第5章 序列比对与分析

北京大学 高歌

gaog@mail.cbi.pku.edu.cn

https://www.gao-lab.org

共享约束需知：使用条款

1. **版权声明**：本PPT及其所有内容（以下简称“本PPT”）仅用于教育和教学用途，版权归属于本PPT作者。
2. **使用要求**：任何使用本PPT的行为均须遵守以下条件：
   1) **致谢和标注**：若部分或全部使用本PPT的内容，请在使用内容的适当位置标注出处，并致谢本PPT作者。
   2) **修改和再分发**：未经作者书面许可，不得对本PPT进行修改或再分发。
3. **禁止商业化使用**：严禁将本PPT用于任何形式的商业化用途，包括但不限于：
   1) 通过网络或其他途径进行付费使用或分发；
   2) 在商业培训、广告或其他商业活动中使用本PPT的内容。
4. **法律责任**：任何违反上述条款的行为，作者保留追究法律责任的权利，包括但不限于：
   1) 要求停止侵权行为；
   2) 追究侵权使用者的经济赔偿责任。
5. **其他规定**：
   1) 本使用条款的解释权归本PPT作者所有。
   2) 作者保留随时更新本使用条款的权利，更新后的条款将即时生效。

本章内容

章节结构

- 第一节：序列特征解析
- 第二节：序列比对和分析
- 第三节：分子演化树构建
- 第四节：讨论与展望

- 本章作者(以拼音序)： 陈士超、高歌、胡德华、田卫东、王明钰
- 本章统稿/协调：高歌

本章内容

本章定位：围绕主流生物学数据(序列)，

- **承上：体现生物信息学“面向数据、方法驱动”的特点**
- **启下：为后续章引入关键概念**

编写原则：突出基本观念与方法，桥接经典问题与最新进展

- **从“小(规模)”入手：主要针对生物信息学经典方法，以一个/几个蛋白/核酸序列为主**
- **以“小”见“大”：承上启下，为后续组学章节引入**

本章导语

包括DNA、RNA和氨基酸在内的生物序列是生物信息学的主要研究对象之一。序列分析的基本思路可以划分为两大类：

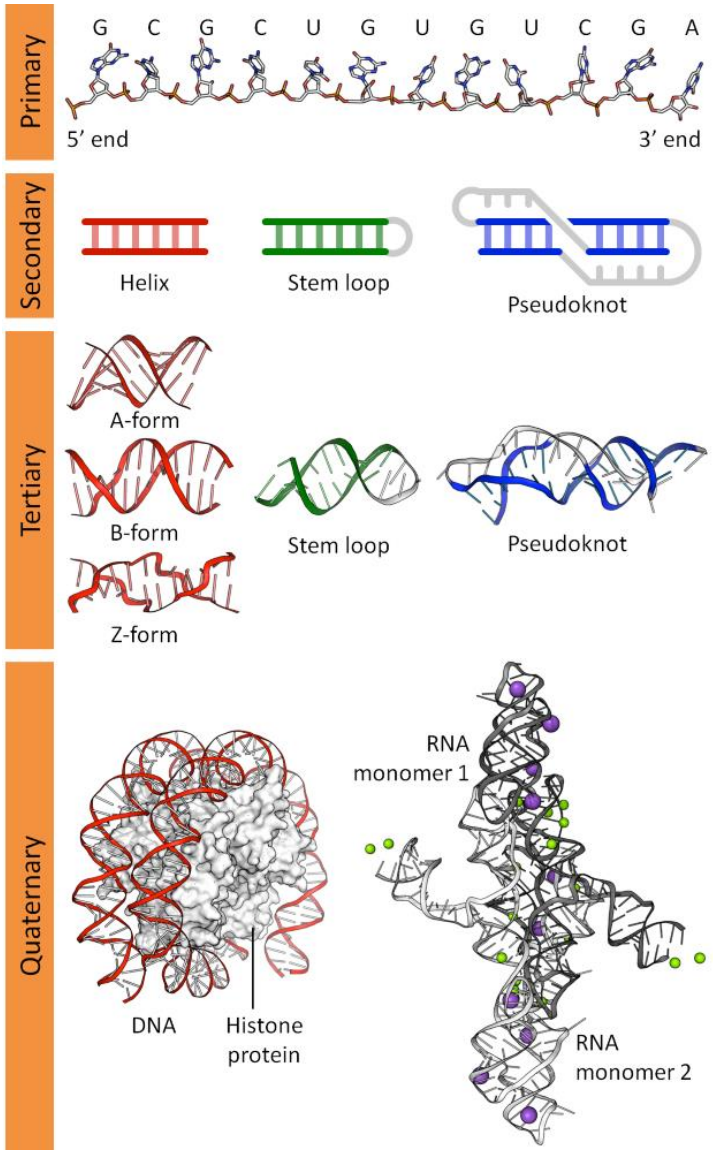
一类是着眼于序列自身，通过多种方法提取并解析其中的结构与功能信息，从而揭示DNA、RNA或蛋白质的生物学特性。例如，分析基因组序列的组成可以帮助我们理解基因的结构，预测潜在的编码区域，并探索基因调控机制；同样，通过分析蛋白质序列，可以预测蛋白质的理化性质及其结构特征，并进一步推测其功能和与其他分子的相互作用。

另一类则是从序列之间的关系出发，通过定量刻画不同的生物分子序列之间的相似性和差异性，来推测它们的功能关联与演化关系。例如，通过寻找与目标序列相似的已知序列，可用于识别未知目标序列的功能和特性；再如通过比较不同物种或不同个体间的序列差异，可以构建演化树并进而解析其间的演化联系。

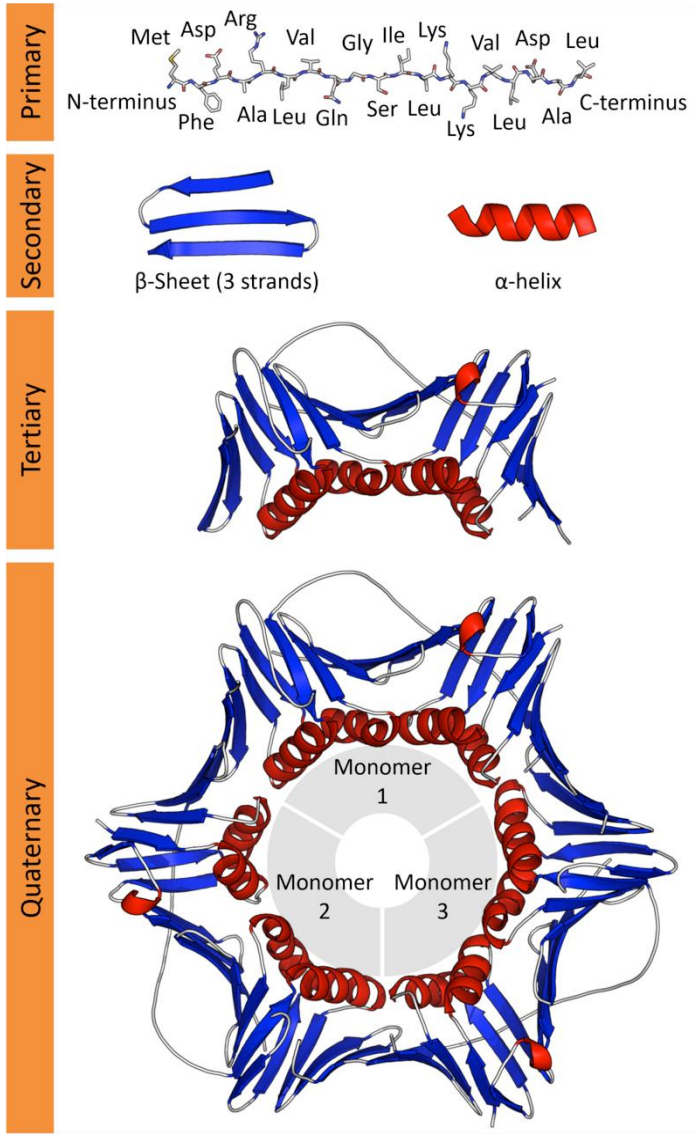
通过将这些不同的分析方法结合，生物信息学家能够在大量的序列数据中发现潜在的生物学模式与规律，并进而为后续的基因功能研究、药物研发和疾病诊断等领域提供关键基础。

第1节 序列特征解析

生物分子的表示



(Image from: https://upload.wikimedia.org/wikipedia/commons/d/da/DNA_RNA_structure_%28full%29.png)



(Modified from: https://commons.wikimedia.org/wiki/Template:Other_versions/Protein_structure_(full)#/media/File:Protein_structure_(full).png)

本节导语

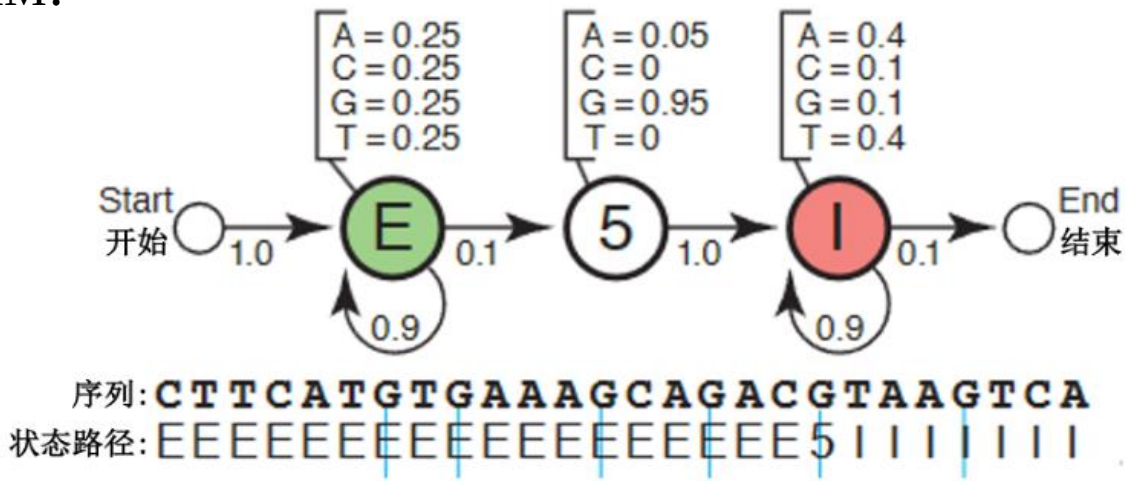
DNA是遗传信息的载体，DNA的碱基组成和排列顺序决定生物的遗传性状。通过对DNA进行基本序列分析不仅可以揭示与特定功能相关的特征信息，也是基因预测的基础。以下以部分典型分析工具与数据库进行介绍，更多内容可见后续章节(如第九章中的蛋白质分析)。

序列比对与分析/序列特征解析

程序或软件名	描述
整合序列分析工具	
BioEdit	用于分析、编辑和处理 DNA 序列数据的生物信息学软件
EMBOSS	综合在线分析软件包
DNAMAN	LynnonBiosoft 公司开发的高度集成化的 DNA 序列编辑软件
DNASTAR	基于 Windows 和 Macintosh 平台的序列分析软件
序列变换	
REVSEQ	EMBOSS 软件包中的序列变换程序之一
Reverse Complement	Sequence Manipulation Suite(SMS)中的序列变换程序
限制性内切酶位点分析	
REBASE	限制性内切酶数据库
NEBcutter	限制性内酶切位点分析工具，整合 REBASE
WebCutter	限制性内酶切位点分析工具，支持线性和环状 DNA 序列分析以及寻找沉默诱变位点
RestrictionMapper	限制性内切酶切位点分析工具，支持线性和环状 DNA 序列分析
重复序列分析	
RepBase	真核生物转座子和重复序列数据库
STRBase	短串联重复序列(STR)数据库
RepeatMasker	散布重复和低复杂性重复序列分析工具，使用 RepBase 和 Dfam 重复序列数据库
CENSOR	使用 RepBase 查找重复序列
Tandem Repeats Finder	串联重复序列分析工具

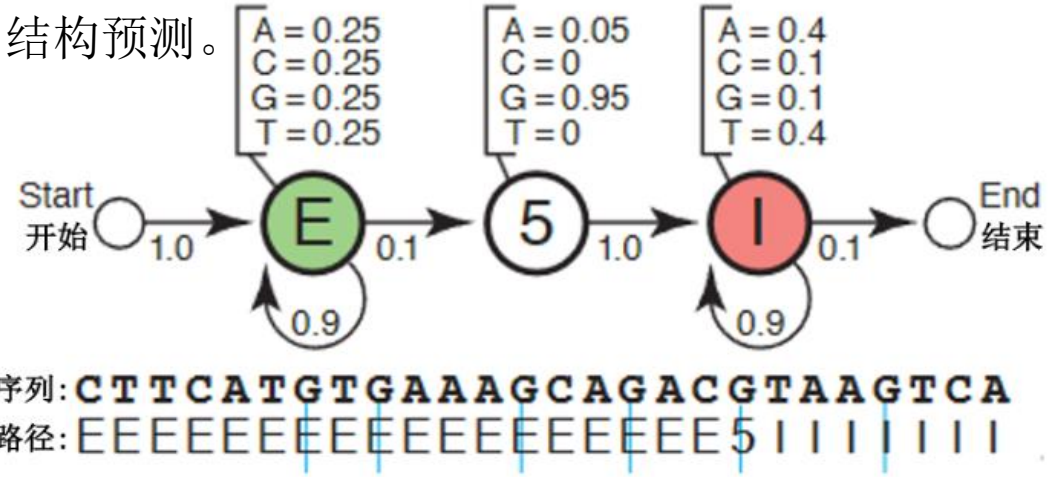
可以采取多种方式来整合这些信号。早期的工作一般是在经典统计推断框架下，通过显式引入特定的统计模型来对DNA序列进行建模与推断。如通过引入隐马尔科夫模型(Hidden Markov Model，HMM)，可以将DNA序列视为由一组包含多个隐状态的马尔可夫过程“生成”的观测序列。具体来说，根据生物学知识，我们猜测不同位置有不同的统计学特性，比如外显子平均碱基组成较均匀(每个碱基25%)，内含子富含A / T(故可假设A / T各40%，C / G各10%)，并且5′SS区域核苷酸几乎总是G(故可假设95%G和5%A)。接下来我们根据上述假设构造HMM：

