

Homology (同源性) 指两个(或多个)基因、蛋白质或序列之间存在共同祖先所导致的进化关系; 也就是说, 它描述的是“是否源自同一个祖先序列”这一**定性概念**, 而不是相似度的高低。根据分化方式不同, 同源通常进一步分为直系同源 (**orthology**) (由物种形成事件导致的分化, 常更可能保留相似功能) 和旁系同源 (**parology**) (由基因复制事件导致的分化, 功能更可能发生分化)。注意: 同源性不是“多少%的同源”, 序列只能说“同源/不同源”, 而“相似度”(similarity/identity)才是可量化的百分比指标。

• When two sequences are homologous, their sequences usually share significant identity.

• Both have some 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. (直系同源)

• Orthologs are presumed to have similar biological functions: human and rat myoglobin both transport oxygen in muscle cells.

• Human α1-globin is paralogous to α-1 globin (sharing 100% identity). (旁系同源)

• 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.

• 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 (代霍夫模型) **是一类用于描述蛋白质序列随时间发生氨基酸替换的进化模型。它基于 Dayhoff 统计得到的替换概率 (PAM 矩阵), 用一个替换率/概率矩阵来刻画某氨基酸在一定进化距离下变成另一种氨基酸的可能性, 常用于计算序列比对的替换得分和构建系统发育分析中的替换模型。

• 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 others due to similarities in biochemical properties.

• Approach: create a method to reflect the probabilities based on observed substitution.

Accepted Point Mutations (APM, 可接受点突变): 在 Dayhoff/PAM 的构建框架中, APM 指的是在一段高度近缘蛋白序列的可靠比对系统发育 (或祖先状态) 推断基础上, 能够从经验数据中识别出“单个位点氨基酸替换事件”。这里 “accepted” 强调的是这些替换确实在现存序列中被观察到/推断到 (即已固定到可见), 而不是先验地以“自然选择是否接受”作概念定义。

• 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.

Frequency of AA (氨基酸频率): 指在用建模的序列集合中, 20 种氨基酸各自出现的背景比例 (边际概率分布)。它刻画随机背景下某氨基酸出现的基线概率, 是构建替换模型与进行 log-odds 打分的关键背景项。

Relative Mutability of AA (氨基酸相对可变性): 衡量不同氨基酸作为起始残基时被替换出去的相对倾向 (相对速率/易变程度) 的参数或指标。它用于刻画“哪些氨基酸更保守、哪些更易变”, 从而帮助确定替换过程的速率结构, 并影响替换概率矩阵与打分矩阵中各类替换的相对权重。

the relative mutability of the amino acid (m_i/f_i)

• Number of times each amino acid was observed to mutate (m_i)

• Overall frequency of occurrence of that amino acid (f_i)

Biology: the less mutable residues probably have important structural or functional roles in proteins.

Mutation Probability Matrix (替换概率矩阵): 一个 20×20 的概率矩阵, 元素 M_{ij} 表示在给定进化距离 (或时间) 下, 氨基酸 j 经过演化后变为 i 的条件概率; 对角线元素表示保持不变的概率。该矩阵是 Dayhoff 模型中描述氨基酸替换过程的核心对象, 并可进一步导出对所用的替换打分矩阵。

• 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 units of years.

PAM1: 指以 1 个 PAM 距离单位标注得到的替换概率矩阵 (及其对应的打分矩阵), 对应“非常短”的演化距离, 通常用在近缘序列的替换矩阵与比对评分体系的基础构建。

PAM250: Dayhoff PAM 模型中表示 250 PAM 进化距离的氨基酸替换矩阵 (概率矩阵/打分矩阵)。它把 PAM1 作为一步马尔可夫替换过程, 通过矩阵乘积 (连乘) 累积得到, 从而在远距离下自动计入同一位置的多次替换 (multiple hits)。在该尺度上序列通常仅有 20% identity (约五分之一位点不变), 例如原为 A 的位点在 PAM250 上仍为 A 的概率可约 13%。

• PAM0: no amino acids have changed.

• PAM00: an equal likelihood of any amino acid being present that approximate the background probability for the frequency occurrence of each amino acid.

Relatedness Odds Matrix (相关性优势矩阵): 对每一对氨基酸 i, j , 比较同源演化模型中 i 与 j 在对齐位置配对出现的概率/与在随机背景分布下 i 与 j 偶然配对的概率的比值所形成的矩阵。该矩阵直观反映某一配对相对随机背景是更常见还是更罕见, 是从概率模型走向比对打分的中间桥梁。

$R_{ij} = \frac{M_{ij}}{f_i f_j}$ 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. The relatedness odds matrix in Equation (3.3) may also be expressed as $R_{ij} = M_{ij}/f_j$, where R_{ii} is the relative odds ratio.

