution matrix): **Henikoff** used the **BLOCKS database** (over 500 groups of **distantly related protein sequences**) to generate the scoring matrices.
- It **merges all proteins** in an alignment that has 62% amino acid identity or greater into **one sequence**.
- The **BLOSUM62 matrix** is useful for scoring proteins that share **less than 62% identity**.

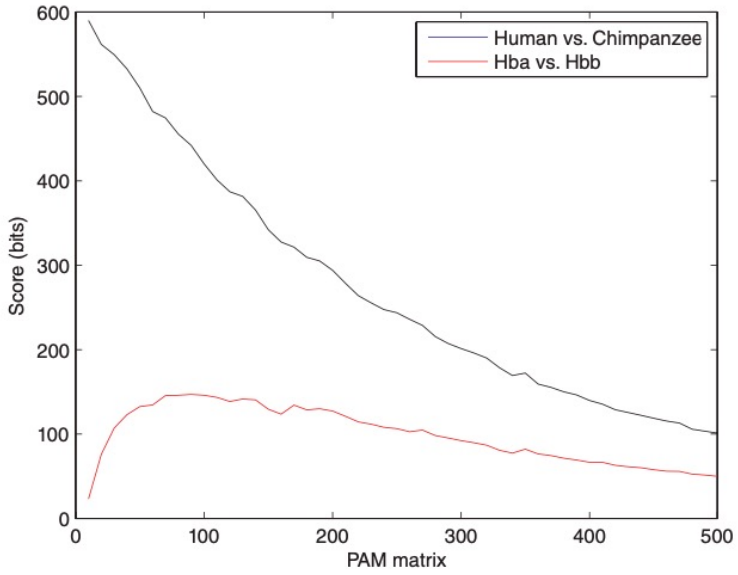


FIGURE 3.16 Global pairwise alignment scores using a series of PAM matrices. Two closely related globins (human and chimpanzee beta globin; black line) were aligned using a series of PAM matrices (*x* axis) and the bit scores were measured (*y* axis). For two distantly related globins (human alpha versus beta globin; red line) the bit scores are smaller for low PAM matrices (such as PAM1 to PAM20) because mismatches are severely penalized.

$$S_{ij} = 2 \times \log_2 \left(\frac{q_{ij}}{p_{ij}} \right).$$

Alternative to PAM: BLOSUM

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-1	1	1	-2	-1	-3	-2	5								
M	-1	-2	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

FIGURE 3.17 The BLOSUM62 scoring matrix of Henikoff and Henikoff (1992). This matrix merges all proteins in an alignment that have 62% amino acid identity or greater into one sequence. BLOSUM62 performs better than alternative BLOSUM matrices or a variety of PAM matrices at detecting distant relationships between proteins. It is therefore the default scoring matrix for most database search programs such as BLAST (Chapter 4).

Source: Henikoff & Henikoff (1992). Reproduced with permission from S. Henikoff.

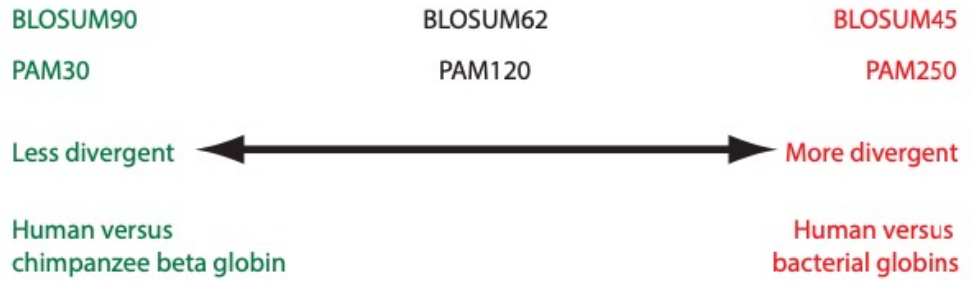


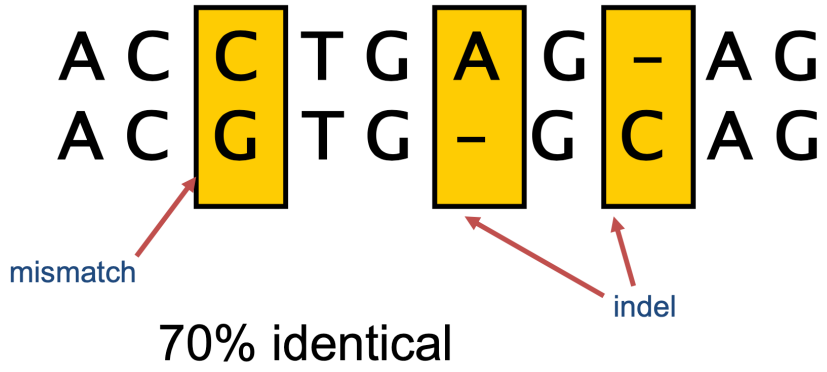
FIGURE 3.18 Summary of PAM and BLOSUM matrices. High-value BLOSUM matrices and low-value PAM matrices are best suited to study well-conserved proteins such as mouse and rat beta globin. BLOSUM matrices with low numbers (e.g., BLOSUM45) or high PAM numbers are best suited to detect distantly related proteins. Remember that in a BLOSUM45 matrix all members of a protein family with greater than 45% amino acid identity are grouped together, allowing the matrix to focus on proteins with less than 45% identity.

Pairwise alignment

- **Lining up two sequences** to achieve maximal levels of identity (and conservation) for the purpose of assessing the degree of similarity and the possibility of homology
- **Align by Hand:**

seq1: ACCTGAGAG

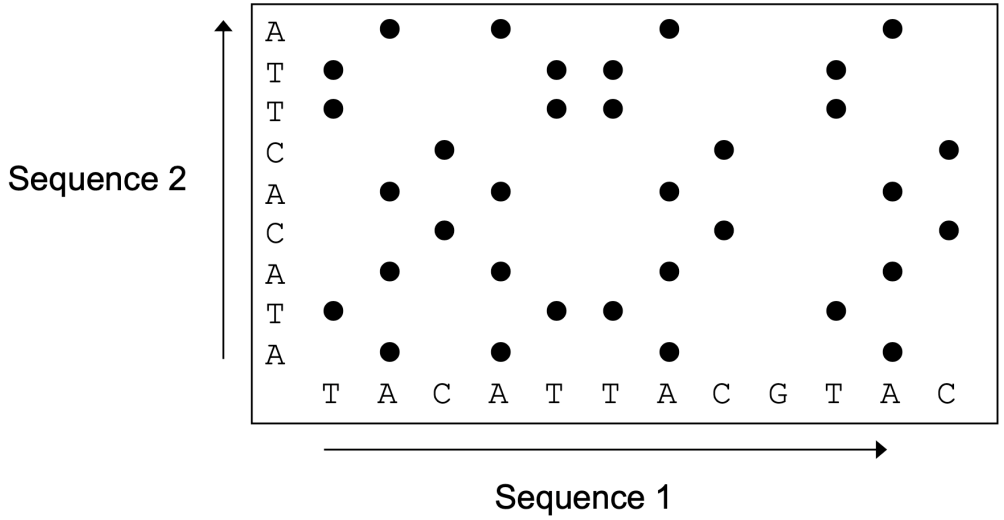
seq2: ACGTGGCAG

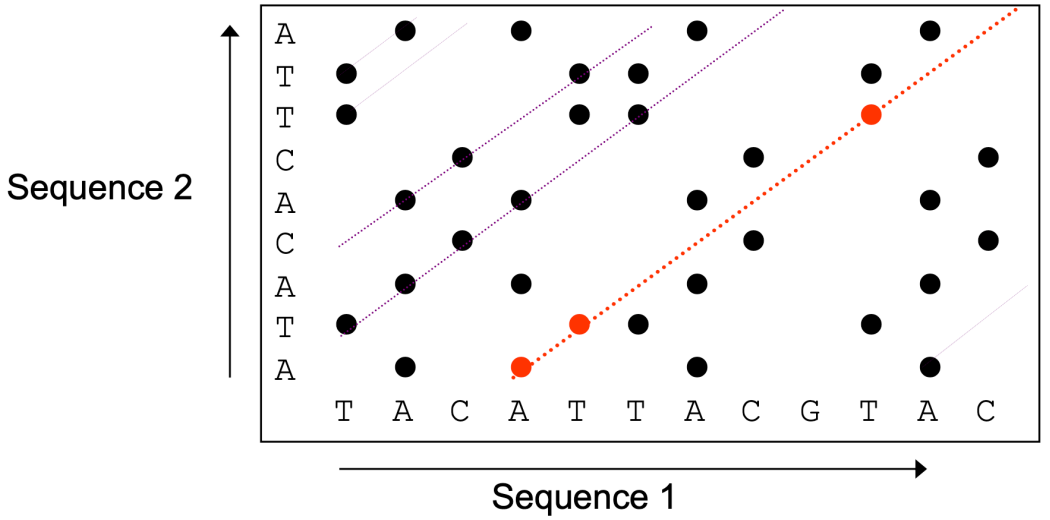


- Need some kind of **system** to find the best alignment

Pairwise alignment

- **Lining up two sequences** to achieve maximal levels of identity (and conservation) for the purpose of assessing the degree of similarity and the possibility of homology
- **Align by Dotplot: (ungapped)**



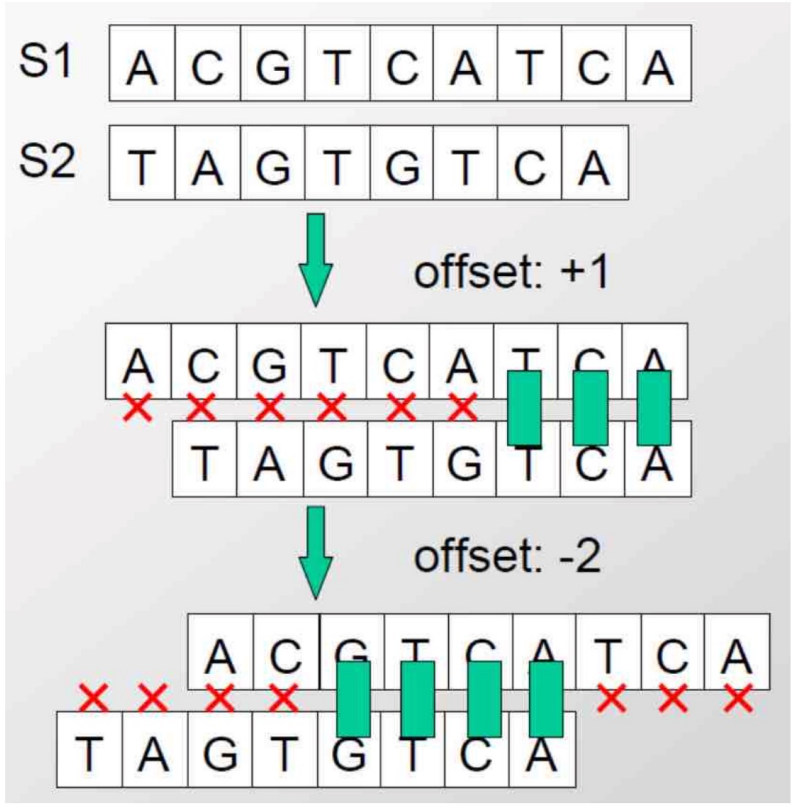


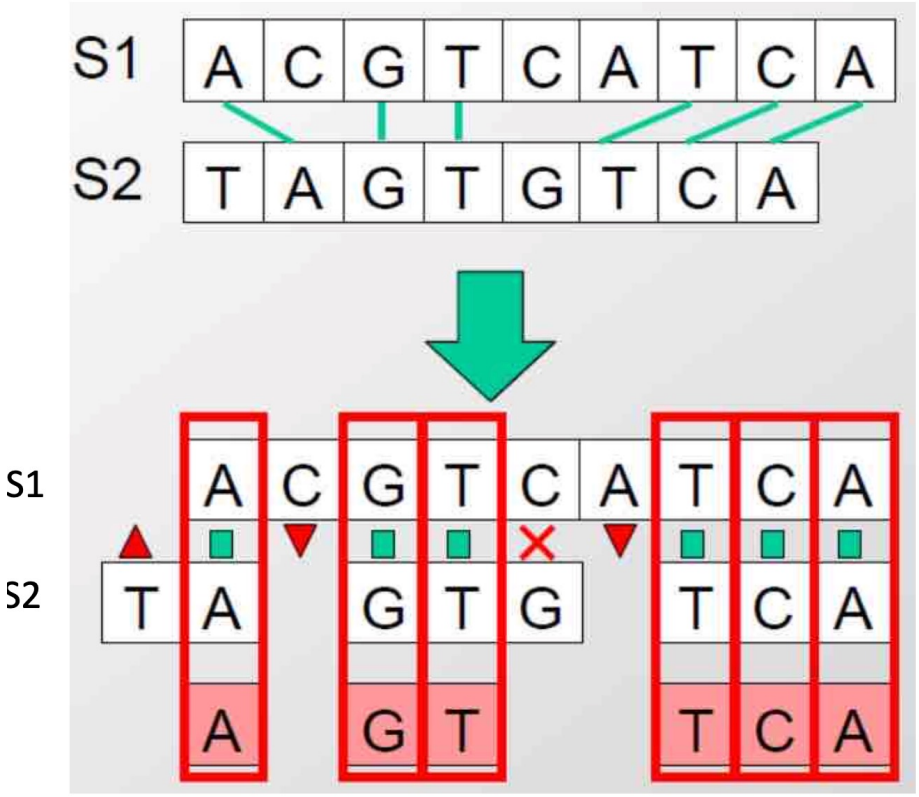
One possible alignment:

```
T A C A T T A C G T A C
  | |       |
A T A C A C T T A
```

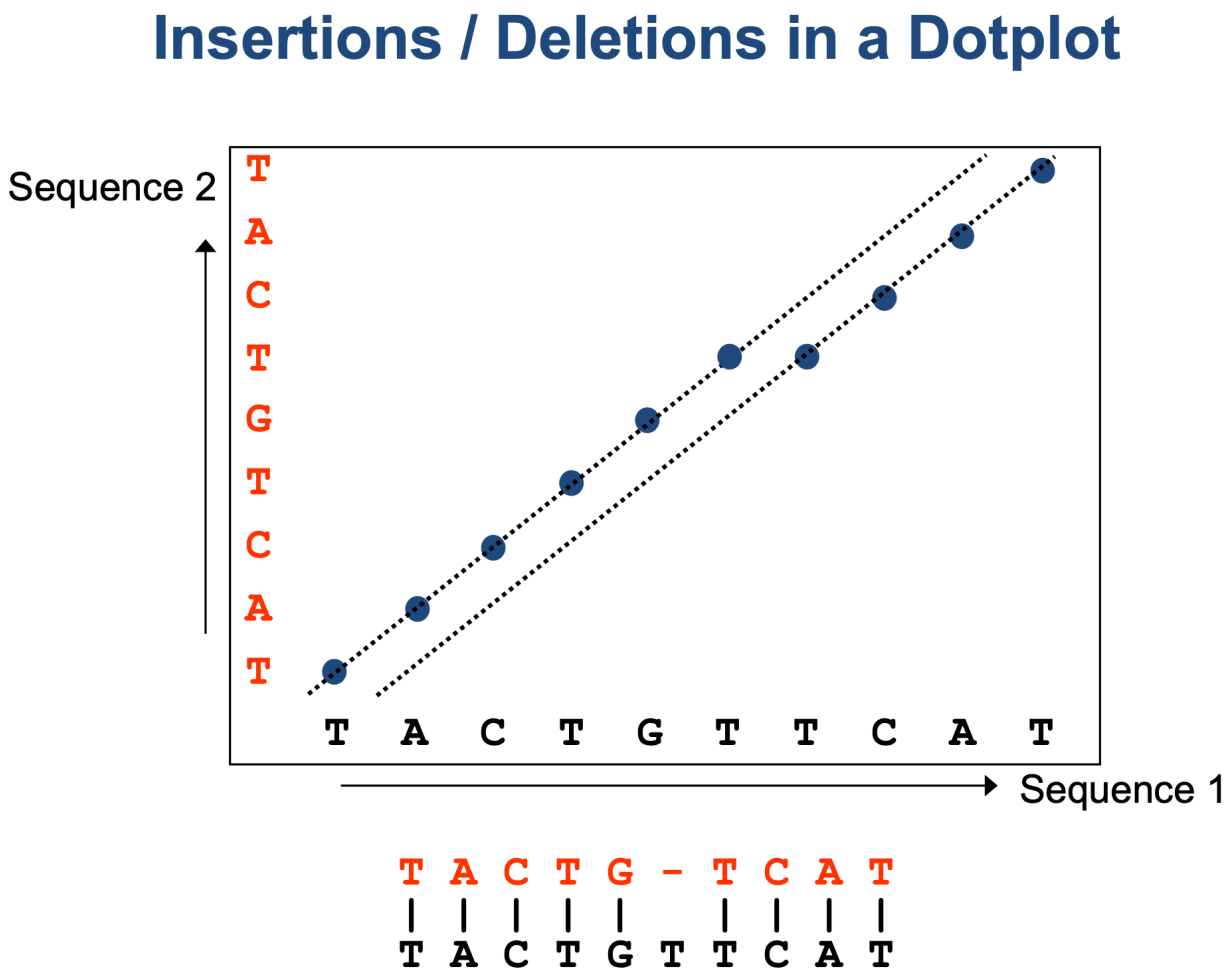
- Need some kind of **system** to find the best alignment

Longest common subsequence





Longest common subsequence



Possible Alignment with Gap

[1] identity (stay along a diagonal)
[2] mismatch (stay along a diagonal)
[3] gap in one sequence (move vertically!)
[4] gap in the other sequence (move horizontally!)

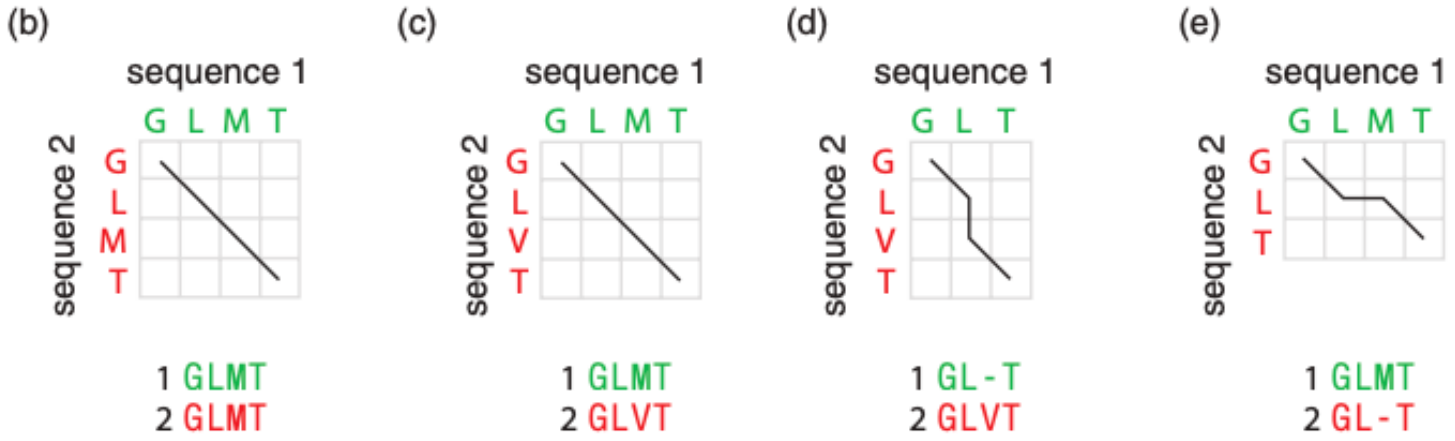
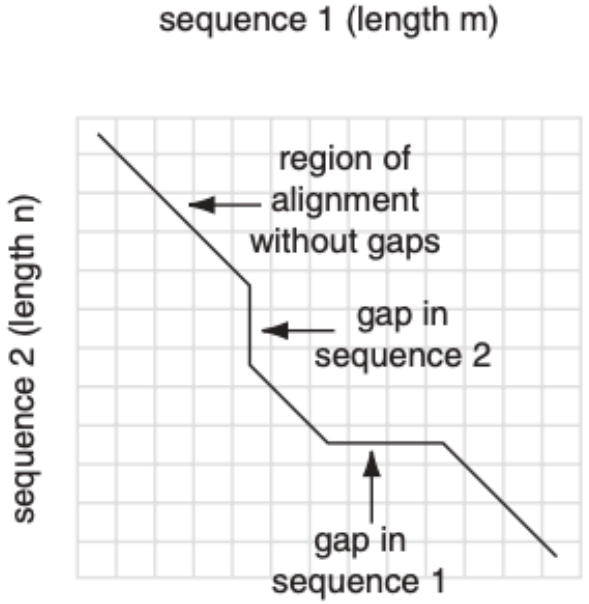


FIGURE 3.20 Pairwise alignment of two amino acid sequences using a dynamic programming algorithm of Needleman and Wunsch (1970) for global alignment. (a) Two sequences can be assigned a diagonal path through the matrix and, when necessary, the path can deviate horizontally or vertically, reflecting gaps that are introduced into the alignment. (b) Two identical sequences form a path on the matrix that fits a diagonal line. (c) If there is a mismatch (or multiple mismatches), the path still follows a diagonal, although a scoring system may penalize the presence of mismatches. If the alignment includes a gap in (d) the first sequence or (e) the second sequence, the path includes a vertical or horizontal line.



Compute best alignment

S1	A	C	G	T	C	A	T	C	A
S2	T	A	G	T	G	T	C	A	

Rigorous algorithms = Dynamic Programming
- Needleman-Wunsch (global; 1970)
- Smith-Waterman (local; 1981)

Enumerate all possible alignments:

There are

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

possible global alignments for 2 sequences of length *n*

For two sequences of length n:

n	Enumeration
10	184,756
20	1.40E11
100	9.00E58

- Need to figure out how to use solution to smaller problems for solving larger problem.
- We need to keep a reasonable bound on how many sub-problems we solve
- Make sure that each sub-problem is solved only once

Seq1: AAACG
Seq2: AAGCG

Alignment based on prefixes

Global Sequence Alignment

- **Needleman-Wunsch approach**: optimal alignment, for DNA/protein, allowing gaps.
- **Three steps:** 1) setting up a matrix; 2) scoring the matrix; 3) identifying the alignment.

Setting up a matrix

Identify positions of identity

(a)

Sequence 1		Sequence 2							
			F	M	D	T	P	L	N	E
		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								

- Set up two matrices: an amino acid **identity matrix** and then **a scoring matrix**.
- A series of grey-filled cells along the diagonal for two sequences is called identity matrix.

Global Sequence Alignment

Scoring the matrix

- To determine the path through the matrix that **maximizes the score**.

(a)

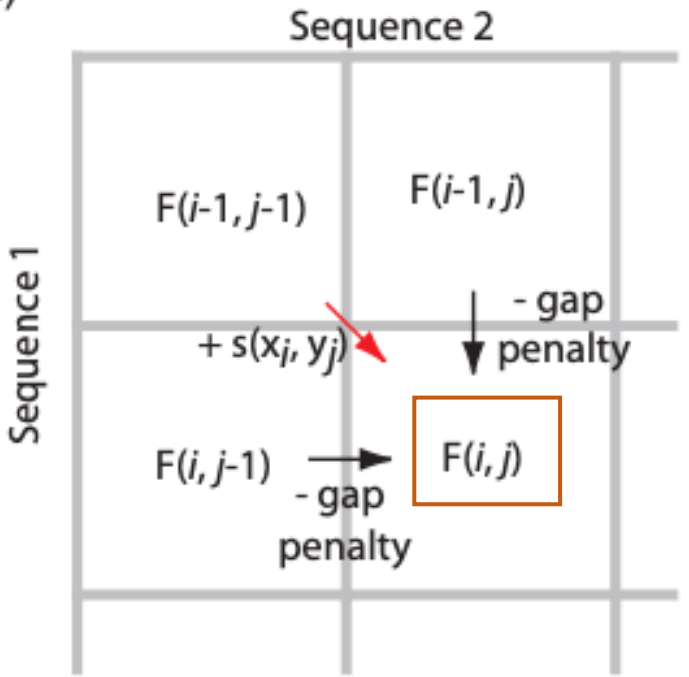
Sequence 1 \ Sequence 2		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2								
K	-4								
H	-6								
M	-8								
E	-10								
D	-12								
P	-14								
L	-16								
E	-18								

(b)

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

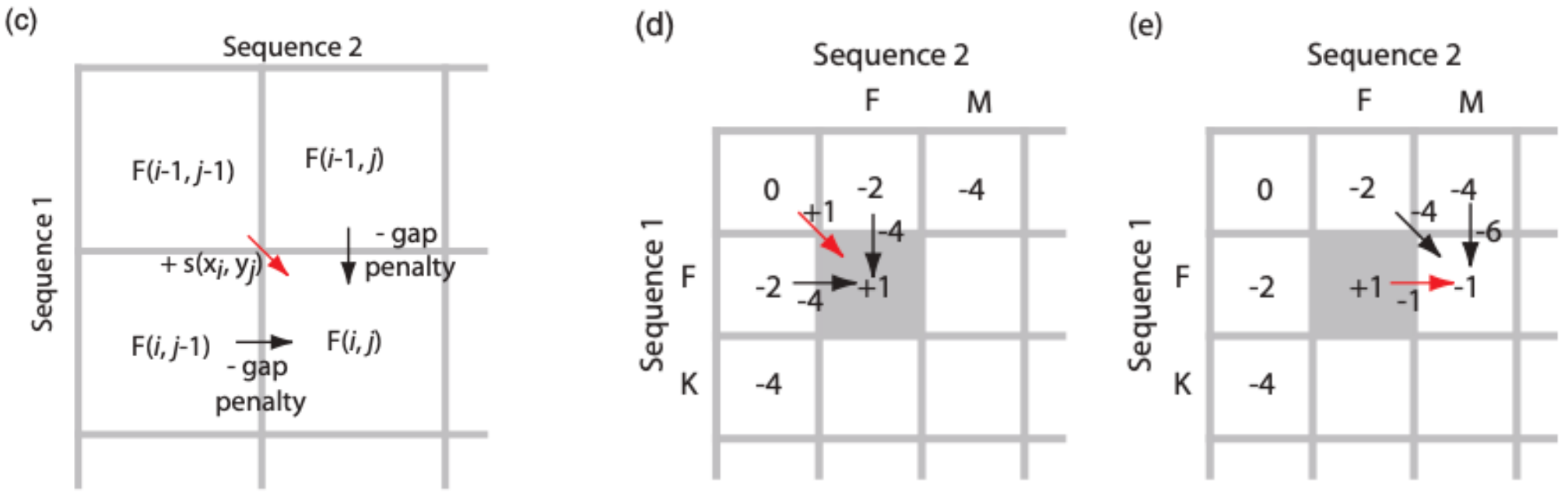
Score (this example) = +1 (match)
-2 (mismatch)
-2 (gap penalty)

(c)



Global Sequence Alignment

Scoring the matrix



Global Sequence Alignment

Scoring the matrix

(f)

Sequence 1 \ Sequence 2		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							
H	-6								
M	-8								
E	-10								
D	-12								
P	-14								
L	-16								
E	-18								

(g)

Sequence 1 \ Sequence 2		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

Global Sequence Alignment

Identifying the Optimal Alignment

- After the matrix is filled, the alignment is determined by a **trace-back** procedure.
- For every cell, we can determine the **best score** derived from three adjacent cells.
- We therefore define a **path** (pink-shaded).
- The **final alignment** is guaranteed to be optimal, given this scoring system.