承上启下



针对图中给出的碱基序列和状态转移路径π，可以得到在参数为 θ 的HMM条件下，总体概率log P(S，π|HMM，θ)= –41.22。因此，通过使用Viterbi算法即可得到最可能的状态序列，并进而预测这段序列中5’剪接序列的位置(5'SS)。

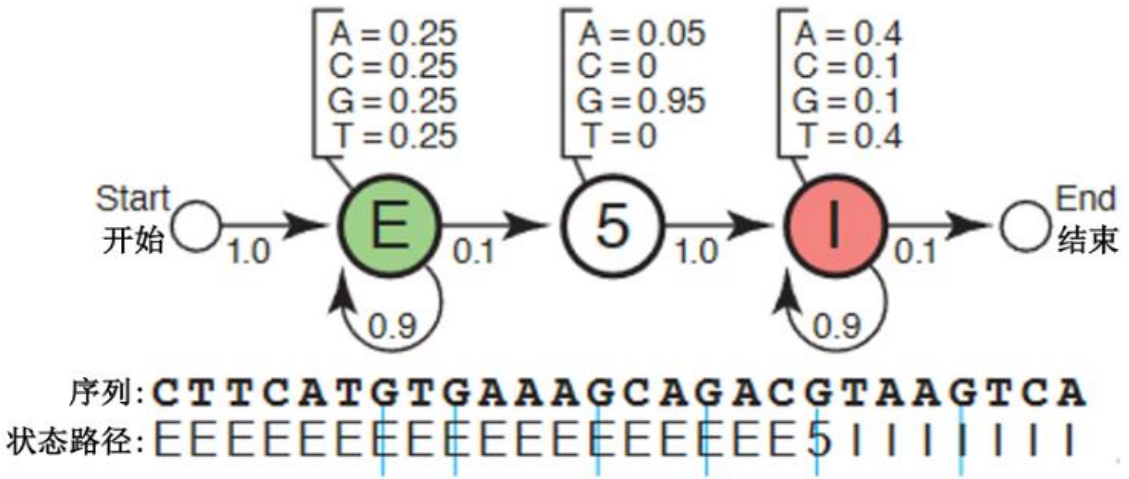
与之类似，美国麻省理工大学的Burge和Karlin于1997年开发的GenScan，是基于广义隐马尔可夫模型的人类及脊椎动物基因预测软件。GenScan通过识别序列中的统计特征，如密码子使用频率、外显子和内含子边界信号的共识序列等，对基因进行预测。它还考虑了基因的起始和终止区域，以及潜在的启动子信号，以提供全面的基因结构预测。



序列比对与分析/序列特征解析

近年来，随着数据的积累与算力的提升，支持向量机(SVM)、随机森林或神经网络等机器学习方法也被广泛应用于基因预测。与早期的显式建模不同，这些方法通过对已知的基因组注释、转录组测序数据等训练数据的学习以“自动” 识别出序列中的特定模式，并用这些模式来预测新序列中的基因。

值得注意的是，虽然类似的模型与算法可以以 “从头预测”的方式在未知序列上工作，但考虑到上述序列特征在物种间的变化，这些预测算法往往需要事先设定物种参数。



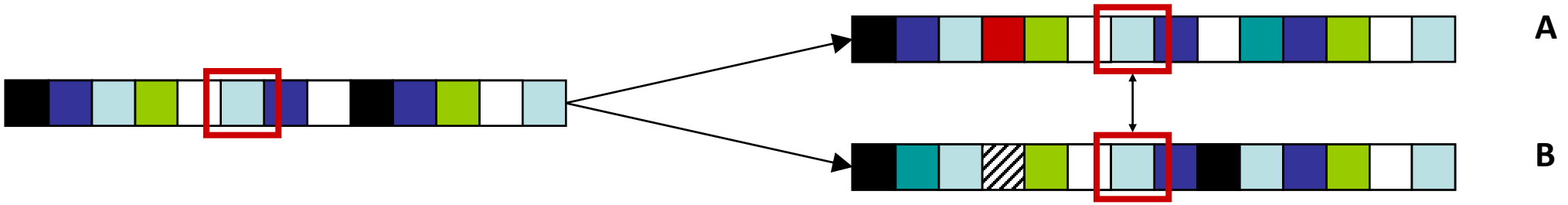
第2节 序列比对和分析

本节导语

量化衡量序列之间的相似性和差异性是解析序列间关系的核心前提。为此，需要首先厘清几个概念：同源(Homologous)是一个演化概念。序列之间“同源”即指它们拥有一个或多个共同的起源(祖先)。在实践中，通常通过序列相似(Similarity)来检定序列之间的同源关系。值得指出的是，根据“序列-结构-功能”这一关联链条，具有相似序列的分子往往也具有相似的功能，因此与已知功能的序列相似也常常被用来推断未知序列的功能。

Sequence Alignment: in Biology

- **The purpose of a sequence alignment is to line up all residues in the inputted sequence(s) that based on their functional or evolutionary relationship.**



Sequence Alignment: in Maths

- **Input:**
  - **Two (or more) sequences S1, S2, …, Sn,**
  - **and a scoring function f.**

- **Output:**
  - **The alignment of S1, S2, …, Sn, which has the optimal score.**

$$\arg\max_{ali}(f(ali(S_1, S_2, \cdots, S_n)))$$

Scoring Alignment


```
GAATC      GAAT-C      -GAAT-C
CATAC      C-ATAC      C-A-TAC

GAATC-     GAAT-C      GA-ATC
CA-TAC     CA-TAC      CATA-C
```

Scoring function: measure the quality of a given alignment.

- **Scoring matrix,**
- **Gap penalty**

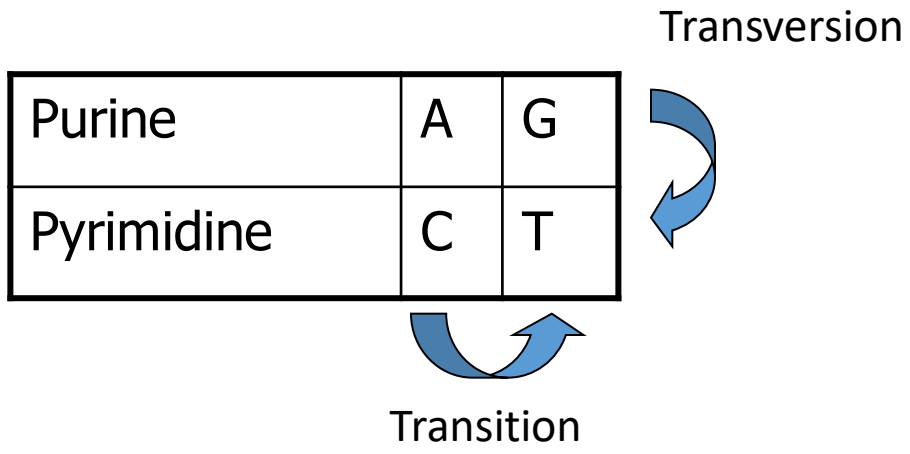
Scoring Matrix

- Scoring a substitution

- Measure the likelihood of a given substitution happened in the real world.
  - Substitutions that are more likely should get a higher score
  - Substitutions that are less likely should get a lower score

- Scoring Matrices are designed to detect signal above background, i.e. to detect similarities beyond what would be observed by chance alone

Nucleotide Scoring Matrix



GAATC

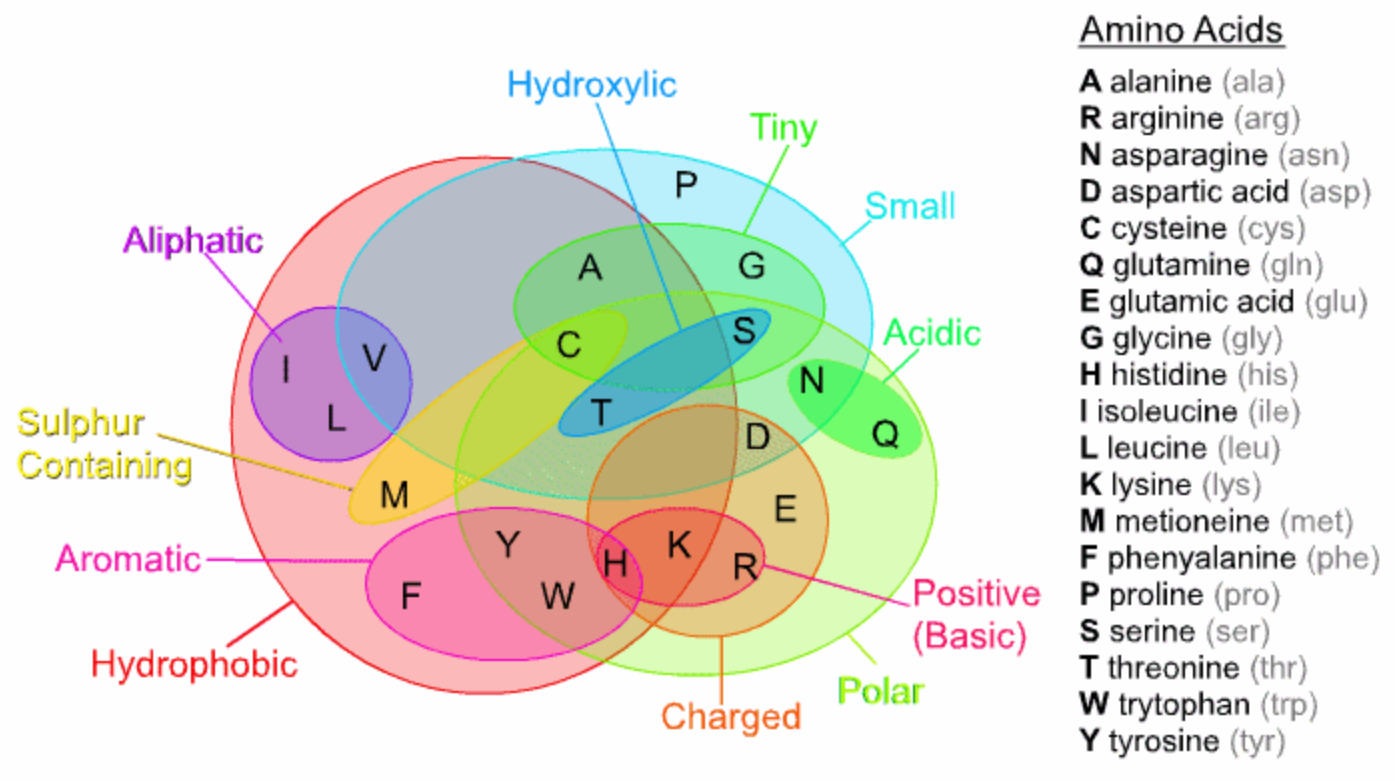
CATCC

-7 + 2 + (-7) + (-5) + 2 = -15

A hypothetical substitution matrix:

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

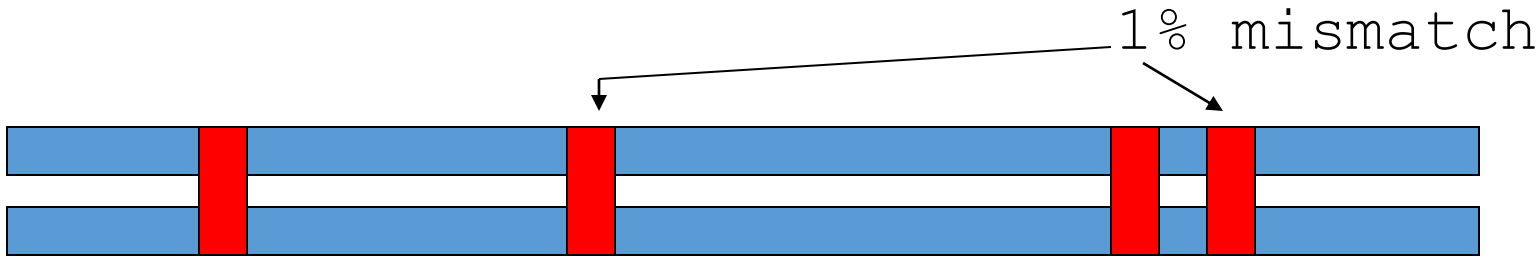
Amino acid Scoring matrix



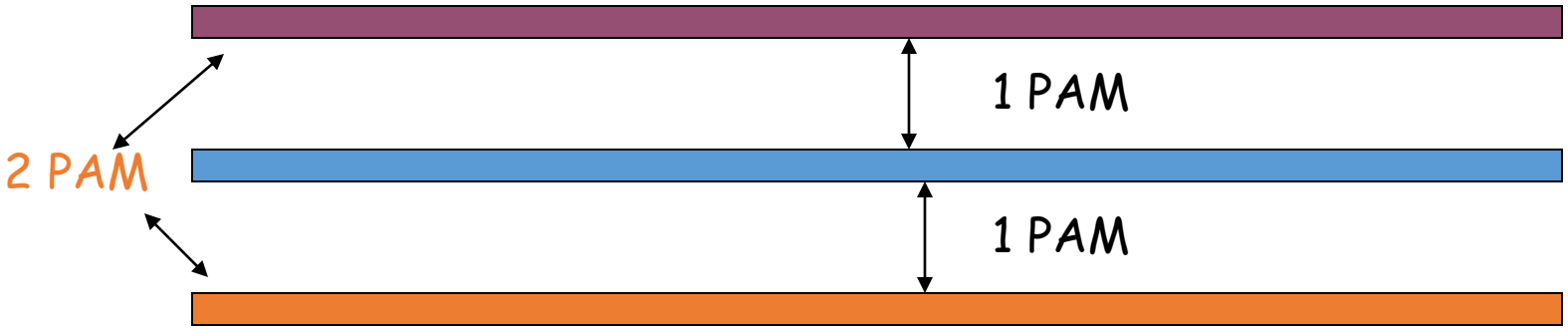
(Adopted from Prof. Jingchu Luo)

