5

Multiple Sequence Alignment

5.1 Pairwise Alignment Among Multiple Sequences

In previous chapters, the structural features of pairwise sequence alignment and the features of mutations were discussed. Based on these features, dynamic programming-based algorithms and the statistical decision-based algorithm (SPA) were presented. These algorithms are restricted to performing alignments for a pair of sequences. However, to apply these alignment methods to bioinformatics, they must be able to process a family of sequences (many more than two sequences) simultaneously. The alignment algorithms that can accomplish this are called multiple sequence alignment, or simply MA. When studying these alignments, pairwise alignment are the best reference. Therefore, we begin this chapter by discussing the structure resulting from mutations, as well as the structure by alignment of multiple sequences. As we attempt to develop MA, we discuss how to use pairwise alignment to process multiple sequences, and we consider what types of problems this raises.

5.1.1 Using Pairwise Alignment to Process Multiple Sequences

Both dynamic programming-based algorithms and statistical decision-based algorithms for pairs of sequences are fast and programmable. Until true MA are developed, researchers must use pairwise alignment methods to process multiple sequences. Many current MA software packages, such as Clustal-W [102] etc., are in fact based on pairwise alignment. That is, the pairwise alignment is an important component of these software packages. The use of pairwise alignment methods to process multiple sequences is important for a rough analysis of the common structure of multiple sequences. For example, we can analyze affinity relationships and evolution between each pair of sequences, based on pairwise alignment methods. This demonstrates that pairwise alignment of multiple sequences is important enough to consider specifically.

Let m be the number of sequences in a multiple sequence set; the computational complexity then ranges from $O(m^2 n)$ to $O(m^2 n^2)$ if we use pairwise alignment methods to process the set of sequences. For example, if we use the SPA, then the complexity is $O(m^2 n)$, while the computational complexity is $O(m^2 n^2)$ for a dynamic programming-based algorithm such as the Smith–Waterman algorithm. Therefore, if the size of the benchmark set is not overly large, pairwise alignment is acceptable in the sense of complexity. Of course, the use of pairwise alignment methods to build the homologous family of a sequence is only a stopgap measure, not the final goal.

5.1.2 Topological Space Induced by Pairwise Alignment of Multiple Sequences

We maintain the notations in (1.1) and (1.2) such that

$$\mathcal{A} = \{A_s = (a_{s,1}, a_{s,2}, \cdots, a_{s,n_s}), \quad s = 1, 2, \cdots, m\} \tag{5.1}$$

is a multiple sequence in which A_s is the sth gene sequence whose length is n_s, and $a_{s,j} \in V_4 = \{0, 1, 2, 3\}$ is the state space of nucleotides in the gene sequence, and m is its multiplicity. In addition, we still must introduce the following notations and point out some problems.

The Matrices Induced by Pairwise Alignment of Multiple Sequences

If $\mathcal{A}$ is a multiple sequence defined in (5.1), then for any $s, t \in M = \{1, 2, \cdots, m\}$, we find the result of pairwise alignment of (A_s, A_t) as follows:

$$(C_{s,t}, C_{t,s}) = ((c_{s,t;1}, c_{s,t;2}, \cdots, c_{s,t;n_{s,t}}), (c_{t,s;1}, c_{t,s;2}, \cdots, c_{t,s;n_{t,s}})) . \tag{5.2}$$

Then, $C_{s,t}, C_{t,s}$ is the expansion of (A_s, A_t) in which, $c_{s,t;j}, c_{t,s;j} \in V_5$ and $n_{s,t} = n_{t,s}$ is the common length of the sequences ($C_{s,t}$ and $C_{t,s}$).

We then obtain a matrix induced by pairwise alignment of multiple sequences as:

$$\bar{\mathcal{C}} = (C_{s,t})_{s,t=1,2,\cdots,m} , \tag{5.3}$$

in which $C_{s,t}$ is defined by (5.2). For simplicity, we refer to $\bar{\mathcal{C}}$ as the alignment matrix induced by multiple sequences $\mathcal{A}$, or the simpler alignment matrix. It is easy to find that changing the order of the pairwise alignments for the multiple sequences will result in a different matrix.

Definition 25. *Let $\bar{\mathcal{C}}$ be the alignment matrix induced by the multiple sequences $\mathcal{A}$. Let $T_{s,t}$ be the shifting mutation mode from A_t to A_s, and let $W = \{w(a,b),\ a, b \in V_5\}$ be the penalty function on V_5. Then:*

1. *$\bar{\mathcal{C}}$ is the minimum penalty alignment matrix if the expansion $(C_{s,t}, C_{t,s})$ of (A_s, A_t) has the minimum penalty score.*

2. *$\bar{\mathcal{C}}$ is the uniform alignment matrix if $(C_{s,t}, C_{t,s})$ is the uniform alignment of (A_s, A_t) based on the mode $T_{s,t}$ for every pair $s, t \in M = \{1, 2, \cdots, m\}$.*

The definitions of the minimum penalty alignment and uniform alignment, as well as the relationship between these two kinds of pairwise alignments, are outlined in Chaps. 1 and 4, respectively. In this chapter, we focus on the case of minimum penalty alignment. We will discuss the uniform alignment case in Chap. 7.

Penalty Matrix Induced by Pairwise Alignment of Multiple Sequences

Let $\bar{\mathcal{C}}$ be the alignment matrix induced by the multiple sequence $\mathcal{A}$. If the penalty function $W = w(a, b)$ defined on V_5 is given, then for any $s, t \in M$, we have two expansions $C_{s,t}$ and $C_{t,s}$ based on the pair of sequences A_s and A_t. The penalty score for the pair $C_{s,t}$ and $C_{t,s}$ is defined by:

$$w_{s,t}(\bar{\mathcal{C}}) = w(C_{s,t}, C_{t,s}) = \sum_{j=1}^{n_{s,t}} w(c_{s,t;j}, c_{t,s;j}) \ . \tag{5.4}$$

Definition 26. *Let $\bar{\mathcal{C}}$ be the alignment matrix induced by multiple sequences $\mathcal{A}$, and let W be the penalty function defined on V_5. The matrix*

$$\bar{W}(\bar{\mathcal{C}}) = [w_{s,t}(\bar{\mathcal{C}})]_{s,t=1,2,\cdots,m} \tag{5.5}$$

is then the penalty matrix induced by pairwise alignment of multiple sequences $\mathcal{A}$, where $w_{s,t}(\bar{\mathcal{C}})$ is defined by (5.4). It is acceptable to simply call this the penalty matrix.

A fixed penalty matrix $\bar{W}^o = (w^o_{s,t})_{s,t=1,2,\cdots,m}$, is the minimum penalty matrix if each $w^o_{s,t}$ is the score of the minimum penalty alignment of (A_s, A_t).

For the sake of simplicity, we use $\bar{W}(\bar{\mathcal{C}})$ to replace

$$\bar{W} = (w_{s,t})_{s,t=1,2,\cdots,m} \ .$$

Theorem 22. *If the multiple sequence $\mathcal{A}$ and the penalty function W on V_5 are given, then the minimum penalty matrix $\bar{W}^o$ is uniquely determined, and denoted by $\bar{W}^o = \bar{W}^o(\mathcal{A})$.*

Proof. It is sufficient to prove that the score $w^o_{s,t}$ of the minimum penalty alignment is uniquely determined for any sequence pair (A_s, A_t). To do this, we check the definition

$$w^o_{s,t} = \min\{w_{s,t} = w(C_{s,t}, C_{t,s}) \colon (C_{s,t}, C_{t,s}) \text{ is the alignment of } (A_s, A_t)\} \tag{5.6}$$

in which the set at the right-hand side of expression (5.6) has a lower bound. It follows that the minimum value is unique. Hence, the minimum penalty matrix $\bar{W}^o$ is also unique.

We then have the relationship

$$\mathcal{M} = \{\mathcal{A}, \bar{W}^o\} \ .$$

This is called the minimum penalty representation of pairwise alignment of multiple sequences. We will prove later that $\mathcal{M}$ forms a metric space.

Metric Space Defined on a Finite Set

The metric space is a fundamental concept in mathematics. To show that $\mathcal{M} = \{\mathcal{A}, \bar{W}^o\}$ is a finite metric space, we present the general definition of a metric space defined on a finite set.