Log-Odds Scoring Matrix The logarithmic form of the relatedness odds matrix is called a log-odds matrix.

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

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

• The values for M_{ij} (q_{ij} , 叫作“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.

做什么

用到的符号/计算 (按图中记号; 列=from j , 行=to i)

统计 APM 替换计数 近缘蛋白的经验替换计数: C_{ij} ($i \neq j$) 表示 from j to i 的 accepted substitution counts; 设 $C_{jj} = 0$ 。

估计背景氨基酸频率统计每种氨基酸出现次数 n_i , 得 $f_i = \frac{n_i}{\sum_k n_k}$, 且 $\sum_i f_i = 1$ 。

计算相对可变性 (易对每个原始氨基酸 j : 总离开量 $m_j = \sum_{i \neq j} C_{ij}$ 。

变程度) 相对可变性常写 $M_j^{\text{rel}} = \frac{m_j}{f_j}$ (并常再归一化)。

构造末缩放的一步 令 $\hat{M}_{ij} = \frac{C_{ij}}{\sum_{k \neq j} C_{kj}} = \frac{C_{ij}}{m_j}$ ($i \neq j$), 并置 $\hat{M}_{jj} = 0$ 。

“替换偏好”矩阵 每一行 (固定 j) 在 $i \neq j$ 上求和为 1。

$\frac{C_{ij}}{m_j}$ (读为 M_j^{rel})

得到 PAM1 概率矩阵 (全局尺度 δ 使其对应 1 PAM (约 1% 期望变化)。更严谨的做法:

阵 (含不变概率) $M_{ij}^{(1)} = \delta M_j^{\text{rel}} \hat{M}_{ij}$ ($i \neq j$); PAM1 由求得

$M_{jj}^{(1)} = 1 - \sum_{i \neq j} M_{ij}^{(1)}$ (保证每列和为 1)。 $\sum_{ij} f_j M_{ij} = 0$ 其中 $M_{ij} = \frac{C_{ij}}{f_j}$

外推得到 $PAM_n / PAM250$: 马尔可夫累积: $M^{(n)} = (M^{(1)})^n$ (如 PAM250: $M^{(250)} = (M^{(1)})^{250}$)。

计算 relatedness (A) 方向性 (与图一致) : 以 j 为“原始”、 i 为“终态”, 常写 odds 与 log-odds

打分矩阵 $R_{ij}^{(n)} = \frac{M_{ij}^{(n)}}{f_i}$,

$S_{ij}^{(n)} = \text{round}(\beta \log_b R_{ij}^{(n)})$ (Dayhoff 常取 $\beta = 10, b = 10$)。

(B) 实用对称版本 (做比对更常用) : 先对联合概率做对称化 (使

$S_{ij} = S_{ji}$), 再取 log-odds。

BLOSUM (BLOCKS Substitution Matrix): 一类关于蛋白质序列比对的替换打分矩阵, 由 Henikoff & Henikoff 基于保守片段 (blocks) 的多序列比对统计得到。其核心做法是在同源蛋白的保守区段中统计氨基酸配对共现频率, 并与背景频率比较, 构建 **log-odds** 得分 $S_{ij} = \log \frac{q_{ij}}{p_{ij}}$ (再缩放取整), 反映 i 与 j 在同源区中是否比随机更常出现。BLOSUM 的编号表示构成它的聚类区间 (如 BLOSUM62, 使用 62% 相似性阈值把过于相近的序列先聚类以减少偏差); 一般来说, 编号越大更适合远缘序列。

$S_{ij} = 2 \times \log_2 \left(\frac{q_{ij}}{p_{ij}} \right)$ q_{ij} 是“同源/真实对齐模型”(target model) 下, 氨基酸 i 与 j 在同一对齐列中配对出现的概率; p_{ij} 是“随机背景模型”(null / background) 下, i 与 j 偶然配对出现的概率。

• q_{ij} 通常假设两侧残基独立, 先求单个氨基酸的背景频率 p_i , 再得到 p_{ij}

维度 **PAM 矩阵** **BLOSUM 矩阵**

构建数据来源 由近缘蛋白的全长/可靠对比中推断的替换 (Dayhoff 的 APM 统计) 由蛋白家族的保守区段 blocks (局部保守片段) 中直接统计配对替换