PAM: Scoring by evolutionary distance

- **PAM: Percent Accepted Mutation**
- **Two sequences are 1 PAM apart if they differ in 1 % of the residues.**
- **1 PAM = one step of evolution**



From PAM2 to PAM250



- 1 PAM = one step of evolution
- PAM_2 = two step of evolution = PAM_1 * PAM_1
- PAM_{250}
  - = PAM_1*PAM_{249}
  - = PAM_1^{250}

PAM250

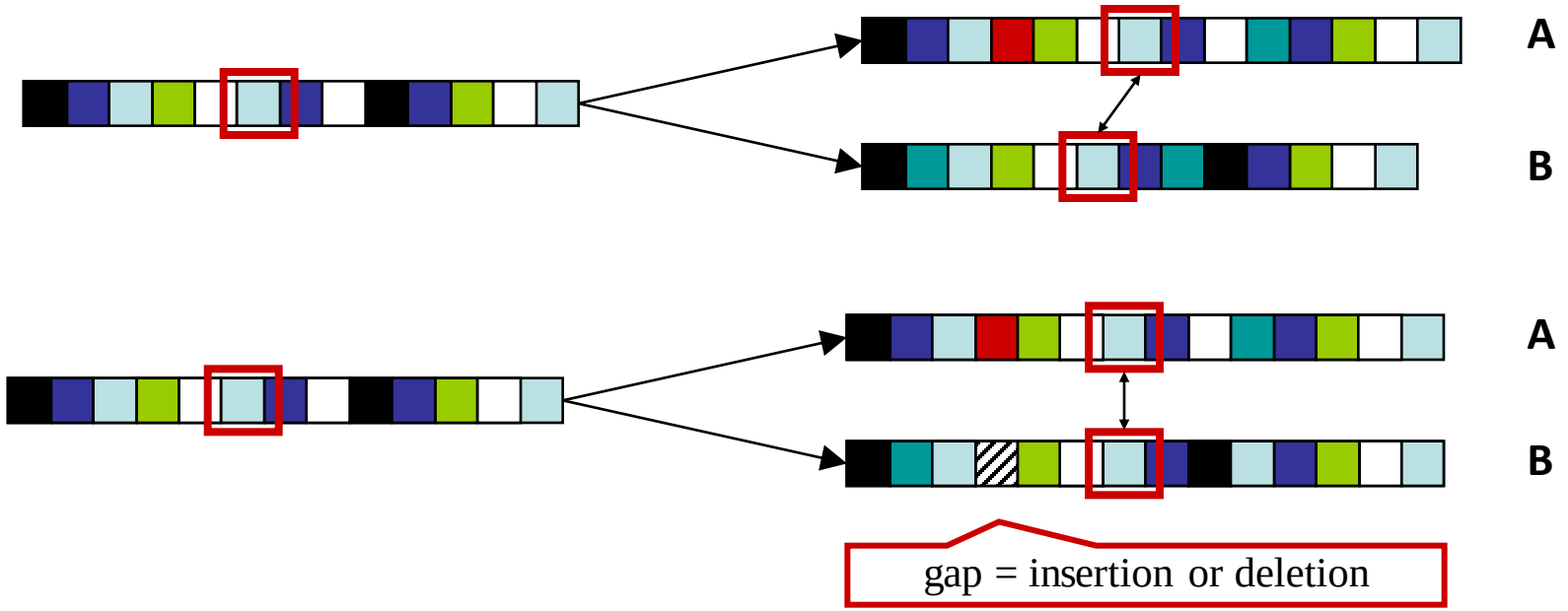
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1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

BLOSUM62

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

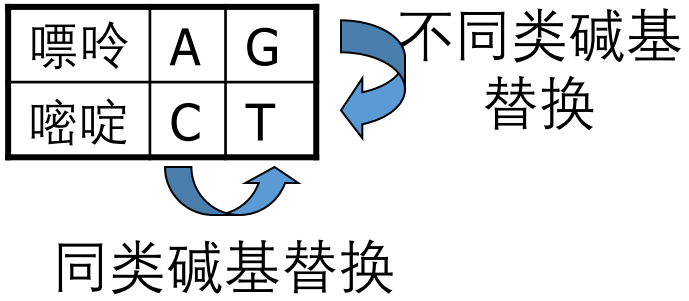
Scoring gaps

- **Gap ➔ an Insertion or Deletion during the evolution.**
  - **Much less frequent than residue substitution, due to the function constrain.**
  - **Often have a negative score as “penalty”.**



- 碱基替换打分

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



- 每个空位罚5分

GAAT−C
C−ATAC

(-7) + (-5) + (2) + (2) + (-5) + 2= -11

Sequence Alignment: Enumerate?

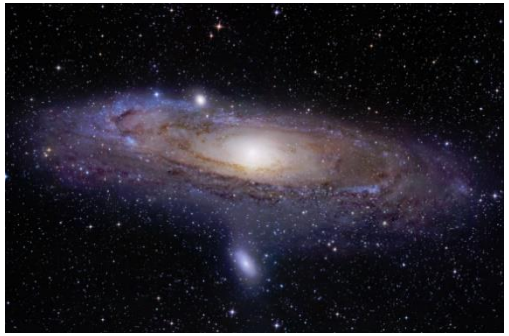

```
GAATC      GAAT-C     -GAAT-C
CATAC      C-ATAC     C-A-TAC

GAATC-     GAAT-C     GA-ATC
CA-TAC     CA-TAC     CATA-C
```

Sequence Alignment: Enumerate?

- **If gaps are allowed in every position and of every length, naive enumeration is exponential in the length of the sequences.**

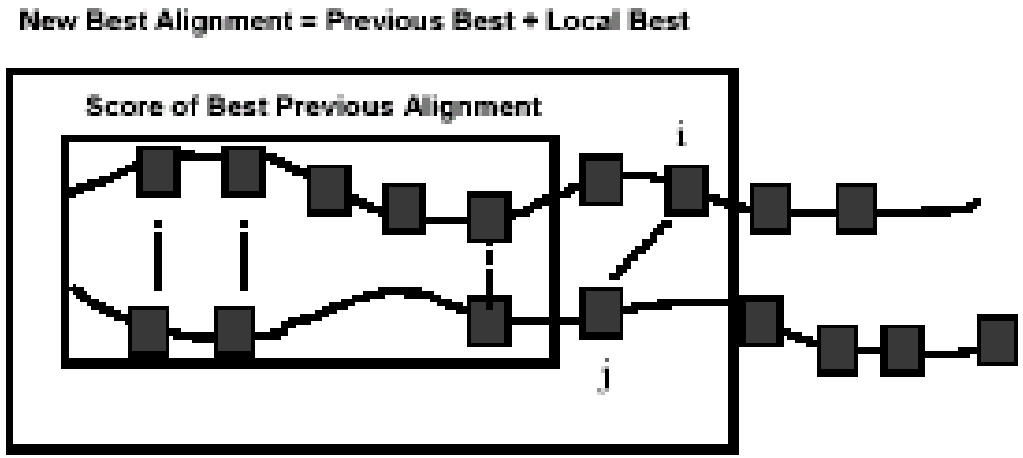
$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

- **For two sequences with 300 letters, 10^{88} possible alignments exist.**
- **The visible universe is estimated to contain 10^{78} ~ 10^{80} atoms (from: wikianswers)**



Sequence Alignment: What is the computational Algorithm?

- **The best alignment that ends at a given pair of letters is the best alignment of the sequences up to that point, plus the best alignment for the two additional letters.**



G G –
C – C

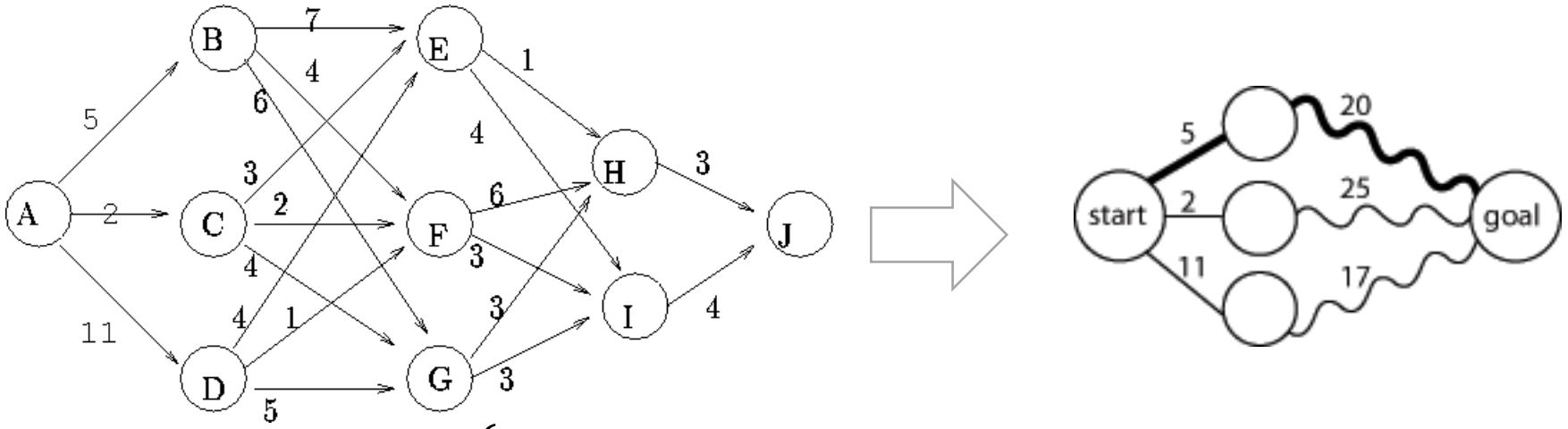
(Russ Altman BMI214)

Sequence Alignment: What is the computational Algorithm?

动态规划(Dynamics Programming)，一种用来解决具有最优子结构(optimal substructure)性质优化问题的计算机算法

- 大问题问题的最优解可以从其子问题的最优解来有效地构建

➔ 将原始问题分解为若干个规模较小的同构子问题



$$F(A,J) = \min \begin{cases} F(A,H) + Dist(H,J) \\ F(A,I) + Dist(I,\mathrm{J}) \end{cases} \qquad F(A,A) = 0$$

双序列比对动态规划算法公式

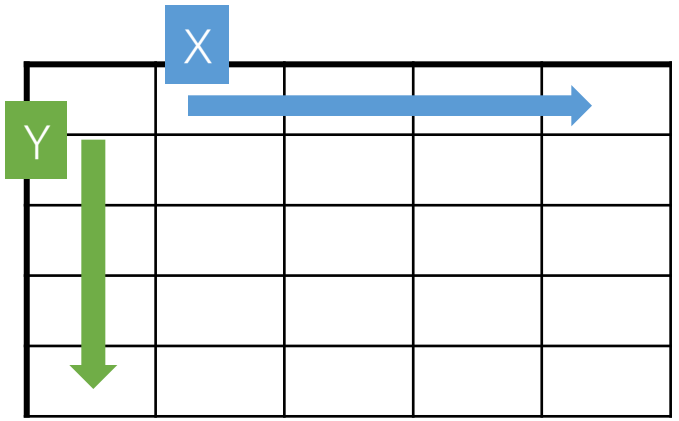
假设x和y是需要比对的两个序列
F(i,j) 是$x_{1...i}$ and $y_{1...j}$之间最优比对的得分
s(A,B)是用A替换B(错配)的得分；d 是空位得分(线性)