Let $M = \{1, 2, \cdots, m\}$ be a finite set, and let $w_{s,t}$ be a function defined on $M \times M = \{(s,t) \colon s, t \in M\}$.

Definition 27. *A function $w_{s,t}$ defined on $M \times M$ is a measure (or metric or distance), if the following conditions hold:*

1. ***Nonnegative property:*** *$w_{s,t} \geq 0$ holds all $s, t \in M$ and $w_{s,t} = 0$ if and only if $s = t$.*
2. ***Symmetry property:*** *$w_{s,t} = w_{t,s}$ holds for all $s, t \in M$.*
3. ***Triangle inequality:*** *$w_{s,r} \leq w_{s,t} + w_{t,r}$ holds for all $s, t, r \in M$.*

If the distance function $w_{s,t}$ defined on a finite set M is given, then the M endowed with this distance forms a metric space, and it is called a finite metric space, or finite distance space.

The Fundamental Theorem of Minimum Penalty Alignment

Let $\mathcal{M}$ be the minimum penalty representation of pairwise alignment of multiple sequences defined as above. It is a finite metric space under the natural distance induced by the minimum penalty matrix $\bar{W}^o$, although this is not obvious. In fact, we can not build the relationship among

$$w(C_{s,t}, C_{t,s}) \,, \quad w(C_{s,r}, C_{r,s}) \,, \quad w(C_{t,r}, C_{r,t})$$

directly because the expansions $C_{s,t}$ and $C_{s,r}$ based on A_s and A_t, A_r are not unique. The fundamental theorem of the minimum penalty alignment is that the minimum penalty representation $\mathcal{M}$ is a finite metric space if the penalty matrix $\bar{W}^o = [w^o_{s,t}]_{s,t \in M}$ satisfies the three conditions defined above.

Theorem 23. *(Fundamental theorem of minimum penalty alignment.)*
Let $\mathcal{A}$ be a multiple sequence and $\bar{W}^o = (w^o_{s,t})_{s,t \in M}$ be the minimum penalty matrix of the multiple sequences $\mathcal{A}$ defined by (5.5) under a given penalty function $W = w(a,b)$, $a, b \in V_5$. Then, the $\mathcal{M}$ endowed with the natural distance induced by $\bar{W}^o = (w^o_{s,t})_{s,t \in M}$ is a finite metric space.

Proof. For simplicity, let the penalty function $w(a,b)$ be the Hamming matrix on V_5. Then, let $\bar{W}^o = (w^o_{s,t})_{s,t\in M}$ be the minimum penalty matrix based on the multiple sequences $\mathcal{A}$ defined by (5.5). We consider the natural distance induced by this minimum penalty matrix as follows: $d(A_s, A_t) = w^o(s,t)$. Using the definitions of the Hamming matrix and $\bar{\mathcal{C}}$, we find that $d(\cdot)$ satisfies both the nonnegative and symmetry properties. Therefore, we need only prove that $d(\cdot)$ satisfies the triangle inequality. Alternatively, we prove that $w^o_{s,r} \le w^o_{s,t} + w^o_{t,r}$ holds for all $s,t,r \in M$. For an arbitrary three $s,t,r \in \{1,2,\cdots,M\}$, we may assume that these three subscripts are different from each other. For simplicity, we also omit the subscripts of the three vectors A_s, A_t, A_r. Let

$$Z = (z_1, z_2, \cdots, z_{n_z}), \quad Z = A, B, C, \quad z = a, b, c \tag{5.7}$$

be the uniform representation of the three sequences, and let

$$\begin{pmatrix} A' \\ B' \end{pmatrix}, \quad \begin{pmatrix} A^* \\ C^* \end{pmatrix}, \quad \begin{pmatrix} B^o \\ C^o \end{pmatrix} \tag{5.8}$$

be the minimum penalty alignments of all possible combined pairs in A, B, C. Thus, we alternatively prove that

$$w(B^o, C^o) \le w(A', B') + w(A^*, C^*) \tag{5.9}$$

holds, by using the steps outlined below:

1. Following from the definition in (5.7), we know that the sequences A', A^* are the expansions of A, B', B^o are the expansions of B, and C^*, C^o are the expansions of C, with the corresponding expanded modes given as

$$\begin{cases} H'_a = (\gamma'_{a,1}, \gamma'_{a,2}, \cdots, \gamma'_{a,n_a}) = \left\{\left(i'_{a,k}, \ell'_{a,k}\right),\ k = 1,2,\cdots,k'_a\right\}, \\ H^*_a = (\gamma^*_{a,1}, \gamma^*_{a,2}, \cdots, \gamma^*_{a,n_a}) = \left\{\left(i^*_{a,k}, \ell^*_{a,k}\right),\ k = 1,2,\cdots,k^*_a\right\}, \\ H'_b = \left(\gamma'_{b,1}, \gamma'_{b,2}, \cdots, \gamma'_{b,n_b}\right) = \left\{\left(i'_{b,k}, \ell'_{b,k}\right),\ k = 1,2,\cdots,k'_b\right\}, \\ H^o_b = \left(\gamma^o_{b,1}, \gamma^o_{b,2}, \cdots, \gamma^o_{b,n_b}\right) = \left\{\left(i^o_{b,k}, \ell^o_{b,k}\right),\ k = 1,2,\cdots,k^o_b\right\}, \\ H^*_c = (\gamma^*_{c,1}, \gamma^*_{c,2}, \cdots, \gamma^*_{c,n_c}) = \left\{\left(i^*_{c,k}, \ell^*_{c,k}\right),\ k = 1,2,\cdots,k^*_c\right\}, \\ H^o_c = (\gamma^o_{c,1}, \gamma^o_{c,2}, \cdots, \gamma^o_{c,n_c}) = \left\{\left(i^o_{c,k}, \ell^o_{c,k}\right),\ k = 1,2,\cdots,k^o_c\right\}, \end{cases} \tag{5.10}$$

   where n_a, n_b, n_c are the lengths of sequences A, B, C, respectively.
2. Let $n'_a, n^*_a, n'_b, n^o_b, n^*_c$ and n^o_c be the lengths of the sequences A', A^*, B', B^o, C^* and C^o respectively, where $n'_a = n'_b$, $n^*_a = n^*_c$, $n^o_b = n^o_c$. Then, following from $(H'_a, H^*_a, H'_b, H^o_b, H^*_c, H^o_c)$, the decompositions of N'_a, N^*_a,

N'_b, N^o_b, N^*_c and N^o_c are as follows:

$$\begin{cases} N'_a = \left\{\delta'_{a,1}, \delta'_{a,2}, \cdots, \delta'_{a,2k'_a+1}\right\} , \\ N^*_a = \left\{\delta^*_{a,1}, \delta^*_{a,2}, \cdots, \delta^*_{a,2k^*_a+1}\right\} , \\ N'_b = \left\{\delta'_{b,1}, \delta'_{b,2}, \cdots, \delta'_{b,2k'_b+1}\right\} , \\ N^o_b = \left\{\delta^o_{b,1}, \delta^o_{b,2}, \cdots, \delta^o_{b,2k^o_b+1}\right\} , \\ N^*_c = \left\{\delta^*_{c,1}, \delta^*_{c,2}, \cdots, \delta^*_{c,2k^*_c+1}\right\} , \\ N^o_c = \left\{\delta^o_{c,1}, \delta^o_{c,2}, \cdots, \delta^o_{c,2k^o_c+1}\right\} , \end{cases} \tag{5.11}$$