核心思想 先进化替换模型 (马尔可夫过程), 再用模型 直接用数据统计得到经验 log-odds, 不依赖外生成不同距离的矩阵推的进化距离模型

距离/外推 PAM1 为基准, 再用矩阵外推: $P(n) = P(1)^n$ (显式考虑 multiple hits) 不做从 1 步到 n 的矩阵幕外推; 每个编号是一套独立统计结果

编号含义 PAM_n 表示进化距离约为 n PAM (n 越大越远缘) $BLOSUM_x$ 表示聚类区间 (%identity) (x 越大越近缘; x 越小越远缘)

适用序列相似度趋势 PAM 数值越大 → 适合更远缘 (如 PAM250) BLOSUM 数值越小 → 适合更近缘 (如 BLOSUM45); 数值越大 → 更近缘 (如 BLOSUM80)

“背景频率”处理 依赖估计的氨基酸背景频率与替换过程参数 (可变性等) 背景频率通常从 blocks 统计中导出, 并用于 log-odds

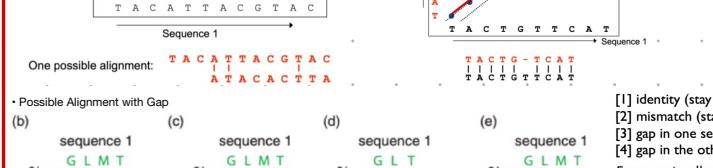
典型使用场景 强调“进化距离”的建模/推断; 也用于比对打分 蛋白数据库检索与比对中非常用 (如 BLASTP 常用 BLOSUM62)

• 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)

Longest common subsequence

Insertions / Deletions in a Dotplot



One possible alignment: T A C T G T A C G T A C

Possible Alignment with Gap

(b) sequence 1 G L M T sequence 2 G L M sequence 1 G L M T sequence 2 G L V (c) sequence 1 G L M T sequence 2 G L M sequence 1 G L T sequence 2 G L V (d) sequence 1 G L M T sequence 2 G L M sequence 1 G L V sequence 2 G L V (e) sequence 1 G L M T sequence 2 G L M sequence 1 G L T sequence 2 G L V

[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)

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

Rigorous algorithms = Dynamic Programming

- Needleman-Wunsch (global; 1970)

- Smith-Waterman (local; 1981)

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

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

(a) Sequence 2 F M D T P L N E

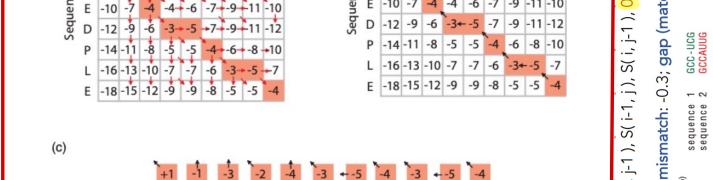
(b) Score = Max $\left\{ F(i-1, j-1) + s(x_i, y_j) \right. \left. , F(i-1, j) - \text{gap penalty} , F(i, j-1) - \text{gap penalty} \right\}$

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

② To determine the path through the matrix that maximizes the score.

(c) Sequence 2 F M D T P L N E

F(i-1, j-1) + gap penalty F(i, j-1) - gap penalty F(i, j) - gap penalty



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

③ To determine the path through the matrix that maximizes the score.

(a) Sequence 2 F M D T P L N E

(b) Sequence 2 F M D T P L N E

(c) Sequence 2 F M D T P L N E

④ To determine the path through the matrix that maximizes the score.

⑤ To determine the path through the matrix that maximizes the score.

⑥ To determine the path through the matrix that maximizes the score.

⑦ 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.

• The final alignment is derived from the path.

⑧ 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.

• The final alignment is derived from the path.

⑨ 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.

• The final alignment is derived from the path.

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

T-Coffee 的三步流程:

1. 构建成对比对库:
- 在第一步, T-Coffee 会先计算所有序列的成对比对 (pairwise alignments), 并为每一对比对分配权重。
- 比对权重反映了比对的相似度 (similarity)。如果两个序列在比对中匹配较好, 它们的权重就较高, 反之则较低。
- 例如, 在图中的 (A) 部分, Seq x 和 Seq y 的比对权重为 88, 而 Seq x 和 Seq v 的比对权重为 77。权重高的比对代表了两个序列之间的较高相似度。

圆圈标注解释:

- 每行代表两个序列的比对, 且显示了比对结果 (如匹配、错配和缺口)。每对比对都有一个权重值, 权重越高表示比对质量越高。

2. 一致性比对 (Consistency Alignment):