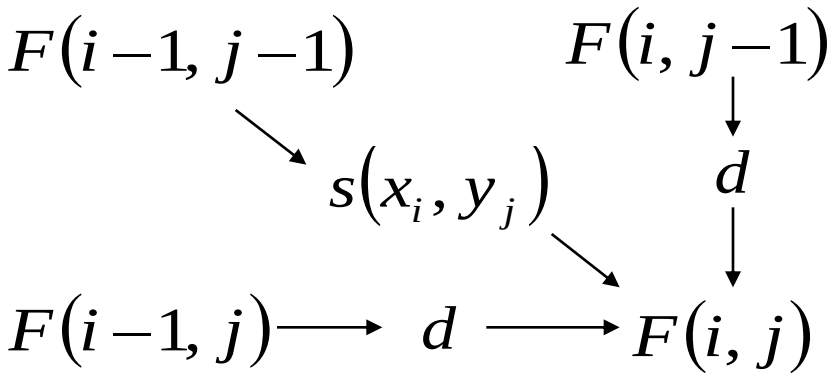
$$F(0,0)=0$$

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j) & x_i \text{ 比对到} y_j\\ F(i-1,j)+d & x_i \text{ 比对到空位}\\ F(i,j-1)+d & y_j \text{ 比对到空位}\end{cases}$$

$$F(0,0)=0$$

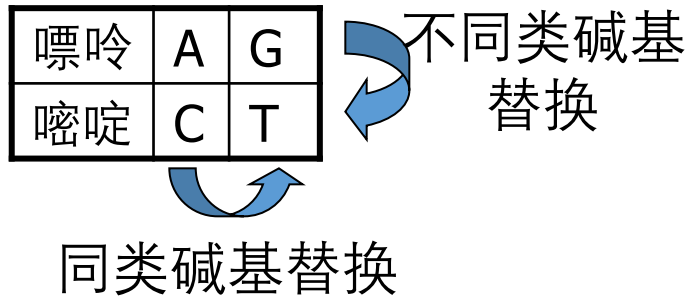
$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j) & x_i \text{ 比对到} y_j\\ F(i-1,j)+d & x_i \text{ 比对到空位}\\ F(i,j-1)+d & y_j \text{ 比对到空位}\end{cases}$$





- 碱基替换打分

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



- 每个空位罚5分

GAAT−C

C−ATAC

(-7) + (-5) + (2) + (2) + (-5) + 2= -11

输入序列 S1: AAG

输入序列 S2: AGC

		A	A	G
A				
G				
C				

序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5

		A	A	G
	0			
A				
G				
C				

$$F(0,0)=0$$

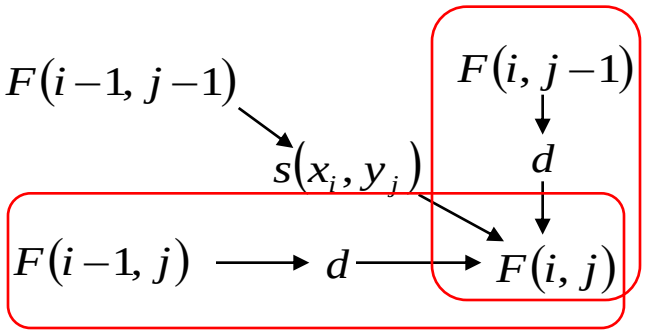
$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j)\\F(i-1,j)+d\\F(i,j-1)+d\end{cases}$$

序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5

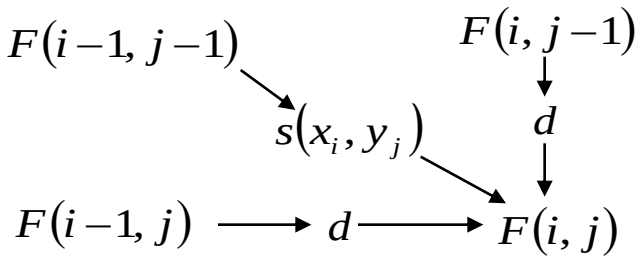
		A	A	G
	0 →	-5 →	-10 →	-15
A	-5			
G	-10			
C	-15			



序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5



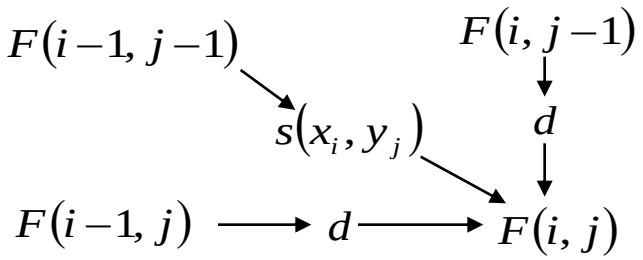
		A	A	G
	0 →	-5 →	-10 →	-15
A	-5	2 →	-3 →	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5

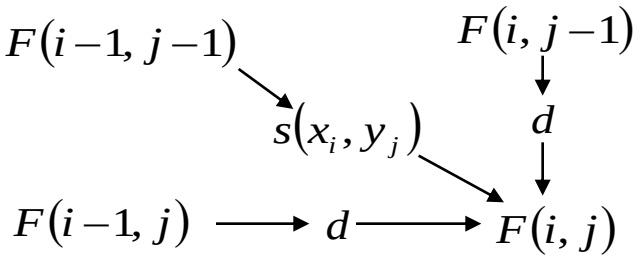
		A
	0 →	-5
A	-5	2

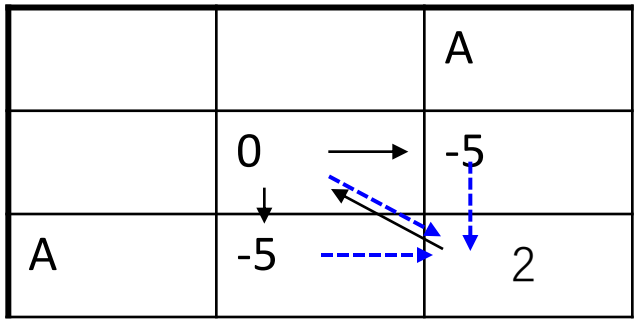


序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5





-5 + (-5) = -10

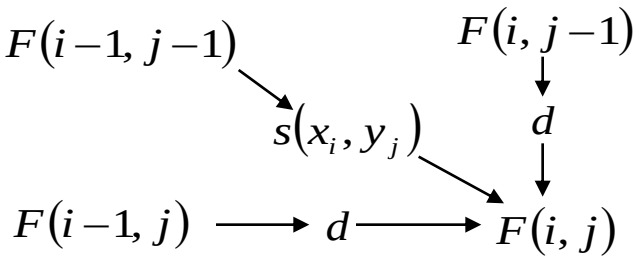
0 + 2 = 2

-5 + (-5) = -10

序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5

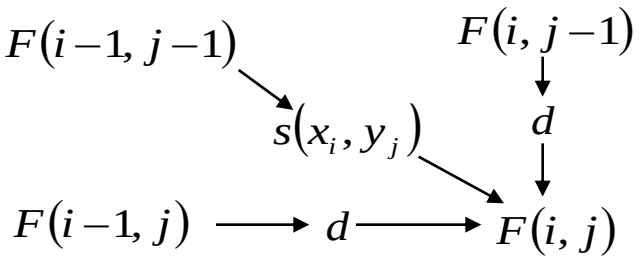


		A	A	G
	0 →	-5 →	-10 →	-15
A	-5	2 →	-3 →	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

序列比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优比对
空位得分d=-5



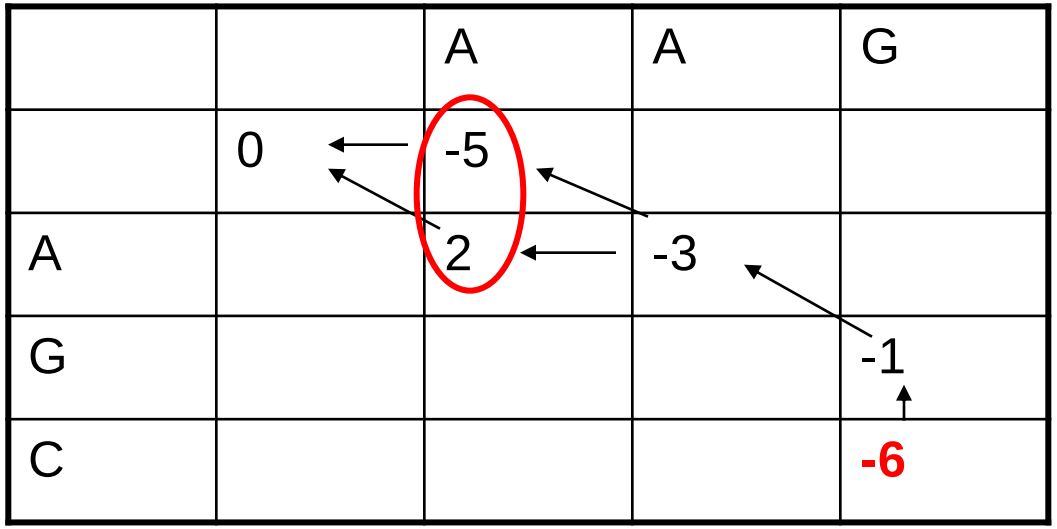
		A	A	G
	0 →	-5 →	-10 →	-15
A	-5	2 →	-3 →	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

回溯得到最终比对结果

从矩阵右下角逐步回溯至左上角。每个箭头指向序列中前一个位置的符号

A A G –
– A G C

A A G –
A – G C



对两条序列从头到尾全部的残基进行比对：全局比对

Needleman–Wunsch 算法：端到端全局比对

J. Mol. Biol. (1970) **48**, 443–453

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNSCH

Department of Biochemistry, Northwestern University, and Nuclear Medicine Service, V. A. Research Hospital Chicago, Ill. 60611, U.S.A.

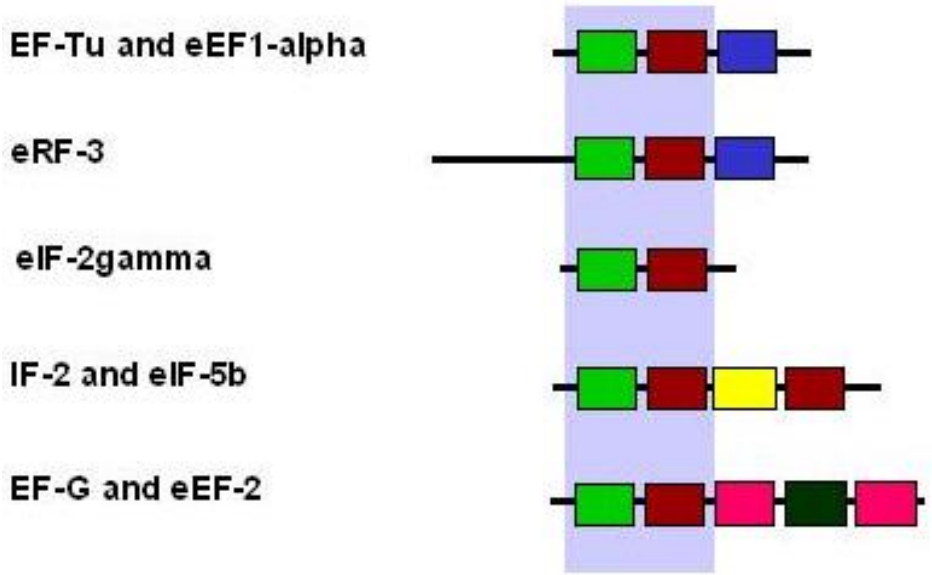
(*Received 21 July 1969*)

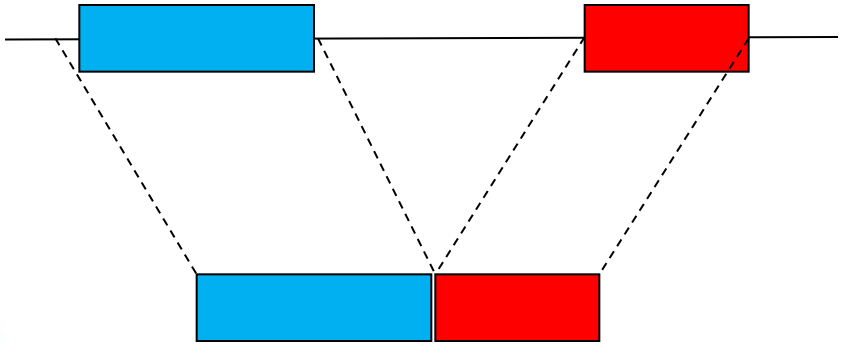
A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for all possible interruptions in either of the sequences. While the interruptions give rise to a very large number of comparisons, the method efficiently excludes from consideration those comparisons that cannot contribute to the maximum match.

Comparisons are made from the smallest unit of significance, a pair of amino acids, one from each protein. All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through the array. For this maximum match only certain of the possible pathways must be evaluated. A numerical value, one in this case, is assigned to every cell in the array representing like amino acids. The maximum match is the largest number that would result from summing the cell values of every pathway.