where the intervals δ's are connected in order. If we let

$$\begin{array}{ll} \Delta'_{a,1} = \left\{\delta'_{a,1}, \delta'_{a,3}, \cdots, \delta'_{a,2k'_a+1}\right\} & \Delta'_{a,2} = \left\{\delta'_{a,2}, \delta'_{a,4}, \cdots, \delta'_{a,2k'_a}\right\} \\ \Delta^*_{a,1} = \left\{\delta^*_{a,1}, \delta^*_{a,3}, \cdots, \delta^*_{a,2k^*_a+1}\right\} & \Delta^*_{a,2} = \left\{\delta^*_{a,2}, \delta^*_{a,4}, \cdots, \delta^*_{a,2k^*_a}\right\} \\ \Delta'_{b,1} = \left\{\delta'_{b,1}, \delta'_{b,3}, \cdots, \delta'_{b,2k'_b+1}\right\} & \Delta'_{b,2} = \left\{\delta'_{b,2}, \delta'_{b,4}, \cdots, \delta'_{b,2k'_b}\right\} \\ \Delta^o_{b,1} = \left\{\delta^o_{b,1}, \delta^o_{b,3}, \cdots, \delta^o_{b,2k^o_b+1}\right\} & \Delta^o_{b,2} = \left\{\delta^o_{b,2}, \delta^o_{b,4}, \cdots, \delta^o_{b,2k^o_b}\right\} \\ \Delta^*_{c,1} = \left\{\delta^*_{c,1}, \delta^*_{c,3}, \cdots, \delta^*_{c,2k^*_c+1}\right\} & \Delta^*_{c,2} = \left\{\delta^*_{c,2}, \delta^*_{c,4}, \cdots, \delta^*_{c,2k^*_c}\right\} \\ \Delta^o_{c,1} = \left\{\delta^o_{c,1}, \delta^o_{c,3}, \cdots, \delta^o_{c,2k^o_c+1}\right\} & \Delta^o_{c,2} = \left\{\delta^o_{c,2}, \delta^o_{c,4}, \cdots, \delta^o_{c,2k^o_c}\right\} \end{array} \tag{5.12}$$

then we have

$$a'_{\Delta'_{a,1}} = a^*_{\Delta^*_{a,1}} = A , \quad b'_{\Delta'_{b,1}} = b^o_{\Delta^o_{b,1}} = B , \quad c^*_{\Delta^*_{c,1}} = c^o_{\Delta^o_{c,1}} = C , \tag{5.13}$$

and all components in the following vectors

$$a'_{\Delta'_{a,2}} , \quad a^*_{\Delta^*_{a,2}} , \quad b'_{\Delta'_{b,2}} , \quad b^o_{\Delta^o_{b,2}} , \quad c^*_{\Delta^*_{c,2}} , \quad c^o_{\Delta^o_{c,2}}$$

are the inserted symbol "–".

3. Following from the union $H''_a = H'_a \vee H^*_a$ of the expanded modes H'_a, H^*_a, we may expand sequence A to A'' under the mode H''_a. Then, A'' is actually the virtual expansion of both A' and A^*, whose extra regions are $H''_a - H'_a$ and $H''_a - H^*_a$, respectively. Therefore, we get the expanded modes on A' and A^* as $(H''_a - H'_a)_{a'}$ and $(H''_a - H^*_a)_{a^*}$, respectively. $(H''_a - H'_a)_{a'}$, $(H''_a - H^*_a)_{a^*}$ are then two different quadratic expansions of sequence A, whose evolution process is given as:

$$A \xrightarrow{H'_a} A' \xrightarrow{(H''_a - H'_a)_{a'}} A'' , \quad A \xrightarrow{H^*_a} A^* \xrightarrow{(H''_a - H^*_a)_{a^*}} A'' . \tag{5.14}$$

4. Since the lengths of sequences B', C^* are equal to the lengths of sequences A', A^*, respectively, the expansions of B', C^* under the expanded mode $(H''_a - H'_a)_{a'}$, $(H''_a - H^*_a)_{a^*}$ are denoted by B'', C'', respectively. Then, $(H''_a - H'_a)_{a'}$, $(H''_a - H^*_a)_{a^*}$ are the common expanded regions of pair A'' and B'' and pair A'' and C'' respectively. Hence, we find that

$$w(A'', B'') = w(A', B') , \quad w(A'', C'') = w(A^*, C^*) . \tag{5.15}$$

On the other hand, based on the definition of $w(A'', B'') = \sum_{j=1}^{n''_a} w(a''_j, b''_j)$, we derive the following relationships:

$$\begin{aligned} w(A'', B'') + w(A'', C'') &= \sum_{j=1}^{n''_a} \left[w\left(a''_j, b''_j\right) + w\left(a''_j, c''_j\right) \right] \\ &\geq \sum_{j=1}^{n''_a} w\left(b''_j, c''_j\right) = w(B'', C'') , \end{aligned} \tag{5.16}$$

where the inequality holds due to the following expression:

$$w\left(a''_j, b''_j\right) + w\left(a''_j, c''_j\right) \geq w\left(b''_j, c''_j\right) \quad \forall j = 1, 2, \cdots, n''_a ,$$

where $w(a, b)$ is a measurement defined on V_5, and n''_a is the length of the sequence A''.

5. In view of (5.15) and (5.16), we have the following inequality:

$$\begin{aligned} w(B^o, C^o) \leq w(B'', C'') &\leq w(A'', B'') + w(A'', C'') \\ &= w(A', B') + w(A^*, C^*) . \end{aligned} \tag{5.17}$$

Similarly, we can prove that

$$\begin{aligned} w(A^*, C^*) &\leq w(A', B') + w(B^o, C^o) , \\ w(A', B') &\leq w(A^*, C^*) + w(B^o, C^o) . \end{aligned}$$

Hence, the required triangle inequality relationship of $\bar{W}^o = (w^o_{s,t})_{s,t \in M}$ holds. This is equivalent to saying that $\bar{W}$ is a distance function defined on $\mathcal{A}$. This ends the proof.

Next, we denote the finite metric space with a minimum penalty matrix by $\mathcal{M} = \{M, W\}$, and we may call it the **metric space of pairwise alignment** for short, where $M = \{1, 2, \cdots, m\}$ is the subscript of $\mathcal{A}$, and $W = (w_{s,t})_{s,t=1,2,\cdots,m}$ is the minimum penalty matrix induced by the pairwise alignment of the multiple sequences $\mathcal{A}$ under a given penalty function. This metric space $\mathcal{M}$ is useful in the clustering of multiple sequences, and in the analysis of the evolution of multiple sequences. Clustering analysis is useful in many aspects of sequence analysis. We will discuss this further in later sections.

5.2 Optimization Criteria of MA

5.2.1 The Definition of MA

Using pairwise alignment to process multiple sequences is not a true multiple alignment approach, as several problems cannot be solved through use of this strategy. For example:

1. **To search for common stable regions of a family of sequences**. In other words, in determining the common region of many biological sequences, the pairwise alignment methods do not work.
2. **For an overview of the characteristics and trends of multiple sequences**. The stable regions of multiple sequences do not perfectly coincide. Frequently, there are sequences in a multiple sequence set such that the stable regions are different. This difference often cannot be found through pairwise alignment, only by MA.
3. **Analyze these types of mutation comprehensively**. Structure of the sequence before mutation and prediction problems. In the mutating processes, many important mutation types will occur, for example, independent mutation and transitional mutation, etc. To analyze these types of mutation comprehensively, and to predict the trend of changes, we must involve MA.

In conclusion, MA are vitally important tools in the analysis of the common structure of a family of sequences, and their usefulness is not limited to the research on mutations in and evolution of biological sequences. It is used comprehensively to solve bioinformatics problems, for example, as a main tool for predicting the secondary structure of proteins.

To create MA, we begin by building the optimization criteria of MA methods, and then attempt the optimization of MA.

5.2.2 Uniform Alignment Criteria and SP-Optimization Criteria for Multiple Sequences

Definition of Uniform Alignment of Multiple Sequences

Uniform alignment of a pair of sequences was addressed when we discussed mutation and pairwise alignment. We now generalize this concept to fit multiple sequences.

Let $\mathcal{A} = \{A_1, A_2, \cdots, A_n\}$ be a multiple sequence and $\mathcal{C} = \{C_1, C_2, \cdots, C_m\}$ be the alignment of $\mathcal{A}$. Typically,

$$A_s = (a_{s,1}, a_{s,2}, \cdots, a_{s,n_s}) , \quad C_t = (c_{t,1}, c_{t,2}, \cdots, c_{t,n'_t}) , \quad s, t \in M ,$$

$a_{s,j}$, $c_{t,j}$ are elements of V_4, V_5, respectively, and each C_s is virtual expansion of A_s.

Within the multiple sequence $\mathcal{A}$, every pair A_s, A_t are mutated sequences acted on by shifting and nonshifting mutations. Let $T_{s,t}$ be the mutation mode for A_s, A_t mutating to C_s, C_t, respectively, and let (C'_s, C'_t) be the compressed sequences of (C_s, C_t). If $(c_{sj}, c_{tj}) = (4, 4)$, then delete these two components from C_s, C_t, respectively, so that the rest of (C'_s, C'_t) is still the expansion of (A_s, A_t).