- 在第二步, T-Coffee 采用一种一致性比对方法。对于每一对比对 (如图 A 和 B), 它会考虑第三个序列 C 来形成新的比对 A-C-B, 即使用 C 作为中介。增强 A 和 B 之间比对的一致性。
- 这种一致性比对的核心思想是: 如果 C 同时与 A 和 B 都具有较高的相似度, 那么 A 和 B 之间的比对就更有可能是准确的。因此, T-Coffee 会用 C 作为桥梁, 帮助提升 A-B 比对的精确性。

3. 渐进式比对 (Progressive Alignment):

- 最后, T-Coffee 会使用树结构进行渐进式比对。在这个阶段, 所有之前的比对结果 (包括由中介序列带来的信息) 会被整合到一个加权的比对树中。
- T-Coffee 会根据成对比对库的进展信息, 通过树结构逐步将各个序列合并, 最终完成多序列比对。
- 在进行渐进式比对时, T-Coffee 会使用权重信息来确定哪个序列应该先对齐, 以提高整体比对的一致性和精度。

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

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.

In 1977, Frederick Sanger developed Sanger Sequencing via chain-termination method, and sequenced the first complete genome: one of bacteriophage ϕ X174.

Sanger Sequencing

1.DNA polymerase extension with dNTP → DNA Replication and Chain Termination.

2.Extension terminated when ddNTP incorporated.

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

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 MSA 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

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)

Why Realign Between Subgroups?

- 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.
- 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.
- 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.
- 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.
- 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.

特性/算法

原理	Star Alignment	Clustal W	T-Coffee
基于将一个中心序列与其他序列渐进式比对方法, 首先进行成对比对, 然后逐步合併序列, 生成最终的比对结果。	- 选择一个中心序列。 - 将其他序列与中心序列对比。 - 逐步合併最相似的序列, 直至所有序列对比完成。	- 计算所有序列的成对比对。 - 对序列进行成对比对。 - 合并所有比对结果。	- 计算所有序列的成对比对。 - 对序列进行成对比对。 - 逐步合併最相似的序列, 直至所有序列对比完成。
大致思路	- 选择一个中心序列。 - 将其他序列与中心序列对比。 - 逐步合併最相似的序列, 直至所有序列对比完成。	- 计算所有序列的成对比对。 - 对序列进行成对比对。 - 合并所有比对结果。	- 选择一个中心序列。 - 将其他序列与中心序列对比。 - 逐步合併最相似的序列, 直至所有序列对比完成。
计算顺序	- 计算所有序列的成对比对。 - 逐步合併最相似的序列, 直至所有序列对比完成。	- 计算所有序列的成对比对。 - 逐步合併最相似的序列, 直至所有序列对比完成。	- 计算所有序列的成对比对。 - 逐步合併最相似的序列, 直至所有序列对比完成。
计算方法	- 成对比对 → 逐步合併比对结果, 成对比对 → 渐进式合併比对 - 进式合併比对 → 对比结果	- 成对比对 → 逐步合併比对结果, 成对比对 → 渐进式合併比对 - 进式合併比对 → 对比结果	- 成对比对 → 逐步合併比对结果, 成对比对 → 渐进式合併比对 - 进式合併比对 → 对比结果
优点	- 结构简单, 易于实现。 - 对小数据集适用。	- 实现简单, 适用于一般数据集。 - 比较快速。	- 在大部分情况下比 Clustal W 快速, 尤其适用于大数据集。 - 提供了比 Clustal W 更高的比对量应用。
缺点	- 对大数据集的处理不够高效。 - 精度较低, 尤其是对大差异序列, 太耗时。 - 列比对时容易引入错误。	- 计算量较大, 速度较慢。 - 对大数据集或高通量数据的“扩”和“比”优化, 较为耗时。 - 对结构数据集的支持不如 T-Coffee。	
适用场景	- 小型数据集 (10–50个序列)。 - 大数据集 (10–50个序列)。 - 对比相似的序列。	- 适用于需要高精度比对的场 景。 - 适用于常规序列对比任务。 - 对于处理序列间差异较大的情况, 需要较快速比对速度的高通量应 用。	
处理差异大的序列	较差	良好	
速度	中等	慢	
扩展性	差		
蛋白质结构数据支持	不支持		
系统发育分析	支持 (Expresso 插件)		
RNA二级结构分析	不适用		
高通量应用	不适用		

Diagram of the MUSCLE algorithm flow:

- 1.1 k-mer counting: Unaligned sequences are processed to create a k-mer distance matrix D1.
- 1.2 UPGMA: TREE1 is constructed using the k-mer distance matrix D1.
- 1.3 progressive alignment: MSA1 is generated using TREE1.
- 2.1 compute %ids from MSA1: The percentage identity is calculated from MSA1.
- 2.2 UPGMA: TREE2 is constructed using the Kimura distance matrix D2.
- 2.3 progressive alignment: MSA2 is generated using TREE2.
- 3.1 delete edge from TREE2: Subtrees are deleted from TREE2.
- 3.2 compute subtree profiles: Subtree profiles are computed for MSA3.
- 3.3 re-align: MSA3 is re-aligned.
- 3.4 SP score better?: If the SP score is better, save the alignment; otherwise, repeat the process.
- 3.5 repeat: The process repeats until no improvement is made.
- Group-to-group alignment: The final alignment is grouped into subalignments.
- Tree-dependent partitioning: The tree is partitioned into subalignments based on the guide tree.
- Divide into subalignments: The final alignment is divided into subalignments.

Diagram of the MUSCLE algorithm flow:

- 1.1 k-mer counting: Unaligned sequences are processed to create a k-mer distance matrix D1.
- 1.2 UPGMA: TREE1 is constructed using the k-mer distance matrix D1.
- 1.3 progressive alignment: MSA1 is generated using TREE1.
- 2.1 compute %ids from MSA1: The percentage identity is calculated from MSA1.
- 2.2 UPGMA: TREE2 is constructed using the Kimura distance matrix D2.
- 2.3 progressive alignment: MSA2 is generated using TREE2.
- 3.1 delete edge from TREE2: Subtrees are deleted from TREE2.
- 3.2 compute subtree profiles: Subtree profiles are computed for MSA3.
- 3.3 re-align: MSA3 is re-aligned.
- 3.4 SP score better?: If the SP score is better, save the alignment; otherwise, repeat the process.
- 3.5 repeat: The process repeats until no improvement is made.
- Group-to-group alignment: The final alignment is grouped into subalignments.
- Tree-dependent partitioning: The tree is partitioned into subalignments based on the guide tree.
- Divide into subalignments: The final alignment is divided into subalignments.

MLP (多层感知机)

MLP是最基础的一类前馈神经网络，由输入层、若干隐藏层和输出层组成，相邻两层之间通常是“全连接”。它通过在每一层做线性变换（加权求和加偏置）再接非线性激活函数，从而拟合复杂的非线性关系。MLP常用于表格数据、简单分类/回归等任务，也是很多更复杂网络结构的“积木”。

RNN (循环神经网络)

RNN会处理序列数据（如文本、语音、时间序列），其核心特点是当前时刻的输出不仅依赖当前输入，还依赖前一时刻的隐状态（相当于“记忆”）。这种循环连接使它能建模上下文与时序依赖，但基础RNN容易出现梯度消失/爆炸问题。因此实践中常用LSTM、GRU等改进结构来增强长期依赖建模能力。

CNN (卷积神经网络)

CNN主要用于图像等具有局部结构的数据，通过卷积核在局部区域滑动提取特征，具备“局部连接、权重共享”的特点，因此参数更少、对平移等变化更稳定。它通常会配合池化等操作逐步扩大感受野，从低级别到高级逐级抽象，也广泛用于视频、语音识别、部分NLP等场景。

Diffusion (扩散模型)

扩散模型是一类生成模型，常用于高质量图像生成。其思想通常是：先把真实数据逐步扩散到接近纯噪声（前向过程），再训练模型学习“去噪”或“反向还原”（后向过程）。推理时从随机噪声出发，通过多次去噪生成样本，质量高但传统采样步数较多。近年来也有加速采样的方法。

LLM (大语言模型)

LLM是以海量文本数据训练的语言模型，常基于Transformer架构，目标是学习语言的统计规律，从而实现文本生成、问答、总结、翻译、代码辅导等功能。它通常通过自监督任务（如预测下一个词）进行预训练，再通过指令微调、对齐（如RLHF等）提升“按人类意图回答”的表现。LLM擅长语言理解与生成，但也可能出现幻觉、偏差与对新鲜事实不了解等问题。

full connection (全连接层 / Fully Connected, FC)

全连接层指上一层的每个神经元都与下一层的每个神经元相连，输出是输入向量的线性变换再加偏置： $y = Wx + b$ ，表达能力强，但参数量随输入维度快速增长，容易过拟合、计算开销巨大。因此在CNN中常用于最后的分类头，在现代模型里也常与正则化、Dropout等一起使用。

activation function (激活函数)