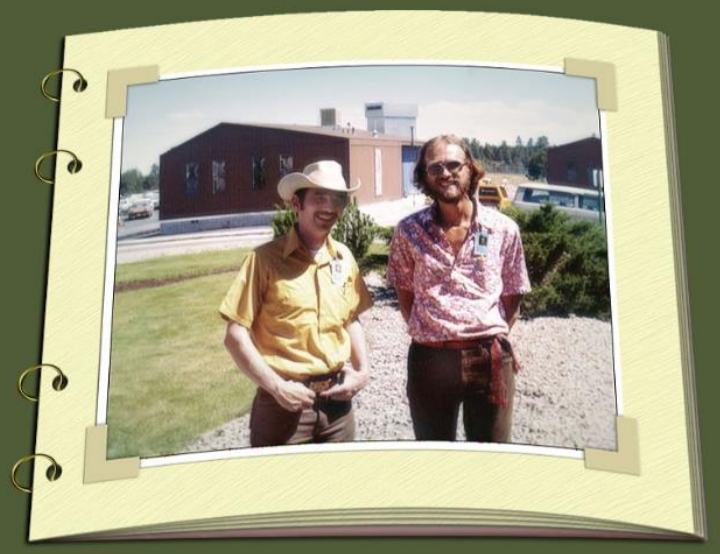
发现相似的局部序列 ➔ 局部比对





Smith-Waterman 算法：局部比对

Smith and Waterman at Los Alamos, New Mexico
Photo by David Lipman, taken summer of 1980



(http://www.cmb.usc.edu/people/msw/SmithWaterman.html)

J. Mol. Biol. (1981), **147**, 195–197

Identification of Common Molecular Subsequences

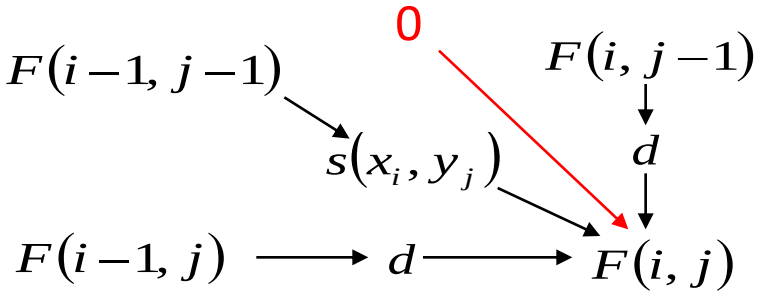
The identification of maximally homologous subsequences among sets of long sequences is an important problem in molecular sequence analysis. The problem is straightforward only if one restricts consideration to contiguous subsequences (segments) containing no internal deletions or insertions. The more general problem has its solution in an extension of sequence metrics (Sellers 1974; Waterman *et al.*, 1976) developed to measure the minimum number of "events" required to convert one sequence into another.

These developments in the modern sequence analysis began with the heuristic homology algorithm of Needleman & Wunsch (1970) which first introduced an iterative matrix method of calculation. Numerous other heuristic algorithms have been suggested including those of Fitch (1966) and Dayhoff (1969). More mathematically rigorous algorithms were suggested by Sankoff (1972), Reichert *et al.* (1973) and Beyer *et al.* (1979), but these were generally not biologically satisfying or interpretable. Success came with Sellers (1974) development of a true metric measure of the distance between sequences. This metric was later generalized by Waterman *et al.* (1976) to include deletions/insertions of arbitrary length. This metric represents the minimum number of "mutational events" required to convert one sequence into another. It is of interest to note that Smith *et al.* (1980) have recently shown that under some conditions the generalized Sellers metric is equivalent to the

双序列局部比对动态规划算法公式

$$F(0,0)=0$$

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

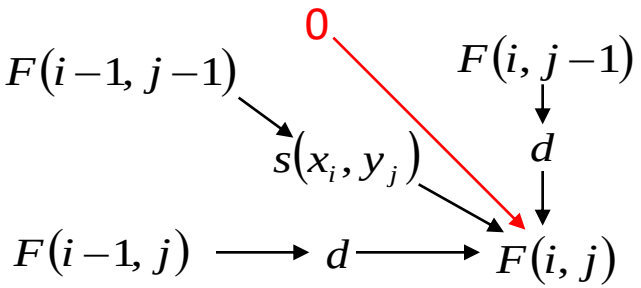


局部比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优局部比对
空位得分d=-5

		A	A	G
A				
G				
C				

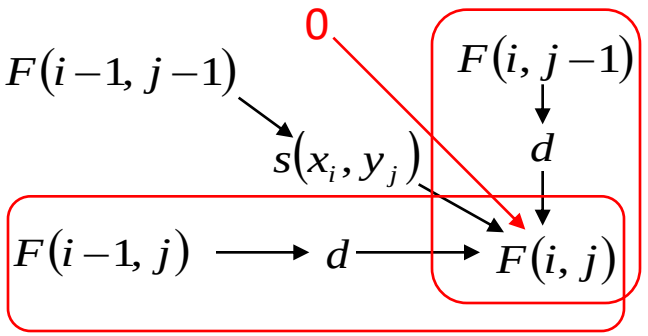


局部比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优局部比对
空位得分d=-5

		A	A	G
	0	0	0	0
A	0			
G	0			
C	0			

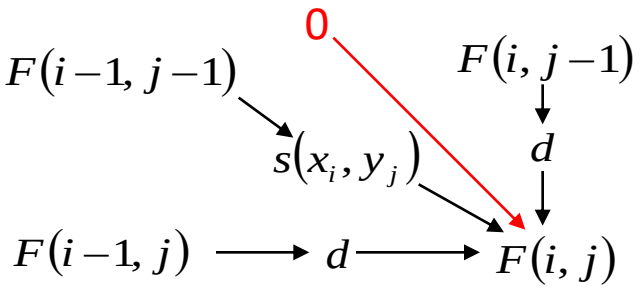


局部比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优局部比对
空位得分d=-5

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

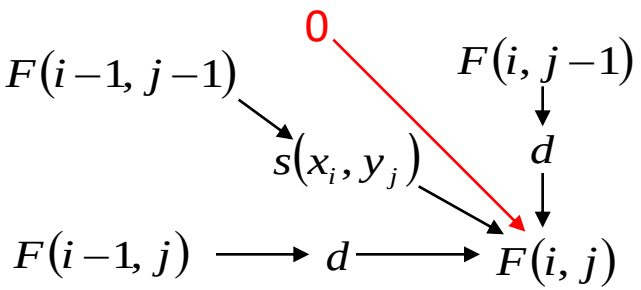


局部比对动态规划算法示例

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

寻找AAG and AGC的最优局部比对
空位得分d=-5

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0



回溯得到最终比对结果

从矩阵中最大值所在位置开始回溯到0为止

A G
A G

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

回溯得到最终比对结果

另一个最优局部比对结果

A
A

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

$$F(0,0)=0$$

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

全局比对

$$F(0,0)=0$$

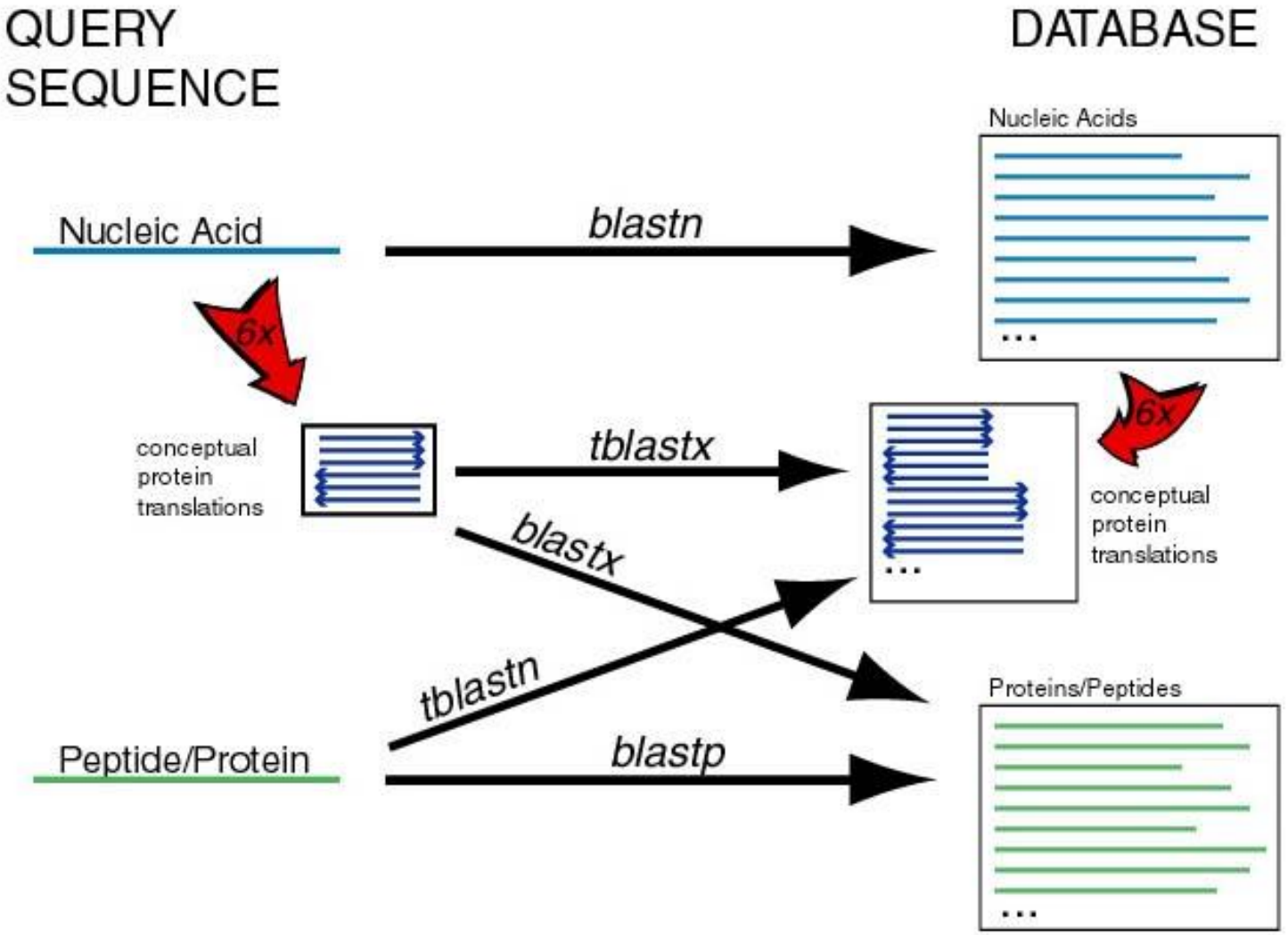
$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j)\\F(i-1,j)+d\\F(i,j-1)+d\\0\end{cases}$$

局部比对

序列比对与分析/应用与拓展

- 序列数据库检索
- 基于序列比较的序列聚类
- 多重序列比对

序列数据库检索



(Joel H. Graber)

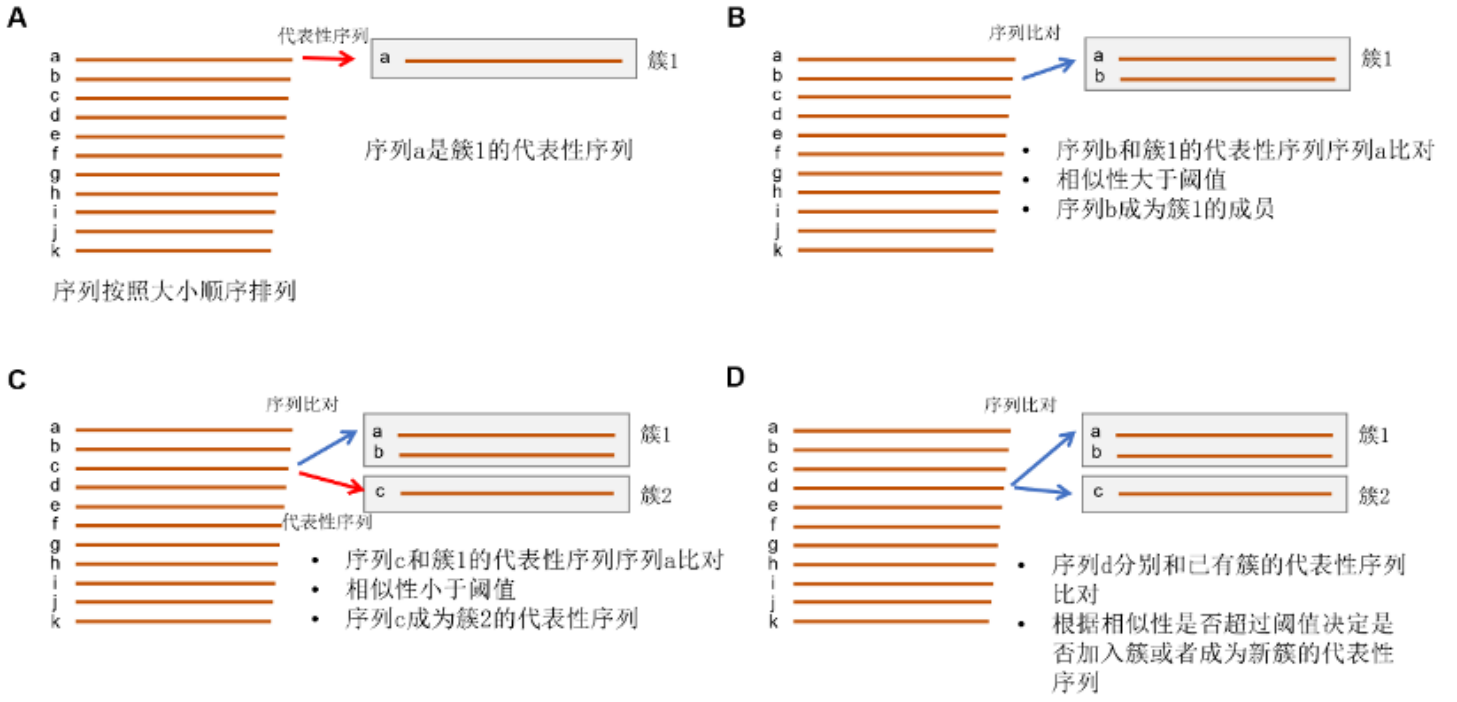
E-Value: an assessment

- **E-value: Expectation value**
  - **the number of alignments with a given score that would be expected to occur at random in the database that has been searched**
  - **e.g. if E=10, 10 matches with scores this high are expected to be found by chance**