Definition 28. *Let $\mathcal{C}$ be the multiple expansion of $\mathcal{A}$. Then $\mathcal{C}$ is the uniform alignment of $\mathcal{A}$, if for every $s \neq t \in M$, the following conditions are satisfied:*

1. *For every expansion C'_s of A_s, the added part just consists of the regions resulting from type-III mutation so that A_s to A_t.*
2. *For every expansion C'_t of A_t, the added part just consists of the regions resulting from type-III mutation so that A_t to A_s.*

Calculation of Uniform Alignment of Multiple Sequences

In Sects. 3.1 and 3.2, we mentioned the mutation mode of multiple sequences and their envelope. If a multiple sequence $\mathcal{A}$ has only shifting mutations, then the uniform alignment $\mathcal{C}$ of $\mathcal{A}$ can be computed by the following steps:

1. Calculate the minimum envelope C_0 of $\mathcal{A}$.
2. For each s, since C_0 is the expansion of A_s, we compare C_0 and A_s. For the extra coordinates of C_0 relative to A_s, we replace them with "$-$", and then renew the sequence denoted by C_s. The collection of all renewed sequences $\mathcal{C} = \{C_s,\ s \in M\}$ is the uniform alignment of the multiple sequence.
3. If $\mathcal{A}$ is a multiple sequence involving both shifting and nonshifting mutations, then the minimum envelope C_0 involves type-I and type-II mutations, and C_0 relative to A_s can be divided into two parts, namely, the expansion and nonexpansion parts as follows:
$$C_0 = \left(c_{\Delta'_{s,0}}, c_{\Delta'_{s,1}} \right) ,$$
where $c_{\Delta'_{s,0}}$ is the expansion part and $c_{\Delta'_{s,1}}$ is the nonexpansion part of A_s.
4. The uniform alignment of a multiple sequence $\mathcal{A}$ is the result processed the following way: replace the corresponding coordinates in the region of $c_{\Delta'_{s,0}}$ by the elements of A_s, and replace the coordinates in the region of $c_{\Delta'_{s,1}}$ by the virtual symbol "$-$". The renewed multiple sequence is then the uniform alignment of $\mathcal{A}$.

Example 19. Let $\mathcal{A}$ be a triple of sequences given by:

$$\begin{cases} A_1 = \text{aactg()ggga[tagat]gguuuaacgta\{aauau\}accgt}, \\ A_2 = \text{aactg(gta)ggga[]gguuuaacgta\{aauau\}accgt}, \\ A_3 = \text{aactg(gta)ggga[tagat]gguuuaacgta\{\}accgt} . \end{cases}$$

Comparing these three sequences, we find the following mutation relationships:

Based on A_1, we insert gta after position 5 and delete tagat after position 9, so that A_1 mutates to A_2.

Also based on A_1, we insert gta after spot 5 and delete aauau after position 25, so that A_1 mutates to A_3.

Based on A_2, we insert tagat after position 12 and delete aauau after position 23, so that A_2 mutates to A_3.

Obviously, the triple sequence $\mathcal{A}$ has shifting mutations only. Therefore, each sequence in $\mathcal{A}$ can be mutated from another sequence in $\mathcal{A}$ by type-III and type-IV mutations.

The minimum envelope and maximum core are given by:

$$\begin{cases} C_0 = \text{aactg(gta)ggga[tagat]gguuuaacgta\{aauau\}accgt}\,, \\ D_0 = \text{aactg()ggga[]gguuuaacgta\{\}accgt}\ , \end{cases}$$

in which the data in parentheses, brackets, or braces are the deleted segments in A_1, A_2, A_3, respectively. We set

$$\begin{cases} C_1 = \text{aactg(----)ggga[tagat]gguuuaacgta\{aauau\}accgt}\,, \\ C_2 = \text{aactg(gta)ggga[-------]gguuuaacgta\{aauau\}accgt}\,, \\ C_3 = \text{aactg(gta)ggga[tagat]gguuuaacgta\{-------\} accgt}\ . \end{cases}$$

This renewed triple sequence is the uniform alignment of the triple sequence $\mathcal{A}$.

If A_1, A_2, A_3 have nonshifting mutations, they have no unified envelope and core. We construct an envelope and core with type-I and type-II mutations, and then construct the corresponding uniform alignment. For example, if

$$\begin{cases} A'_1 = \text{aaatg()ggga[tagat]gguuuaacgta\{aauau\}accgt}\,, \\ A'_2 = \text{aactg(gta)ggga[]gguaauucgta\{aauau\}accgt}\,, \\ A'_3 = \text{aactg(gta)ggga[tagat]gguuuaacgtaaccgg}\ . \end{cases}$$

then besides the shifting mutation like the one in A_1, A_2, A_3, there are also type-I and type-II mutations in A'_1, A'_2, A'_3.

At position 3 of A'_1, c was mutated to a, relative to A_2. This is a type-I mutation. At positions 16–19 of A'_2, aauu was mutated to uuaa, relative to A_1. This is a type-II mutation. At the last position of A'_3, the t was mutated to t, relative to A_1 and A_2. This again is type-I mutation.

After this preprocessing, we denote the envelope and core of A'_1, A'_2, A'_3 by C_0, D_0, and let the uniform alignment be:

$$\begin{cases} C'_1 = \text{aaatg(----)ggga[tagat]gguuuaacgta\{aauau\}accgt}\ , \\ C'_2 = \text{aactg(gta)ggga[-------]gguaauucgta\{aauau\}accgt}\ , \\ C'_3 = \text{aactg(gta)ggga[tagat]gguuuaacgta\{-------\}accgg}\ . \end{cases}$$

Problems in Uniform Alignment of Multiple Sequences

Based on the definition of uniform alignment of multiple sequences, we know that all of the shifting mutations among the multiple sequences can be determined by uniform alignment of multiple sequences, after which all the mutations between every pair in the multiple sequence can be determined. Thus, uniform alignment of multiple sequences is the ultimate goal. On the other hand, there are several difficult problems involved in uniform alignment of multiple sequences which must be solved, such as:

1. Example 19 is a special case that may be solved. For general cases, the calculation of the uniform alignment is too complex; so we must still find a systematic algorithm in order to solve it.
2. It is difficult to judge whether a MA is a uniform alignment or not. We cannot establish a unified indexing system to judge uniform alignment.

This shows that the uniform alignment of multiple sequences is simply an ideal optimization criterion, which is in reality difficult to perform. So, we must still find other optimization criteria.

SP-Criterion of MA

The SP-penalty functions of MA presented in (1.9), are frequently involved in current literature. The involved notations are stated as follows:

1. Let $w(a,b)$, $a,b \in V_5$ be the metric function defined on V_5, which is also called the difference degree, or penalty matrix. The most popular penalty matrices for DNA (or RNA) are the Hamming matrix, the WT-matrix, etc. The definition of the WT-matrix is presented in (1.7). Generally, a metric function $w(a,b)$, $a,b \in V_5$ should satisfy the three axioms, namely, nonnegativity, symmetry and the triangle inequality.
2. The SP-function is the function most frequently used as the penalty function for multiple sequences. The definition of the SP-function is presented in (1.9).
3. A generalized form of the SP-function is the weighted WSP-function defined as follows:

$$w_{\rm WSP}(\mathcal{C}) = \sum_{j=1}^{n'} \sum_{t>s} \sum_{s=1}^{m-1} \theta_{s,t} w(c_{s,j}, c_{t,j}) = \sum_{j=1}^{n'} \sum_{s=1}^{m-1} \sum_{t>s} \theta_{s,t} w(c_{s,j}, c_{t,j}) \ , \tag{5.18}$$

   where $\theta_{s,t}$ is a weighting function.