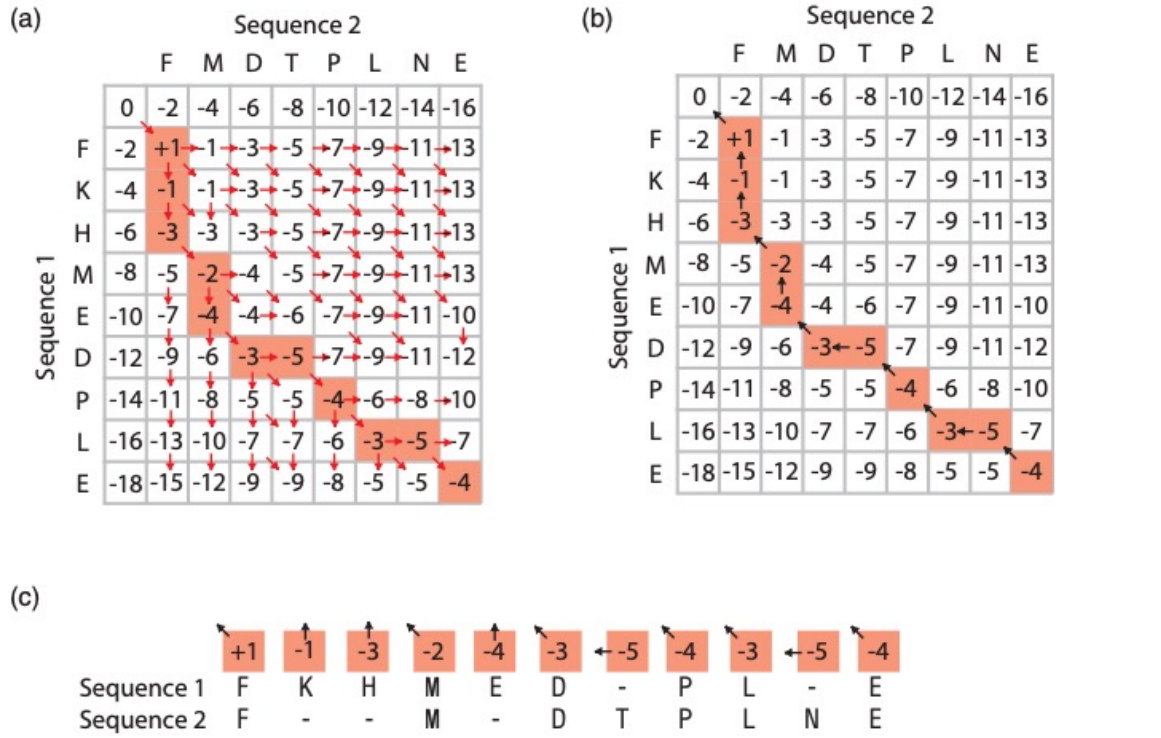


FIGURE 3.22 Global pairwise alignment of two amino acid sequences using a dynamic programming algorithm: scoring the matrix and using the trace-back procedure to obtain the alignments. (a) The alignment of **Figure 3.21(g)** is shown. The cells highlighted in pink represent the source of the optimal scores. (b) In an equivalent representation, arrows point back to the source of each cell's optimal score. (c) This trace-back allows us to determine the sequence of the optimal alignment. Vertical or horizontal arrows correspond to the positions of gap insertions, while diagonal lines correspond to exact matches (or mismatches). Note that the final score (–4) equals the sum of matches ($6 \times 1 = 6$), mismatches (none in this example), and gaps ($5 \times -2 = -10$).

Basic principles of dynamic programming

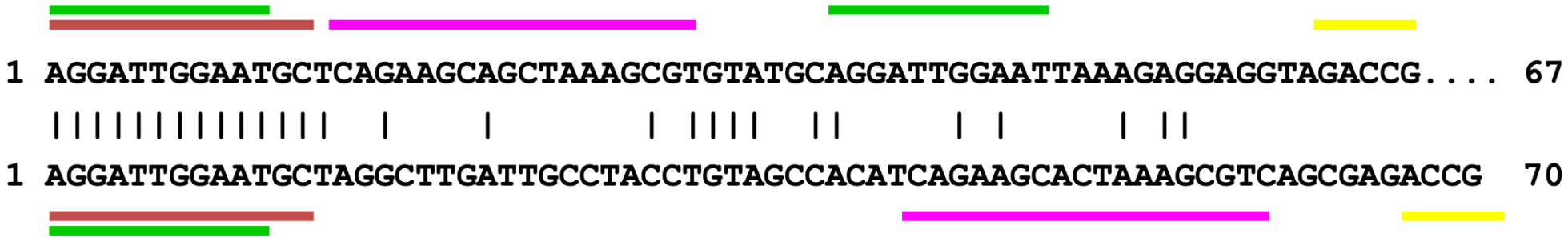
- Dynamic programming（动态规划）: an optimal path is detected by incrementally extending optimal sub-paths by making a series of decisions at each step of the alignment for the best score.

  - Creation of an **alignment path matrix**

  - **Stepwise** calculation of score values

  - **Backtracking** (evaluation of the optimal path)

How about local similarity:



Global alignment methods try to find the best alignment over whole length can miss local similarity region. In this way, we will want a local alignment, the best match between subsequences of x and y

- Original formulation: Smith & Waterman, *Journal of Molecular Biology*, 1981
- interpretation of array values is somewhat different
  - $F(i, j)$ = score of the best alignment of a suffix of $x[1...i]$ and a suffix of $y[1...j]$

Local Sequence Alignment

- **Smith-Waterman approach**: no penalty for **starting the alignment** at internal position.

(a)

Sequence 2 \ Sequence 1		C	A	G	C	C	U	C	G	C	U	U	A	G
	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
A	0.0	0.0	1.0	0.0	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	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	0.0	0.7
G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	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	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	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	0.0
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	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	1.0
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	2.7
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	2.0
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	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	2.0
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	2.0

Max { S(i-1, j-1), S(i-1, j), S(i, j-1), 0 }

Match: +1; mismatch: -0.3; gap (match to mismatch): -1.3

(b)

```
sequence 1   GCC-UCG
sequence 2   GCCAUUG
```

(c)

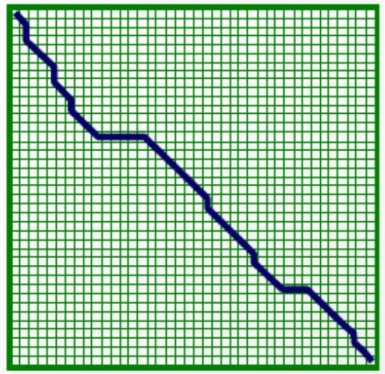
```
sequence 1   CA-GCC-UCGCUUAG
sequence 2   AAUGCCAUUGACG-G
```

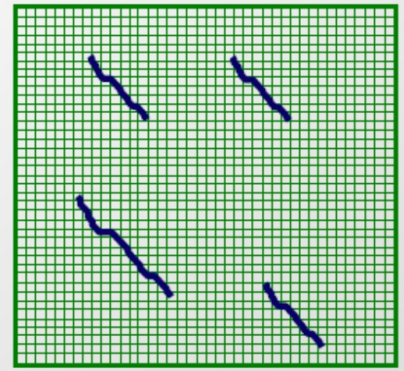
FIGURE 3.24 Local sequence alignment method of Smith and Waterman (1981). (a) In this example, the matrix is formed from two RNA sequences (CAGCCUCGCUUAG and AAUGCCAUUGACGG). While this is not an identity matrix (such as that shown in **Fig. 3.21a**), positions of nucleotide identity are shaded gray (or shaded pink in the region of local alignment). They scoring system here is +1 for a match, minus one-third for a mismatch, and a gap penalty of the difference between a match and a mismatch (–1.3 for a gap of length one). The matrix is scored based on finding the maximum of four possible non-negative values. The highest value in the matrix (3.3) corresponds to the beginning of the optimal local alignment, and the aligned residues (green font) extend up and to the left until a value of zero is reached. (b) The local alignment derived from this matrix is shown. Note that this alignment includes identities, a mismatch, and a gap. (c) A global alignment of the two sequences is shown for comparison to the local alignment. Note that it encompasses the entirety of both sequences.

Comparison

- **Global alignment algorithms** start at the beginning of two sequences and add gaps to each until the end of one is reached.
- **Local alignment algorithms** finds the region of highest similarity between two sequences and build the alignment outward from there.

Global Alignment vs. Local alignment





Needleman-Wunsch algorithm

Initialization: $F(0, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right

Smith-Waterman algorithm

Initialization: $F(0, j) = F(i, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(xi, yj) \end{cases}$$

Termination: Anywhere

Generalized gap penalties （惩罚）

- Which alignments is better?

```
ATTTTAGTAC
ATT - -AGTAC
```

```
ATTTTAGTAC
A- TT -AGTAC
```

Generalized gap penalties （惩罚）

- **Gap penalties** determine the score calculated for a subsequence and thus affect **which alignment is selected**.
- The **normal model** is that gaps of length *k* is penalized equally with value *p*.

$$\text{This penalty can be modeled as } w(k) = k * p.$$

- It could be a good idea to **penalize differently for gaps** of different lengths (incremental penalty decreases as the size of gap grows).

$$\text{This can be modeled as } w(k) = p + q * k + r * k^2.$$

- **Simpler approximation:** have a fixed penalty to start a gap and a linear cost to add a gap.

$$\text{as } w(k) = p + q * k.$$

- More **complex functions** for protein-coding sequence: a gap of length mod 3 can be less.

BIO309: Computational Biology, 2025

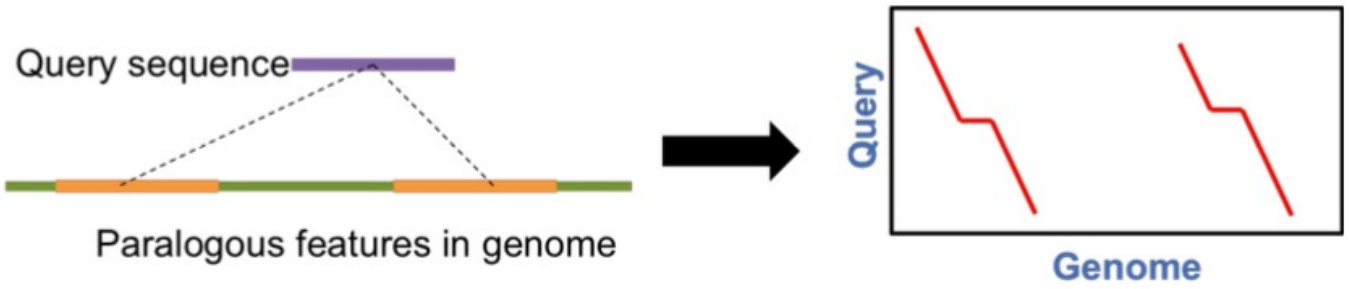
Lecture 10: BLAST

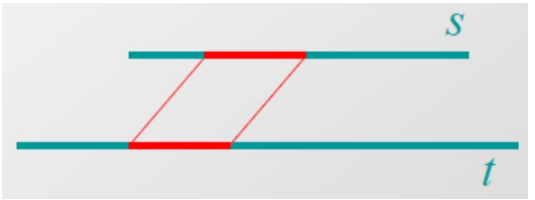
Zhengyu LIANG, PhD (梁征宇)

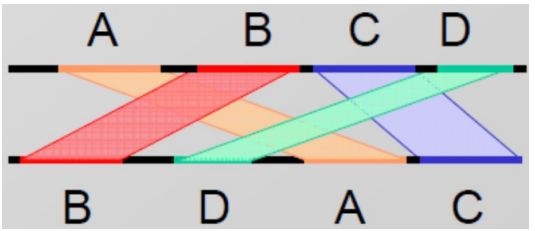
南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

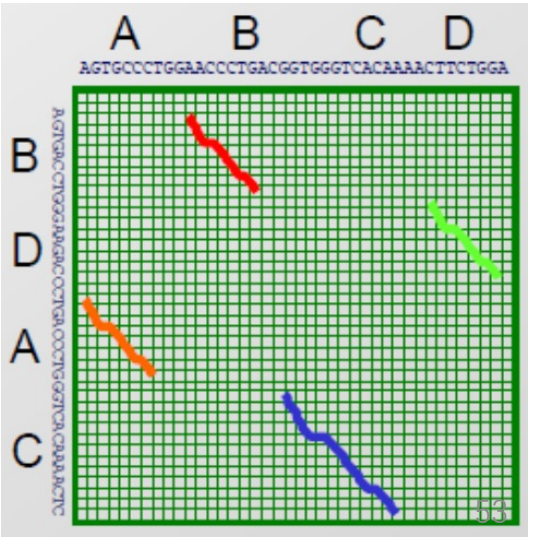
Why local alignment

- **Statement** of local alignment:
  - A local alignment of string **s** and **t** is an alignment of a substring of **s** with a substring of **t**.
- **Applications** of local alignment:
  - Small domains of a gene may be only conserved portions
  - Looking for a small gene in a large chromosome (search)
  - Large segments often undergo rearrangements



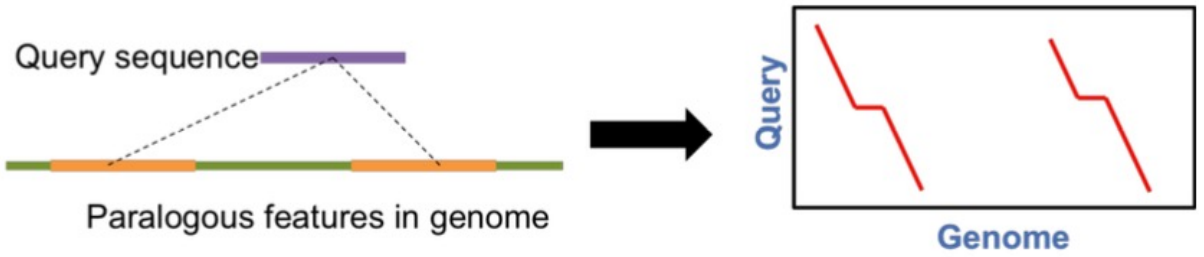






Basic Local Alignment Search Tool (BLAST)

- **Basic Local Alignment Search Tool (BLAST)** is the main NCBI tools for comparing a protein or DNA sequence to other sequences in **databases**.
- BLAST searching allows to select **one sequence** (as *query*) and perform **pairwise sequence alignment** against an **entire database** (as *target*).
- (X) The global alignment: identifying **locally matched regions** (i.e., protein domain).
- (X) The local alignment: **optimal pairwise alignment** (but computationally intensive).
- **BLAST** offers a local alignment strategy having both **speed and sensitivity**.



BLAST feature: How to speed up?

(when dealing with 50 billion targets)

BLAST Algorithm: List, Scan, Extend

- The **BLAST search algorithm** finds a match between a query and a database sequence, and then **extends the match** in either direction:
  - Phase 1: Compile a **list of words** (w=3) above **threshold T** of pairwise alignment.
  - Phase 2: When occurs, BLAST **extends the word** pairs that surpass a **cut-off score S**.
  - Phase 3: A **trace-back** procedure to assign the alignment (with insertion, deletion, mismatch)
- The **search results** consist:
  - **Highly related sequences** from the database
  - **Marginally related sequences** along with **a scoring scheme** to describe the degree of relatedness.

Algorithm

List

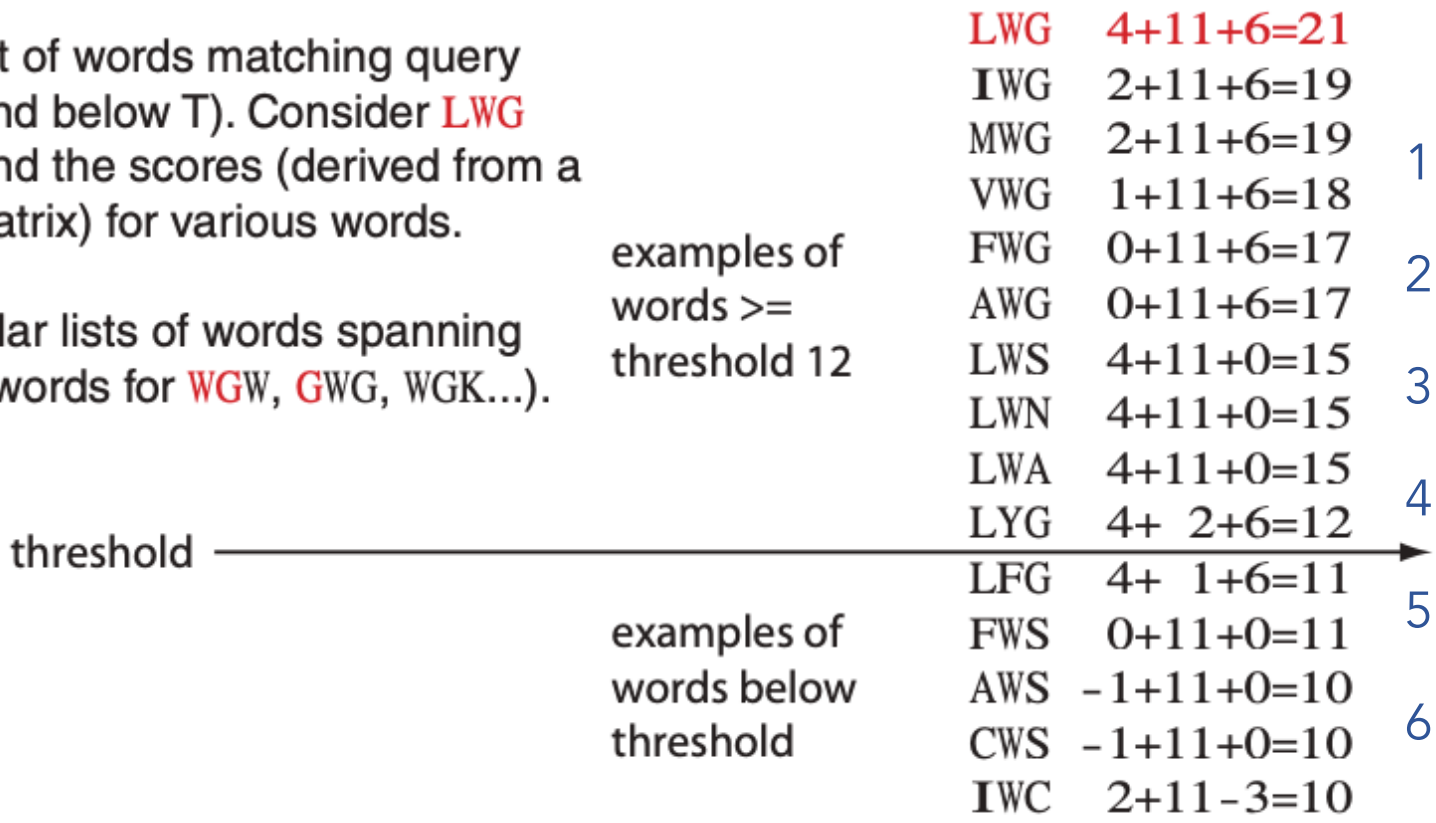
Phase 1: Setup: compile a list of words (w=3) above threshold T

- Query sequence: human beta globin NP_000509.1 (includes ...VTALWGKVNVD...).
  This sequence is read; low complexity or other filtering is applied; a "lookup" table is built.

- Words derived from query sequence (HBB): VTA TAL ALW LWG WGK GKV KVN VNV NVD

- Generate a list of words matching query (both above and below T). Consider LWG in the query and the scores (derived from a BLOSUM62 matrix) for various words.

- Generate similar lists of words spanning the query (e.g. words for WGW, GWG, WGK...).



1. Fixed length w of "words"
2. Scoring with "BLOSUM62"
3. Establish threshold value T
4. w=3 for protein; w=11 for DNA
5. "word" above T are collected
6. We call the "words" as "hits"

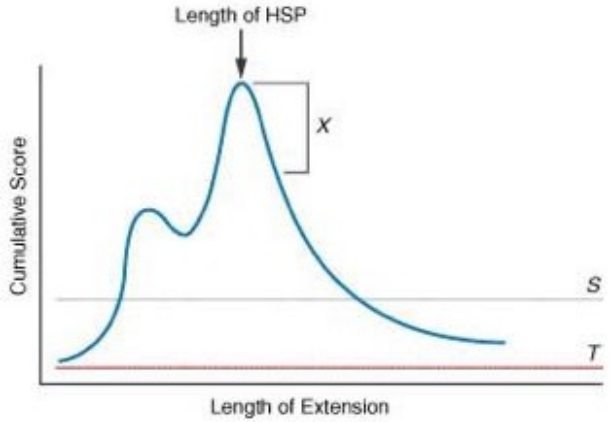
Scan & Extend

Phase 2: Scanning and extensions

- Select all the words above threshold T (LWG, IWG, MWG, VWG, FWG, AWG, LWS, LWN, LWA, LYG)
- Scan the database for entries ("hits") that match the compiled list
- Create a hash table index with the locations of all the hits for each word
- Perform gap free extensions
- Perform gapped extensions

```
LTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV  HBB
L+P +K+ V A WGKV  +  E G EAL R+ + +P T+ +F  F      D   G+ +V
LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF------DLSHGSAQV  HBA
```

extension — word pair from first phases of search "hits" alpha globin, triggers extension — extension



1. Search an **index** of database to find **entries** for the "word"
2. One-hit method: perform **ungapped** extension
3. Two-hits method: generate **two word pairs** with distance A from each other, and perform **ungapped** extension.
4. Extension process is **terminated** when score falls below cutoff
5. Gapped extension for sufficiently HSP.

BLAST extends hits to find alignments called **High-scoring Segment Pair (HSP)**

Scan & Extend

1. One-Hit Method

The **one-hit method** relies on identifying a single high-scoring word pair or seed match between the query and database sequence to initiate an ungapped extension. This approach works as follows:

- **Step 1: Seed Identification**: BLAST identifies a word pair (a short, exact match of a specified length) with a score above a given threshold.
- **Step 2: Ungapped Extension**: When a single high-scoring word pair is found, BLAST begins extending the match in both directions without introducing gaps, calculating the alignment score at each step.
- **Scoring and Termination**: The ungapped extension continues as long as the alignment score improves or remains above a certain drop-off threshold. Once the score falls below the threshold, the extension stops, and the alignment is recorded if it is significant.

The one-hit method is faster but can lead to false positives, as a single seed may not be as reliable for finding true alignments in complex sequences.

- **One-Hit Method**:
  - Requires only one high-scoring word pair to start ungapped extension.
  - Faster but can lead to false positives, especially in large and complex databases.
- **Two-Hit Method**:
  - Requires two word pairs within a distance A, leading to fewer false positives.
  - More accurate for finding biologically meaningful alignments.

The **two-hit method** is more stringent and is the default in many BLAST implementations because it reduces false positives. It requires finding **two high-scoring word pairs** (seeds) that are within a specified distance A of each other on the same diagonal (i.e., in a compatible alignment frame). Here's how it works:

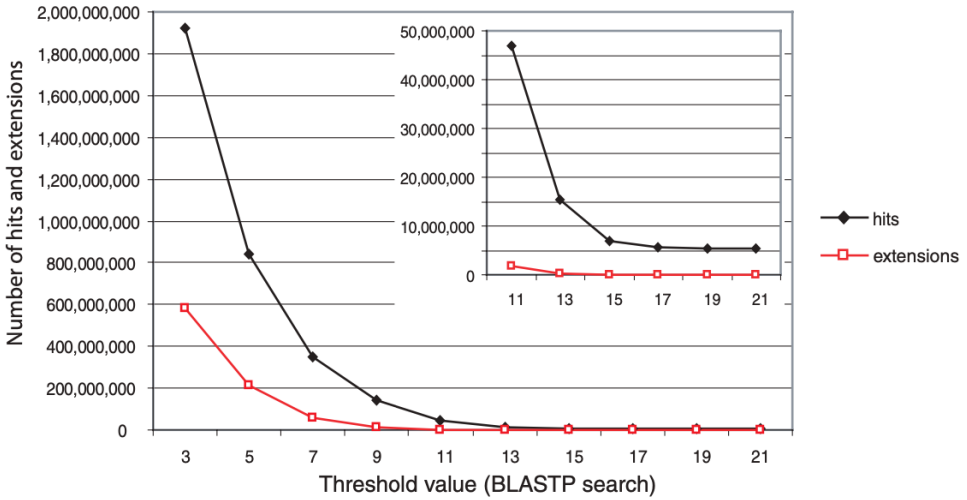
- **Step 1: First Seed Identification**: BLAST initially identifies a word pair with a score above a set threshold.
- **Step 2: Second Seed Search within Distance A**: Instead of extending immediately, BLAST searches for a **second high-scoring word pair** within a predefined distance A from the first seed. The distance A is set to ensure that the two hits are likely to be part of the same alignment region.
  - For example, if A is set to 40, BLAST will search for another hit within 40 residues or nucleotides from the first seed.
- **Ungapped Extension**: Once two seeds are identified within distance A of each other, BLAST initiates an ungapped extension from the first hit through to the second and extends outward in both directions.
- **Scoring and Termination**: As with the one-hit method, extension continues until the score begins to drop below a threshold or falls too far below the best score reached so far.

The two-hit method is more computationally intensive but results in higher alignment accuracy. By requiring two nearby hits, it is more likely that the extension will represent a meaningful alignment rather than a random match.

Trace-back

Phase 3: Traceback
- Calculate locations of insertions, deletions, and matches (for alignments saved in Phase 2)
- Apply composition-based statistics (for BLASTP, TBLASTN)
- Generate gapped alignment

FIGURE 4.12 Schematic of the original BLAST algorithm. In the setup phase a query sequence (such as human beta globin) is analyzed with a given word size (e.g., $w = 3$), and a list of words is compiled having a threshold score (e.g., $T = 11$). Several possible words derived from the query sequence are listed in the figure (from LWG to IWC); in a BLAST search there are 8000 words compiled for $w = 3$. For a given word, such as the portion of the query sequence consisting of LWG, a list of words is compiled with scores greater than or equal to some threshold T (e.g., 12). In this example, 15 words are shown along with their scores from a BLOSUM62 matrix; 10 of these are above the threshold, and 5 are below. In phase 2, a database is scanned to find entries that match the compiled word list. Ungapped and gapped extensions are performed, although (to increase efficiency) positions are not saved. The database hits are extended in both directions to obtain high-scoring segment pairs (HSPs). If a HSP score exceeds a particular cutoff score S, it is reported in the BLAST output. In phase 3, a trace-back is performed and locations of insertions and deletions are recorded. Note that in this particular example the word pair that triggers the extension step is not an exact match (see boxed residues LWG aligned to AWG). The main idea of the threshold T for protein searches is to also allow both exact and related but nonexact word hits to trigger an extension. For nucleotide BLASTN searches, exact matches are required rather than words above a threshold.