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

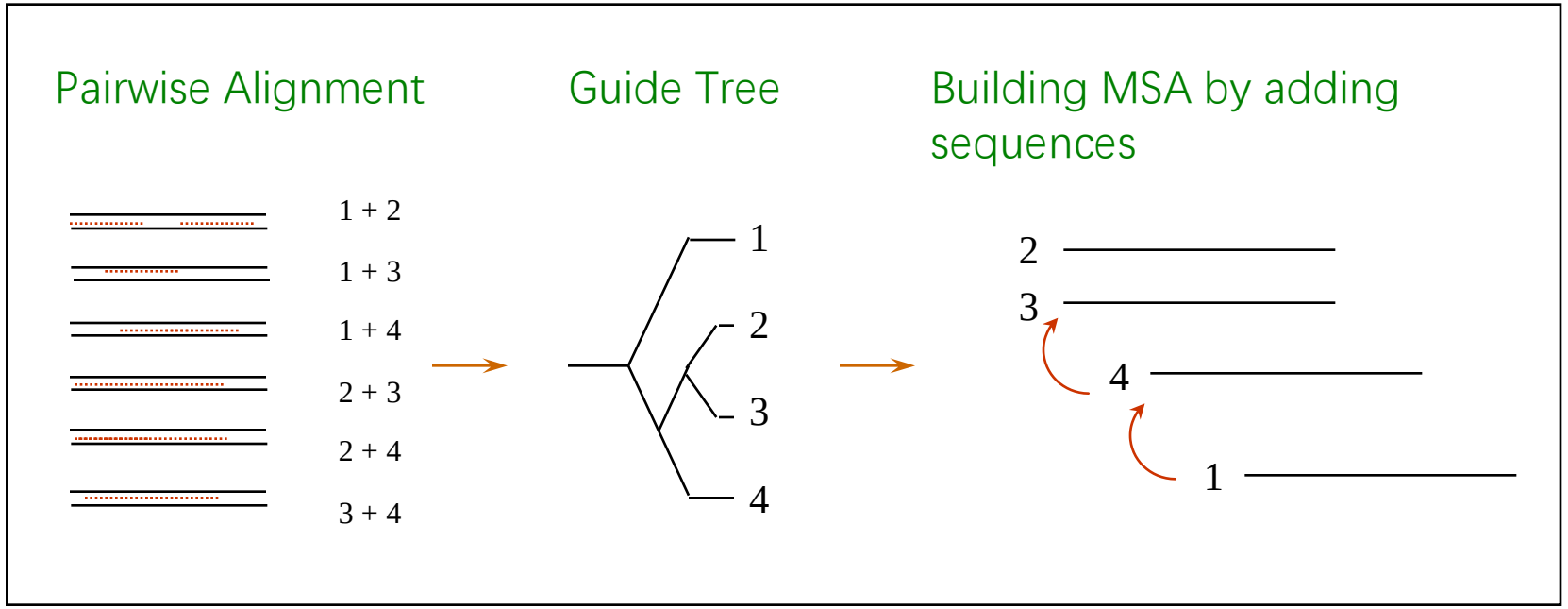
基于序列比较的序列聚类

通常认为，序列上相类似的蛋白质或者基因往往具有相类似的功能，而且相似的程度往往和功能相似的程度正相关；类似的，序列相似程度在不考虑趋同进化等特殊事件的前提下也可以作为序列间进化关系远近的表征。因此，通过将蛋白或者DNA序列按照互相之间相似的程度分成多个簇(cluster)进行序列聚类(sequence clustering)，可以实现对蛋白质或DNA分子的功能与演化分类。其中具有代表性的算法包括CD-HIT、UCLUST和Linclust。CD-HIT和UCLUST使用了相类似的贪心增量策略(greedy incremental strategy)来实现序列的聚类。



多重序列比对:ClustalW

1. Distance matrix construction, by pairwise alignment of each pair of sequences
2. Guide tree construction from the distance matrix
3. Progressive alignment of the sequences according to the branches in the guide tree



第3节 分子演化树构建

本节导语

通过系统比较不同物种或不同个体间的序列差异，可以构建演化树以解析其间的演化联系，并进而表征特定DNA/蛋白质分子从共同祖先逐渐演化分化的过程。演化树(又称系统发育树，Phylogenetic Tree)是一种针对特定演化关系的图形表示，类似于家谱图，演化树提供了生物多样性和演化历史的重要信息，可用于解释生物特征的起源和演化。

Inferring Phylogenetic Tree by UPGMA

UPGMA (Unweighted Pair-Group Method using arithmetic Average)

- **Algorithm:**
  - **From distance matrix, cluster pair of Units with smallest distance, and re-calculate new distance for other Units**
  - **Repeat previous step until clusters converge**

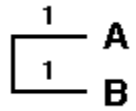
Hierarchical Clustering (by Average Linkage)

- **Algorithm:**
  - **From distance matrix, cluster pair of Units with smallest distance, and re-calculate new distance for other Units**
  - **Repeat previous step until clusters converge**

UPGMA Demo

- **Cluster pair with smallest distance, then**
- **Recalculate distance matrix**

	A	B	C	D	E
B	2				
C	4	4			
D	6	6	6		
E	6	6	6	4	
F	8	8	8	8	8

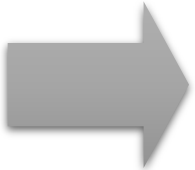


- **Calculate new distance using composite Unit(A,B):**
  - **Distance between a simple Unit and a composite Unit is the average of the distances between the simple Unit and the constituent simple Unit of the composite Unit**

```
dist (A,B),C = (dist A,C + dist B,C) / 2 = (4 + 4) / 2 = 4
dist (A,B),D = (dist A,D + dist B,D) / 2 = (6 + 6) / 2 = 6
dist (A,B),E = (dist A,E + dist B,E) / 2 = (6 + 6) / 2 = 6
dist (A,B),F = (dist A,F + dist B,F) / 2 = (8 + 8) / 2 = 8
```

- **Calculate new distance using composite Unit(A,B):**
  - **Distance between a simple Unit and a composite Unit is the average of the distances between the simple Unit and the constituent simple Units of the composite Unit**

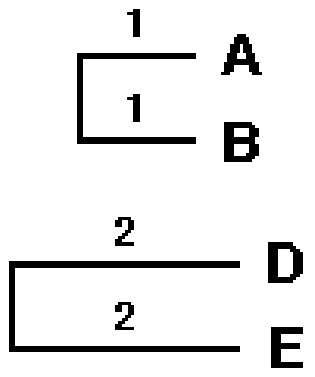
	A	B	C	D	E
B	2				
C	4	4			
D	6	6	6		
E	6	6	6	4	
F	8	8	8	8	8



	A,B	C	D	E
C	4			
D	6	6		
E	6	6	4	
F	8	8	8	8

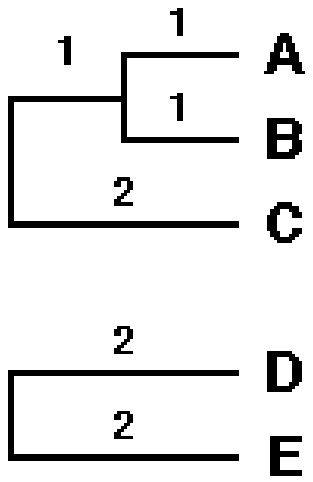
- **Second Iteration**

	A,B	C	D	E
C	4			
D	6	6		
E	6	6	4	
F	8	8	8	8



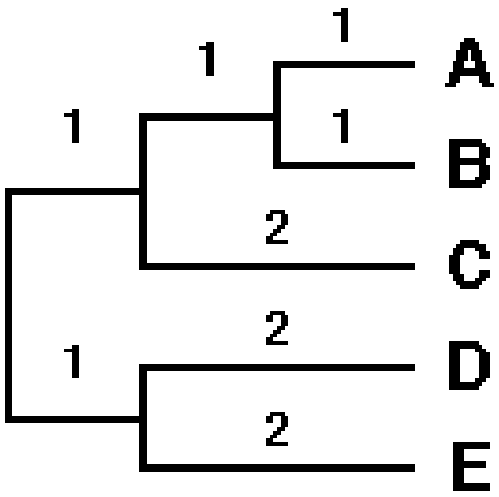
- **Third Iteration**

	A,B	C	D,E
C	4		
D,E	6	6	
F	8	8	8



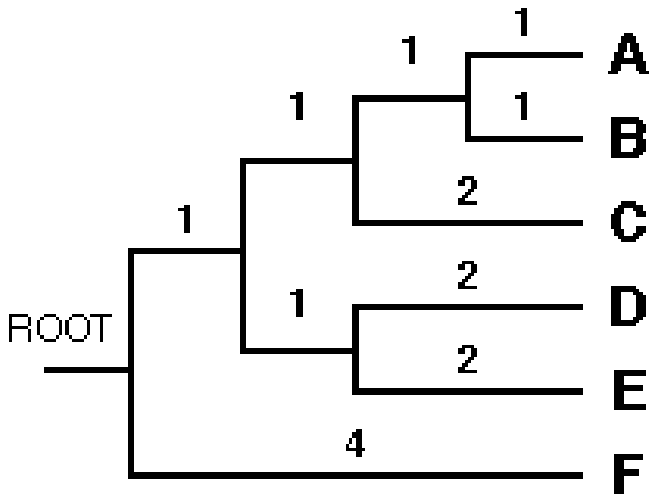
- **Fourth Iteration**

	AB,C	D,E
D,E	6	
F	8	8



- **Fifth Iteration**

	ABC,DE
F	8



Neighbors Joining

- **Proposed by Saitou and Nei in 1987**

Masatoshi Nei, Kyoto prize-winning evolutionary geneticist, dies at 92

BY SUDHIR KUMAR AND GREG FORNIA · 5/19/23

Neighbors Joining

- **Rate (can) varies among branches**
  - **minimum-evolution tree: the tree with the smallest sum of branch lengths**
  - **Pairing sequences based on the effect of the pairing on the sum of the branch lengths of the tree**

Neighbor Joining Algorithm

- For each node i the distance from the rest of the tree is estimated by

$$r_i = \frac{1}{N-2}\sum_{k \neq i} d_{ik}$$

- Choose the node i and j for which $D_{ij} = d_{ij} - r_i - r_j$ is smallest (neighbors)
- Compute branch length from i and j to ij

$$d_{i,(ij)} = \frac{1}{2}d_{ij} + \frac{1}{2}(r_i - r_j), \quad d_{j,(ij)} = \frac{1}{2}d_{ij} + \frac{1}{2}(r_j - r_i)$$

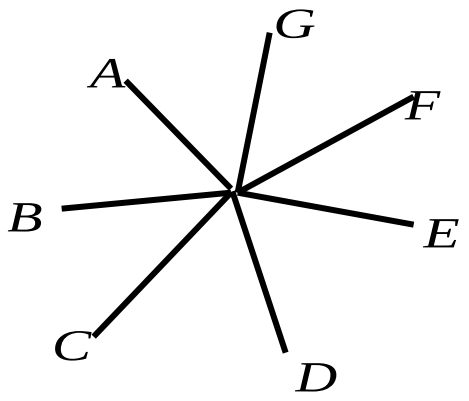
- Compute the distance between new cluster and each other cluster

$$d_{(ij),k} = \frac{d_{ik} + d_{jk} - d_{ij}}{2}$$

Neighbour joining algorithm(1)

	A	B	C	D	E	F	G	r
A	NA							93.0
B	63	NA						80.8
C	94	79	NA					87.0
D	111	96	47	NA				96.0
E	67	16	83	100	NA			84.8
F	23	58	89	106	62	NA		88.0
G	107	92	43	20	96	102	NA	92.0

Start from the star-like tree
Calculate r_i



Neighbour joining algorithm(2)

	A	B	C	D	E	F	G
A	NA	-110.8	-86.0	-78.0	-110.8	-158.0	-78.0
B		NA	-88.8	-80.8	-149.6	-110.8	-80.8
C			NA	-136.0	-88.8	-86.0	-136.0
D				NA	-80.8	-78.0	-168.0
E					NA	-110.8	-80.8
F						NA	-78.0
G							NA

Calculate D_{ij} , D and G are the closest

Calculate the branch lengths of D and G

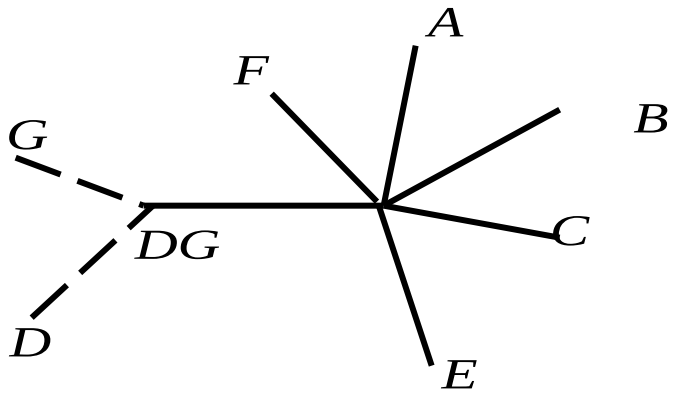
$$d_{d,dg} = \frac{1}{2} d_{d,g} + \frac{1}{2}(r_d - r_g) = \frac{1}{2} * 20 + \frac{1}{2}(96 - 92) = 12$$

$$d_{g,dg} = \frac{1}{2} d_{d,g} + \frac{1}{2}(r_g - r_d) = \frac{1}{2} * 20 + \frac{1}{2}(92 - 96) = 8$$