激活函数用来给神经网络引入非线性，使网络能拟合复杂的非线性关系；若没有激活函数，多层线性变换仍等价于单层线性模型。常见激活函数有ReLU、Sigmoid、Tanh、GELU等，它们影响梯度传播、收敛速度和最终性能，是网络设计的重要组成部分。

robust (鲁棒性)

鲁棒性描述模型在输入扰动、噪声、缺失、分布偏移或对抗样本等不理想条件下仍能保持稳定性的能力。鲁棒的模型不容易因小的变化输出大幅错误结果，通常可通过数据增强、正则化、对抗训练、稳健损失函数、模型集成等手段提升。

泛化能力 (generalization)

泛化能力是指模型在未见过的新数据（测试集/真实环境数据）上仍能保持良好表现的能力。高泛化意味着模型学到的是数据背后的规律，而不是训练数据的偶然噪声，泛化与模型复杂度、数据量与质量、正则化方法、训练策略等密切相关。

过拟合 (overfitting)

过拟合是指模型在训练集上表现很好，但在测试集或新数据上表现变差，原因通常是模型过于复杂或训练过久，记住了训练数据中的噪声和偶然性。常见应对方法包括增加数据、数据增强、正则化（L2、Dropout）、早停（early stopping）、降低模型复杂度等。

训练集 (training set)

训练集是用来学习模型参数的数据集合，训练过程中模型通过最小化训练集上的损失函数来更新权重。通常还会配套验证集用于调参/早停，以及测试集用于最终评估，以避免“在测试集上调参”的评估偏差。

loss function (损失函数)

损失函数用于衡量模型预测与真实标签之间的差距，是训练优化的直接目标。不同任务对应不同损失：回归常用MSE/MAE，分类常用交叉熵，生成/对比学习也有各自损失，选择合适的损失函数会显著影响训练稳定性及最终效果。

梯度下降 (gradient descent)

梯度下降是一类优化方法，通过计算损失函数对参数的梯度，沿着“负梯度方向”更新参数以减少损失。实际训练常用随机梯度下降 (SGD) 或小批量梯度下降 (mini-batch)，并搭配动量、学习率调度或自适应优化器（如Adam）来提升收敛速度与稳定性。

backpropagation (反向传播)

反向传播是计算神经网络梯度的高效算法，本质是链式法则在计算图上的系统应用。它先做前向传播得到输出与损失，再从损失开始逐层向后计算各层参数的梯度，为梯度下降等优化算法提供更新方程，是深度学习训练的核心机制。

MSE (均方误差, Mean Squared Error)

MSE是回归任务常用的损失函数，定义为预测值与真实值差的平方的平均： $\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ ，平方项会对较大的误差给予更高惩罚，因此对异常值更敏感。它在高斯噪声假设下与最大似然估计有对应关系，计算与求导也较方便。

forward (前向传播)

前向传播是指把输入数据从网络的输入层依次经过各层的线性变换与激活函数，最终得到模型输出（预测值）的过程。在训练时，前向传播还会用于计算损失函数，为后续反向传播计算梯度提供基础。

optimizer (优化器)

优化器是根据损失函数的梯度来更新模型参数（权重、偏置）的算法组件，决定“怎么走”才能让损失下降。常见优化器有SGD、带动量的SGD、RMSprop、Adam等，它们主要区别在于是否使用动量、是否自适应调整学习率以及对梯度历史信息的利用方式。

LSTM (长期短期记忆)

LSTM是RNN的改进结构，通过引入门控机制（遗忘门、输入门、输出门）来控制信息的保留与丢弃，从而缓解普通RNN的梯度消失问题，提升对长期依赖的建模能力。它常用于序列任务，如语言建模、机器翻译、时间序列预测等。

自编码器 (Autoencoder, AE)

机器学习 (Machine Learning)

机器学习是一类让计算机从数据中学习规律并进行预测/决策的方法体系，强调用数据驱动的方式替代手工编写规则，它包含监督学习、无监督学习、半监督学习、强化学习等多种范式，常见模型有线性模型、决策树、SVM、神经网络等。

深度学习 (Deep Learning)

深度学习是机器学习的一个分支，核心是使用具有多层次的神经网络从数据中自动学习分层特征表示，相比传统方法，它在大数据与算力支持下能在视觉、语音、NLP等领域取得更强性能，但通常需要更多数据、计算资源与更复杂的训练技巧。

神经网络 (Neural Network)

神经网络是一类受生物神经系统启发的函数逼近模型，由大量“神经元”通过权重连接构成。每个神经元对输入做加权求和并经过非线性激活，从而可以拟合复杂的非线性关系；根据连接方式不同可形成MLP、CNN、RNN、Transformer等多种结构。