When threshold is raised:

Speed: increased **Hits:** fewer

Distantly related match: missed

Conclusion: the lower threshold yields a more **accurate search**

BLAST Algorithm: Raw Score

- 1) Raw Scores (S values) from an Alignment

(a)

```
 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    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
```

(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  VTALWGKVNVD--EVGGEALGRLL  33
           V  +WGKV  D    G E L RL
Sbjct  11  VLNVWGKVEADIPGHGQEVLIRLF  34

match       4  11 5   6    6 5 4 5    sum of matches: +60 (round up to +61)
              6 4                 4
mismatch   -1 1    0   -2 -2  -4  0   sum of mismatches: -13
            -2      0   -3   0
gap open             -11              sum of gap penalties: -13
gap extend            -2
                                      total raw score: 61 - 13 - 13 = 35
```

$$S = (\Sigma Mij) - cO - dG,$$

where

M = score from a similarity matrix for a particular pair of amino acids (ij)
c = number of gaps
O = penalty for the existence of a gap
d = total length of gaps
G = per-residue penalty for extending the gap

FIGURE 3.5 Pairwise alignment of human beta globin (the "query") and myoglobin (the "subject"). (a) The alignment from the search shown in **Figure 3.4**. Note that this alignment is local (i.e., the entire lengths of each protein are not compared), and there are many positions of identity between the two sequences (indicated with amino acids intervening between the query and subject lines; see rows with arrows). The alignment contains an internal gap (indicated by two dashes). (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.

BLAST Algorithm: Statistics and *E* value

- The **statistical significance** of a BLAST search:
  - Whether the alignment present **significant matches** or **occurs by chance**.
- **2) Expect value (*E* values)** from an Alignment
  - A **query sequence** to a database of **random sequence** (uniform length), the scores have the shape of an **extreme value distribution** (not normal distribution)
  - The extreme value distribution is **skewed to the right**, with tail that decays in x (x^2 in normal distribution)
  - The **formula** that describe the likelihood that a particular **BLAST score occurs by chance**:

$$E = Kmne^{-\lambda S}$$

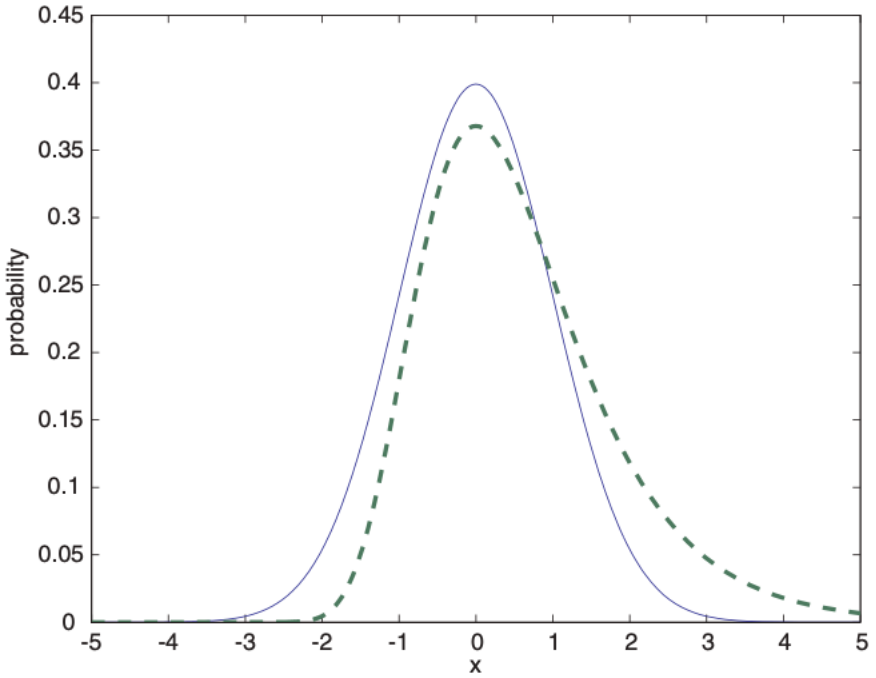


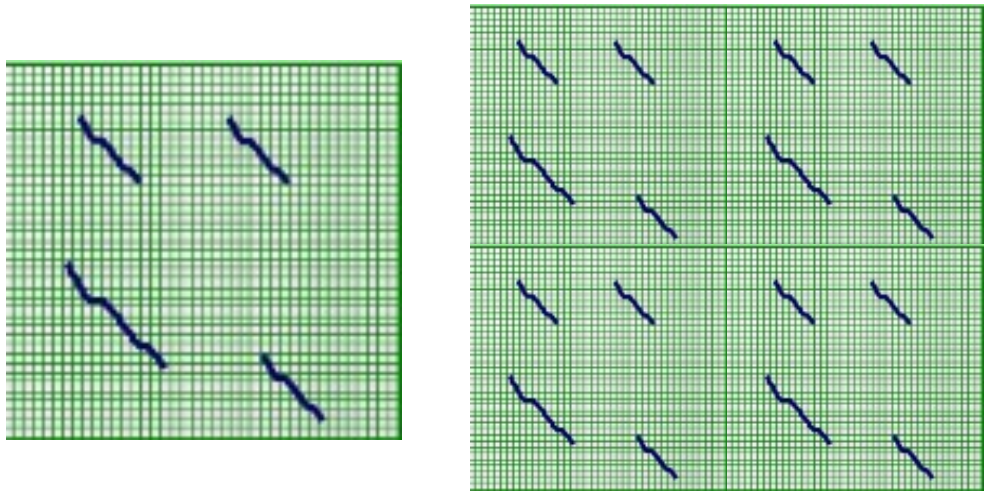
FIGURE 4.14 Normal distribution (solid line) is compared to the extreme value distribution (dotted line). Comparing a query sequence to a set of uniform-length random sequences usually generates scores that fit an extreme-value distribution (rather than a normal distribution). The area under each curve is 1. For the normal distribution, the mean (μ) is centered at zero, and the probability Z of obtaining some score x is given in terms of units of standard deviation (σ) from x to the mean: $Z = (x - \mu)/\sigma$. In contrast to the normal distribution, the extreme value distribution is asymmetric with a skew to the right. It is fit to the equation $f(x) = (e^{-x})(e^{-e^{-x}})$. The shape of the extreme value distribution is determined by the characteristic value u and the decay constant λ ($u = 0$; $\lambda = 1$).

BLAST Algorithm: Statistics and *E* value

- The **expected number** of HSPs having score S (or better) by chance alone is defined:

$$E = Kmne^{-\lambda S}$$

- ***S*: raw score**, similarity of each pairwise comparison (HSPs) and partly based upon the scoring matrix selected.
- ***E*: expect value**, number of different alignment with scores (>= **S**) that are expected to occur by chance.
- **λ, K:** Karlin Altschul statistics, constant
- ***m*:** length of query sequence
- ***n*:** length of entire database
- ***m, n*:** defines the size of the search space



search space

BLAST Algorithm: properties

- An ***E* value** is related to a probability **value *p***.

$$E = Kmne^{-\lambda S}$$

- The **value of *E*** decreases exponentially with **increasing *S***.
- **Higher *S* value** correspond to **better alignments**.
- The ***E* value** for aligning a pair of **random sequences** must be **negative**
- **Parameter *K*** describes the **searching space** (database).
- For **E=1**, in a database of this particular size, one match with a given score is expected to occur by chance.
- If the database were **twice as big**, there would be **twice** the likelihood of finding a score equal to **S** by chance
- The **formula** is developed for ungapped alignment

BLAST Algorithm: Bit Score

- 3) Bit score (*S'* score) from an Alignment
  - Raw scores are calculated from the substitution matrix and gap penalty parameters that are chosen.
  - The bit score S' is calculated from the raw score by normalizing with the statistical variables that define a given scoring system.
- Bit score from different alignment, even those employing different scoring matrices can be compared.

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

$$E = mn \times 2^{-S'} \qquad E = Kmne^{-\lambda S}$$

  - S' is bit score
  - E value corresponding to a given bit score as defined
- Bit score account for the scoring system and describe the information content inherent
- The Bit Score does not depend on the search space size.

BLAST Algorithm: *E* value and *p* value

- The ***p* value** is the **probability** of a chance alignment occurring with the score in question
- The ***p* value** is calculated by relating the observed **alignment score *S*** to the expected distribution of **HSP scores** by comparisons of **random sequences** in same database.
- The ***p* and *E* values** are different ways of representing the significance of the alignment.

$$p = 1 - e^{-E}$$

$$E = Kmne^{-\lambda S}$$

E values of about 1 to 10 are far easier to interpret than corresponding *p* values.

Very small *E* values are very similar to *p* values.

E	*p*
10	0.99995460
5	0.99326205
2	0.86466472
1	0.63212056
0.1	0.09516258 (about 0.1)
0.05	0.04877058 (about 0.05)
0.001	0.00099950 (about 0.001)
0.0001	0.00010000

BLAST Scoring System

- **Raw score (S):** Sum of scores for each aligned position and scores for gaps
  - S = **λ**(matches) - **λ**(mismatches) - **λ**(gap penalties)
  - **note:** this score varies with the scoring matrix used and thus may not be meaningfully compared for different searches
- **Bit score (S'):** Version of the raw score that is normalized by the scale of the scoring matrix (**λ**) and the scale of the gap penalty (K)
  - $S' = (\lambda S - \ln(K)) / \ln(2)$
  - **note:** because it is normalized the bit score can be meaningfully compared across searches
- **E value:** Number of alignments with bit score S' or better that one would **expect to find by chance** in a search of a database of the same size
  - $E = mn2^{-S'}$ m, n = effective length of query sequence, database
  - **note:** E values may change if databases of different sizes are searched

BIO309: Computational Biology, 2025

Lecture 11: Multiple Sequence Alignment

Zhengyu LIANG, PhD (梁征宇)

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Sum of Pairs Score (SP-Score)

- Consider pairwise alignment of sequence

  a_i and a_j

  imposed by a multiple alignment to of k sequence

- Denote the score of this suboptimal (not necessarily optimal) pairwise alignment as

  $S^*(a_i, a_j)$

- Sum up the pairwise scores for a multiple alignment:

  $$S^*(a_1, \ldots, a_k) = \sum_{i,j} S^*(a_i, a_j)$$

	/
a_i	ATG-CA-AT
a_j	A-G-CATAT
a_1	ATG-CA-AT
...	A-G-CATAT
a_k	ATCCCATTT

S : substitution matrix*

Sum of Pairs Score: Example

- Each column is scored by summing the scores of all pairs of symbols in that column:

$$S_{l=1}\begin{pmatrix} A \\ A \\ A \end{pmatrix} = Score(A, A) + Score(A, A) + Score(A,A)$$

- and extend to :

$$S_{l=3}\begin{pmatrix} G \\ G \\ C \end{pmatrix} = Score(G, G) + Score(G, C) + Score(G,C)$$

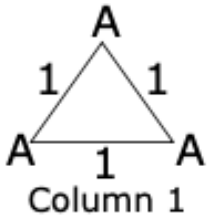
- Remark:

$$Score(-, -) = 0$$

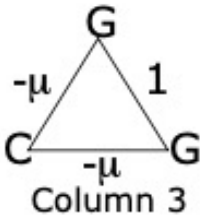
/

a_1	ATG-CA-AT
…	A-G-CATAT
a_k	ATCCCATTT

S* *: substitution matrix*



Score=3



Score = $1 - 2\mu$

Sum of Pairs Score: Interpretation

- **No theoretical justification** for the score.

	/
a_1	ATG-CA-AT
…	A-G-CATAT
a_k	ATCCCATTT

A A A A
A A A A
A A C G
A C C C

6 0 -2 -4

match: +1; *mismatch: -1*

S* : substitution matrix

$$S^*(a_1, \ldots, a_k) = \sum_{i,j} S^*(a_i, a_j)$$

Focusing more on the mutations of SP-Score

A A A A
A A A A
A A A A
A A A I
A A I I
A I I I

15 10 7 6

match: +1; *mismatch: 0*

Information Entropy（熵）

The use of $\log_2 p(x)$ in the formula for **information entropy** stems from its foundation in information theory, introduced by Claude Shannon. The logarithmic function provides a natural way to quantify the uncertainty or "surprise" associated with observing a particular outcome. Here's a detailed explanation of why this format is used:

1. Logarithm Captures "Surprise" or Information Content

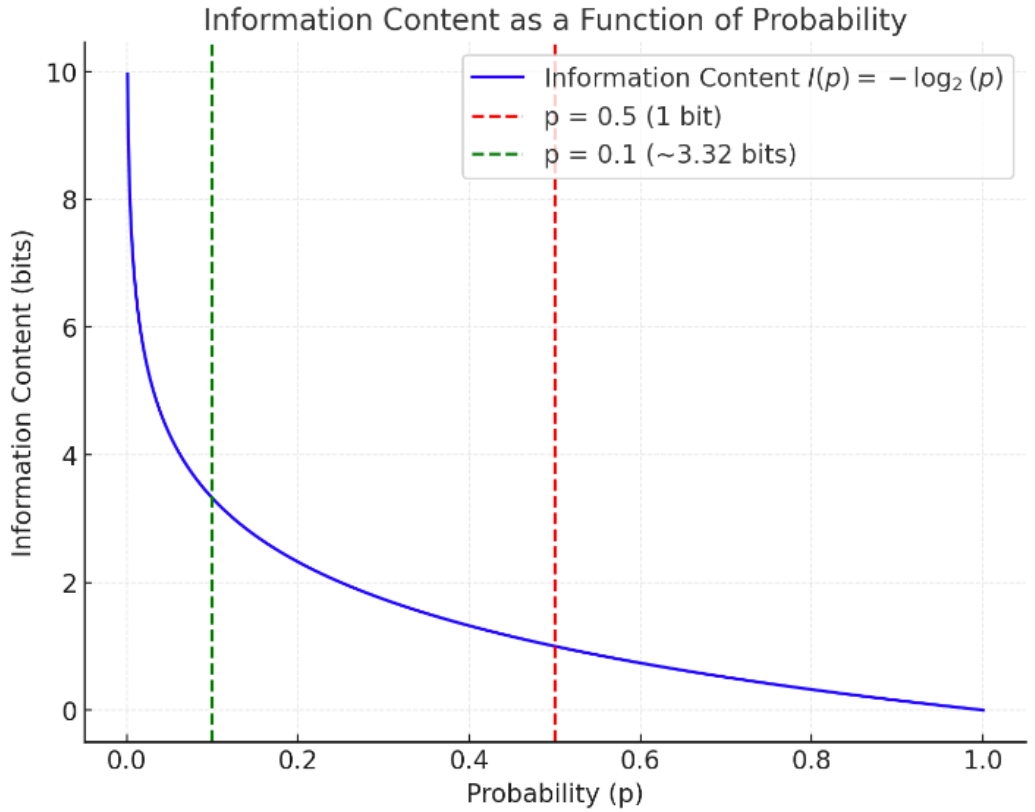
- The **information content** of an event is inversely related to its probability:

$$I(x) = -\log_2 p(x)$$

  - Rare events (low $p(x)$) have high information content, as they are surprising
  - Common events (high $p(x)$) carry less information, as they are expected.

For example:

- If $p(x) = 0.5$: $I(x) = 1$ bit of information.
- If $p(x) = 0.1$: $I(x) = 3.32$ bits (more surprising and informative).



Information Entropy（熵）

3. Logarithm is Additive for Independent Events

- Logarithms have the property:

$$\log(ab) = \log(a) + \log(b)$$

This reflects how probabilities combine for independent events and allows entropy to be an additive measure over a set of possible outcomes.

Example:

- For independent outcomes x_1 and x_2, with probabilities $p(x_1)$ and $p(x_2)$:

$$I(x_1, x_2) = -\log_2(p(x_1) \cdot p(x_2)) = -\log_2 p(x_1) - \log_2 p(x_2)$$

Why Not Linear Functions?

- A linear function like $p(x)$ does not adequately reflect the diminishing marginal contribution of highly probable outcomes or the high surprise of rare events.
- Logarithmic functions naturally compress scales, giving appropriate weight to rare and frequent events.

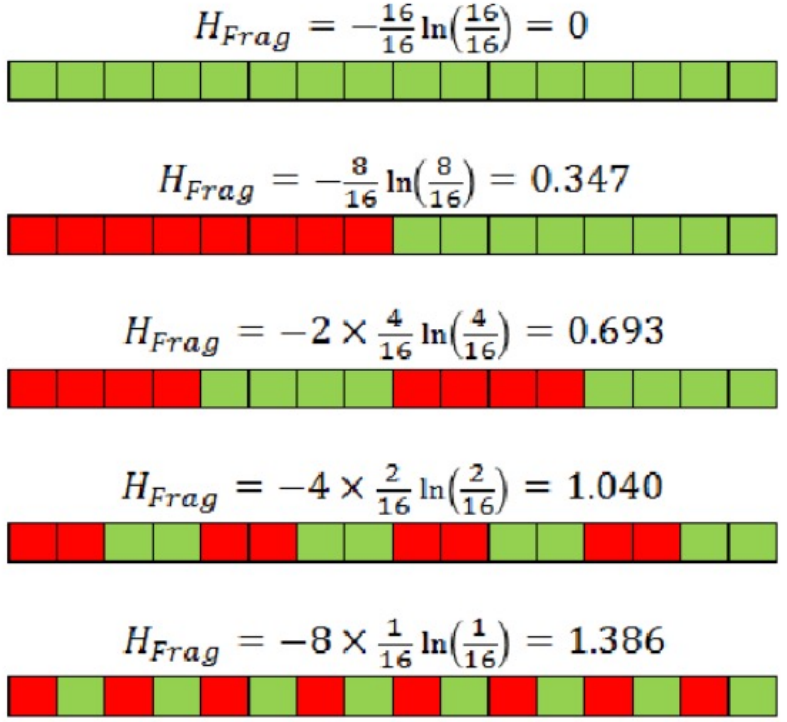
Entropy（熵）based Score

- Columns that can be communicated using few bits are good
- Information theory tells us that an optimal code uses $-\log_2 p$ bits to encode a message of **probability** p (larger p ~ less information)

$$Entropy = - p(x_i) \log p(x_i)$$

- Each column of entropy score:

$$Entropy = -\sum_{x=A,T,G,C} p_x \log p_x$$



A A A A
A A A A
A A A A
A A A I
A A I I
A I I I

15 10 7 6

A A A A
A A A A
A A C G
A C C C

6 0 -2 -4

Entropy（熵）based Score

$$Entropy = -\sum_{x=A,T,G,C} p_x \log p_x$$

- Define frequencies for occurrence of each letter in each column

  $p_A = 1$, $p_T = p_G = p_C = 0$ (1st column)

  $p_A = 0.75$, $p_T = 0.25$, $p_G = p_C = 0$ (2nd column)

  $p_A = p_T = p_G = p_C = 0.25$ (3rd column)

- Each column of entropy score:

  -[1 * log(1) + 0 * log(0) + 0 * log(0) + 0 * log(0)] = 0

  -[3/4 * log(3/4) + 1/4 * log(1/4) + 0 * log(0) + 0 * log(0)] = 0.811

  -[1/4 * log(1/4) + 1/4 * log(1/4) + 1/4 * log(1/4) + 1/4 * log(1/4)] = 2.0

- Multiple Alignment Entropy Score:

  Alignment Entropy: 0 + 0.811 + 2.0 = 2.811

Best

A A A
A A G
A A T
A T C

$$entropy\begin{pmatrix}A\\A\\A\\A\end{pmatrix} = 0$$

Worse

$$entropy\begin{pmatrix}A\\T\\G\\C\end{pmatrix} = -\sum \frac{1}{4}\log\frac{1}{4} = -4(\frac{1}{4} * -2) = 2$$

$$\sum all\ columns = -\sum_{x=A,T,G,C} p_x \log p_x$$

Heuristic

Multiple Alignment from Pairwise Alignment

- Every **multiple alignment** induces **pairwise alignments**

```
x  AC-GCGG-C
y  AC-GC-GAG
z  GCCGC-GAG
```

```
x  AC-GCGG-C        x  AC-GCGG-C        y  AC-GC-GAG
y  AC-GC-GAG        z  GCCGC-GAG        z  GCCGC-GAG
```

- Given 3 arbitrary **pairwise alignment**, can we construct **a multiple alignment** that induce them?

NOT ALWAYS: pairwise alignment may be inconsistent

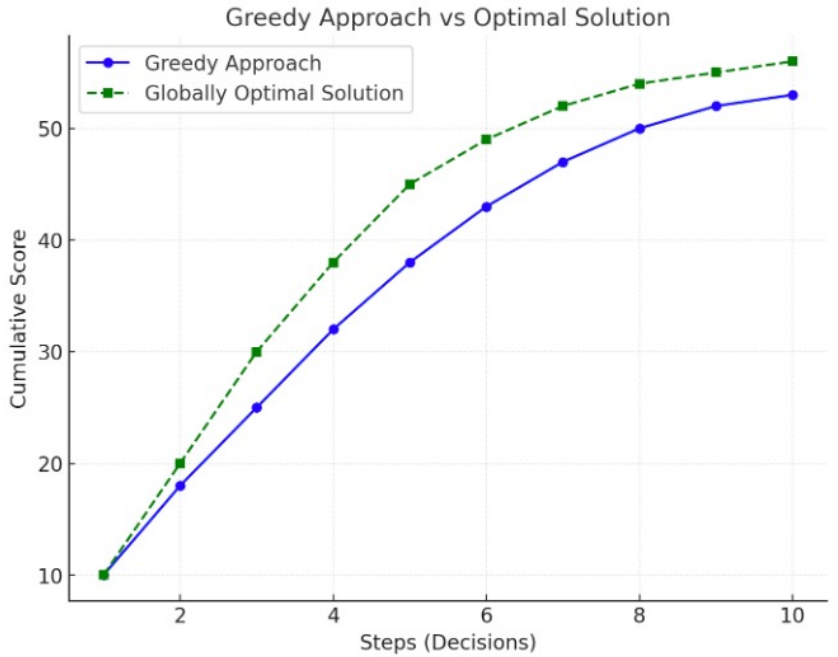
- From **an optimal multiple alignment**, we can infer pairwise alignment between all pairs of sequences, but they are **not necessarily optimal**.
- It is difficult to **infer a "good" multiple alignment** from optimal pairwise alignment between all sequences.
- **Can we have another angle question?** Optimal for **base** → optimal **sequence**

Multiple Alignment: Greedy Approach

The **Greedy Approach** is a problem-solving paradigm used in algorithms where decisions are made step-by-step, choosing the most optimal option at each stage, with the hope that this local optimization leads to a globally optimal solution.

Key Characteristics of the Greedy Approach:

1. **Local Optimal Choice:** At every step, the algorithm selects the best possible option based on the current situation without considering the entire problem.
2. **Irrevocable Decisions:** Once a choice is made, it is final and cannot be changed later.
3. **Feasibility:** The chosen option must be valid within the constraints of the problem.
4. **Optimal Substructure:** The problem should have a structure such that the globally optimal solution can be constructed from locally optimal solutions.



Key Observations:

1. The greedy approach performs well initially but may deviate from the global optimum as decisions accumulate.
2. The gap between the two lines illustrates the potential for suboptimal solutions when using a greedy approach.

Star alignment: pairwise to heuristic multiple alignment

Given:

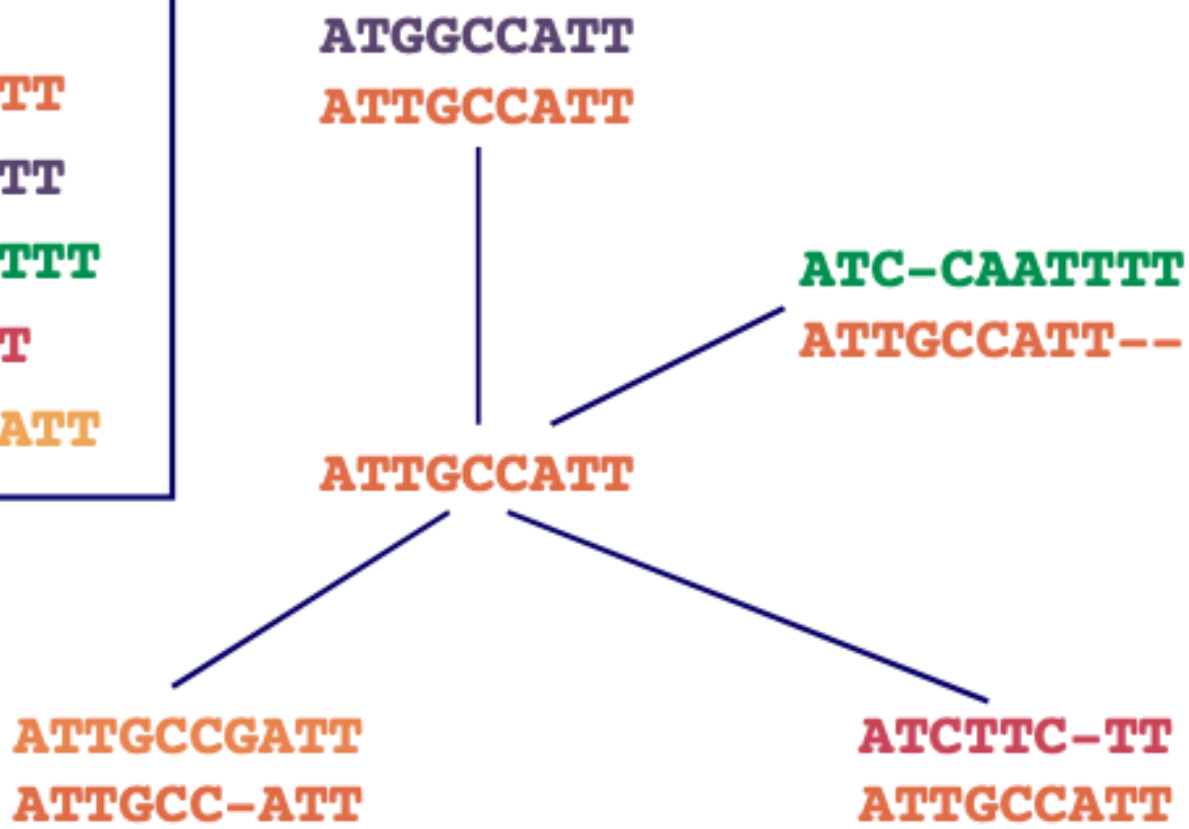
ATTGCCATT

ATGGCCATT

ATCCAATTTT

ATCTTCTT

ATTGCCGATT



Result:

ATTGCC-	A	TT--
ATGGCC-	A	TT--
ATC-CA-	A	TTTT
ATCTTC-	-	TT--
ATTGCCG	A	TT--

Star alignment: Example

The five sequences

S_1	A	T	T	G	C	C	A	T	T
S_2	A	T	G	G	C	C	A	T	T
S_3	A	T	C	C	A	A	T	T	T	T
S_4	A	T	C	T	T	C	T	T	
S_5	A	C	T	G	A	C	C		

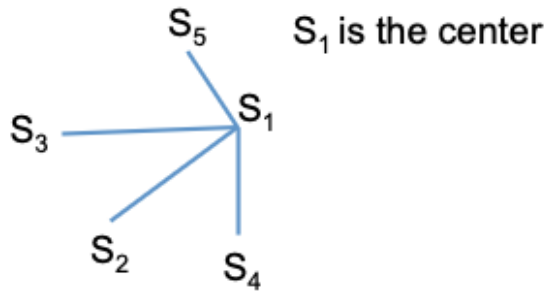
	S_1	S_2	S_3	S_4	S_5	
S_1	-	7	-2	0	-3	**2**
S_2	7	-	-2	0	-4	**1**
S_3	-2	-2	-	0	-7	**-11**
S_4	0	0	0	-	-3	**-3**
S_5	-3	-4	-7	-3	-	**-17**
	2	**1**	**-11**	**-3**	**-17**	

S1 is the sequence

most similar to the rest

S_1	A	T	T	G	C	C	A	T	T	
S_2	A	T	G	G	C	C	A	T	T	
S_1	A	T	T	G	C	C	A	T	T	- -
S_3	A	T	C	-	C	A	A	T	T	T T
S_1	A	T	T	G	C	C	A	T	T	
S_4	A	T	C	T	T	C	-	T	T	
S_1	A	T	T	G	C	C	A	T	T	
S_5	A	C	T	G	A	C	C	-	-	

pairwise alignment to the S1



Let's use the alignment of S_1 and S_2.

S_1	A	T	T	G	C	C	A	T	T		S_1 and S_2 are aligned
S_2	A	T	G	G	C	C	A	T	T		

Now, let's add S_3, using its alignment to S_1.

S_1	A	T	T	G	C	C	A	T	T	-	-	S_1, S_2, and S_3 are aligned
S_2	A	T	G	G	C	C	A	T	T	-	-	
S_3	A	T	C	-	C	A	A	T	T	T	T	

Then, let's add S_4, using its alignment to S_1.

S_1	A	T	T	G	C	C	A	T	T	-	-	S_1, S_2, S_3, and S_4 are aligned
S_2	A	T	G	G	C	C	A	T	T	-	-	
S_3	A	T	C	-	C	A	A	T	T	T	T	
S_4	A	T	C	T	T	C	-	T	T	-	-	

Finally, let's add S_5, using its alignment to S_1.

S_1	A	T	T	G	C	C	A	T	T	-	-	S_1, S_2, S_3, S_4 and S_5 are aligned
S_2	A	T	G	G	C	C	A	T	T	-	-	
S_3	A	T	C	-	C	A	A	T	T	T	T	
S_4	A	T	C	T	T	C	-	T	T	-	-	
S_5	A	C	T	G	A	C	C	-	-	-	-	

Merge multiple alignments

The running time is $k(k-1)*n^2$: $O(k^2 n^2)$

Actually, the star alignment method is not used much.

Star alignment: pairwise to heuristic multiple alignment

- Choose one sequence to be the center
- Align all pairwise sequences with the center
- Merge the alignments: use the center as reference
- **Rule "once a gap, always a gap"**

- **Two ways of choosing the center**
  - Try all possibilities and choose the resulting alignment that **gives highest score**
  - Take sequence S_c that maximizes (need to compute all pairwise alignments)

$$\sum i \neq c \; pairwise\text{-}score\,(S_c, S_i)$$

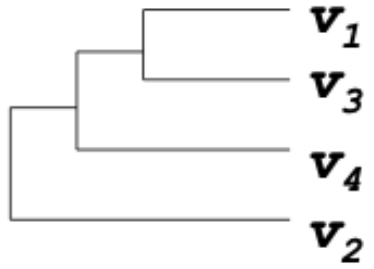
- **Works well for close sequences:**
  - Gaps in consensus string are permanent
  - Use profiles to compare sequences

Clustal W

- **Popular** multiple alignment tool today
- "*W*" stands for "***weighted***" (different parts of alignment are **weighted differently**).
- **Three-step process:**
  - 1) Compute **all pairwise alignments** and store in similarity matrix M: $M[i,j] = sim(s_i, s_j)$
  - 2) Compute the guide tree using **hierarchical clustering**
    - Choose the *smallest M[i,j]*
    - Let S_i and S_j form a **new branch** of the tree by average
    - **Iterate** until M has only **one column/row**
  - 3) **Progressive Alignment** guided by the tree: from leaves towards the root.

	v_1	v_2	v_3	v_4
v_1	-			
v_2	.17	-		
v_3	.87	.28	-	
v_4	.59	.33	.62	-

Similarity: exact matches/sequence length



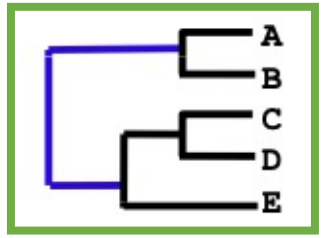
Calculate:

$v_{1,3}$ = alignment (v_1, v_3)

$v_{1,3,4}$ = alignment$((v_{1,3}), v_4)$

$v_{1,2,3,4}$ = alignment$((v_{1,3,4}), v_2)$

Clustal W: Example

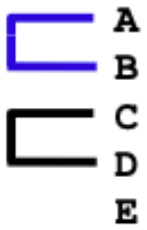


	A	B	C	D	E
A		17	59	59	77
B			37	61	53
C				13	41
D					21


```
A   C   PADKTNVKAAWGKVGAHAGEYGA
B   D   AADKTNVKAAWSKVGGHAGEYGA

A   PEEKSAVTALWGKVNVDEYGG
E   B   GEEKAAVLALWDKVNEEEYGG
```

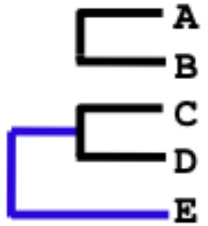
	A	B	E	CD
A		17	77	59
B			53	49
E				31




```
A   C   PADKTNVKAAWG_KVGAHAGEYGA
B   D   AADKTNVKAAWS_KVGGHAGEYGA
C   E   AA__TNVKTAWSSKVGGHAPA__A
D
E

A   PEEKSAV_TALWG_KVN__VDEYGG
B   GEEKAAV_LALWD_KVN__EEEYGG
C   PADKTNVKAA_WG_KVGAHAGEYGA
D   AADKTNVKAA_WS_KVGGHAGEYGA
E   AA__TNVKTA_WSSKVGGHAPA__A
```

	E	CD	AB
E		31	65
CD			54




```
A   ...P...
B   ...G...
C   ...P...