Neighbour joining algorithm(3)

	A	B	C	E	F	DG	r
A	NA						86.5
B	63	NA					75.0
C	94	79	NA				95.0
E	67	16	83	NA			79.0
F	23	58	89	62	NA		81.5
DG	99	84	35	88	94	NA	100.0

Join D and G, calculate the distances r_i from DG to other nodes



Neighbour joining algorithm(4)

	A	B	C	E	F	DG
A	NA	-98.5	-87.5	-98.5	-145.0	-87.5
B		NA	-91.0	-138.0	-98.5	-91.0
C			NA	-91.0	-87.5	-160.0
E				NA	-98.5	-91.0
F					NA	-87.5
DG						NA

Calculate D_{ij} , C and DG are the closest

Calculate the branch lengths of C and DG

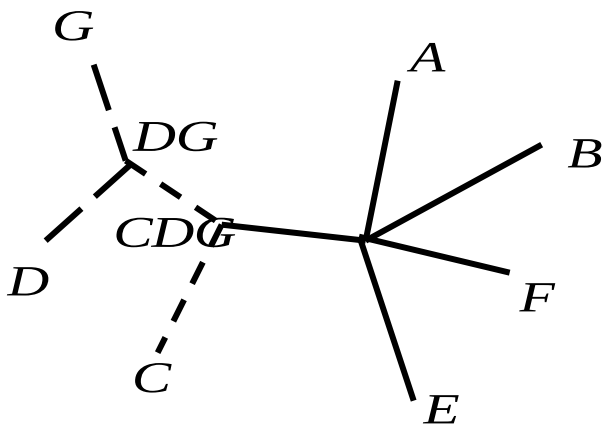
$$c = 15$$

$$dg = 20$$

Neighbour joining algorithm(5)

	A	B	E	F	CDG	r
A	NA					77.33333
B	63	NA				67.00000
E	67	16	NA			71.00000
F	23	58	62	NA		72.33333
CDG	79	64	68	74	NA	95.00000

Join DG and C, calculate the distances r_i from CDG to other nodes



Neighbour joining algorithm(6)

	A	B	E	F	CDG
A	NA	-81.33333	-81.33333	-126.66667	-93.33333
B		NA	-122.00000	-81.33333	-98.00000
E			NA	-81.33333	-98.00000
F				NA	-93.33333
CDG					NA

Calculate D_{ij} , A and F are the closest

Calculate the branch lengths of A and F

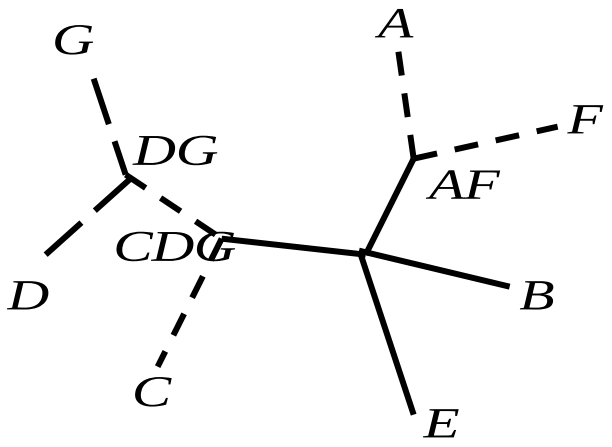
$$a = 14$$

$$f = 9$$

Neighbour joining algorithm(7)

	B	E	CDG	AF	r
B	NA				64.5
E	16	NA			68.5
CDG	64	68	NA		98.5
AF	49	53	65	NA	83.5

Join A and F, calculate the distances r_i from AF to other nodes



Neighbour joining algorithm(8)

	B	E	CDG	AF
B	NA	-117	-99	-99
E		NA	-99	-99
CDG			NA	-117
AF				NA

Calculate D_{ij}, B and E are the closest

Calculate the branch lengths of B and E

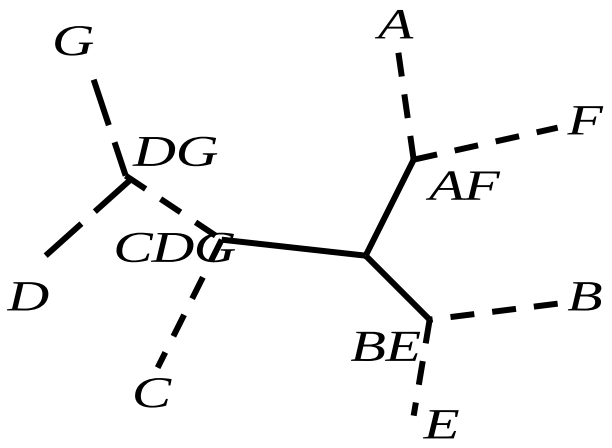
$b = 6$

$e = 10$

Neighbour joining algorithm(9)

	CDG	AF	BE	r
CDG	NA			123
AF	65	NA		108
BE	58	43	NA	101

Join B and E, calculate the distances from BE to other nodes and r_i



Neighbour joining algorithm(10)

	CDG	AF	BE
CDG	NA	-166	-166
AF		NA	-166
BE			NA

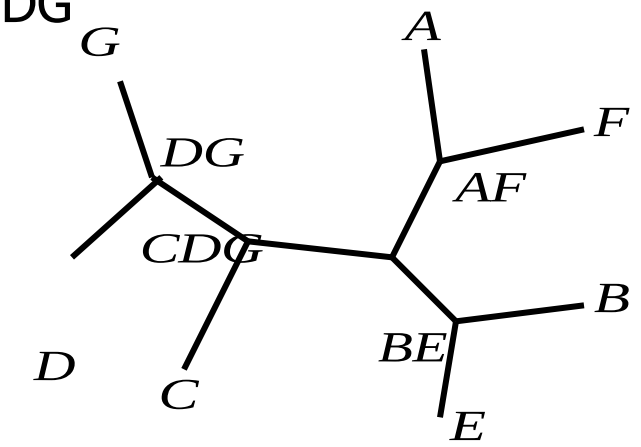
Calculate D_{ij} , AF and CDG are the closest

Calculate the branch lengths of AF and CDG

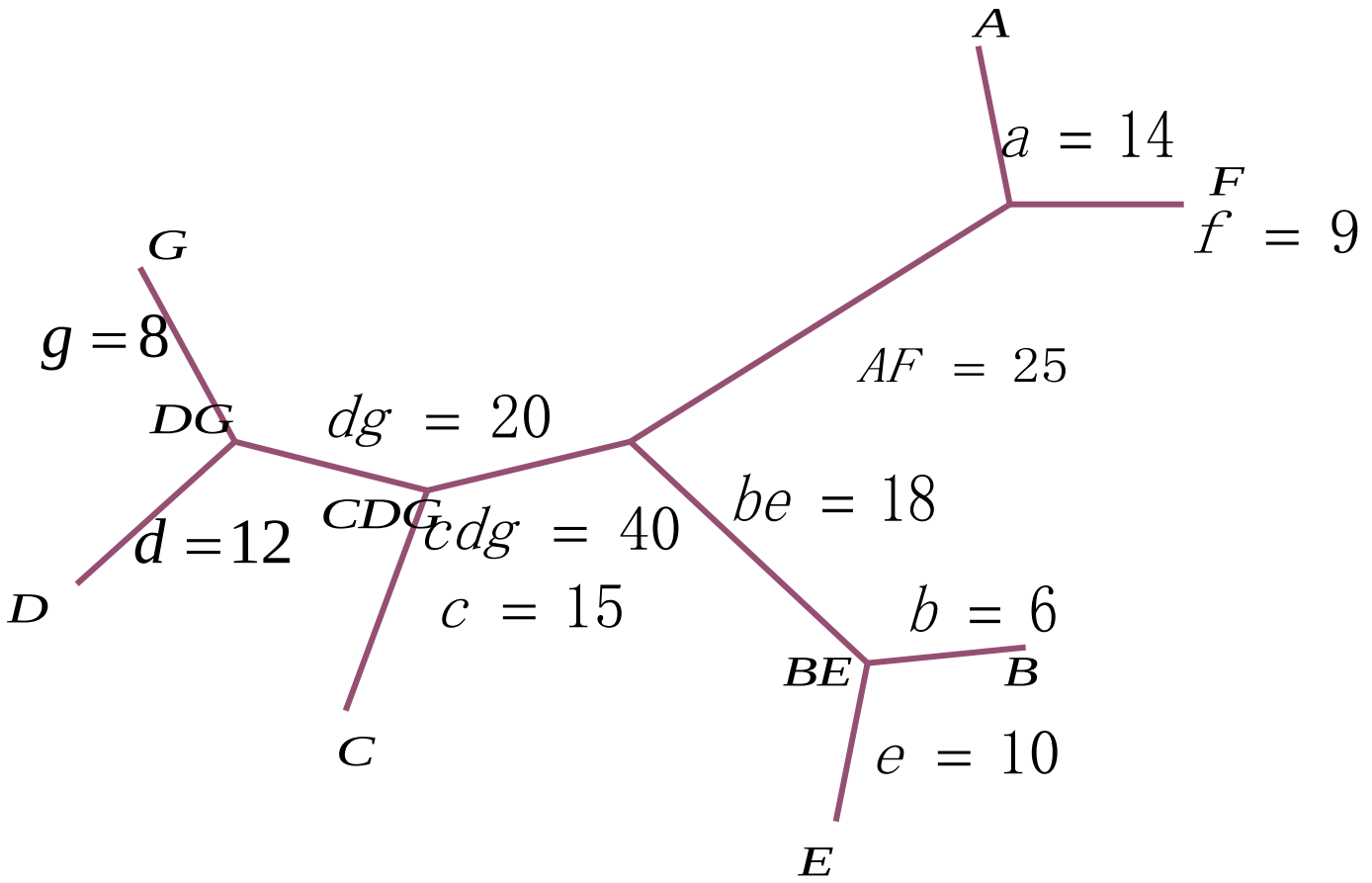
$$\mathrm{cdg} = 40$$

$$\mathrm{af} = 25$$

Join AF and CDG
Finally join BE and CDGAF



Neighbour joining algorithm(11)