回归任务 (Regression)

回归任务的目标是预测连续数值（如房价、温度、销售额等）。模型输出通常是实数，常用评价指标/损失包括MSE、MAE、RMSE、 R^2 等，重点在于预测值与真实值在数值上的误差大小。

分类任务 (Classification)

分类任务的目标是把样本分成离散类别（如猫/狗、正常/异常、多分类标签等）。模型通常输出各类别的概率（如Softmax），常用损失是交叉熵，评价指标包括准确率、精确率、召回率、F1、AUC等。

梯度 (Gradient)

梯度是损失函数对参数的偏导数组成的向量（或张量），指示在当前参数发生变化时损失的方向。训练时通常沿负梯度方向更新参数以降低损失，因此梯度的大小与方向直接影响学习速度与稳定性。

动量法 (Momentum)

动量法是在梯度下降中引入“速度”的思想：更新方向不仅看当前梯度，还综合过去一段时间的梯度累积，从而减少震荡、加速在一致方向上的前进。直观上小球在坡面滚动，能跳出浅小局部震荡，提高收敛效率。

RMSprop

RMSprop是一种自适应学习率优化器，会对每个参数维护梯度平方的指数滑动平均，并用它来缩放当前梯度，使学习率能随参数与训练阶段自动调整。它在处理非平稳目标、RNN训练等场景中常表现稳定，能缓解不同参数尺度差异带来的训练困难。

Adam

Adam结合了动量（梯度的一阶矩估计）与RMSprop风格的自适应缩放（梯度平方的二阶矩估计），对每个参数都计算“带偏置修正”的一阶/二阶矩，从而实现稳定且收敛较快的训练。它是深度学习中最常用的默认优化器之一，对超参数相对不敏感但并非所有任务都最优。

欠拟合 (Underfitting)

欠拟合是指模型过于简单或训练不足，连训练集上的规律都没学好，表现为训练误差和测试误差都较高。常见原因包括模型容量不足、特征表达不够、训练批次数太少或正则化过强；解决通常是增大模型、训练更多、改进特征或降低正则强度。

early stopping (早停)

早停是一种防止过拟合的训练策略：在训练过程中监控验证集指标，当验证集性能在若干轮（patience）内不再提升就停止训练，并通常回滚到验证集表现最好的那一轮参数。它相当于用“何时停止训练”来做正则化。

regularization (正则化)

正则化是通过限制模型复杂度来提升泛化能力的一类方法，常见做法包括在损失中加入参数范数惩罚（L1/L2 weight decay）、数据增强、Dropout、早停、标签平滑等。目标是减少模型对训练集噪声的依赖，避免过拟合。

dropout (随机失活)

Dropout是一种正则化方法：训练时以一定概率随机“丢弃”部分神经元输出（置零），迫使网络不能过度依赖某些特征，从而提升泛化能力。推理时不再丢弃，而是使用完整网络并进行相应的尺度校正（或使用框架默认的等价处理）。

Transformer

Transformer是以注意力机制为核心的序列建模架构，摒弃了RNN的循环结构，主要依赖自注意力并平行处理序列，因此训练效率高、长距离依赖建模能力强。它已成为NLP的主流骨架，并扩展到视觉（ViT）、多模态等领域。

embedding (嵌入表示)

Embedding是把离散符号（如词、字词、类别ID）映射到连续向量空间的表示方式，使模型能通过向量运算学习语义相似性与组合关系。训练中embedding通常是指学习参数；在LLM中，输入token会先通过embedding转成向量序列，再送入Transformer。

self-attention (自注意力)

自注意力是一种让序列中每个位置根据与其他位置的相关性来“聚合信息”的机制：每个token都会对序列里所有token分配权重并加权求和，从而获得上下文相关的表示。它能灵活捕捉长距离依赖，且计算可并行，是Transformer的关键组件。

监督学习

使用对应的训练样本 (x, y) 学习映射 $f(x) \approx y$ ，训练目标是最小化预测与真实标签的误差（如分类用交叉熵、回归用MSE），常见任务包括图像分类、文本分类、房价预测等。

无监督学习

只给定输入数据 x （无标签），目标是发现数据的潜在结构或分布规律。典型任务有聚类（把相似样本分组）、降维（用更低维表示保留主要信息）、密度估计与表示学习等。

强化学习

智能体在环境中按策略选择动作，环境返回新状态与奖励：学习目标是最大化长期回报（期望回报）。核心概念包括状态、动作、奖励、策略、价值函数，常用方法有Q-learning、策略梯度、Actor-Critic等。