D   ...A...
E   ...A...
F   ...Y...
```

Score of this column

```
(2*s(P,A)+s(P,Y)+
 2*s(G,A)+s(G,Y)+
 2*s(P,A)+s(P,Y) ) / 9
```

- Problem with CLUSTAL W (progressive methods)
- Dependence of the initial pairwise sequence alignment
- Propagating errors form initial alignments

Clustal W

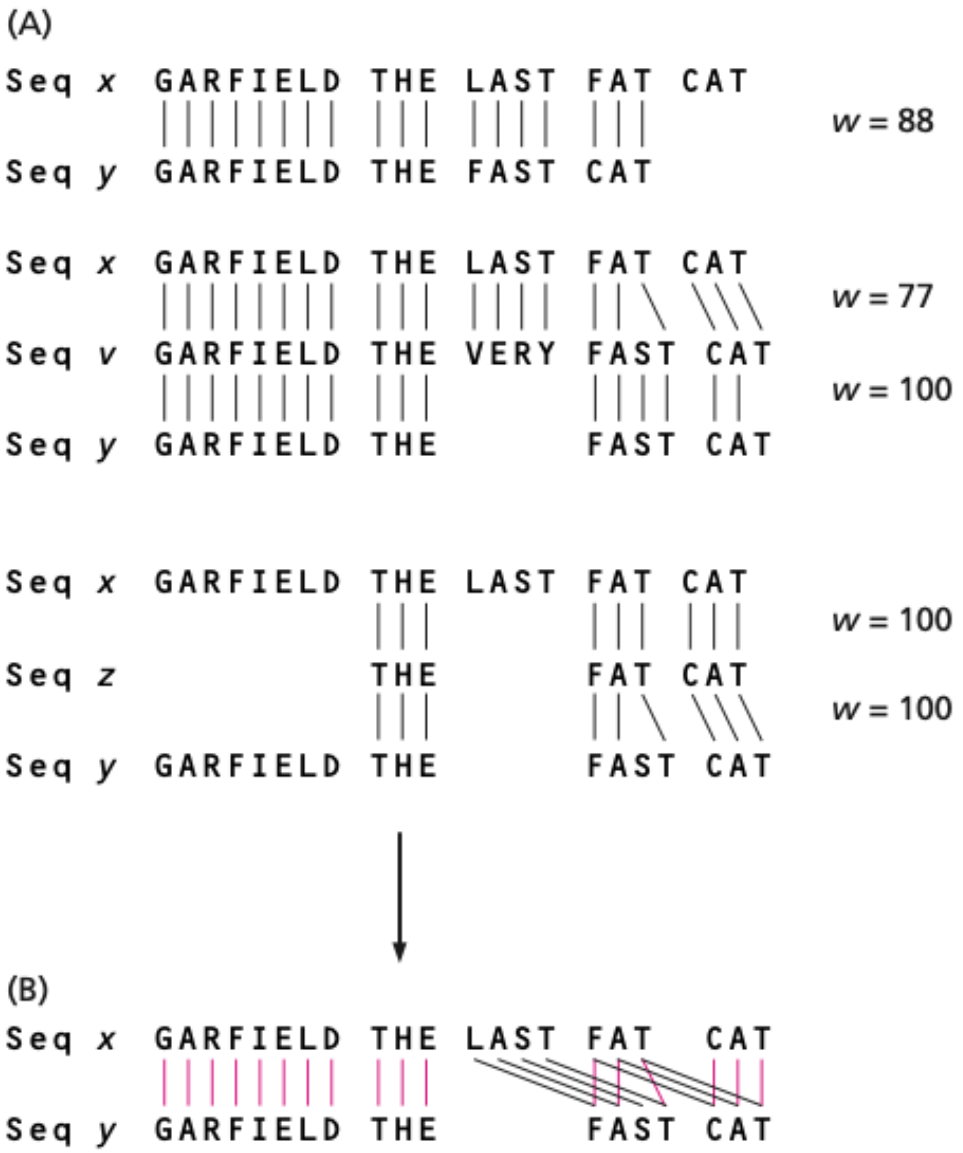
Note how the region of a conserved histidine (▼) varies depending on which of five prominent algorithms is used

- **Histidine (H) residue** has a critical role in binding oxygen in globins, and should be aligned.

Progressive

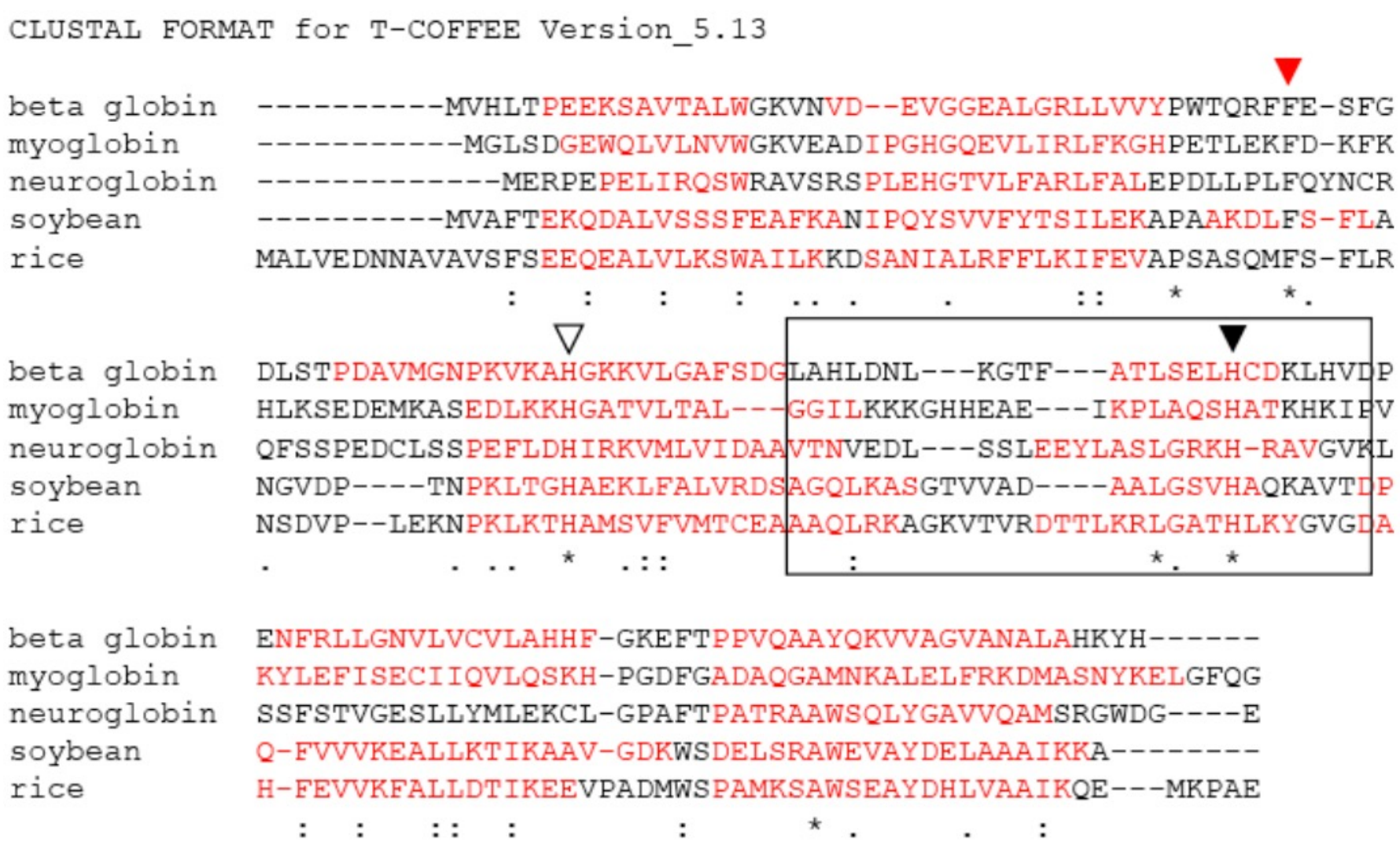
T-Coffee

- **T-Coffee:** Tree-Based Consistency Objective Function For alignment Evaluation
- **Three-step process:**
  - 1) Construct a library of **pairwise alignments** (weight -> similarity)
  - 2) Consistency alignment: for pair (A, B), **consider sequence C to form A-C-B**
  - 3) **Progressive Alignment** using the tree but using weights from extended library
- **More accurate & Slower** than CLUSTALW



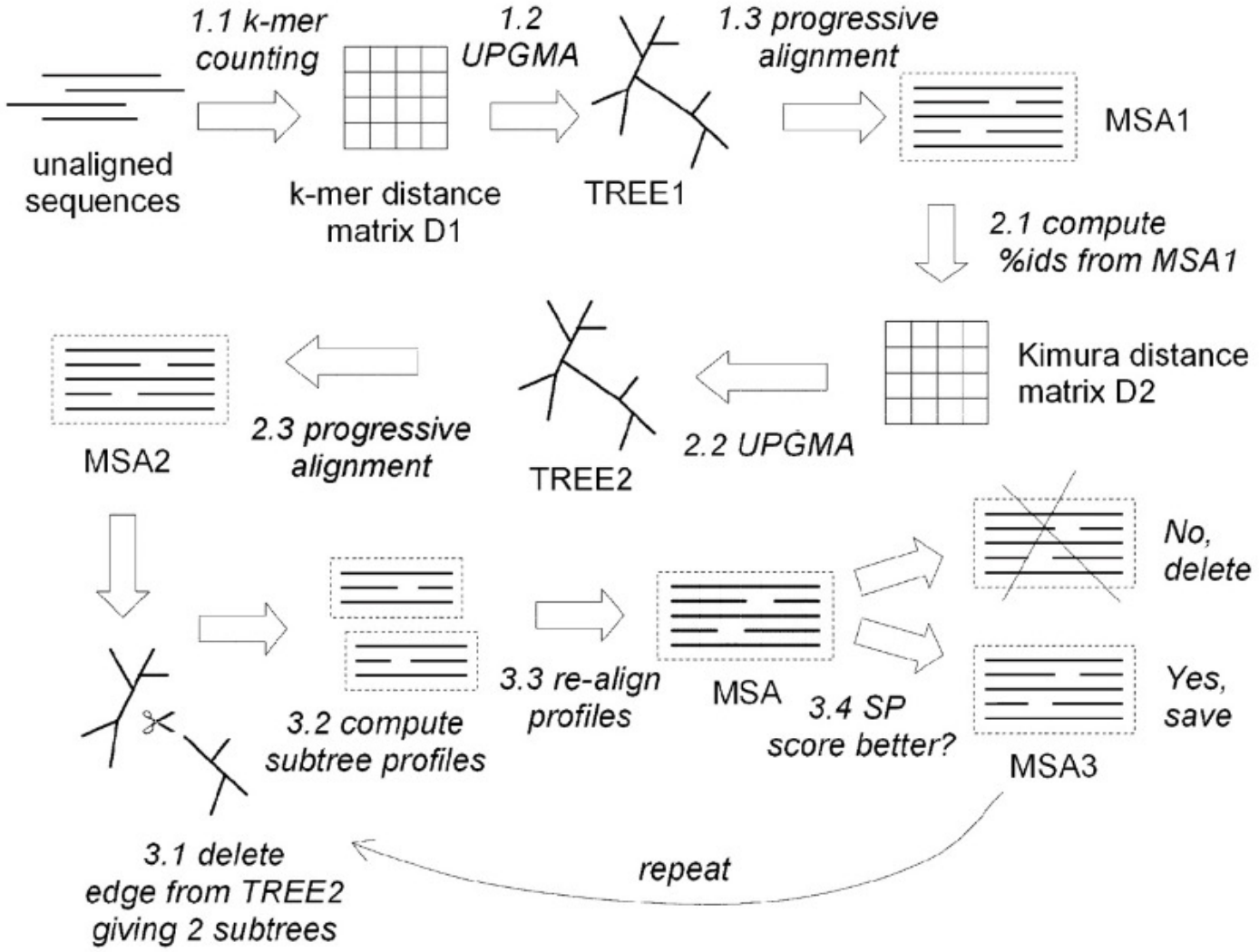
Progressive

T-Coffee



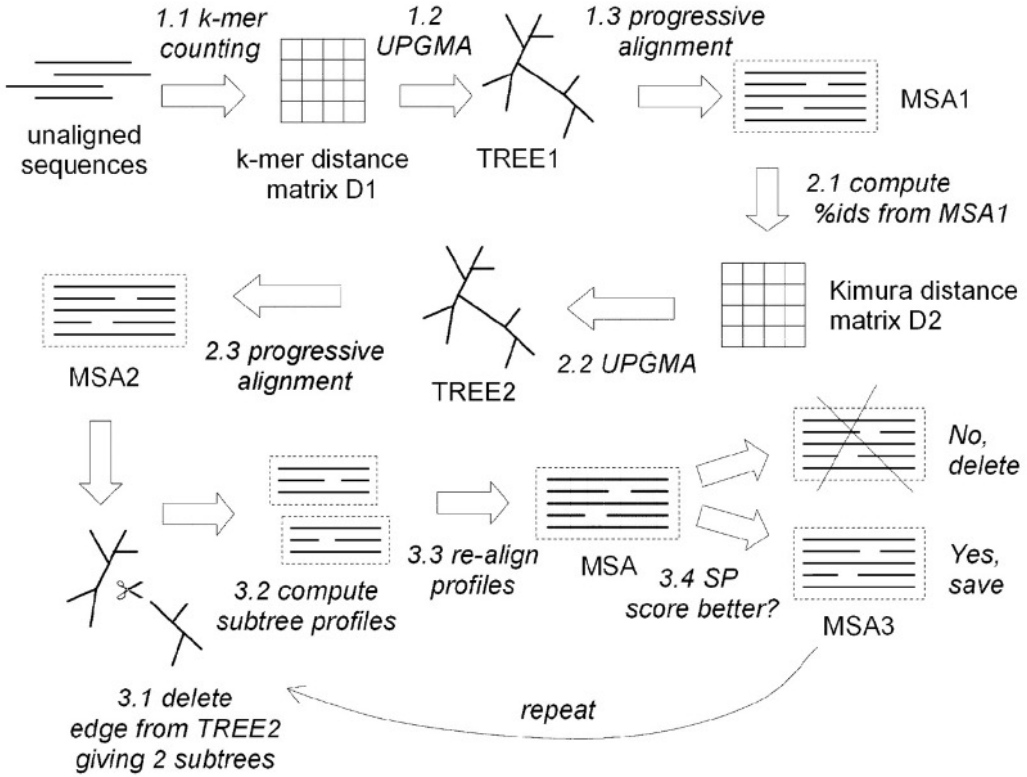
MUSCLE

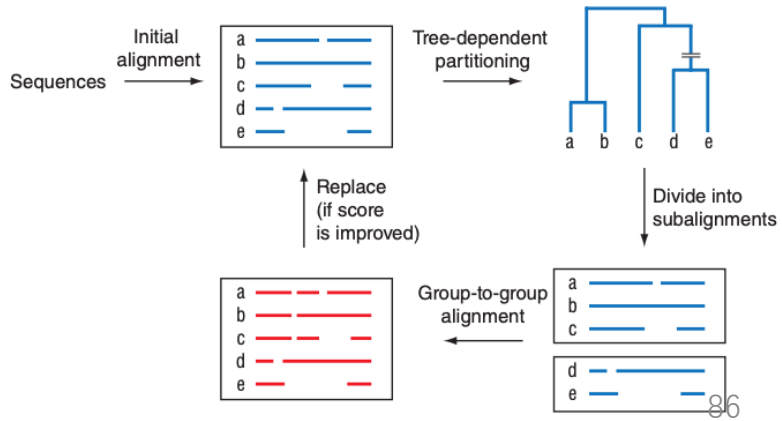
- Build quick approximate sequence **similarity tree** (without pairwise alignment but computing short ***hits*** between any pair of sequence)
- Compute **MSA** using the tree
- Compute **pairwise distances** from MS and new tree
- **Re-compute** MSA using new tree
- Refine the alignment by iteratively **partitioning the sequence into two groups** and merging two MSA groups



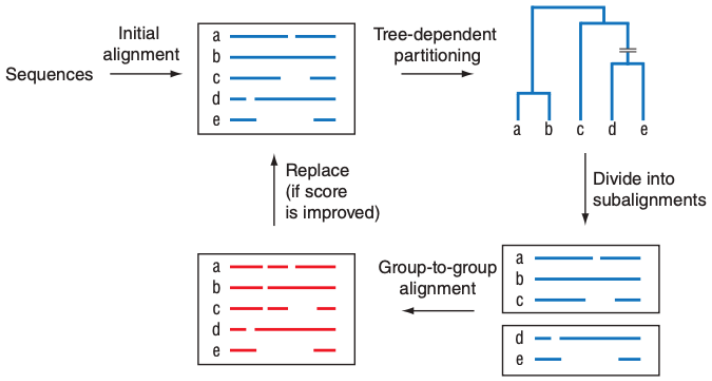
MUSCLE

- **Where the speed-up comes from:**
  - Finding **all short hits is fast** because we can use methods like hashing
  - Only ***n-1*** alignments for a tree
- **Refining multiple sequence alignment** (One Method):
  - Choose a **random** sentence
  - **Remove** from the alignment (***n-1*** left)
  - Align the removed sequence to the ***n-1*** remaining
  - **Repeat**
- Alternatively, **subdivided into two subsets** (MUSCLE)





MUSCLE



Biological Rationale

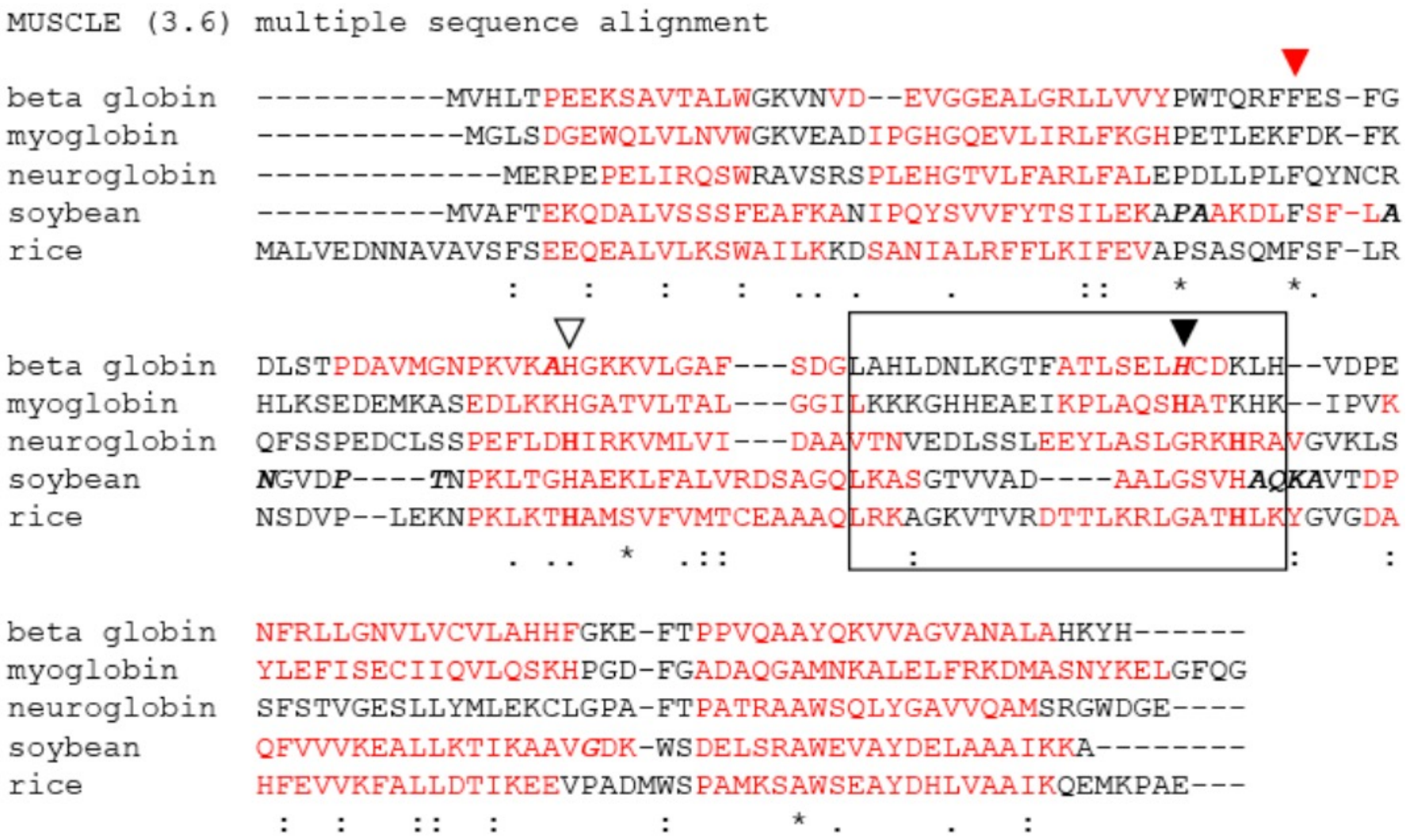
- **Evolutionary Relationships**:
  - Sequences within a subgroup are typically more closely related and easier to align accurately. Between subgroups, however, evolutionary divergence can cause alignment errors, which MUSCLE seeks to correct.
- **Conserved Regions**:
  - Realignment between subgroups helps detect conserved regions across the entire dataset, especially in divergent sequences.
- **Improved Accuracy for Distant Homologs**:
  - By iteratively refining subgroup alignments, MUSCLE enhances its ability to align distant homologs, which often require careful optimization at their boundaries.

Why Realign Between Subgroups?