   The functions $w_{\rm SP}(\mathcal{C})$ and $w_{\rm WSP}(\mathcal{C})$ defined by (1.9) and (5.18) respectively, are both called SP-penalty functions for multiple sequences. Thus, multiple sequence alignment (MSA) may be formed as follows. For a multiple sequences $\mathcal{A}$, search for its expansion $\mathcal{C}_0$ with the minimum penalty (or the maximum similarity) under the given penalty function. Alternatively, solve the expansion $\mathcal{C}_0$ of $\mathcal{A}$ such that

$$w_{\mathrm{SP}}(\mathcal{C}_0) = \min\{w_{\mathrm{SP}}(\mathcal{C})\colon \mathcal{C} \text{ is the expansion of } \mathcal{A}\} \ . \tag{5.19}$$

4. In addition, the SP-scoring function is also commonly used in current literature. We then let $w(a, b)$, $a, b \in V_5$ be a scoring function defined on V_5. Similarly, we have the scoring matrices for DNA (or RNA) sequences based on the Hamming matrix, or the WT-matrix. The scoring matrix based on the Hamming matrix is defined below:

$$W = [w_{\mathrm{WT}}(a, b)]_{a,b\in V_5} = \left(\begin{array}{cccccc} & - & a & c & g & t(u) \\ - & 0 & -2 & -2 & -2 & -2 \\ a & -2 & 1 & 0 & 0 & 0 \\ c & -2 & 0 & 1 & 0 & 0 \\ g & -2 & 0 & 0 & 1 & 0 \\ t(u) & -2 & 0 & 0 & 0 & 1 \end{array}\right) , \tag{5.20}$$

where $-, a, c, g, t(u)$ are $4, 0, 1, 2, 3$ respectively. The definition of the SP-scoring function for multiple sequences is then the same as (1.9) or (5.18). The corresponding optimization criterion is to find the maximum value in (5.19).

5.2.3 Discussion of the Optimization Criterion of MA

Establishing the Optimization Criteria for Multiple Sequences

As presented above, the uniform alignment criterion is reasonable but difficult to judge, while the index of the SP-criterion is easily calculated based on the result $\mathcal{C}$ although its rationality is yet to be demonstrated. In fact, many other optimization criteria for MA have been proposed. Therefore, we first must understand how to find the fundamental rules for judging the quality of specific optimization criteria. For this, we propose the following requirements: rationality, decidability, comparability (with the optimization solution) and helpfulness in calculating the optimization solution. We detail them as follows:

Requirement 1: rationality. Rationality here means whether or not the proposed criterion is related to multiple alignments. We need to know how to judge rationality.

Requirement 2: decidability. Decidability here means it directly and quickly decides the quality of an optimization criterion based on the alignment output. Obviously, a good criterion should be decidable and easy to calculate.

Requirement 3: comparability (with optimization solution). Comparability here means that the alignment result determining the proposed optimization criterion should be comparable with the optimization solution, or should determine the difference between the alignment result and the optimization solution.

Requirement 4: usefulness in optimizing the solution.

Based on the definitions of a uniform alignment criterion and the SP-criterion, we know that a uniform alignment criterion satisfies requirement 1, but it does not satisfy requirements 2 and 3. The SP-criterion satisfies requirements 1 and 2, but does not satisfy requirement 3. Therefore, by using the SP-criterion, we can only judge whether the alignment result is good or bad compared with another result. We cannot calculate the difference between the alignment result and the optimal solution. Later, we may find that the SP-criterion is easily calculated, although it does not really satisfy requirement 1.

Rational Conditions of the Optimization Criteria of MA

As mentioned above, the optimization conditions of MA should relate to the goal of MA. That is, to search for stable regions within multiple sequences and to determine the trend of mutation. Therefore, we should use the "concentration" of alignment results as a basic index.

In mathematics, there are several methods for measuring the relationship between various data. For example, distance, surface area and volume are familiar measurements. As well, the uncertainty of a random variable is a basic element in informatics. The probability distribution is an important factor when determining the uncertainty.

Besides the expressions of metric relations between the data, we also should consider their specific characteristics. For example, in the case of distance, it includes not only the formulas in Euclidean space, but also the three characteristics: nonnegativity, symmetry and the triangle inequality. For measurement, its vital characteristic is its additivity. The uncertainty also has particular characteristics that will be discussed later.

In order to establish the optimization conditions for MA, the concentration is chosen as a candidate index. We add the following conditions on the penalty function of MA.

Condition 5.2.1 Nonnegative property. For any MA $\mathcal{C}$, we always have $w(\mathcal{C}) \geq 0$, and the equality holds if and only if

$$\mathcal{C} = \mathcal{A}\,, \quad \text{and} \quad A_1 = A_2 = \cdots = A_m\ . \tag{5.21}$$

Expression (5.21) means that there are no virtual symbols "$-$" in any sequence.

Condition 5.2.2 Symmetry property. This means that the overall penalty function is invariant if we permute the order of the sequences in $\mathcal{C}$. Generally, let σ_1, σ_2 be two permutations defined on sets

$$M = \{1, 2, \cdots, m\}\,, \quad N' = \{1, 2, \cdots, n'\}\ ,$$

respectively, and let

$$\begin{cases} \sigma_1(\mathcal{C}) = \{A_{\sigma_1(s)}\,, \quad s = 1, 2, \cdots, m\}\,, \\ \sigma_2(\mathcal{C}) = \{\sigma_2(A_s)\,, \quad s = 1, 2, \cdots, m\}\,, \end{cases} \tag{5.22}$$

where

$$\sigma_2(A_s) = (\sigma_2(a_{s,1}), \sigma_2(a_{s,2}), \cdots, \sigma_2(a_{s,n'})) \tag{5.23}$$

and $\sigma_2(a_{s,j}) = a_{s,\sigma_2(j)}$. The symmetry property means that

$$w[\sigma_1(\mathcal{C})] = w(\mathcal{C})\,, \quad w[\sigma_2(\mathcal{C})] = w(\mathcal{C})\ . \tag{5.24}$$

Condition 5.2.3 Maximum–minimum condition. This condition is used to describe the column without the virtual symbol "−". It means that the penalty score is maximum if a, c, g, and u occur in this column obeying uniform distribution, and the penalty score is minimum if only one of a, c, g, or u occurs in this column. Condition 5.2.3 reflects the requirement for uniformity. We use this to find positions that are invariant for all sequences.

Another requirement for uniformity is that the penalty score of the mixed sequence produced by two multiple sequences should be greater than the sum of the penalty scores of two single multiple sequences. For example, if

$$\mathcal{A}_1 = \{A_{1,1}, A_{1,2}, \cdots, A_{1,m_1}\}\,, \quad \mathcal{A}_2 = \{A_{2,1}, A_{2,2}, \cdots, A_{2,m_2}\}\ . \tag{5.25}$$

These two multiple sequences have no common elements, where

$$A_{\tau;s} = (a_{\tau;s,1}, a_{\tau;s,2}, \cdots, a_{\tau;s,n_{\tau,s}})\,, \quad \tau = 1, 2\,, \quad s = 1, 2, \cdots, m_\tau\ . \tag{5.26}$$

The mixed multiple sequence is

$$\mathcal{A}_0 = \{\mathcal{A}_1, \mathcal{A}_2\} = \{A_{1,1}, A_{1,2}, \cdots, A_{1,m_1}, A_{2,1}, A_{2,2}, \cdots, A_{2,m_2}\}\ . \tag{5.27}$$

We denote $\mathcal{A}_0 = \mathcal{A}_1 \otimes \mathcal{A}_2$, and the operation $\otimes$ is called the row superposition of multiple sequences.

Let $\mathcal{C}_\tau$ be the alignments of the multiple sequences $\mathcal{A}_\tau$, $\tau = 1, 2$, and let $\mathcal{C}_0 = \mathcal{C}_1 \otimes \mathcal{C}_2$ be defined in the same way as (5.27). Then, $\mathcal{C}_0$ is the alignment of $\mathcal{A}_0$.

Condition 5.2.4 Convexity of row superposition. Let $\mathcal{C}_\tau$ be the alignment of multiple sequences $\mathcal{A}_\tau$, $\tau = 1, 2$, then $\mathcal{C}_0 = \mathcal{C}_1 \otimes \mathcal{C}_2$ satisfies the following inequality:

$$w(\mathcal{C}_0) \geq \frac{m_1}{m_1 + m_2} w(\mathcal{C}_1) + \frac{m_2}{m_1 + m_2} w(\mathcal{C}_2)\ . \tag{5.28}$$

If $m_1, m_2 > 0$, then the equality in (5.28) holds if and only if there is no "−" in $\mathcal{C}_0$ and if the probabilities of finding a, c, g, t in each column of $\mathcal{C}_1$ and $\mathcal{C}_2$ are the same.