批标准化 (Batch Normalization, BN)

在训练时对每个mini-batch的某层数据做批标准化（减均值、除方差），并加入可学习的缩放 γ 与平移 β ，它能缓解梯度不稳定、加快收敛，并一定程度上起到正则化作用。推理论时使用训练期间累计的均值/方差进行固定调整。

预训练

先在大规模数据上训练模型，使其学到通用特征/表示（例如语言模型预测下一个词、图像模型做自监督比对学习等）。预训练得到的参数可作为下游任务的初始化，通常能显著提升效果并降低标注数据需求。

迁移学习

将源任务所学到的知识（参数、特征表示或结构）迁移到目标任务/领域使用，常见方法为加载预训练模型+微调（fine-tuning），也包括冻结前面层只训练任务头、领域自适应等。对于目标数据少或分布变化的场景。

生成对抗网络 (GAN)

由生成器 G 和判别器 D 组成： G 从噪声生成样本， D 判断样本真伪；二者通过对抗博弈训练，使生成样本逐渐逼近真实数据分布。GAN可生成高质量样本，但训练可能不稳定、易生成模式崩塌（只生成少数模式）。

掩码语言模型 (masked LM)

训练时随机遮住输入中的部分token，让模型利用双向上下文恢复被遮住的token，它擅长学习语义表示与上下文理解（如BERT），但由于训练目标与生成方式不一致，直接做长文本生成通常不如自回归模型自然。

编码器-解码器语言模型 (Encoder-Decoder)

面向“输入序列→输出序列”的条件生成框架：编码器读入源序列形成表示，解码器在该条件下自回归生成目标序列。典型应用是机器翻译、摘要、问答生成等，代表模型如T5、BART、原始Transformer。

foundation model (基础模型)

在超大规模数据上训练得到、具备广泛迁移能力的大模型（语言、视觉或多模态）。它通常通过提示（prompting）、检索增强、微调等方法快速适应多任务，强调“一个模型、多种用途”的通用性。

fine-tuning (微调)

将预训练模型在特定任务/领域数据上继续训练以匹配目标分布与输出格式。可全参数微调，也可用参数高效方法（如LoRA/Adapter）只更新少量参数；微调能显著提升专用任务性能，但也可能带来过拟合或遗忘能力。

benchmark (基准测试)

用于标准化比较模型能力的评测集合，通常包含固定数据集、任务定义、评价指标与评测协议。好的benchmark强调可复现、公平对比与覆盖多种能力（如推理、检索、生成质量、鲁棒性等）。

translationAI (机器翻译 AI)

指实现自动翻译的模型与系统，把源语言文本映射为目标语言文本。现代主流是基于Transformer的神经机器翻译（NMT），通常采用encoder-decoder结构，并用BLEU、COMET等指标或人工评价衡量译文质量。

计算生物学 (Computational Biology)

以数学、统计学与计算方法为核心手段来研究生物学问题的交叉学科，强调“模型与计算机制解释与理论推断”。常见方向包括系统生物学建模、进化与群体遗传分析、蛋白质结构/动力学模拟、干细胞与多组学数据的计数分析等。

生物信息学 (Bioinformatics)

更偏“数据与工具”的生物计算分支，关注生物数据（序列、表达谱、变异、互作网络等）的获取、存储、注释、分析与流程的构建。典型任务包括基因组装与注释、序列检索与比对、变异检测、功能注释、通路/富集分析、数据库与分析软件开发等。实践上常与计算生物学高度重叠，但生物信息学更工程与数据驱动。

序列对比 (Sequence Alignment)

将两条或多条DNA/RNA/蛋白质序列按“插入空位(gap)对齐，寻找相似片段与差异位置的过程，用于推断同源关系、保守位点、结构/功能相似性或进化事件。比对可分为全局比对（整体对齐、适合长度相近且整体同源）与局部比对（找最相似片段，适合只共享局部区域），评分通常由替换矩阵（蛋白）、匹配/错配（核酸）与gap penalty决定。

Sanger sequencing (桑格测序/链终止法)

经典的DNA测序方法：在DNA聚合反应中加入带荧光标记的双脱氧核苷酸（ddNTP），由于ddNTP缺少3'-OH，会随机终止延伸，生成一系列不同长度、末端碱基已知的片段；再通过毛细管电泳按长度分离并读取发光信号，得到序列。其特点是读长较长、准确率高、通量低、成本相对高，常用于小规模测序与验证实验。