1. **Boundary Refinement**:
   - Misalignments often occur at the boundaries where two subgroups meet. By focusing on these interfaces, MUSCLE can correct alignment errors that arise during the progressive alignment stages.
   - This approach minimizes disruptions to already well-aligned regions within subgroups.
2. **Improved Global Consistency**:
   - Realigning between subgroups ensures that the alignment is globally consistent, maintaining coherence across the entire dataset rather than optimizing only local regions.
3. **Preservation of Subgroup Alignments**:
   - Realigning among sequences within a subgroup can disrupt well-aligned regions. By keeping subgroup alignments intact, MUSCLE reduces the risk of introducing new errors in regions that are already accurate.
4. **Tree-Guided Optimization**:
   - The guide tree reflects the evolutionary relationships among sequences. By realigning subgroups based on these relationships, MUSCLE leverages evolutionary signals to improve the overall alignment quality.
5. **Computational Efficiency**:
   - Realigning between subgroups is computationally less expensive than realigning all sequences, especially for large datasets. It focuses resources on the most problematic regions (interfaces between subgroups) where improvements are most likely.

Progressive

MUSCLE



Multiple Sequence Alignment: properties

- not necessarily one "correct" alignment of a protein family
- protein sequences evolve...
- ...the corresponding three-dimensional structures of proteins also evolve
- may be impossible to identify amino acid residues that align properly (structurally) throughout a multiple sequence alignment
- for two proteins sharing 30% amino acid identity, about 50% of the individual amino acids are superposable in the two structures

Scenario	Clustal W	T-Coffee	MUSCLE
Small Datasets (10–50 sequences)	Good	Excellent	Good
Large Datasets (100+ sequences)	Poor scalability	Moderate	Excellent
Divergent Sequences	Moderate	Excellent	Good
Speed Priority	Moderate	Slow	Very Fast
Structural Data (e.g., proteins)	Not supported	Excellent (Expresso)	Not supported
Phylogenetics	Good	Excellent	Good
RNA Secondary Structure Analysis	Not suitable	Excellent (R-Coffee)	Not suitable
High-Throughput Applications	Poor scalability	Moderate	Excellent

Content

- 神经网络基本结构（MLP）
- 其他常见神经网络（RNN, CNN, Diffusion）
- 大语言模型（LLM）

字典

神经网络　　MLP
（多层感知机）
Input 输入
Output-prediction(输出-预测)
FC　full connection（全连接）
Activation function 激活函数

Robust 鲁棒性
泛化能力
过拟合

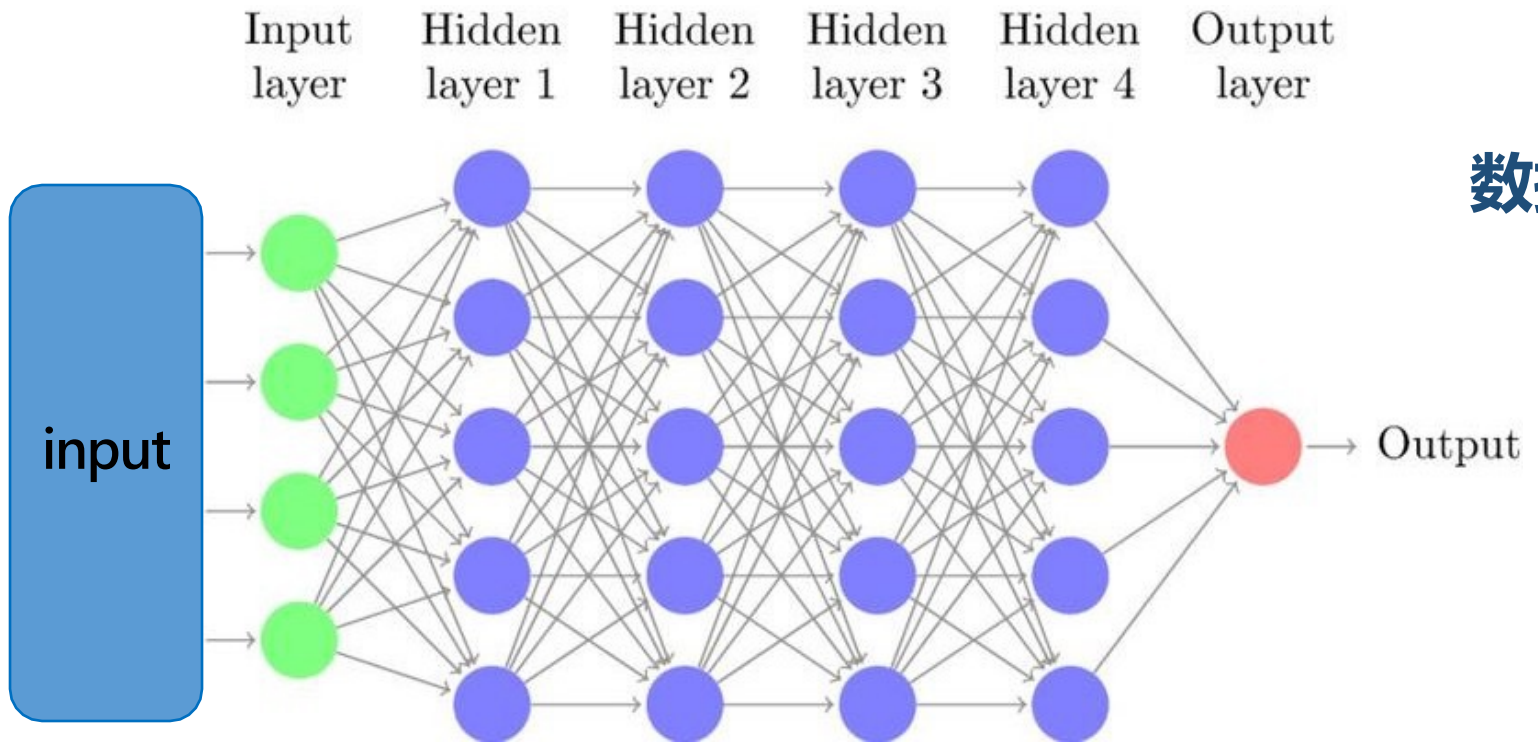
训练集
数据集
Loss function损失函数
梯度下降
Backpropagation 反向传播
MSE 均方差
Forward 前向传播
Optimizer 优化器

CNN卷积神经网络
LSTM长短期记忆神经网络
Diffusion 扩散模型
RNN 循环神经网络

什么是人工智能?

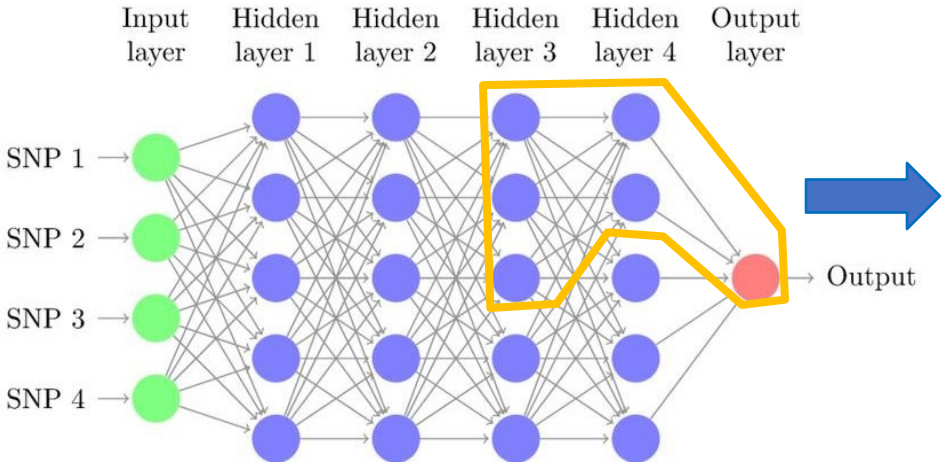