Let the jth column vector of $\mathcal{C}_0, \mathcal{C}_1$, and $\mathcal{C}_2$ be

$$c_{0;\cdot,j} = \begin{pmatrix} c_{1;\cdot,j} \\ c_{2;\cdot,j} \end{pmatrix} , \quad c_{\tau;\cdot,j} = \begin{pmatrix} c_{\tau;1,j} \\ c_{\tau;2,j} \\ \vdots \\ c_{\tau;m_\tau,j} \end{pmatrix} , \quad \tau = 1, 2 \ . \tag{5.29}$$

Then, the equality in Condition 5.2.4 holds if and only if there is no "−" occurring in both $c_{1;\cdot,j}$ and $c_{2;\cdot,j}$ and the probabilities of finding a, c, g, t are the same.

Conditions 5.2.1–5.2.4 are the basic requirements for uncertainty or concentration. We should pay attention when comparing them with other relations (i.e., distance, measurement, etc). These basic relations frequently form an axiomatic system in the mathematical sense. Different axiomatic systems lead to different branches of disciplines, and different branches which have different data structure relations. For example, the difference between Euclidean geometry and non-Euclidean geometry is the fifth postulate (the axiom of parallels).

Besides the uncertainty requirements, we still have several additional requirements. We add the following three conditions:

Condition 5.2.5 The invariance of the penalty function. This means the "penalty function" for MA should be a penalty function for pairwise alignment if it was restricted to a pair of sequences. The popular penalty functions for pairwise alignment include the generalized Hamming function, the WT-matrix, etc.

Condition 5.2.6 The number of virtual symbols "−" is a minimum. If there exists one row of $\mathcal{C}$ where all the elements are "−", i.e., if

$$c_{\cdot j} = \begin{pmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \\ c_{m,j} \end{pmatrix} = \begin{pmatrix} - \\ - \\ \vdots \\ - \end{pmatrix} ,$$

then $\mathcal{C}$ is definitely not the minimum penalty expansion of $\mathcal{A}$.

The row superposition operation $\otimes$ for multiple sequences is defined in expression (5.27). Similarly, we define the column superposition operation for multiple sequences. Let

$$\mathcal{A}_1 = \{A_{1,1}, A_{1,2}, \cdots, A_{1,m}\} , \quad \mathcal{A}_2 = \{A_{2,1}, A_{2,2}, \cdots, A_{2,m}\}$$

be two multiple sequences with the same multiplicity, where

$$A_{\tau,s} = (a_{\tau;s,1}, a_{\tau;s,2}, \cdots, a_{\tau;s,n_{\tau,s}}) , \quad \tau = 1, 2 \ .$$

Then, the column superposition of two multiple sequences is defined as

$$\mathcal{A}_0 = \mathcal{A}_1 + \mathcal{A}_2 = (A_{0,1}, A_{0,2}, \cdots, A_{0,m})^T , \tag{5.30}$$

in which

$$A_{0,s} = (A_{1,s}, A_{2,s}) = (a_{1;s,1}, a_{1;s,2}, \cdots, a_{1;s,n_{1,s}} a_{2;s,1}, a_{2;s,2}, \cdots, a_{2;s,n_{2,s}}) \ . \tag{5.31}$$

For the alignment $\mathcal{C}_1, \mathcal{C}_2$ of $\mathcal{A}_1, \mathcal{A}_2$, we can also define the column superposition operation $\mathcal{C}_0 = \mathcal{C}_1 + \mathcal{C}_2$. If $\mathcal{C}_1, \mathcal{C}_2$ are the alignments of $\mathcal{A}_1, \mathcal{A}_2$, then $\mathcal{C}_0$ is the alignment of $\mathcal{A}_0$.

Condition 5.2.7 Convexity of column superposition. If $\mathcal{C}_\tau$ is the alignment of the multiple sequence $\mathcal{A}_\tau$, $\tau = 1, 2$, then $\mathcal{C}_0 = \mathcal{C}_1 + \mathcal{C}_2$ satisfies

$$w(\mathcal{C}_0) = w(\mathcal{C}_1) + w(\mathcal{C}_2) \tag{5.32}$$

These additional conditions are also the natural requirement for MA.

We can easily verify that the SP-penalty function defined by expression (1.9) does not satisfy the Conditions 5.2.4 and 5.2.7. Therefore, we may arrive at some unreasonable results such as; for example,

$$C_{\mathrm{SP}} \left(\begin{pmatrix} - \\ - \\ \vdots \\ - \\ a \end{pmatrix} \right) = C_{\mathrm{SP}} \left(\begin{pmatrix} a \\ a \\ \vdots \\ a \\ - \end{pmatrix} \right) = m - 1 \ .$$

In addition, if

$$\mathcal{C}_0 = \begin{pmatrix} a \\ a \\ c \\ c \end{pmatrix} , \quad \mathcal{C}_1 = \begin{pmatrix} a \\ c \end{pmatrix} , \quad \mathcal{C}_2 = \begin{pmatrix} a \\ c \end{pmatrix} , \tag{5.33}$$

then following from Condition 5.2.4, the penalty functions of $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ must be the same. Nevertheless, for the SP-function ($w_{\mathrm{SP}}(a, b)$ is assumed to be the Hamming matrix), we have

$$w_{\mathrm{SP}}(\mathcal{C}_0) = 4 \,, \quad \text{while} \quad w_{\mathrm{SP}}(\mathcal{C}_1) = w_{\mathrm{SP}}(\mathcal{C}_2) = 1 \ . \tag{5.34}$$

Obviously, the conclusion in (5.34) does not satisfy Condition 5.2.7. Generally, which conclusion, that of the conclusion of the SP-function or that of Condition 5.2.4, is more reasonable is a good question for discussion.

5.2.4 Optimization Problem Based on Shannon Entropy

The goal of alignment is to keep the corresponding components of multiple sequences as consistent as possible while minimizing the number of virtual symbols "–". Using Conditions 5.2.1–5.2.7, we may find that the penalty function of MA is actually a measure of the complexity or uncertainty of multiple sequences. Since Shannon entropy is a natural measurement to describe the uncertainty, it allows us to use the concept of information to describe the optimization criteria. We now introduce some pertinent notations and properties.

Notations for MA $\mathcal{C}$

Let $\mathcal{C}$ be the alignment of the multiple sequence $\mathcal{A}$, then we introduce the following notation to describe the structure of $\mathcal{C}$:

1. For simplicity, we may assume that the row lengths of $\mathcal{A}$ and $\mathcal{C}$ are the same. Then
$$n_1 = n_2 = \cdots = n_m = n\,, \quad n'_1 = n'_2 = \cdots = n'_m = n'\ .$$
Let $a_{\cdot,j}, a_{i,\cdot}$ be the row vector and column vector of $\mathcal{A}$, respectively, and let $c_{\cdot,j}, c_{i,\cdot}$ be the row vector and column vector of $\mathcal{C}$, respectively.
2. Let
$$\chi_z(c_{s,j}) = \begin{cases} 1\,, & \text{if } c_{s,j} = z\,, \\ 0\,, & \text{otherwise} \end{cases}$$
be the indicator function, here $c_{s,j}, z \in V_5$, and let
$$f_{j,z}(\mathcal{C}) = \sum_{s=1}^{m} \chi_z(c_{s,j})\,, \quad j = 1, 2, \cdots, n'\,, \quad z \in V_5 \tag{5.35}$$
be the frequency distribution function of the value of each component in the column vector of the multiple sequence, then obviously we find that
$$f_{j,z} \geq 0\,, \quad \sum_{z=0}^{4} f_{j,z}(\mathcal{A}) = m$$
holds for any $z \in V_5$, $j = 1, 2, \cdots, n'$.
3. Let $\theta_j(\mathcal{C}) = f_{j,4}(\mathcal{C})$, and let
$$p_{j,z}(\mathcal{C}) = \frac{f_{j,z}(\mathcal{C})}{m}\,, \quad j = 1, 2, \cdots, n'\,, \quad z = 0, 1, 2, 3, 4 \tag{5.36}$$
be the frequency distribution function of the jth column of the multiple sequence, then $\sum_{z=0}^{4} p_{j,z}(\mathcal{C}) = 1$ holds; here $f_{j,z}(\mathcal{C}) p_{j,z}(\mathcal{C})$ are actually functions of $c_{j,z}$.
4. In the definitions of (5.35) and (5.36), we may omit the notation $\mathcal{C}$ sometimes, so
$$p_{j,\cdot}(\mathcal{C}) = p_{j,\cdot} = (p_{j,0}, p_{j,1}, p_{j,2}, p_{j,3}, p_{j,4})\ .$$
We then define a function as follows:
$$w_{HG}(\mathcal{C}) = \sum_{j=1}^{n'} HG(c_{\cdot,j}) = \sum_{j=1}^{n'} \left[H(p_{j,\cdot}) + G(\theta_j)\right]\,, \tag{5.37}$$
where
$$HG(c_{\cdot,j}) = H(p_{j,\cdot}) + G(\theta_j), \quad j = 1, 2, \cdots, n'$$
Then $HG(c_{\cdot,j})$ is called the HG function of $\mathcal{C}$, and $G(\theta_j)$ is a strictly monotonically increasing function with $G(0) = 0$.