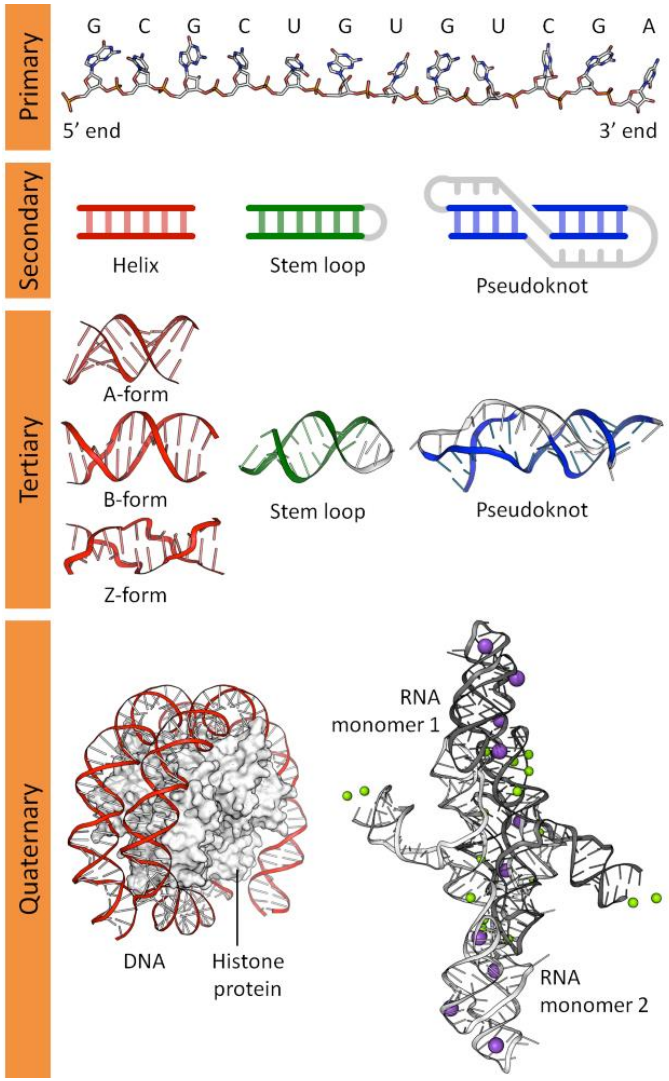
分子演化树构建/应用与拓展

基于距离的方法计算效率高，在简单性和准确性之间提供了良好的平衡，特别适合处理包含成千上万序列的大数据集，已在流行病学、保护生物学等领域获得了广泛应用。

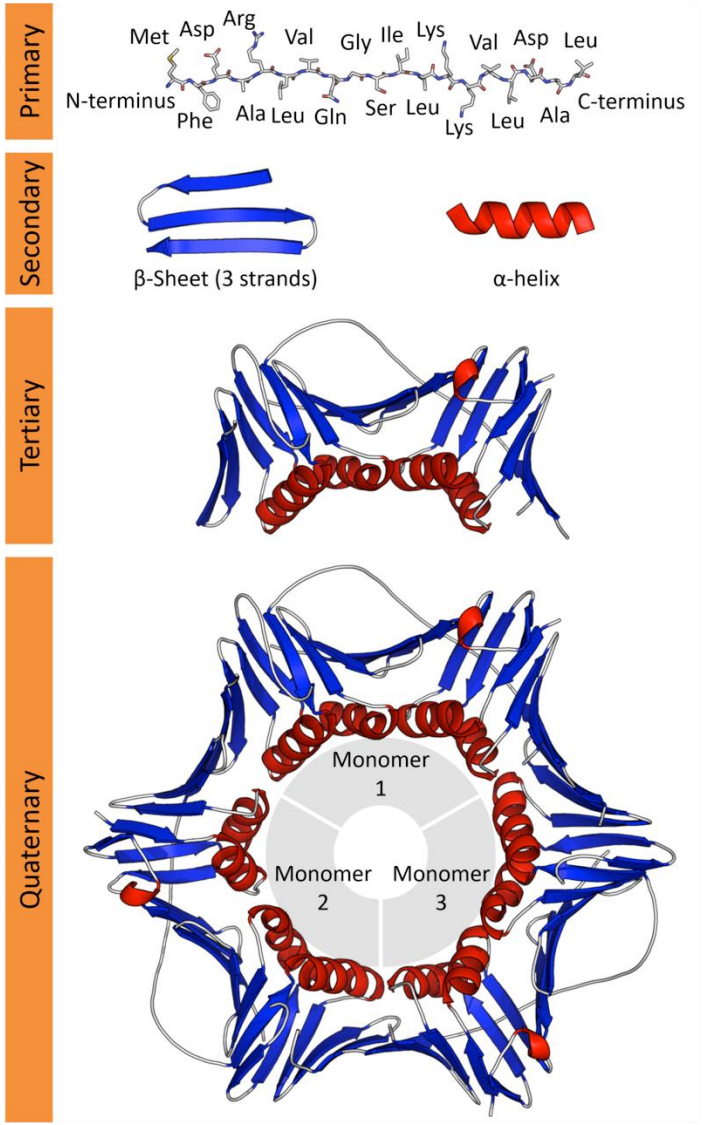
值得一提的是，基于距离的方法将序列数据简化为成对距离，这虽然有效提升了处理效率，但也导致了序列组成等细节的丢失。因此，如果需要对演化细节进行解析，可以考虑采用如基于最大简约法(Maximum Parsimony，MP)、最大似然法(Maximum Likelihood，ML)等基于序列特征的建树方法。

第4节 讨论与展望

生物分子的表示



(Image from: https://upload.wikimedia.org/wikipedia/commons/d/da/DNA_RNA_structure_%28full%29.png)



(Modified from: https://commons.wikimedia.org/wiki/Template:Other_versions/Protein_structure_(full)#/media/File:Protein_structure_(full).png)

生物分子的表示

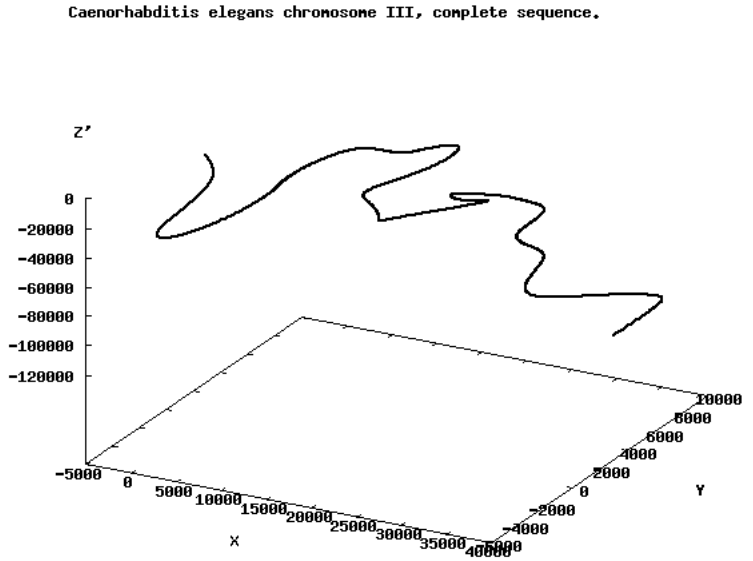
Z curve:

$$x_n = (A_n + G_n) - (C_n + T_n)$$

$$y_n = (A_n + C_n) - (G_n + T_n)$$

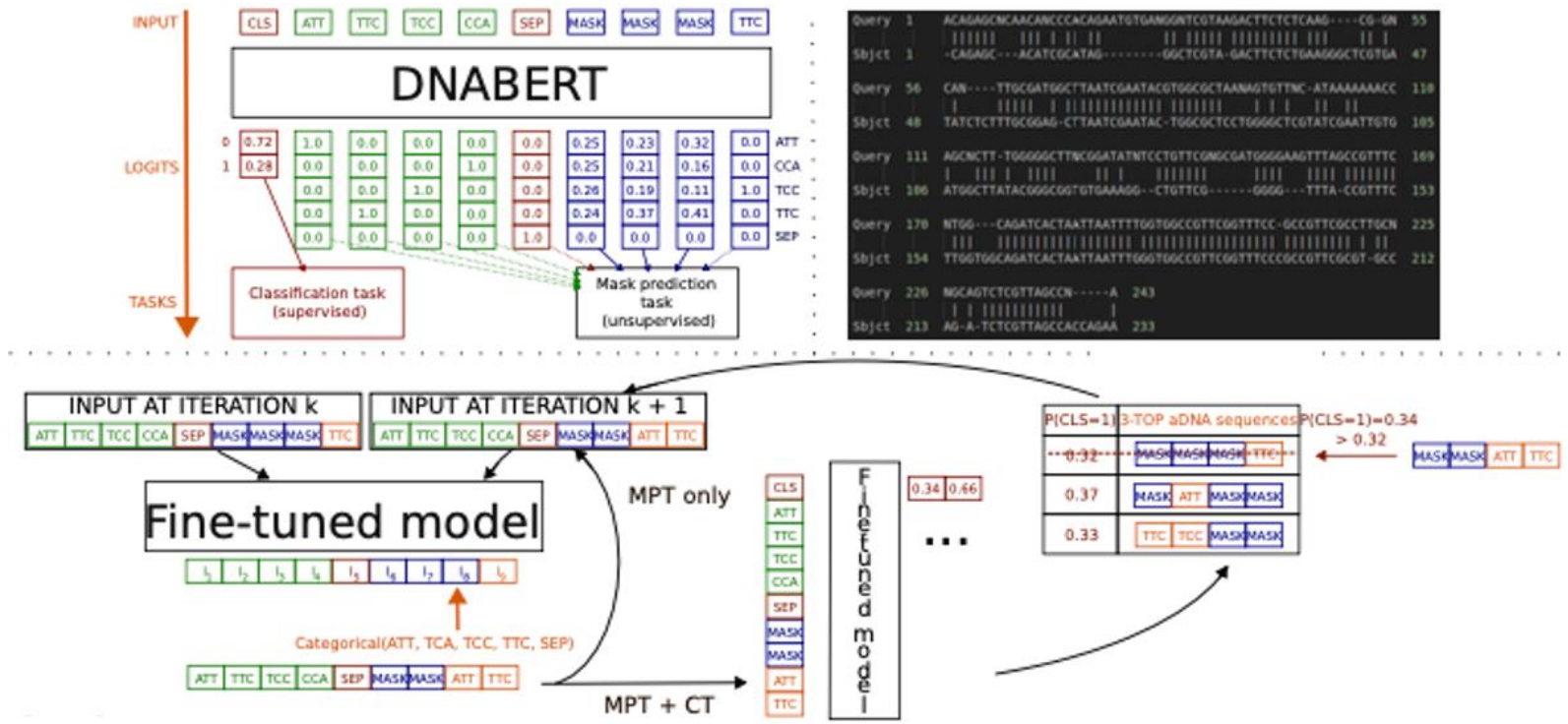
$$z_n = (A_n + T_n) - (C_n + G_n)$$

$$n = 0, 1, 2, \ldots N$$



天津大学张春霆院士于1994年提出DNA序列的Z曲线理论，证明任一DNA序列均可用唯一的一条3维空间曲线表示，从而提示了一条用几何学方法分析DNA序列的新途径。

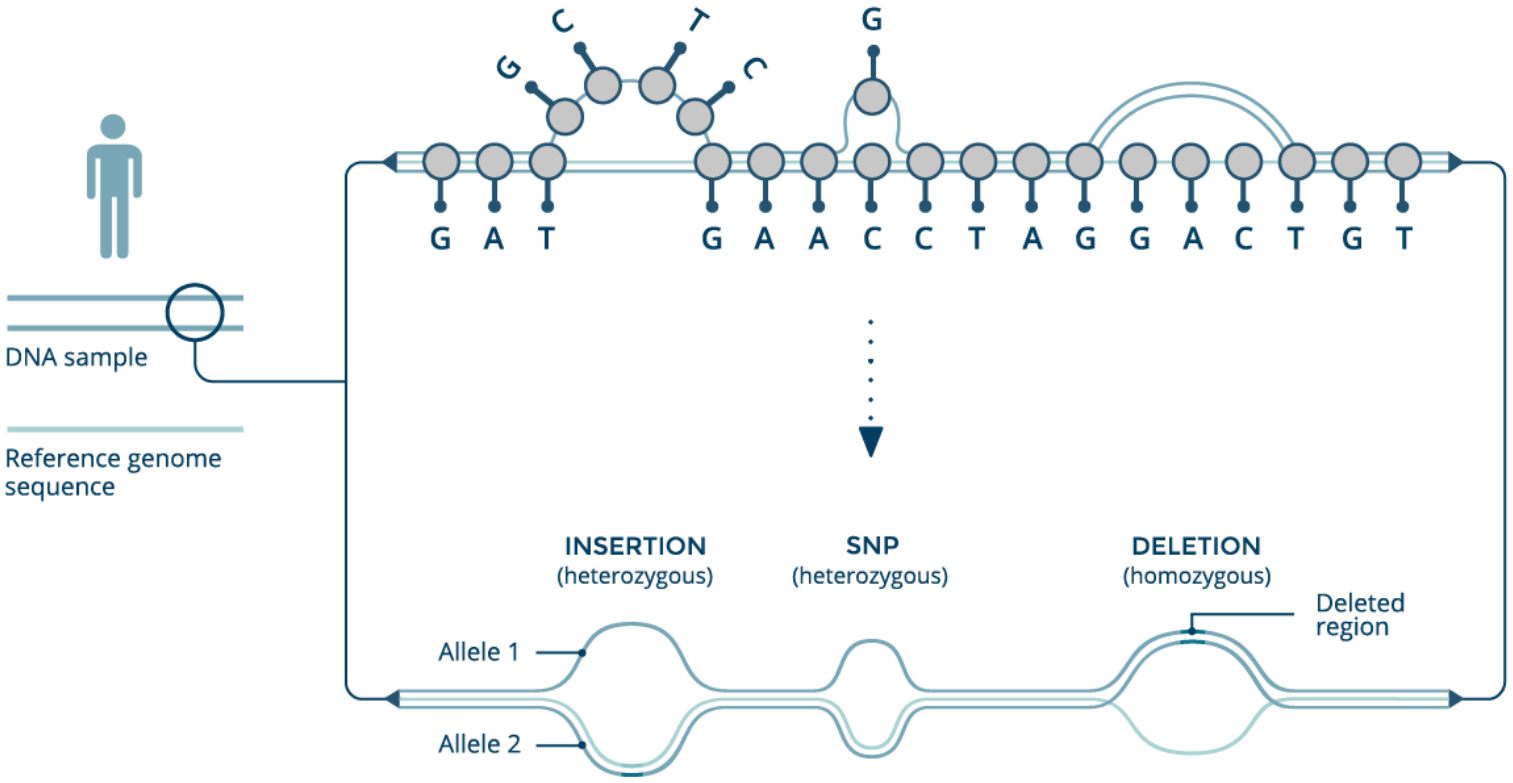
生物分子的表示



在人工智能大语言模型中，可以用one hot等多种编码形式表示特定序列、序列比对及其集合，从而方便后续处理。

(Image modified from http://www.lix.polytechnique.fr/Labo/Theo.BOURY/pdfstorage/AbstractJDSE2023.pdf)

大规模基因组表示与比对



(Image modified from https://www.sevenbridges.com/wp-content/uploads/2016/12/GraphIG-02.png)

- 传统的基因组序列表示方法通常使用线性字符串，这种方法虽然直观，但对于表示遗传变异和复杂的基因组结构有一定的局限性。
- 图形基因组(Graph genome)采取图作为表示，基因组的各个变异和结构差异都可以被建模为图的节点和边。例如，图中的节点可以代表基因组中的不同序列片段或变异，而边则表示这些片段之间的关系或连接方式。通过这种方式，图形基因组能够更全面地描述基因组中的变异、重排、缺失等复杂情况。

本章内容

本章定位：围绕主流生物学数据(序列)，

- **承上：体现生物信息学“面向数据、方法驱动”的特点**
- **启下：为后续章引入关键概念**

编写原则：突出基本观念与方法，桥接经典问题与最新进展

- **从“小(规模)”入手：主要针对生物信息学经典方法，以一个/几个蛋白/核酸序列为主**
- **以“小”见“大”：承上启下，为后续组学章节引入**

谢谢大家

北京大学 高歌

gaog@mail.cbi.pku.edu.cn

https://www.gao-lab.org

感谢关注！

敬请指正！

gaog@mail.cbi.pku.edu.cn