机器学习？　　深度学习？　　神经网络？

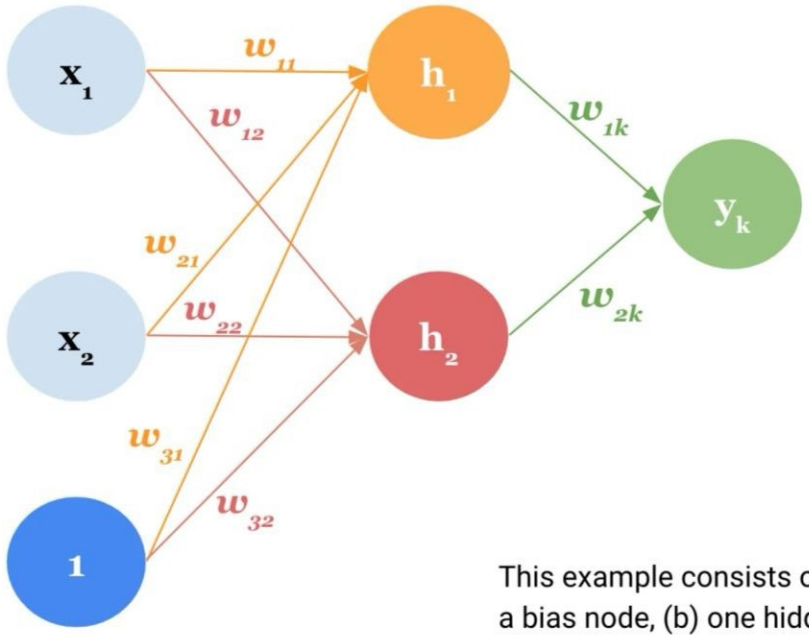
Multilayer perceptron（MLP）　多**层**感知器



数据依赖的表征学习算法

Multilayer perceptron（MLP） 多层感知器





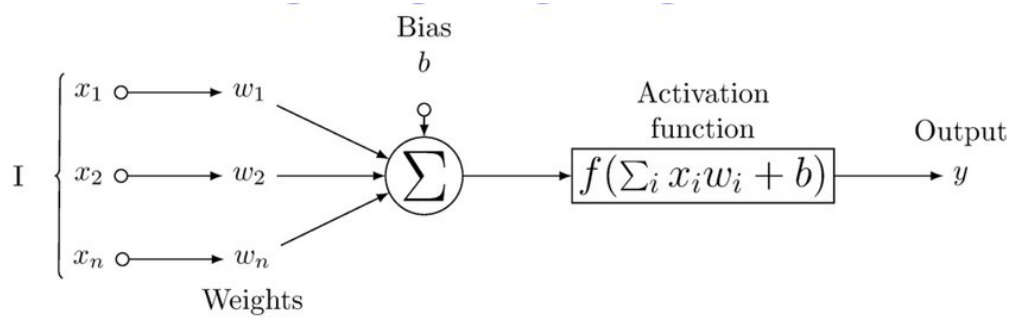
x_1 , x_2 : *input data features*

w_{ij} : *weights of the network*

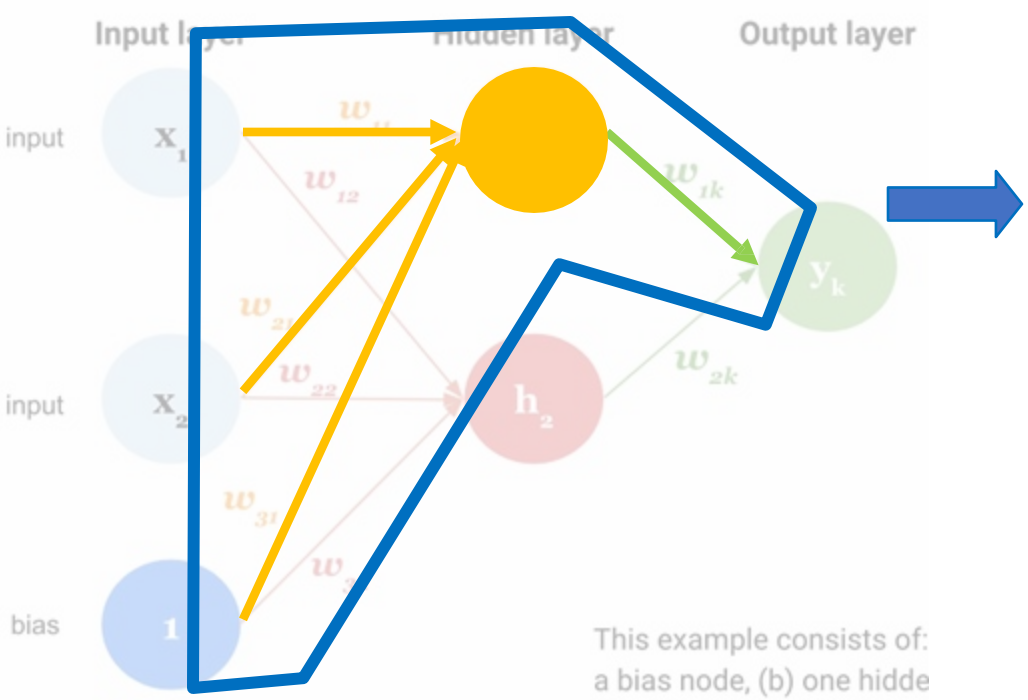
h_1 , h_2 : *nodes in the hidden layer*

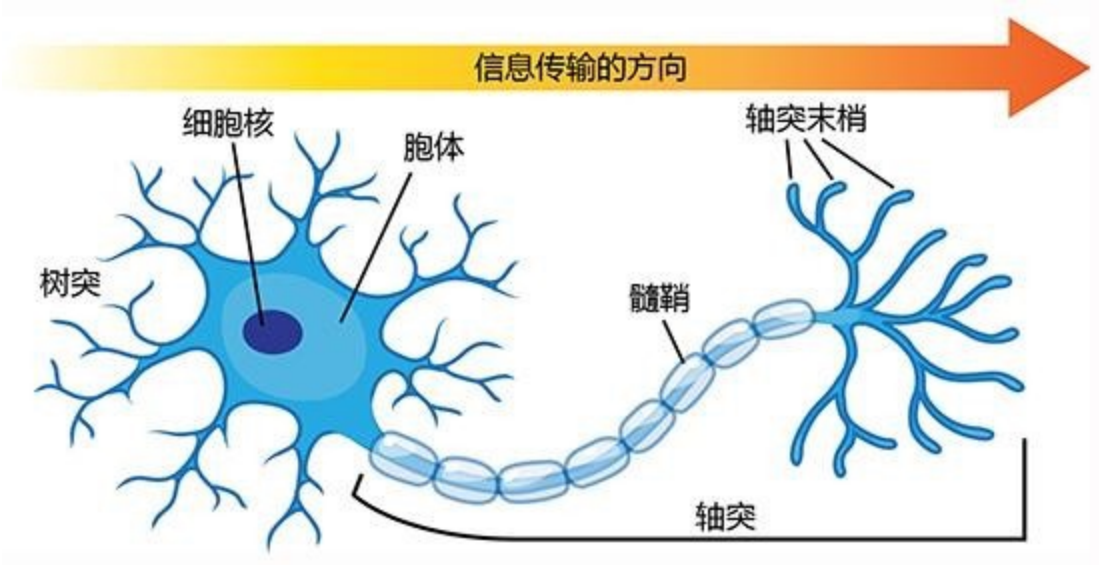
y_k : *output variable*

This example consists of: (a) an input layer with two input nodes and a bias node, (b) one hidden layer with two neurons, and (c) an output layer with one neuron

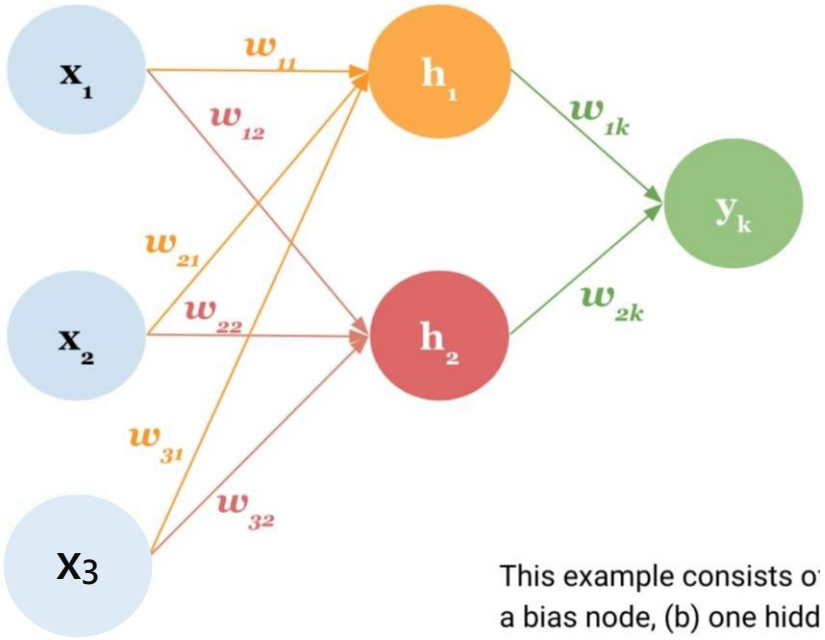


Multilayer perceptron（MLP） 多层感知器





神经网络基本结构：矩阵



This example consists o
a bias node, (b) one hidd

第一层

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \end{bmatrix}$$

第二层

$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} \times \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} = \begin{bmatrix} y_k \end{bmatrix}$$

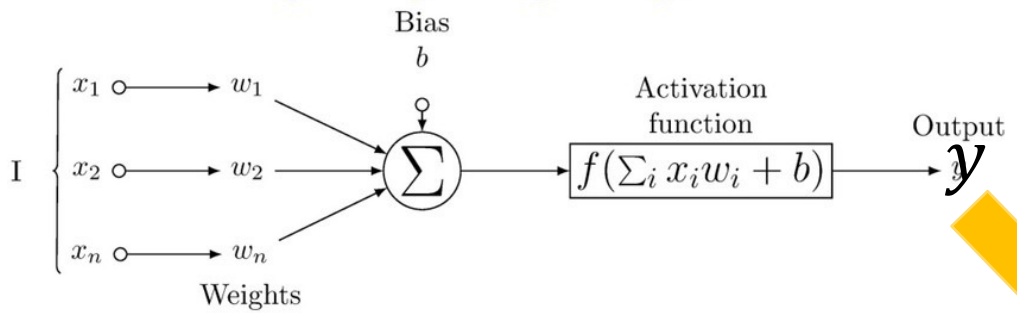
$$y_k = \left(\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \right) \times \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix}$$

$$W_1 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$$

$$W_2 = \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{x} W_1 W_2$$

神经网络基本结构：激活函数



线性

本质为**线性变换**

加入**非线性**

$$f(y) = \sigma(y) = \frac{1}{1+e^{-y}}$$

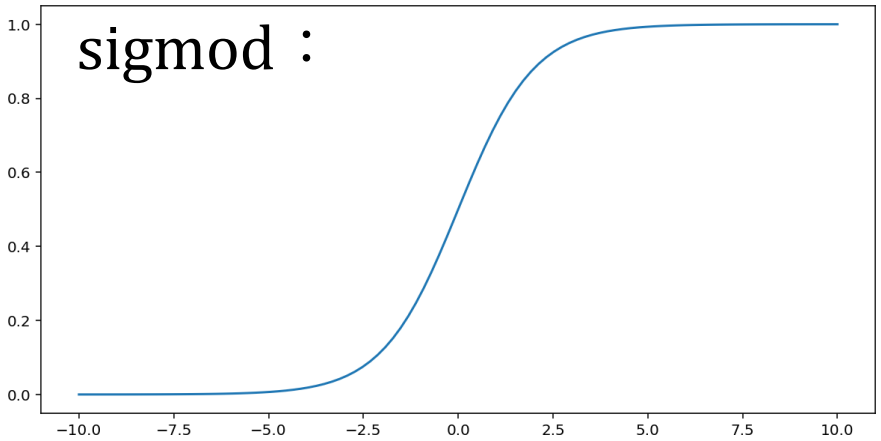
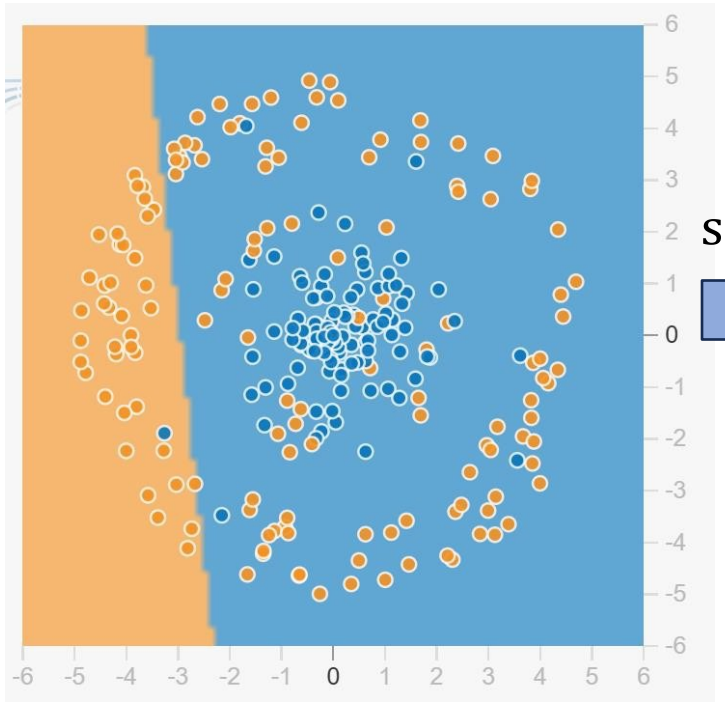
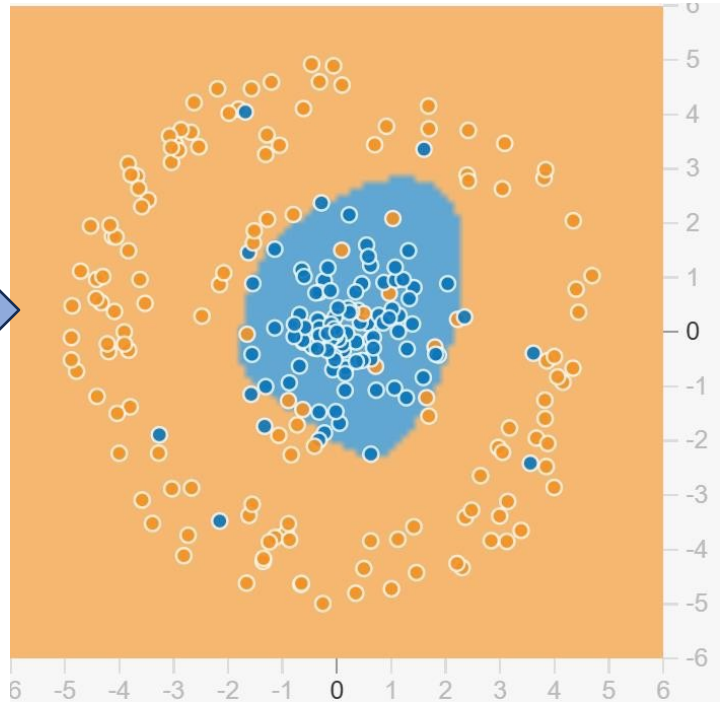
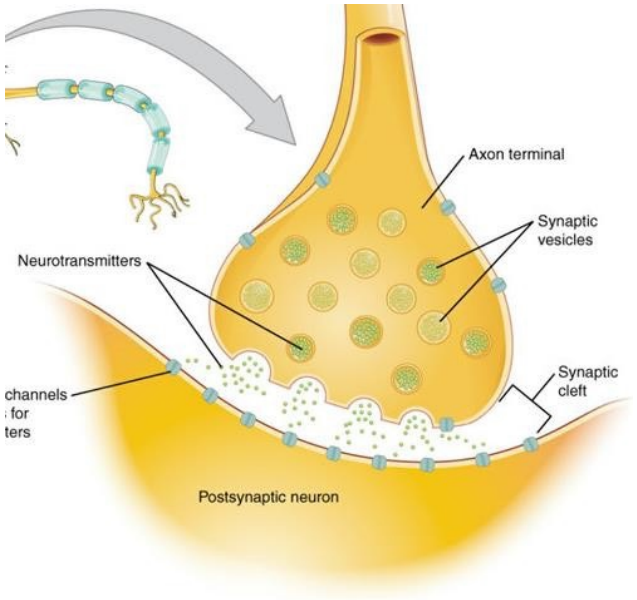


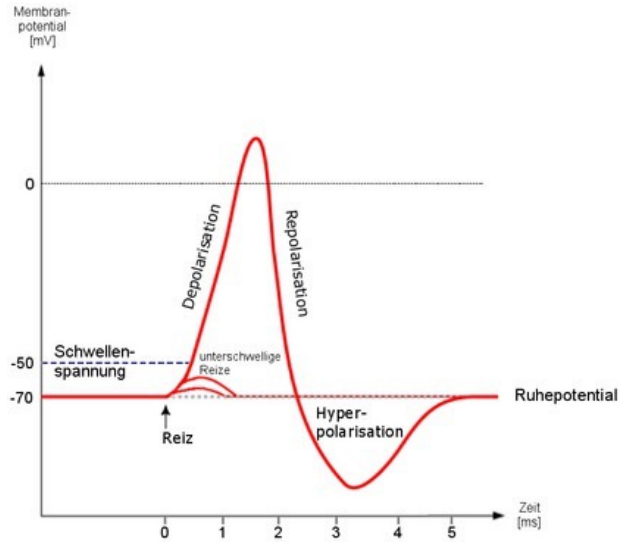
图 4. S 型函数的图表

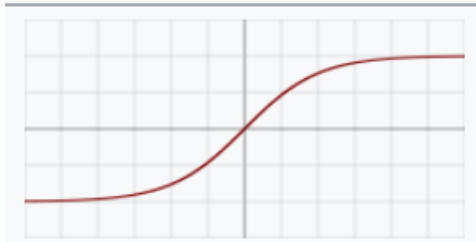


sigmod



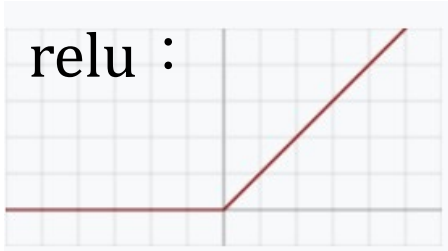






$$f(x) = \tanh(x) = \frac{\left(e^{x} - e^{-x}\right)}{\left(e^{x} + e^{-x}\right)}$$

relu :

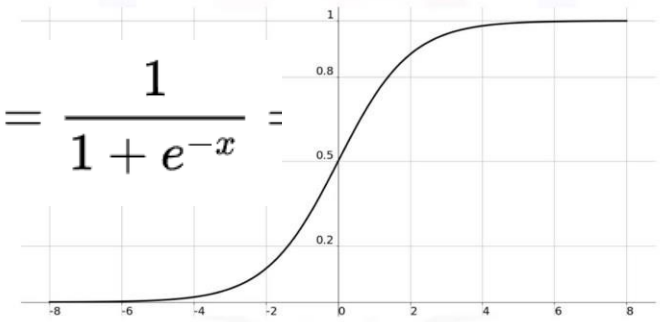


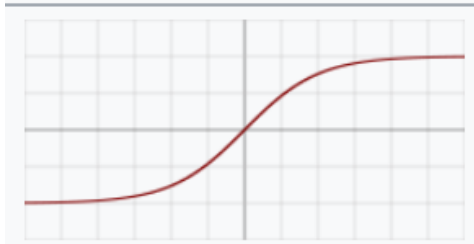
$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

神经网络基本结构：激活函数

Sigmoid Function

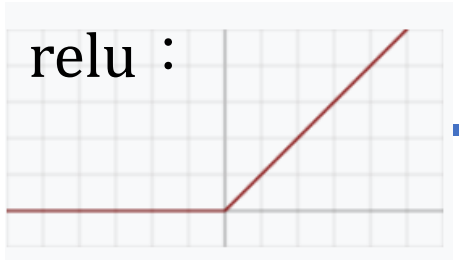
$$\sigma(x) = \frac{1}{1+e^{-x}} =$$

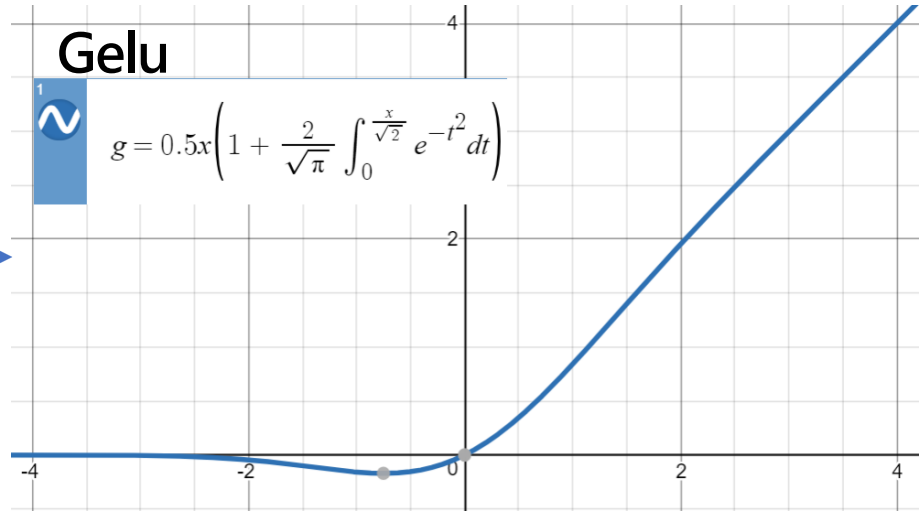


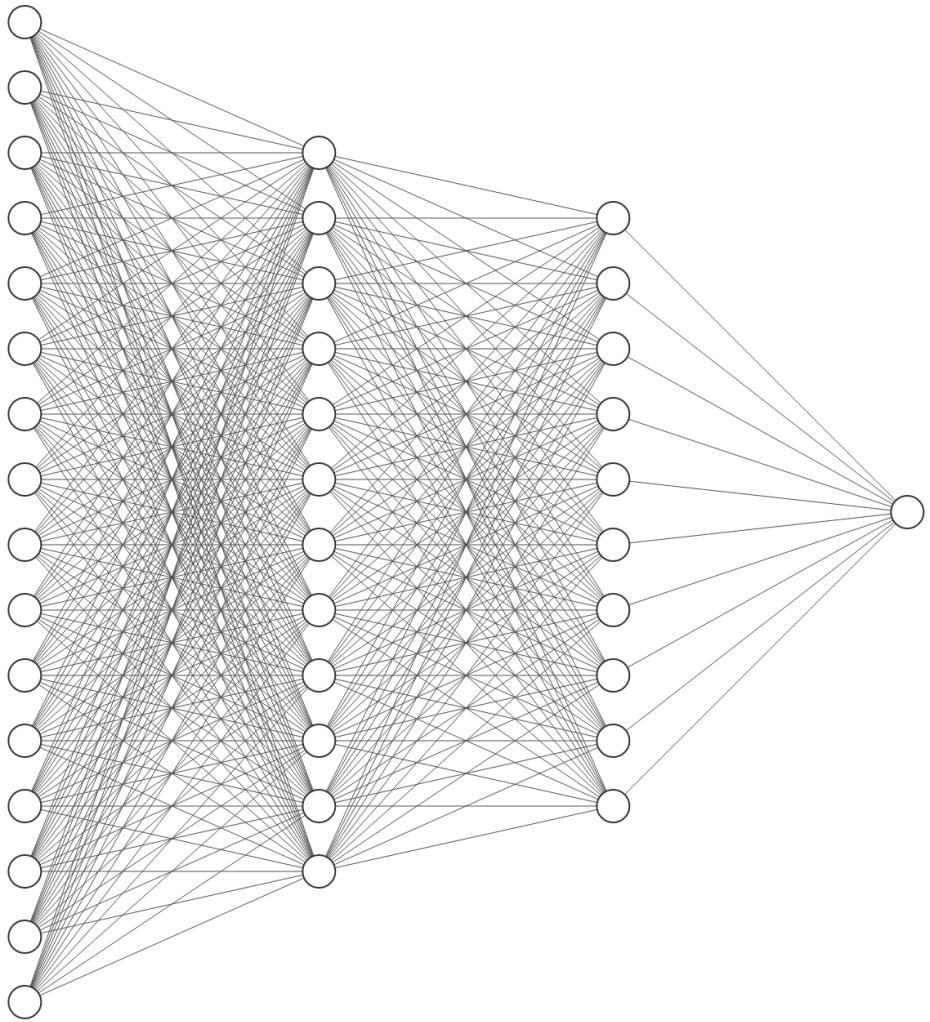


$$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

relu：







Input Layer $\in \mathbb{R}^{16}$ Hidden Layer $\in \mathbb{R}^{12}$ Hidden Layer $\in \mathbb{R}^{10}$ Output Layer $\in \mathbb{R}^{1}$

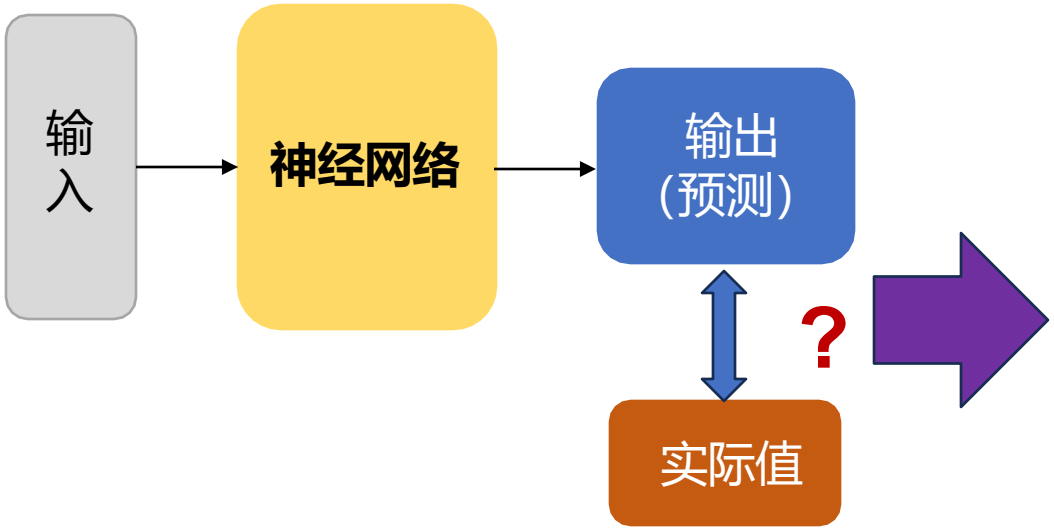
如何训练函数?

1. **设定一个目标** 🎯

2. **狠狠惩罚** 😡

神经网络基本结构：训练&学习

设定一个目标🎯： **损失函数（Loss）**



$$\text{MSE：Loss}=\frac{1}{N}\sum(\text{实际值}-\text{预测值})^2$$

F(预测值，实际值)　**Loss Function**

衡量两者差异

最小化误差 ⟷ **最小化loss**

神经网络基本结构：训练&学习

常见损失函数（Loss）

1. 回归任务(regression)

预测连续数值（例如：房价、股票、气温）。

$$\text{MSE：Loss}=\frac{1}{N}\sum(\text{实际值}-\text{预测值})^2$$