Definition 29. *In the HG function of $\mathcal{C}$, if*

$$H(p_{j,\cdot}) = -\sum_{z=0}^{4} p_{j,z} \log p_{j,z} \tag{5.38}$$

is a Shannon entropy, then the function $w_{HG}(\mathcal{C})$ is called the S-function, or information-based penalty function, and is denoted by $w_S(\mathcal{C})$.

As this is a very important penalty function, the reader should keep it in mind as we will use it in the text below.

The Selection of the Monotonically Increasing Function $G(\theta)$

There are several possible selections for the monotonically increasing function $G(\theta)$, some of which are listed as follows:

1. Linear function: $G_m(\theta) = \frac{\theta}{g(m)}$, where $g(m)$ is a nondecreasing function of m ($g(m) \geq g(m')$ if $m > m'$) which does not depend on θ. In particular, $g(m)$ may be chosen as a constant with respect to m, and $G_m(\theta)$ is a common linear function of θ for the multiple sequences.
2. Power law function: for example, $G(\theta) = \theta^2$.
3. Logarithmic function: i.e., $G(\theta) = \log(1 + \theta)$, etc.

If we choose $w_S(\mathcal{C})$ as the penalty function to process the optimization problems for multiple sequences, then these types of optimal problems are called information-based optimal problems. As well, the corresponding optimization criteria are called information-based criteria.

Information-Based Criteria of Multiple Sequences

Theorem 24. *If $w(\mathcal{C})$ is an information-based function of MA given by (5.37) and (5.38), then Conditions 5.2.1–5.2.7 hold.*

Since the proof of this theorem is long, we will outline for the reader the role played by this theorem, before providing the proof. It follows from this theorem that the information-based penalty function defined by (5.37) and (5.38) is a penalty function satisfying Conditions 5.2.1–5.2.7. That is, this new penalty function is the best one among all penalty functions.

Proof. The proof of this theorem is divided into nine steps as follows:

1. Verifying that Conditions 5.2.1 and 5.2.2 are satisfied. This is trivial because $H(p_{j\cdot})$, $G(\theta_j)$ are nonnegative functions, and $w_S(\mathcal{C}) = 0$ holds if and only if
$$H(p_{j\cdot}) = 0\,, \quad G(\theta_j) = 0\,, \quad j = 1, 2, \cdots n' \tag{5.39}$$
holds. Furthermore, (5.39) holds if and only if (5.21) holds. In addition, based on the definitions of $w_S(\mathcal{C})$ and $H(p_{j\cdot})$, θ_j, we know that they are symmetric functions with respect to the subscripts s, j. Hence, the symmetric Condition 5.2.2 is true.

2. Verifying Condition 5.2.3. Condition 5.2.3 can be directly proved by the properties of Shannon entropy. This is because $H(p_{j\cdot})$ is maximal if $p_{j\cdot}$ obeys uniform distribution, while it is minimum if $p_{j\cdot}$ obeys binary distribution (in other words,
$$p_{j,z} = \begin{cases} 1\,, & \text{if } z = z_0\,, \\ 0\,, & \text{otherwise}\,, \end{cases}$$
for a fixed $z_0 \in V_4$).
3. Verifying Condition 5.2.4. Let $\mathcal{C}_0 = \mathcal{C}_1 \otimes \mathcal{C}_2$, then both $\mathcal{C}_1$ and $\mathcal{C}_2$ are subsets of $\mathcal{C}_0$. Following from (5.26) and (5.27), we have
$$\mathcal{C}_\tau = \{C_{\tau,1}, C_{\tau,2}, \cdots, C_{\tau,m_\tau}\}\,, \quad \tau = 0,1,2\,, \tag{5.40}$$
where $m_0 = m_1 + m_2$, and
$$C_{\tau,s} = (c_{\tau;s,1}, c_{\tau;s,2}, \cdots, c_{\tau;s,n'})\,, \quad \tau = 0,1,2\,, \quad s = 1,2,\cdots,m_\tau\,. \tag{5.41}$$
We then define
$$\begin{cases} f_{\tau,j}(z) = \sum_{s=1}^{m_\tau} \chi_z(c_{\tau;s,j})\,, \\ \theta_{\tau,j} = f_{\tau,j}(4)\,, \\ p_{\tau,j}(z) = \dfrac{f_{\tau,j}(z)}{m_\tau}\,, \end{cases} \tag{5.42}$$
where $\tau = 0,1,2$, $z \in V_5$, $j = 1,2,\cdots,n'$. We then find that the equations
$$\begin{cases} \theta_{0,j} = \theta_{1,j} + \theta_{2,j}\,, \\ p_{0,j}(z) = \mu_1 p_{1,j}(z) + \mu_2 p_{2,j}(z) \end{cases} \tag{5.43}$$
hold for all $z = 0,1,2,3,4$, and $j = 1,2,\cdots,n$, where $\mu_1 = \frac{m_1}{m_0}$, $\mu_2 = \frac{m_2}{m_0}$.
4. Verifying the convexity of $G(\theta)$. Following from formula (5.37) for the S-penalty function for multiple sequences $\mathcal{C}_0$, and the property that $G(\theta)$ is a monotonically increasing function, we obtain the inequality:
$$G(\theta_{0,j}) = \mu_1 G(\theta_{0,j}) + \mu_2 G(\theta_{0,j}) \geq \mu_1 G(\theta_{1,j}) + \mu_2 G(\theta_{2,j})\,. \tag{5.44}$$
5. Verifying the inequality $\mu_1 w_S(\mathcal{C}_1) + \mu_2 w_S(\mathcal{C}_2) \leq w_S(\mathcal{C}_0)$ in Condition 5.2.4. Let $h(p) = -p\log p$ be the entropy density function, then $h(p) = -p\log p$ is a convex function of the variable $p \in (0,1)$. In fact, $\mu_1 + \mu_2 = 1$ and
$$p_{0,j}(z) = \mu_1 p_{1,j}(z) + \mu_2 p_{2,j}(z)$$
imply the inequality:
$$\begin{aligned} -p_{0,j}(z)\log p_{0,j}(z) &= -[\mu_1 p_{1,j}(z) + \mu_2 p_{2,j}(z)]\log[\mu_1 p_{1,j}(z) + \mu_2 p_{2,j}(z)] \\ &\geq -\mu_1 p_{1,j}(z)\log p_{1,j}(z) - \mu_2 p_{2,j}(z)\log p_{2,j}(z)\,. \end{aligned} \tag{5.45}$$

Taking the sum of z by the two sides of expression (5.45), we have

$$\begin{aligned} H(p_{0;j,\cdot}) &= -\sum_{z=0}^{4} p_{0,j}(z)\log p_{0,j}(z) \\ &= -\sum_{z=0}^{4}[\mu_1 p_{1,j}(z)+\mu_2 p_{2,j}(z)]\log[\mu_1 p_{1,j}(z)+\mu_2 p_{2,j}(z)] \\ &\geq -\sum_{z=0}^{4}[\mu_1 p_{1,j}(z)\log p_{1,j}(z)+\mu_2 p_{2,j}(z)\log p_{2,j}(z)] \\ &= \mu_1 H(p_{1;j,\cdot})+\mu_2 H(p_{2;j,\cdot}) \ . \end{aligned} \tag{5.46}$$

Furthermore, using expressions (5.44) and (5.46), we find that the inequality

$$\mu_1[H(p_{1;j,\cdot})+G(\theta_{1,j})]+\mu_2[H(p_{2;j,\cdot})+G(\theta_{2,j})] \leq H(p_{0;j,\cdot})+G(\theta_{0,j}) \tag{5.47}$$

holds for all $j = 1, 2, \cdots, n'$. Again, taking the sum over j on both sides of expression (5.47), we have the inequality

$$\begin{aligned} &\mu_1 w_S(\mathcal{C}_1)+\mu_2 w_S(\mathcal{C}_2) \\ &= \sum_{j=1}^{n'}\{\mu_1[H(p_{1;j,\cdot})+G(\theta_{1,j})]+\mu_2[H(p_{2;j,\cdot})+G(\theta_{2,j})]\} \\ &\leq \sum_{j=1}^{n'}[H(p_{0;j,\cdot})+G(\theta_{0,j})] = w_S(\mathcal{C}_0) \ . \end{aligned}$$

Hence, the inequality $\mu_1 w_S(\mathcal{C}_1)+\mu_2 w_S(\mathcal{C}_2) \leq w_S(\mathcal{C}_0)$ in Condition 5.2.4 holds.