2. 分类任务(Classification)

预测离散类别（例如：猫/狗，区分手写数字 0-9）。

- **二分类 (Binary Cross Entropy, BCE):** 常配合 **Sigmoid** 激活函数使用。

$$L=-[y\log(\hat{y})+(1-y)\log(1-\hat{y})]$$

- **多分类 (Categorical Cross Entropy):** 常配合 **Softmax** 激活函数使用。

$$L=-\sum_i y_i\log(\hat{y}_i)$$

神经网络基本结构：训练&学习

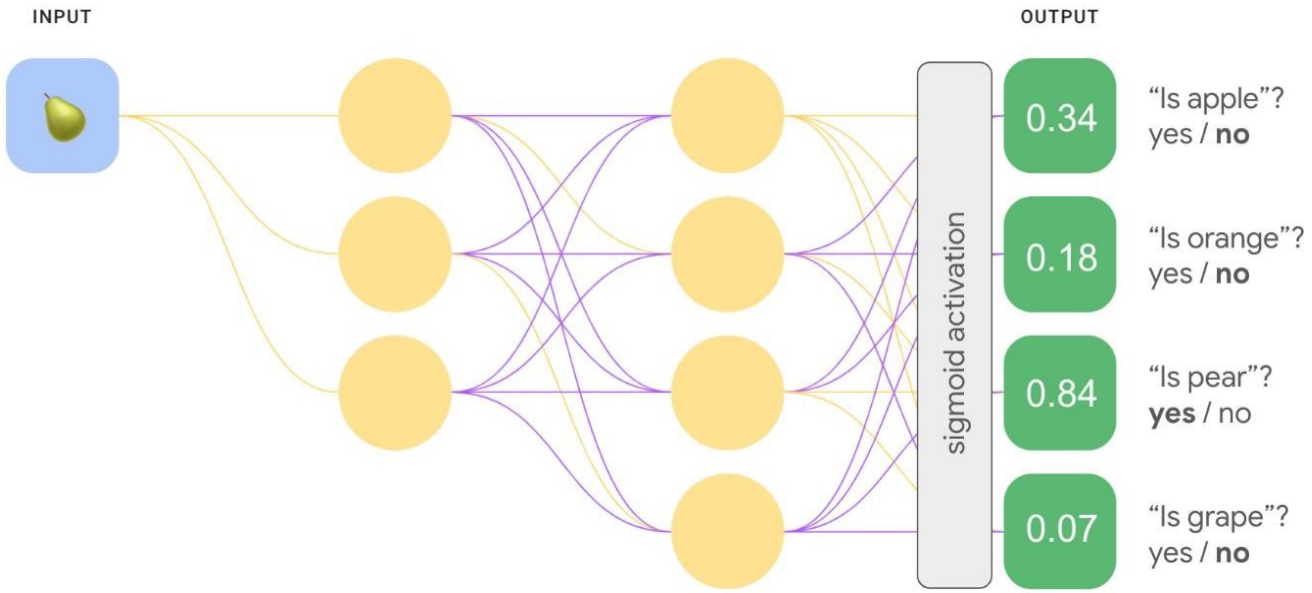
2. 分类任务(Classification)

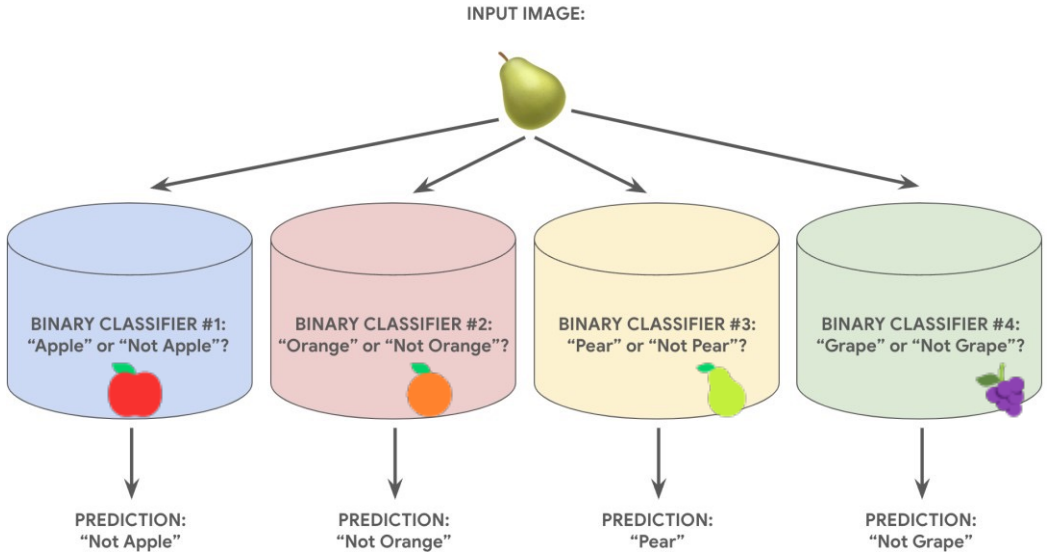
- **二分类 (Binary Cross Entropy, BCE):** 常配合 **Sigmoid** 激活函数使用。

$$L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

- **多分类 (Categorical Cross Entropy):** 常配合 **Softmax** 激活函数使用。

$$L = -\sum_{i} y_i \log(\hat{y}_i)$$





Softmax: $\sigma(z)_i = \dfrac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

- 每个输出值都在 $(0, 1)$ 之间。
- 所有输出值的和等于 1（$\sum \sigma(z)_i = 1$）。

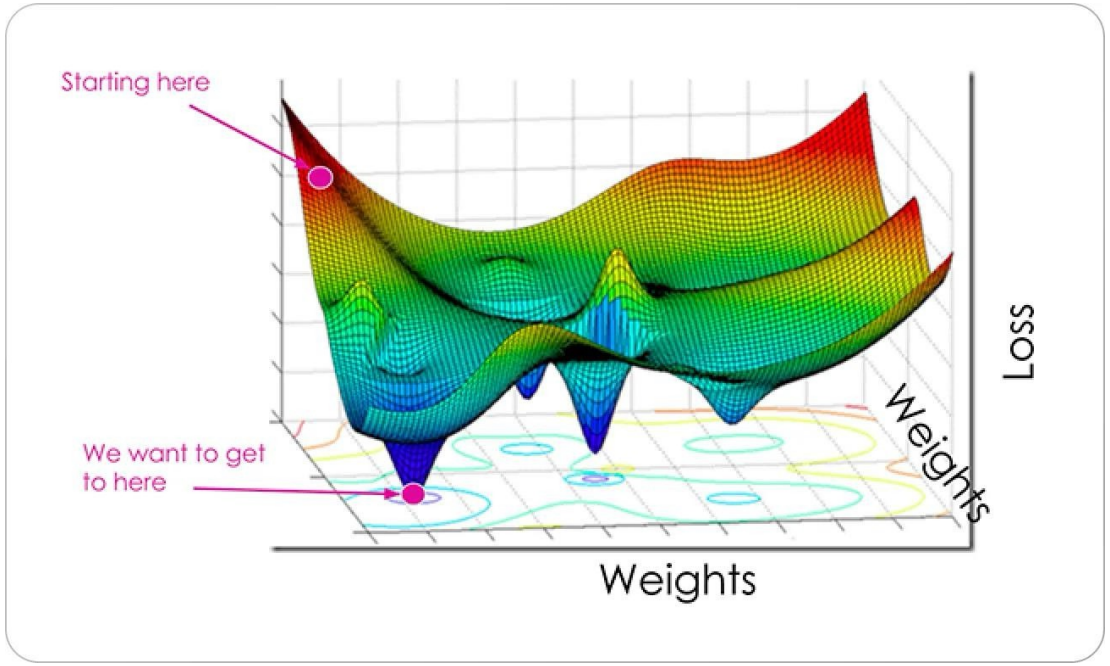
可视化演示

神经网络基本结构：训练&学习

训练模型　　**惩罚?**

最小化误差 ⟺ **最小化loss** ⟶ 求**极值**，找全局最优解

非线性的激活函数嵌套, 计算极其复杂, 实际中几乎不可行

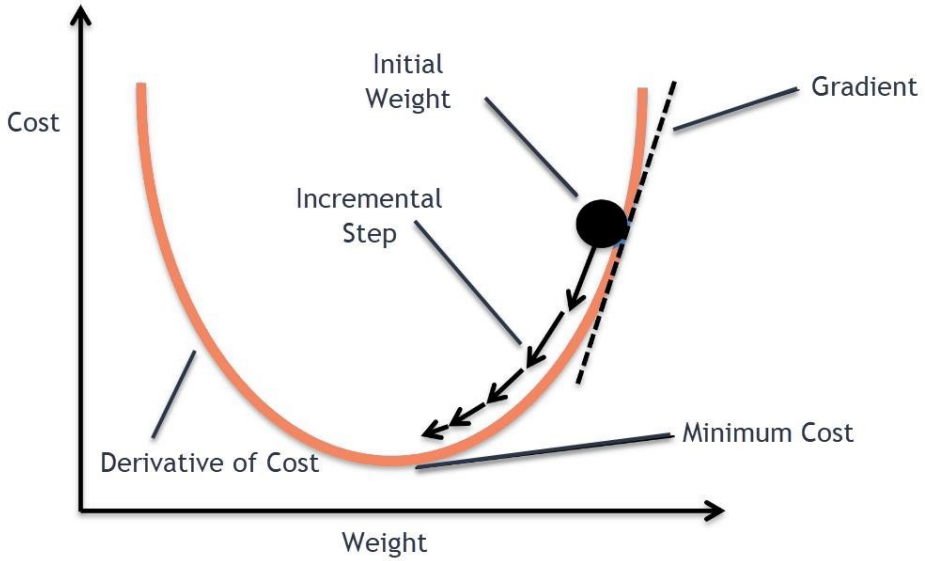


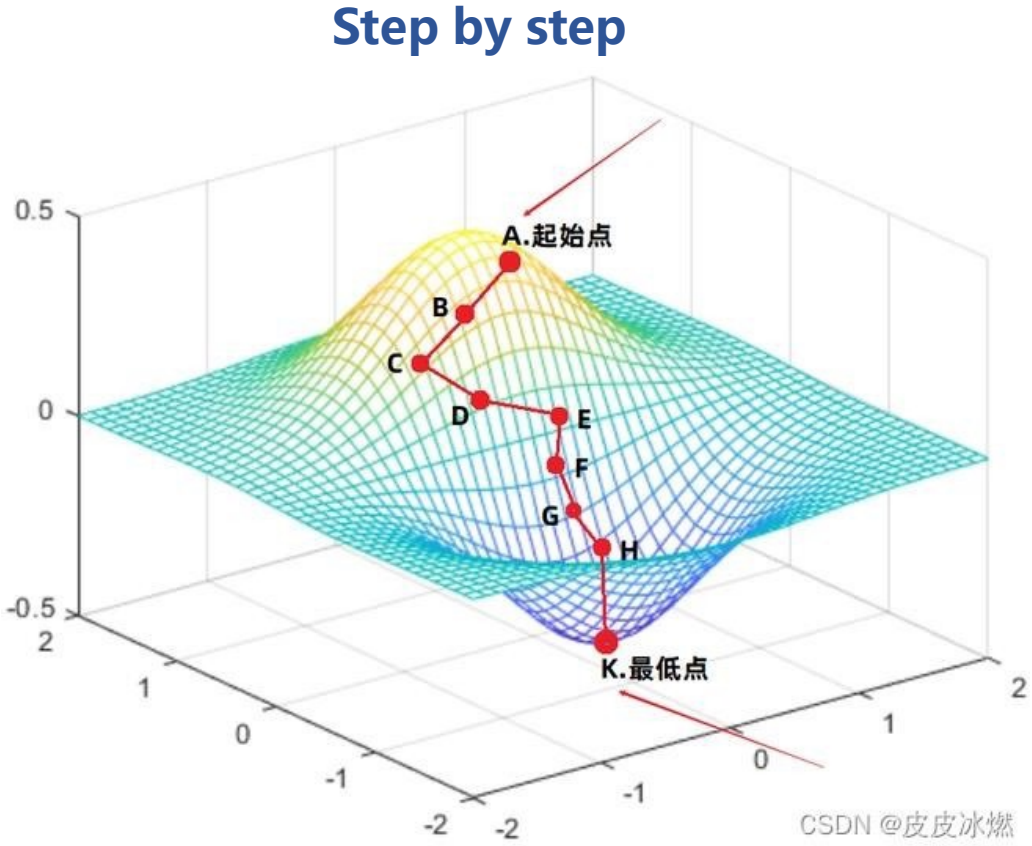
神经网络基本结构：训练&学习

导数(一元)和**偏导数(多元)**的定义:
函数沿坐标轴正方的函数的**变化率**

梯度:一个向量:
1. 函数沿此方向**增长率最大**
2. 大小为最大方向导数的值。

$$gradf(x_0, x_1, \ldots, x_n) = \left(\frac{\partial f}{\partial x_0}, \ldots, \frac{\partial f}{\partial x_j}, \ldots, \frac{\partial f}{\partial x_n}\right)$$





梯度下降与优化器(optimizer)

动量法

$$w_{t+1} = w_t - \alpha m_t$$

- m_t is the gradients at time t (vector)
- α is the learning rate (scale)
- W_t and is the weights at time t *(vector)*)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t}$$

RMSprop (Root Mean Square Propagation)

$$w_{t+1} = w_t - \frac{\alpha_t}{\sqrt{v_t + \epsilon}} \frac{\partial L}{\partial w_t}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{\partial L}{\partial w_t} \right)^2$$

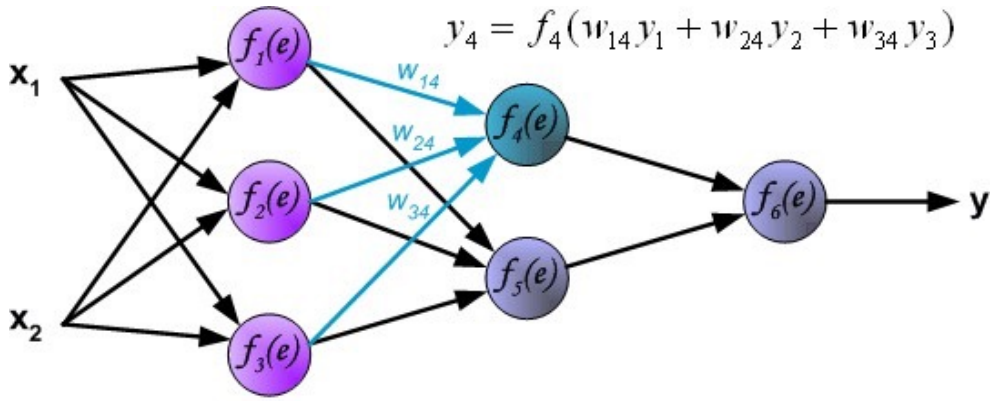
Adam

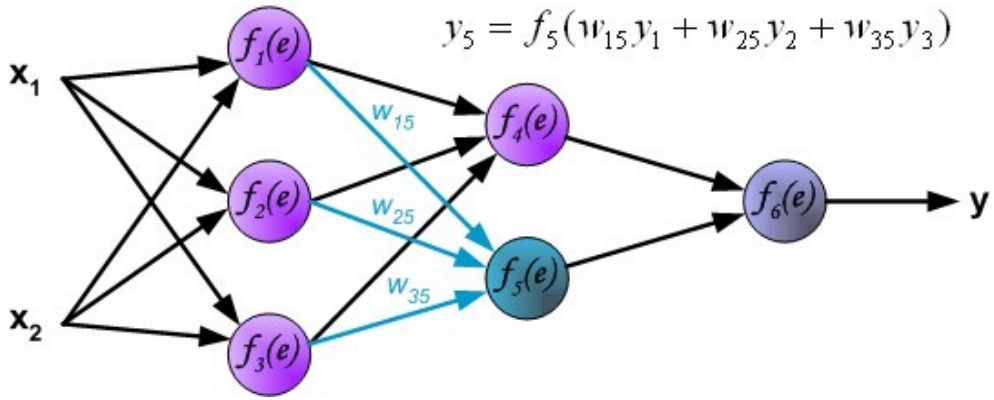
$$w_{t+1} = w_t - \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \alpha$$

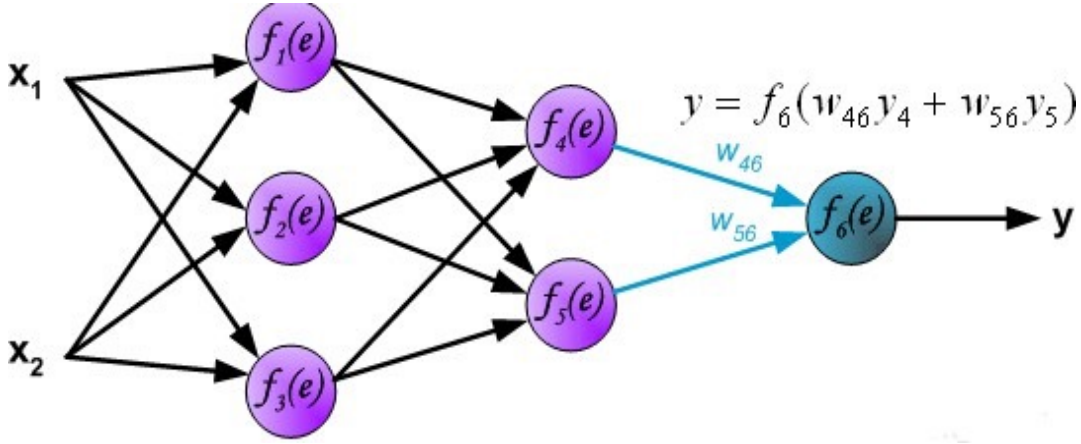
神经网络基本结构：训练&学习

反向传播 Backpropagation

损失函数（Loss）和 **梯度下降（GD）**

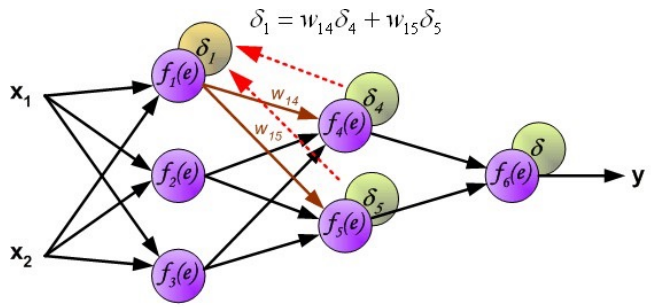


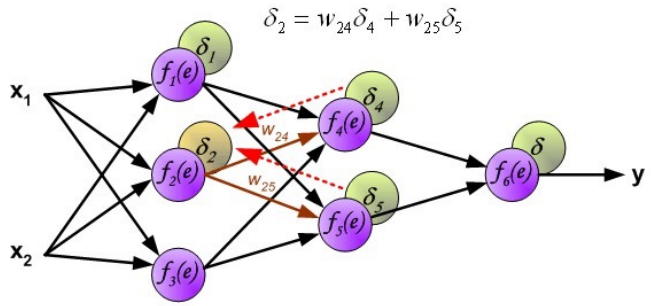


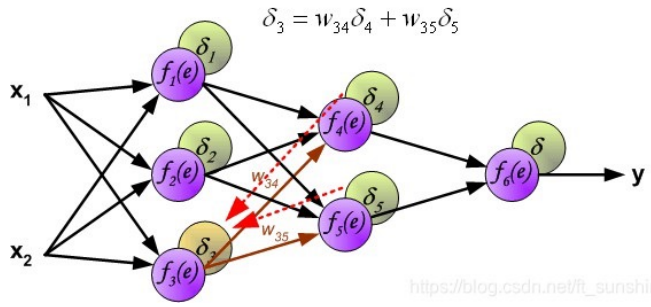


神经网络基本结构：训练&学习

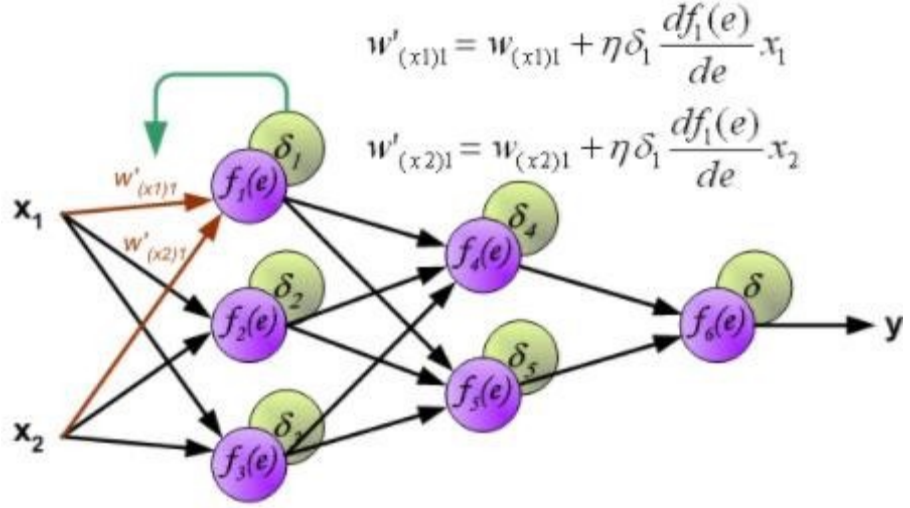
损失函数（Loss）和 梯度下降（GD）

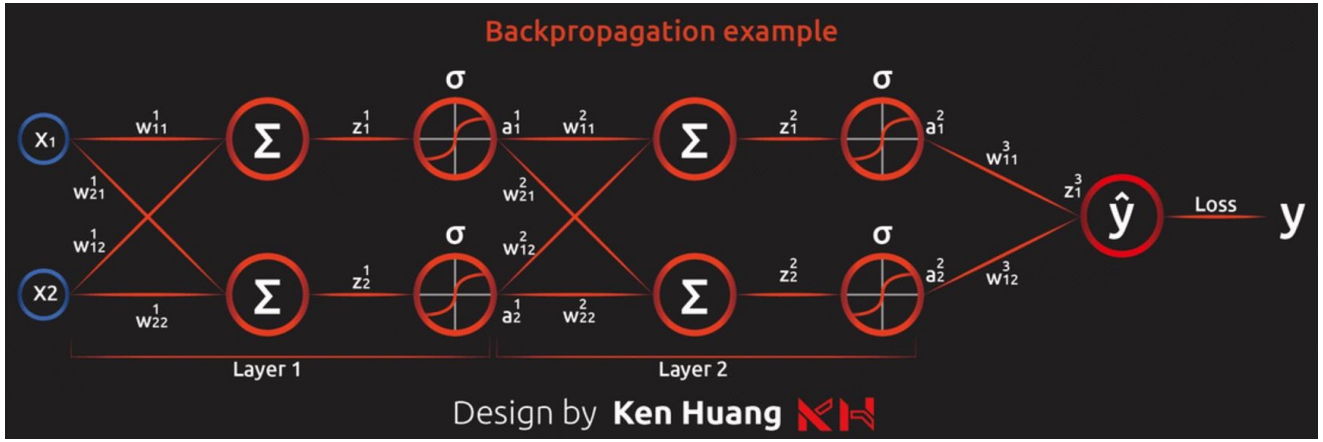




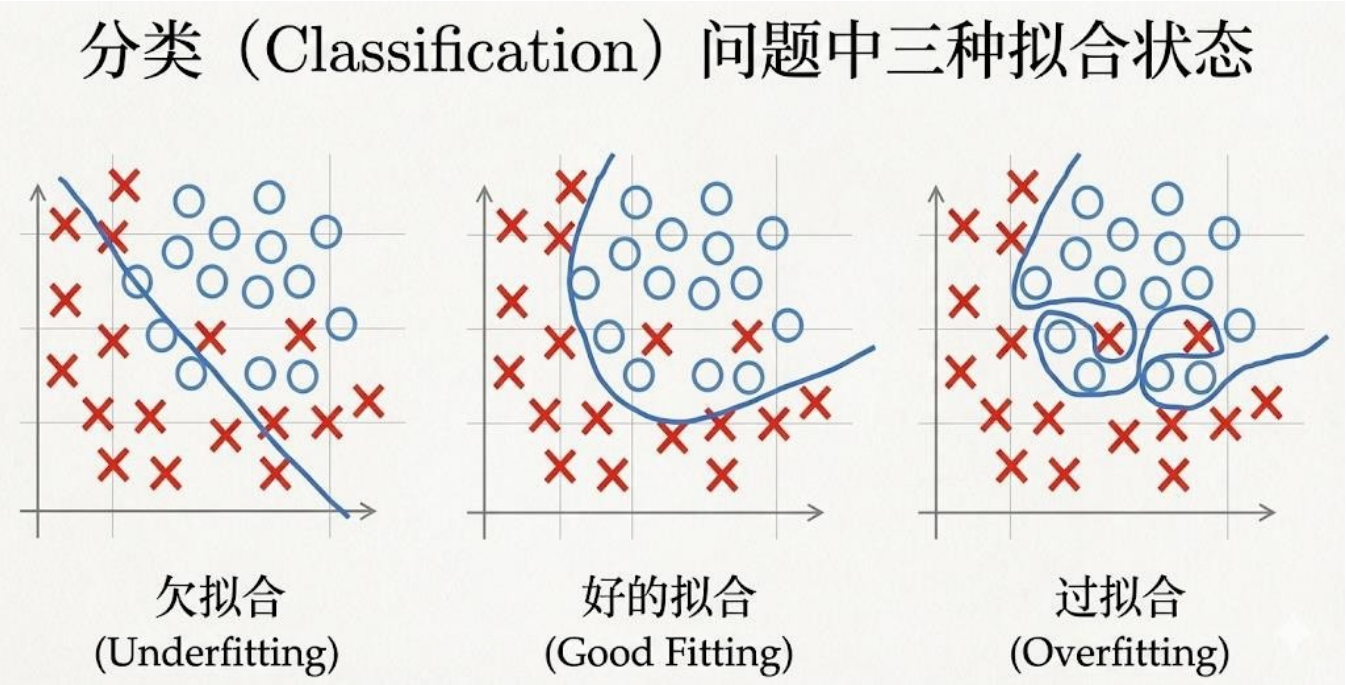


反向传播 Backpropagation





神经网络基本结构：过拟合 & 鲁棒性



- 数据集
  - 数据集过少
  - 噪声过大
  - 噪声过小
- 模型设计
  - 模型过于复杂
- 训练层面
  - 过度训练

广谱抗体 - 极高特异性抗体

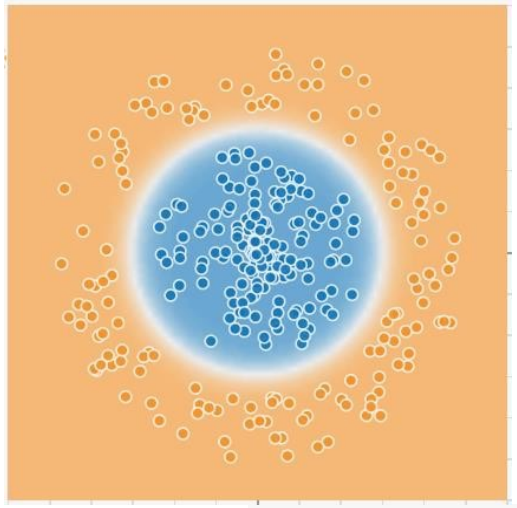
罕见病的患病基因确认

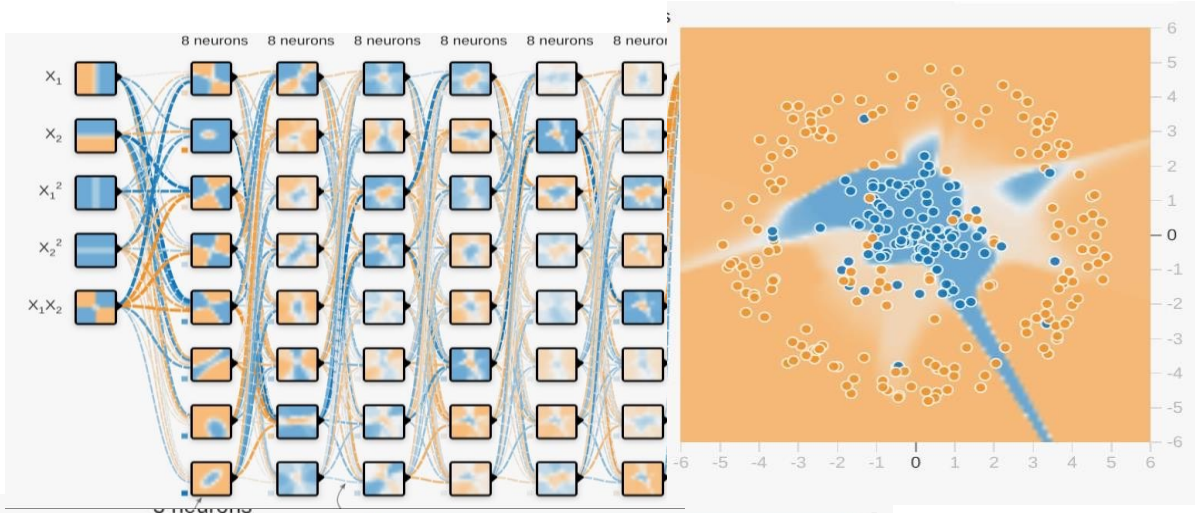
神经网络基本结构：过拟合-模型复杂度

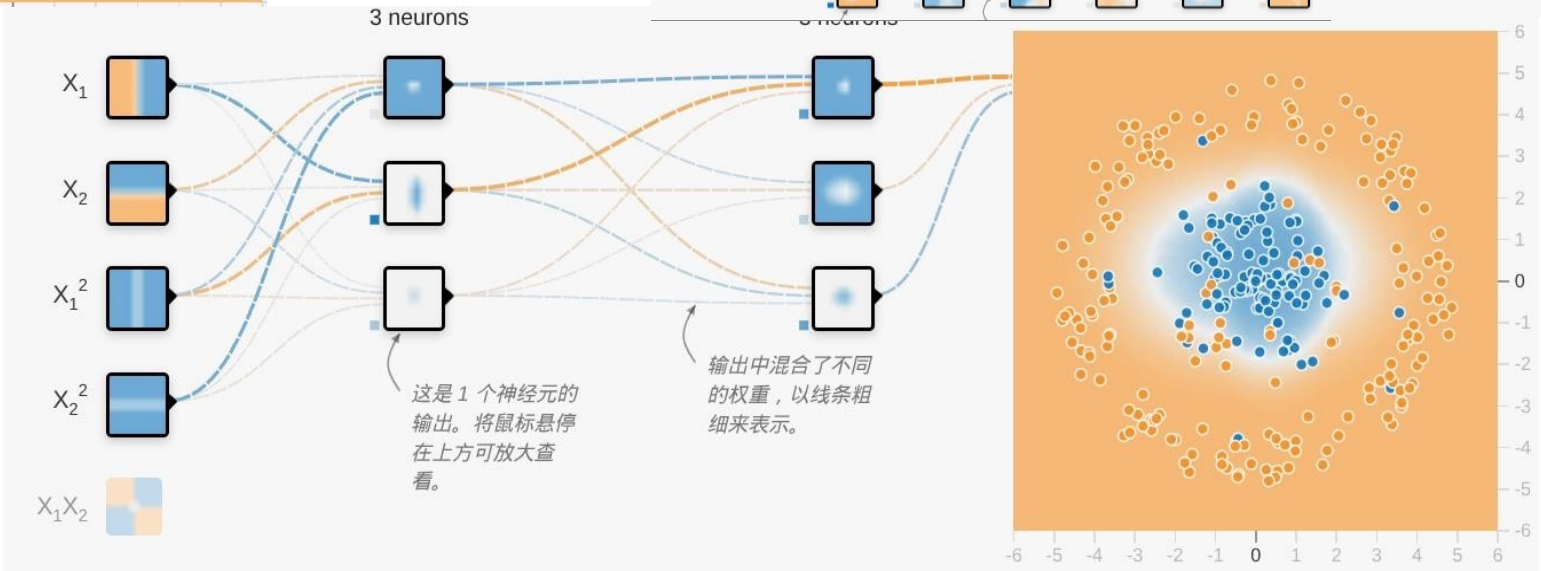
模型复杂度

可视化互动

低噪声数据集





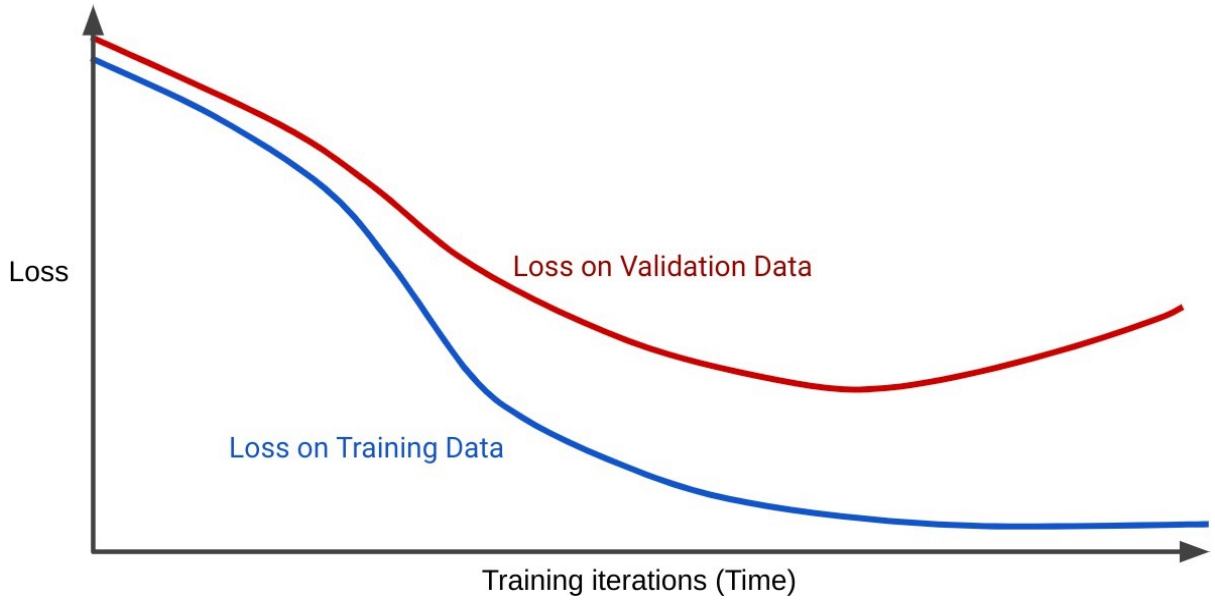


高低噪声数据集


数据
高质量，代表性，
清洗过的特征和标签
有潜力
但未优化
模型架构
网络结构，层数，
连接方式，容量设计
好的模型
高泛化能力，
准确，有效
受限容量
垃圾进，
垃圾出
训练方法 & 正则化
优化算法，损失函数，
Dropout，L1/L2，
Early Stopping

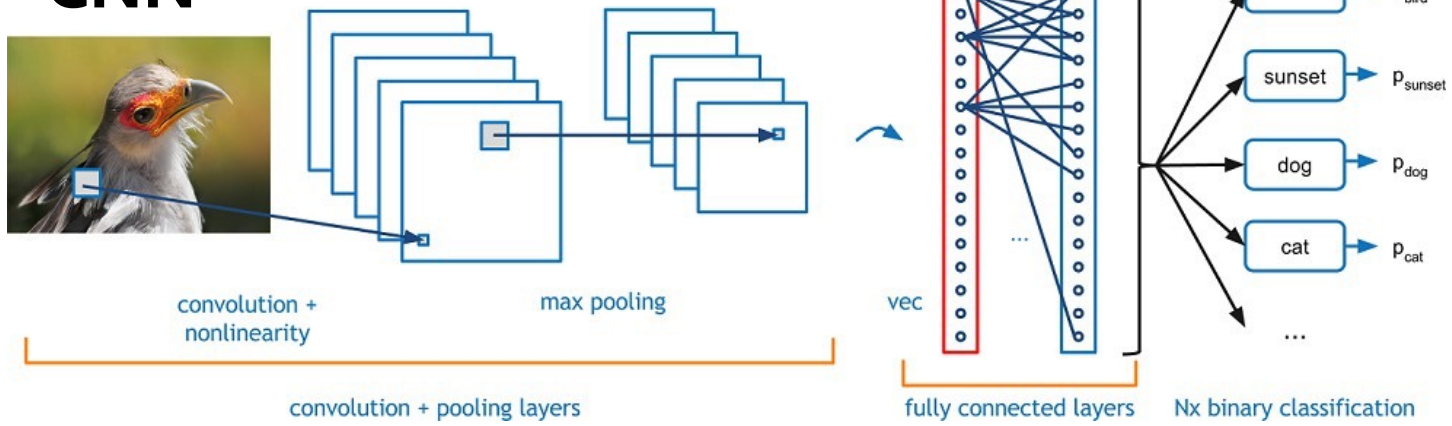
神经网络基本结构：过拟合 & 鲁棒性

- 模型设计
  - 适当**简化**模型结构
- 数据集
  - 增加数据集**规模**
  - 过滤噪声(噪声太大)
  - 适当添加噪声
- 训练层面
  - **Early Stopping**： 看到你刚开始背答案（验证集Loss上升），马上停止考试。
  - **Regularization** (正则化)： 给复杂的解法额外的惩罚（缩减模型参数大小）。
  - **Dropout**： 随机让你脑细胞休息，强迫你不能依赖某几个具体的神经元（死记硬背的路径）。
  - **Loss**: 优化损失函数设计

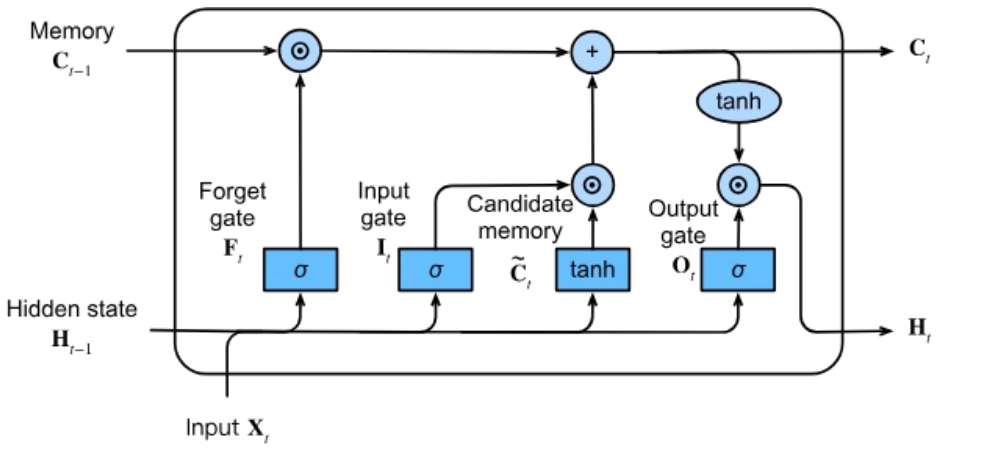


其他神经网络结构

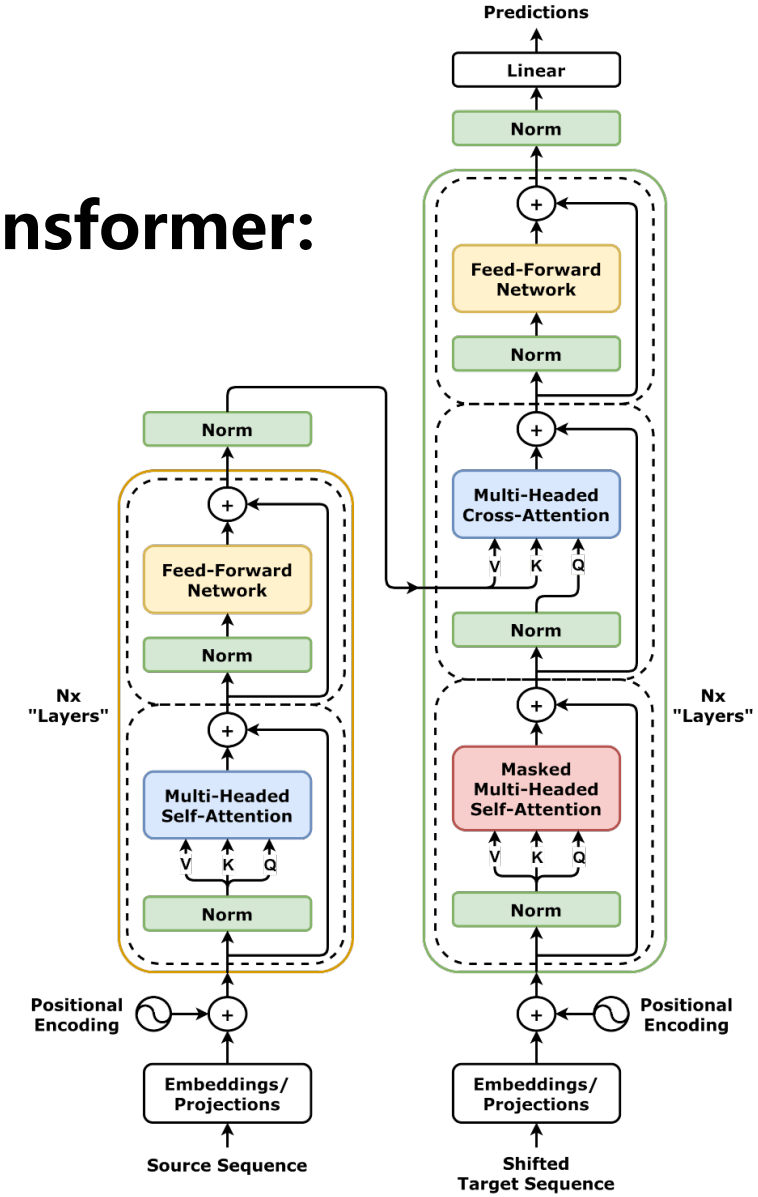
CNN



LSTM(RNN)



Transformer:

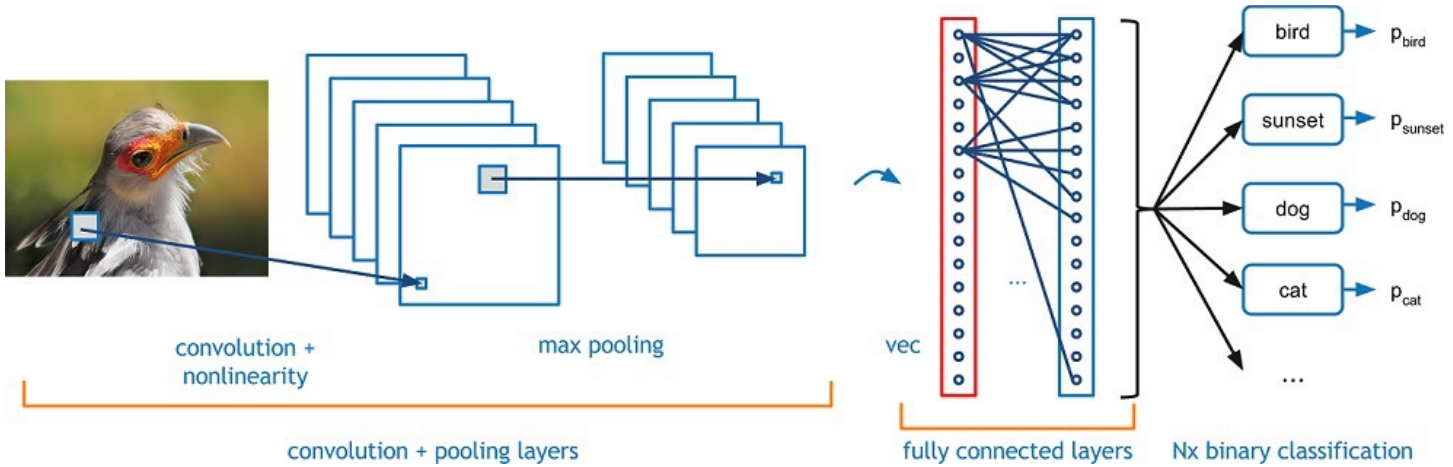


CNN

Convolution neural network(卷积神经网络):

图像识别这一块

卷积?

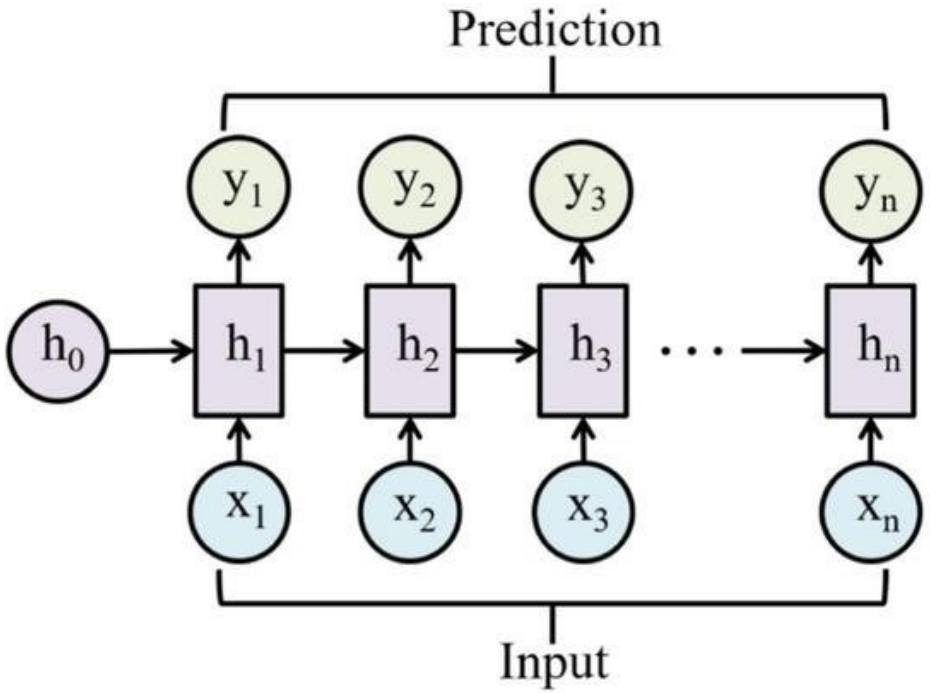


本质可以视为时空上更加**高效**，且连接更加**稀疏**的**MLP**

RNN

recurrent neural network　时序模型

加入**隐藏层**, 总结历史信息



$$h_t = Ah_{t-1} + Bx_t$$

$$y_t = C^\top h_t$$

发展:**LSTM**
视隐藏层(h)为短期记忆, 加入**长期记忆**

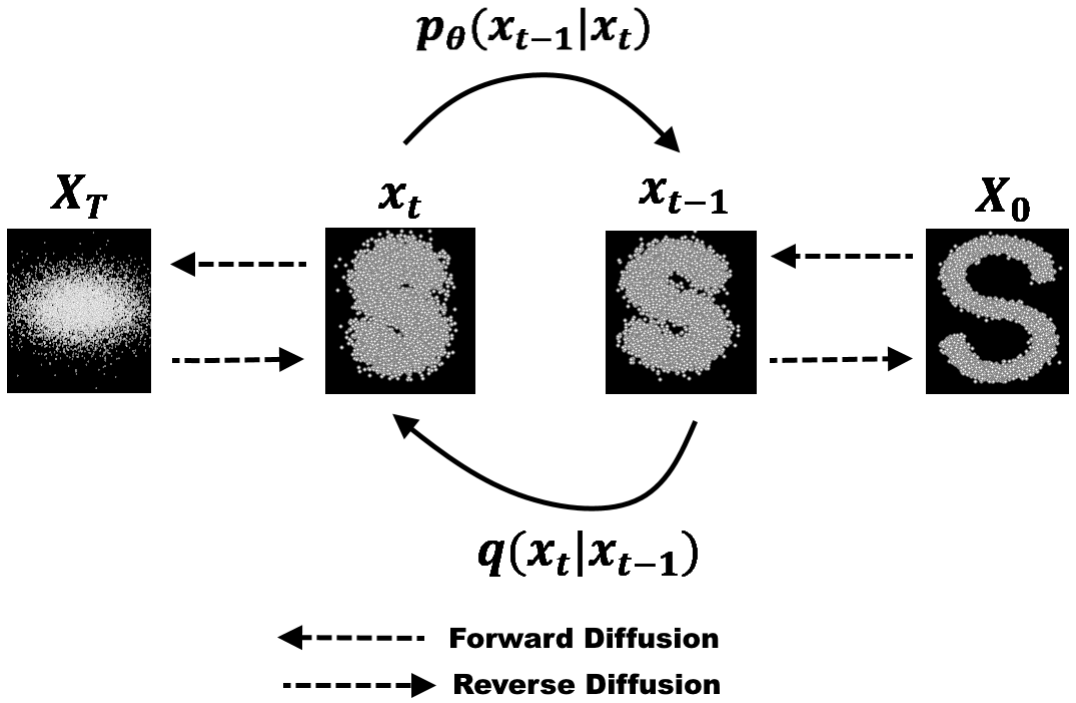
擅长处理时序信息, 模型简单

- 纳米孔基因测序的 Basecalling
- 蛋白质二级结构预测

缺点: 串联结构, 硬件效率低下

结局: transformer

Diffusion



Forward Process (Diffusion)

逐步加入噪声

Reverse Process (Denoising)

逐步消除噪声

Diffusion

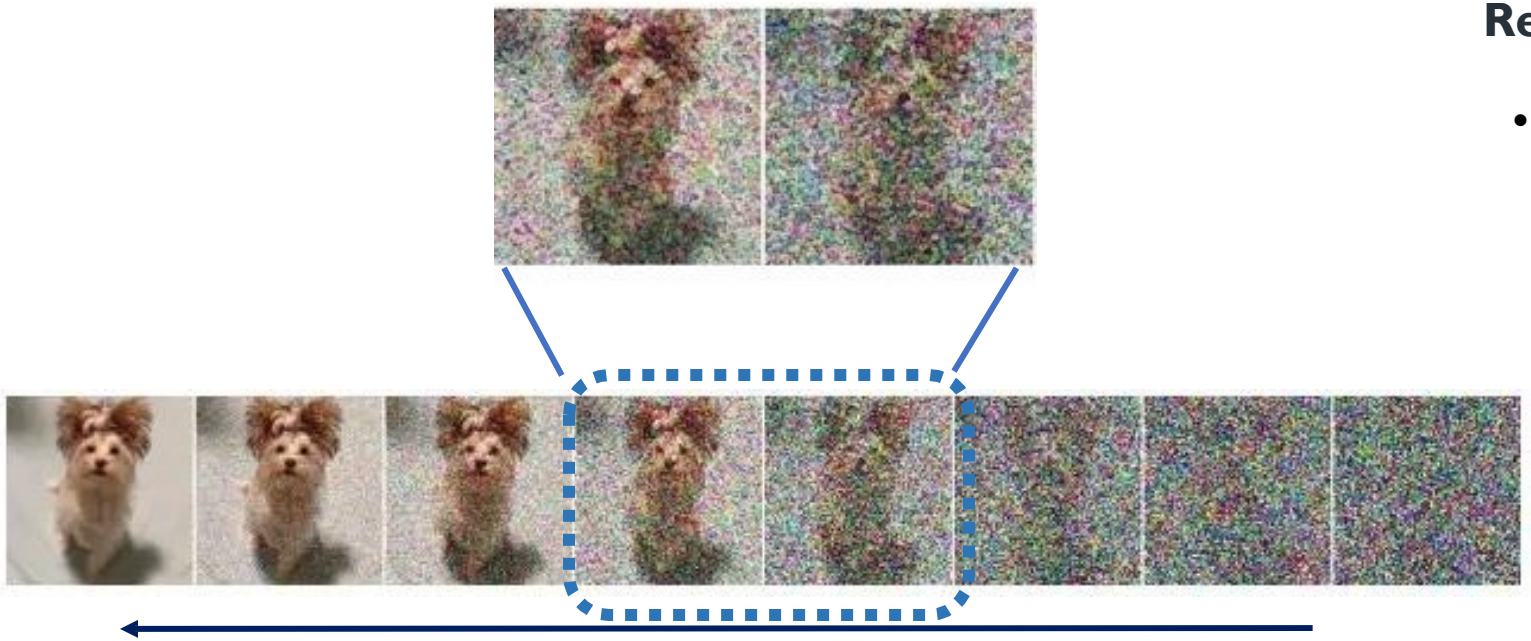
模型如何生成图像?

一步登天

Diffusion

Forward Process (Diffusion)

- 噪声梯度
- 马尔可夫链



Reverse Process (Denoising)

- 逐步去噪

大语言模型 LLM

Large language model

猜词游戏

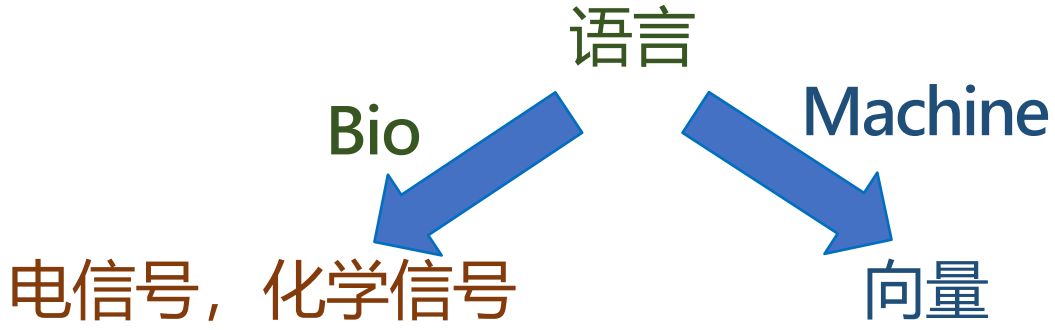
Q: LLM如何让计算机理解人类语言?

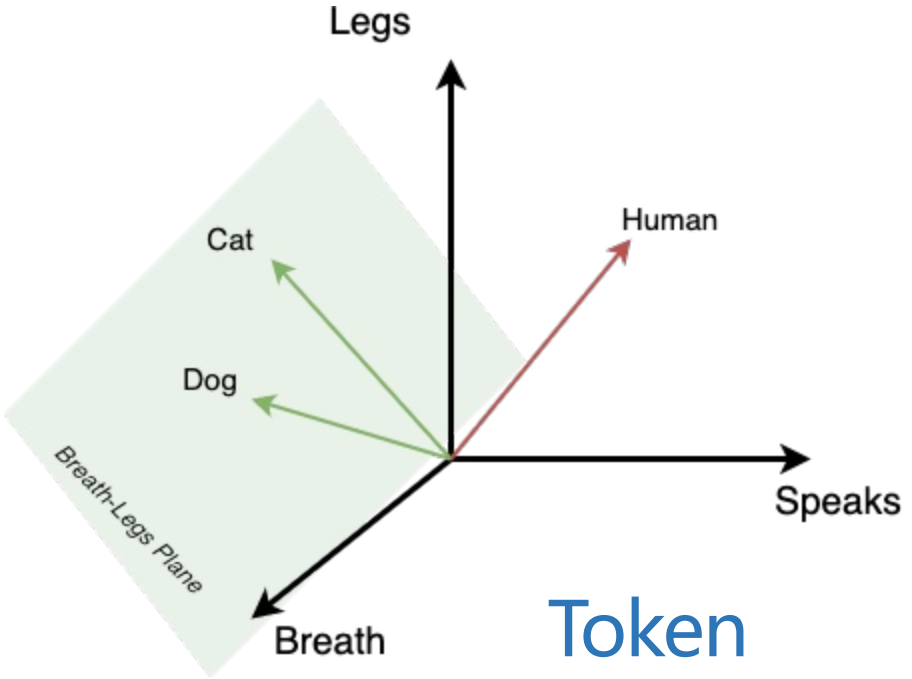
大语言模型 LLM

- LLM如何让计算机理解人类语言？
  - 计算机如何接受语言
  - 计算机如何处理（理解）语言
  - 计算机如何生成回答

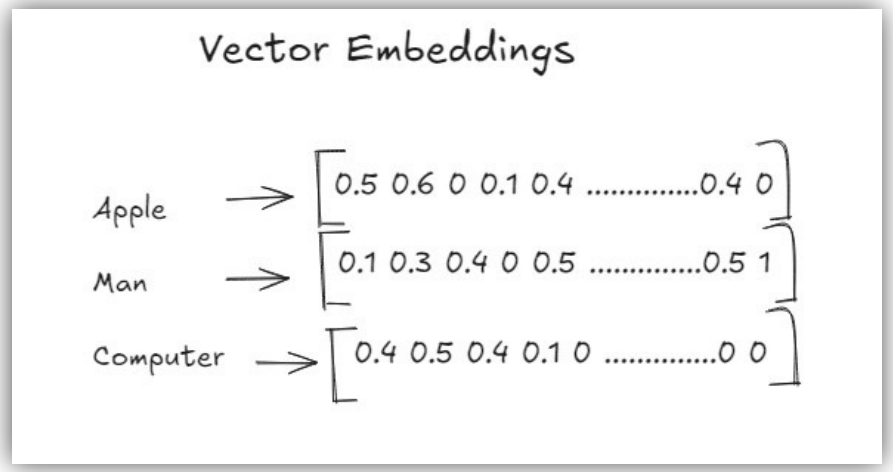
大语言模型 LLM： 接受语言

Embedding 词嵌入



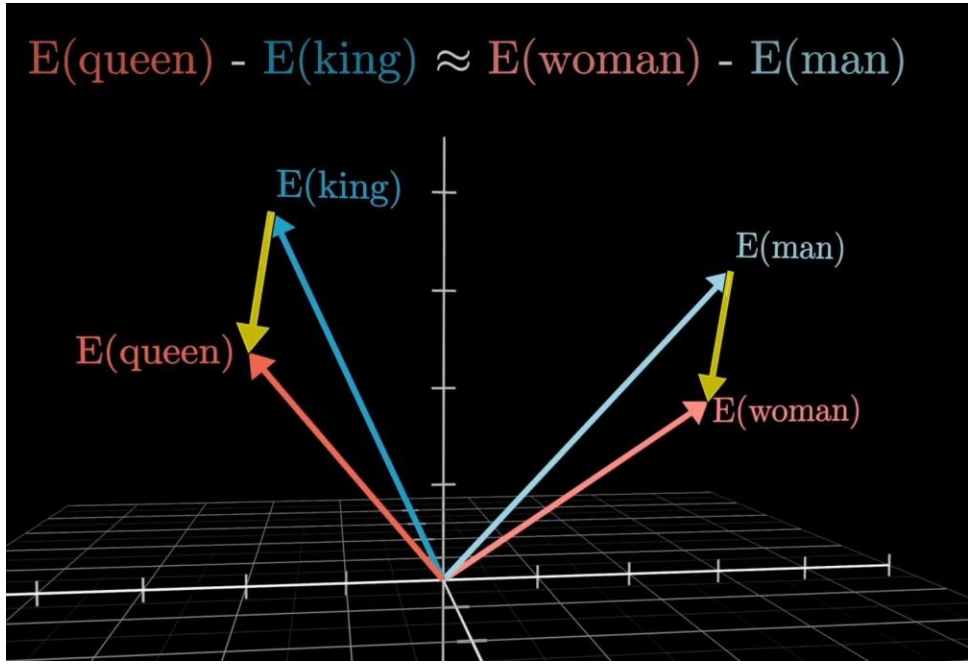


Token



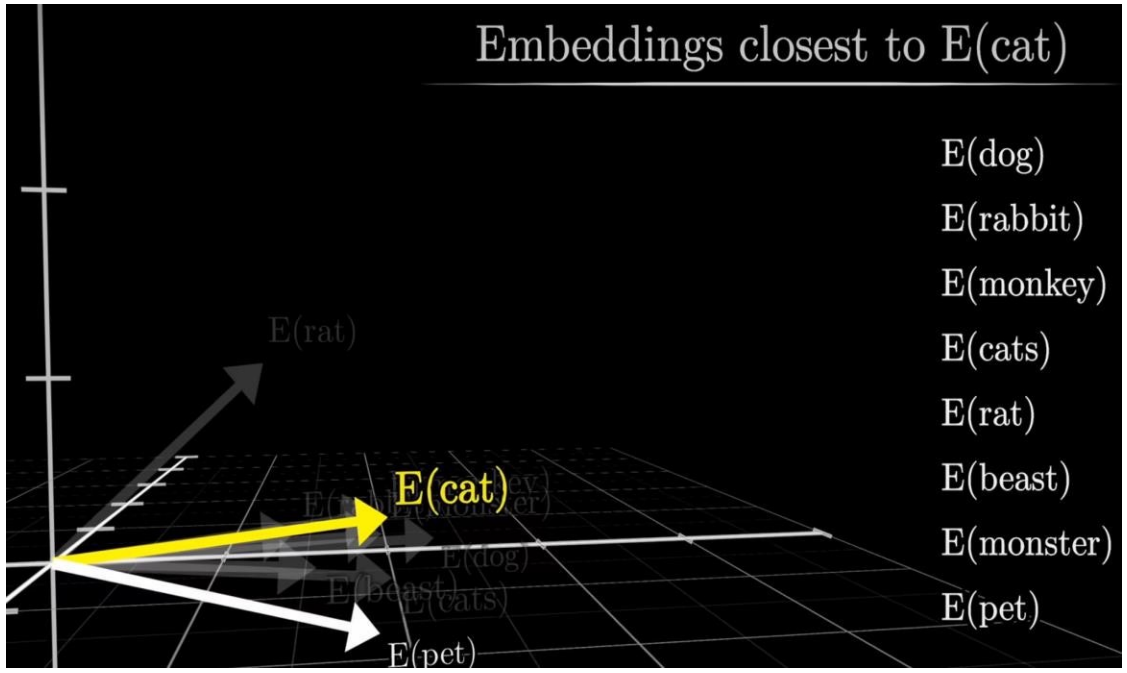
语义空间

词语的差异



向量的**加减**

词语的相似



点积： $\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$

大语言模型 LLM： 理解语言

词汇 to 语义

一词多义

我刚买了个苹果 ， 花子一万块

我买了个17（安卓/苹果）

机器人举不起这个箱子，因为它太重了
它没电了

大语言模型 LLM： 理解语言

词汇 to 语义

一词多义

我刚买了个苹果 ， 花子一万块

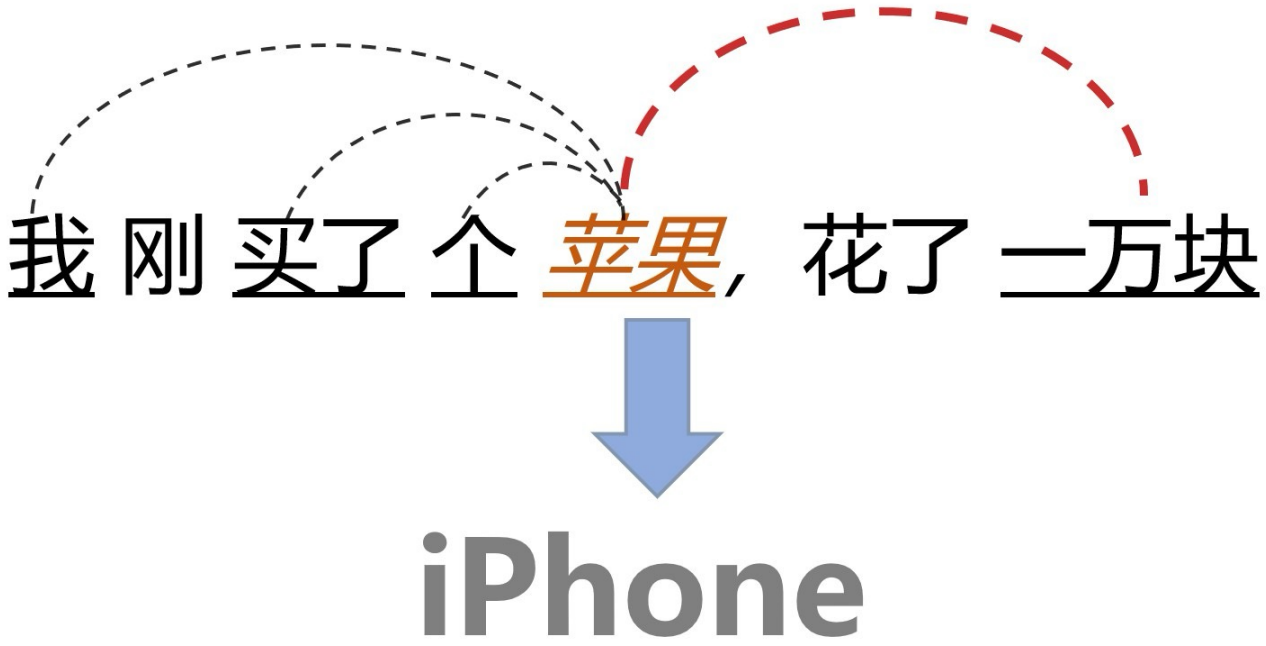
我买了个17（安卓/苹果）

机器人举不起这个箱子，因为它太重了
它没电了

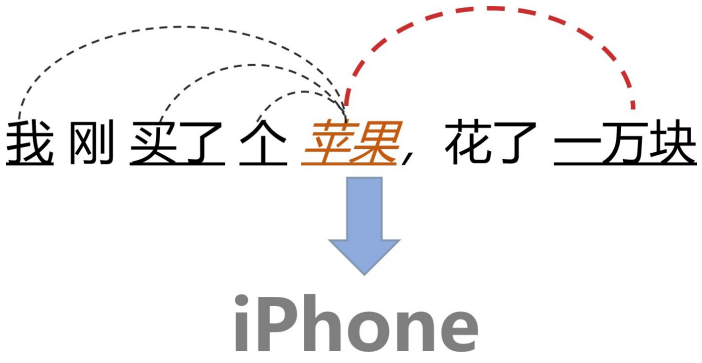
上下文

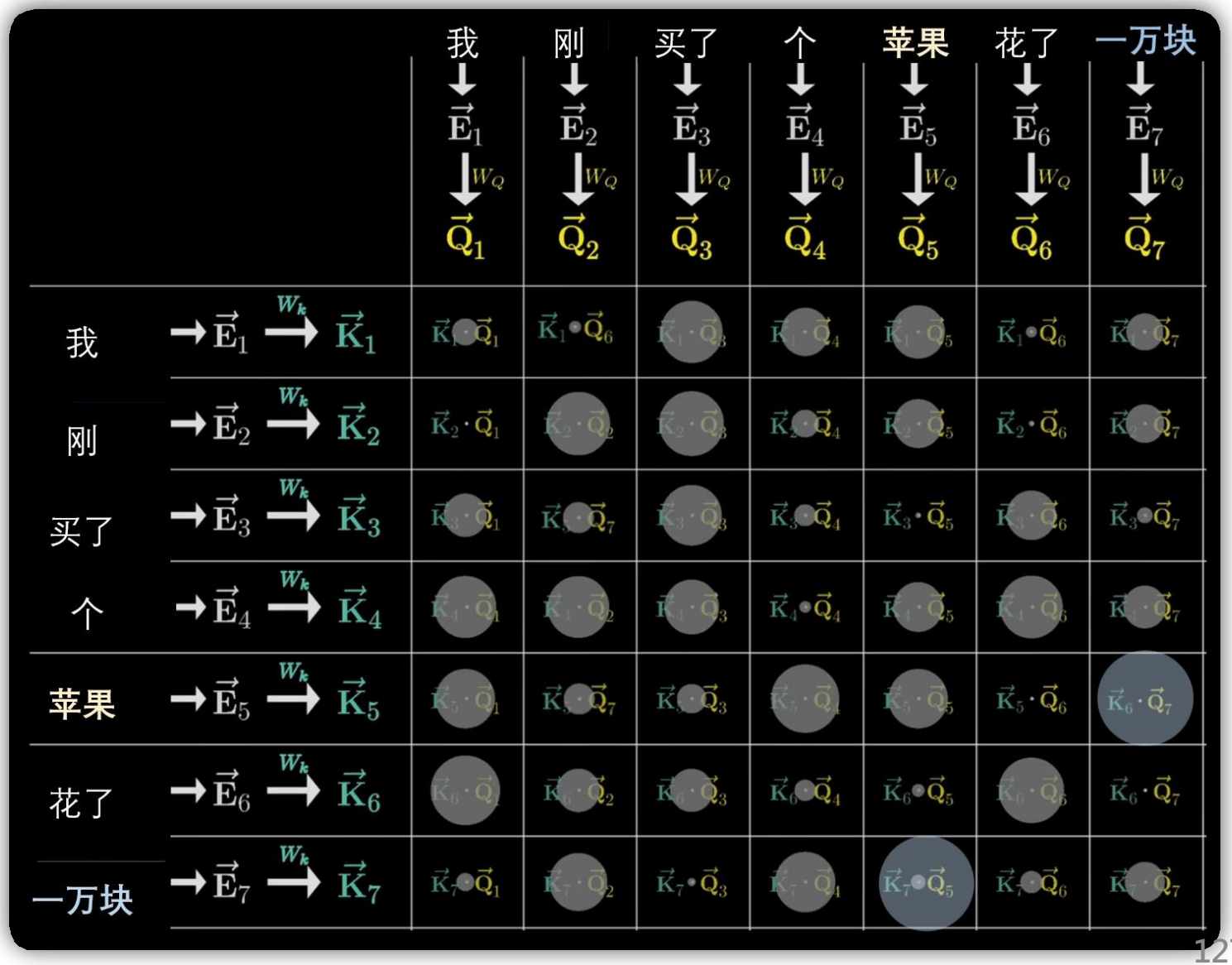
大语言模型 LLM： 理解语言

上下文: Self attention



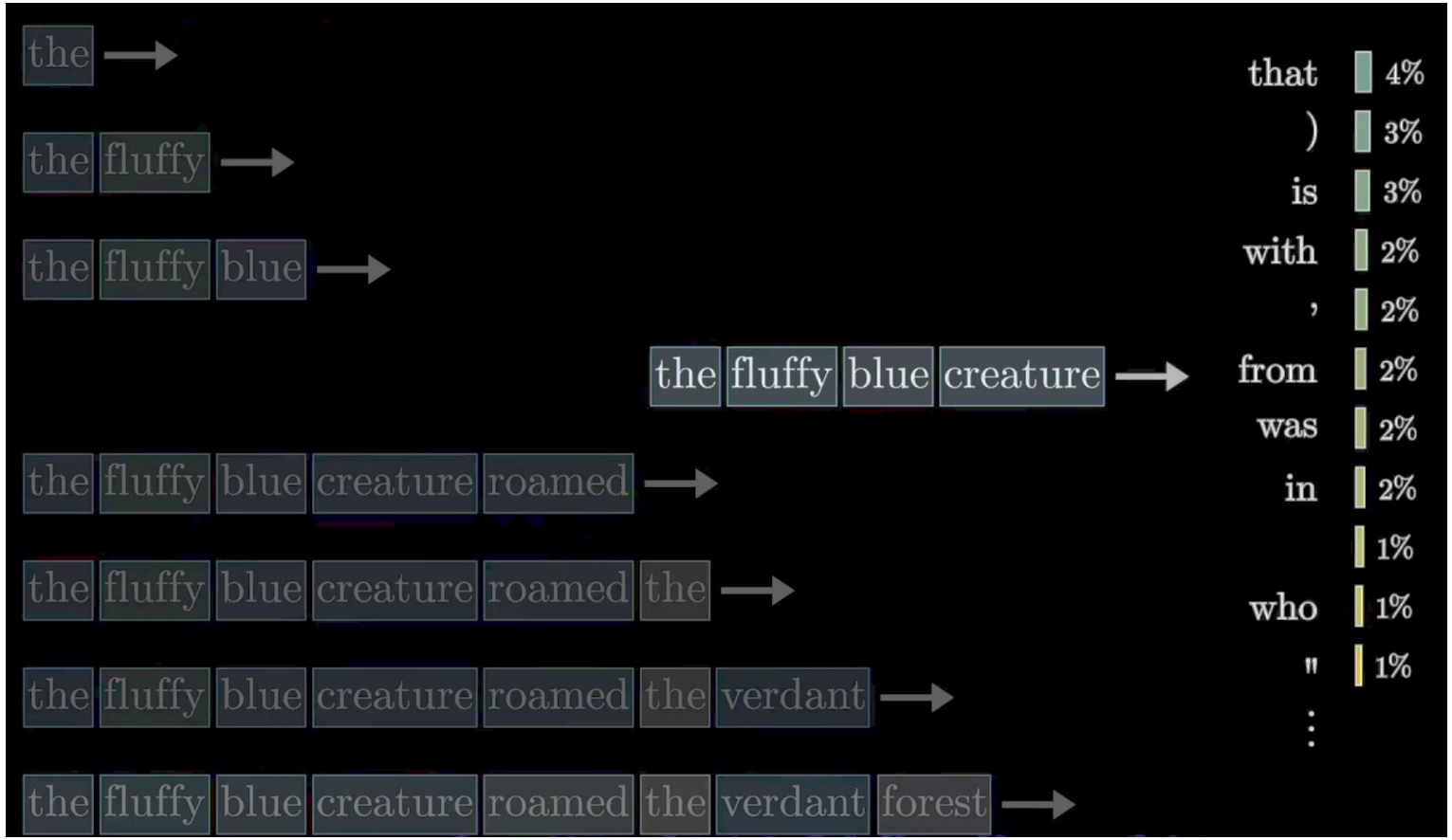
上下文:


我 刚 买了 个 苹果，花了 一万块
iPhone

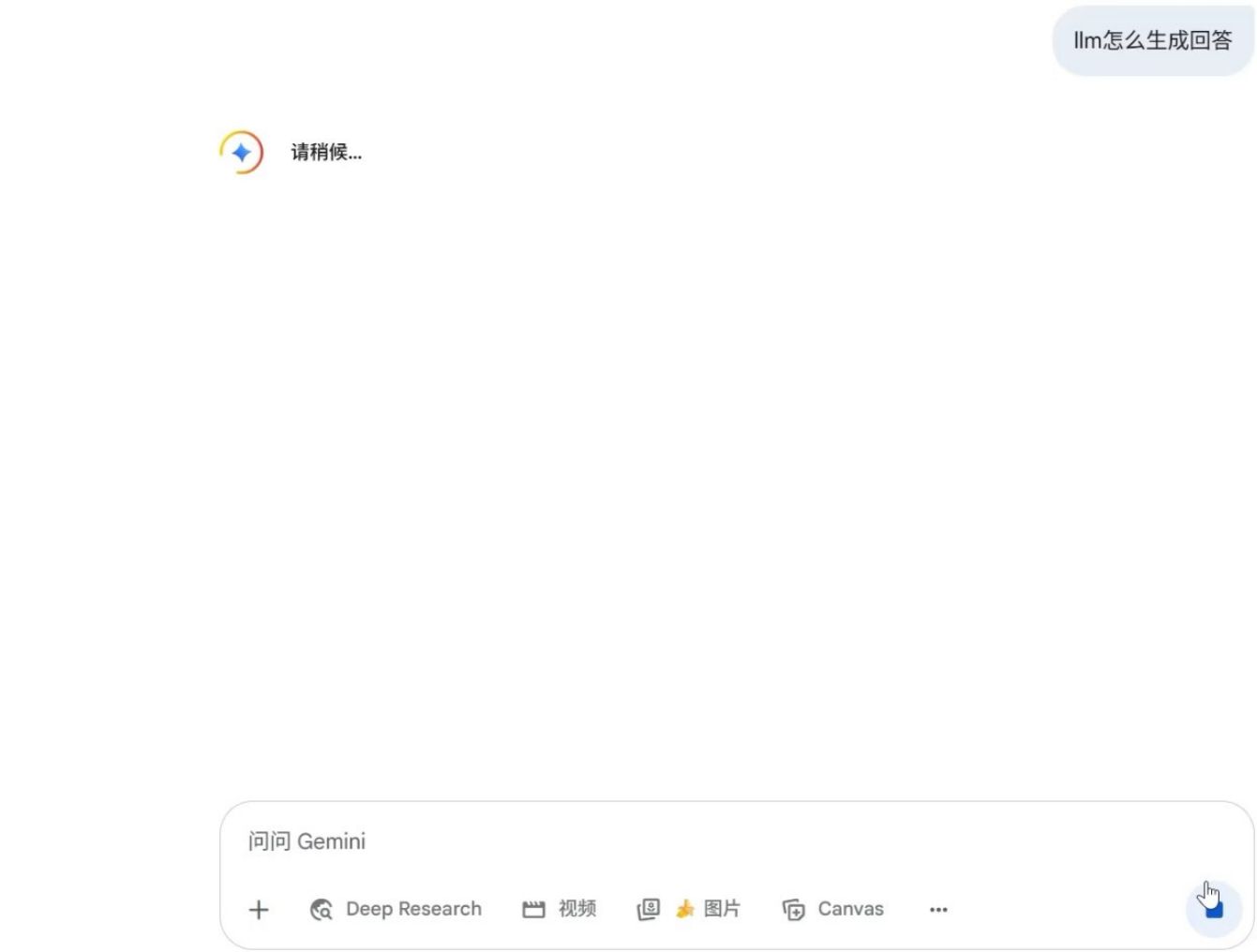

我 刚 买了 个 苹果 花了 一万块
$\vec{E}_1$ $\vec{E}_2$ $\vec{E}_3$ $\vec{E}_4$ $\vec{E}_5$ $\vec{E}_6$ $\vec{E}_7$
W_Q
$\vec{Q}_1$ $\vec{Q}_2$ $\vec{Q}_3$ $\vec{Q}_4$ $\vec{Q}_5$ $\vec{Q}_6$ $\vec{Q}_7$
我 $\vec{E}_1$ W_k $\vec{K}_1$
刚 $\vec{E}_2$ W_k $\vec{K}_2$
买了 $\vec{E}_3$ W_k $\vec{K}_3$
个 $\vec{E}_4$ W_k $\vec{K}_4$
苹果 $\vec{E}_5$ W_k $\vec{K}_5$
花了 $\vec{E}_6$ W_k $\vec{K}_6$
一万块 $\vec{E}_7$ W_k $\vec{K}_7$

大语言模型 LLM： 生成回答

The fluffy blue creature roamed the verdant forest ___?___



预测下一个词的**可能性**


llm怎么生成回答
请稍候...
问问 Gemini
Deep Research
视频
图片
Canvas
Gemini 的回答未必正确无误，请注意核查