6. Verifying the sufficient condition for the equation in Condition 5.2.4. If $\theta_{0,j} = 0$ and the relationship

$$p_{1,j}(z) = p_{2,j}(z) = p_{0,j}(z)\,, \quad \forall z \in V_5 \tag{5.48}$$

holds for all $j = 1, 2, \cdots, n'$, then $\theta_{1,j} = \theta_{2,j} = 0$ and then $G(\theta_{1,j}) = G(\theta_{2,j}) = 0$ holds. Furthermore, following from (5.48), we have $H(p_{1;j,\cdot}) = H(p_{2;j,\cdot}) = H(p_{0;j,\cdot})$. It implies that

$$H(p_{0;j,\cdot}) = \mu_1 H(p_{1;j,\cdot})+\mu_2 H(p_{2;j,\cdot}) \ .$$

Thus,

$$H(p_{0;j,\cdot})+G(\theta_{0,j}) = \mu_1[H(p_{1;j,\cdot})+G(\theta_{1,j})]+\mu_2[H(p_{2;j,\cdot})+G(\theta_{2,j})] \ . \tag{5.49}$$

If we sum over j on both sides of (5.49), then we have

$$w_S(\mathcal{C}_0) = \mu_1 w_S(\mathcal{C}_1)+\mu_2 w_s(\mathcal{C}_2) \ ,$$

showing that the equality in expression (5.28) holds.

7. Verifying the necessary condition for the equation in Condition 5.2.4. If the equal sign in expression (5.28) holds and $m_1, m_2 > 0$, then we have $\theta_{0,j} = \theta_{1,j} = \theta_{2,j} = 0$ and (5.48) hold. Since $G(\theta)$ is a strictly monotonically increasing function and $\theta_{0,j} = \theta_{1,j} + \theta_{2,j}$, it follows that
$$G(\theta_{0,j}) \geq \mu_1 G(\theta_{1,j}) + \mu_2 G(\theta_{2,j})$$
holds. Furthermore, the equality holds if and only if $\theta_{0,j} = 0$. On the other hand, following from the strictly convex property of function $H(p_{0;j,\cdot})$, we have
$$H(p_{0;j,\cdot}) \geq \mu_1 H(p_{1;j,\cdot}) + \mu_2 H(p_{2;j,\cdot}) \ .$$
The equality holds if and only if expression (5.48) is true. If the equal sign in expression (5.28) holds and $m_1, m_2 > 0$, then $\theta_{0,j} = 0$ and expression (5.48) holds. In conclusion, the function $w_S(\mathcal{C})$ satisfies Condition 5.2.4.
8. Verifying Conditions 5.2.5–5.2.7. Since Condition 5.2.7 can be directly verified using the definition of the penalty function, we only check that Conditions 5.2.5 and 5.2.6 hold. For Condition 5.2.6, we may assume that the jth row of $\mathcal{C}$ is such that all the elements are "$-$" in the form
$$\begin{pmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \\ c_{m,j} \end{pmatrix} = \begin{pmatrix} - \\ - \\ \vdots \\ - \end{pmatrix} \ .$$
We obtain a new multiple expansion $\mathcal{C}'$ by deleting this purely "$-$" column from $\mathcal{C}$, and then we have $C_S(\mathcal{C}) > C_S(\mathcal{C}')$. Therefore, $\mathcal{C}$ is definitely not the minimum penalty alignment, and Condition 5.2.6 holds. Since verifying Condition 5.2.5 is a long process, we do this in the next step.
9. For verifying Condition 5.2.5, on the one hand, we begin by calculating $HG(c_{1,j}, c_{2,j})$ defined in (5.37) in the case $m = 2$. We then have the following subcases:
   (a) If $(c_{1,j}, c_{2,j}) = (-,-)$, then $\theta_j = 2$. Therefore,
   $$H(p_{j,\cdot}) = 0 \, , \quad G(\theta_j) = G(2) \, , \quad HG(c_{1,j}, c_{2,j}) = G(2) \ .$$
   (b) If $(c_{1,j}, c_{2,j}) = (-, c)$ $\forall c \in \{0,1,2,3\}$, then $\theta_j = 1$. Therefore,
   $$H(p_{j,\cdot}) = 0 \, , \quad G(\theta_j) = G(1) \, , \quad HG(c_{1,j}, c_{2,j}) = G(1) \ .$$
   (c) If $(c_{1,j}, c_{2,j}) = (c, c')$ $\quad \forall c = c' \in \{0,1,2,3\}$, then $\theta_j = 0$. Therefore,
   $$H(p_{j,\cdot}) = 0 \, , \quad G(\theta_j) = G(0) = 0 \, , \quad HG(c_{1,j}, c_{2,j}) = 0 \ .$$
   (d) If $(c_{1,j}, c_{2,j}) = (c, c')$ $\quad \forall c \neq c' \in \{0,1,2,3\}$, then $\theta_j = 0, p_{j,0} = p_{j,1} = 1/2$. Therefore,
   $$H(p_{j,\cdot}) = 1 \, , \quad G(\theta_j) = G(0) = 0 \, , \quad HG(c_{1,j}, c_{2,j}) = 1 \ .$$

As a result, we find the penalty matrix of $HG(c, c'), \forall c, c' \in V_5$ as follows:

$$w(c, c') = \left(\begin{array}{cccccc} & a & c & g & u & - \\ a & 0 & 1 & 1 & 1 & G(1) \\ c & 1 & 0 & 1 & 1 & G(1) \\ g & 1 & 1 & 0 & 1 & G(1) \\ u & 1 & 1 & 1 & 0 & G(1) \\ - & G(1) & G(1) & G(1) & G(1) & G(2) \end{array}\right) . \tag{5.50}$$

On the other hand, to get the penalty matrix for multiple sequences, we choose a function $G(\theta)$ such that $G(2) \geq G(1) \geq 1$. The penalty matrix $w(c, c')$ coincides with the generalized Hamming penalty matrix that is commonly used for pairwise alignment. This ends the proof of this theorem.

Discussion of the Converse Theorem 24

In Theorem 24, we proved that the information-based criterion satisfies Conditions 5.2.1–5.2.7 of the penalty function. We now consider the inverse proposition: what kind of conditions will imply the information-based function defined in (5.38). To solve this problem, we use the definition and properties of Shannon entropy.

Condition 5.2.8 The penalty function $w(\mathcal{C})$ is formed by HG defined by (5.37), $H(p_0, \cdots, p_4)$ is a continuous function of $(p_1, \cdots, p_4)$, and $G(\theta_j)$ is a strictly monotonically increasing function with $G(0) = 0$.

Condition 5.2.9 If $\mathcal{C}_0 = \mathcal{C}_1 \otimes \mathcal{C}_2$ is defined as in Condition 5.2.4, and function $H(\cdot)$ satisfies:

$$H(p_{0;j,\cdot}) = H(\mu_1) + \mu_1 H(p_{1;j\cdot}) + \mu_2 H(p_{2;j,\cdot}) . \tag{5.51}$$

where $h(p) = -p \log p - (1 - p) \log(1 - p)$.

Theorem 25. *If the penalty function $w(\mathcal{C})$ satisfies Conditions 5.2.1–5.2.3, 5.2.8, and 5.2.9, then $w(\mathcal{C})$ is definitely the information-based penalty function defined by (5.37) and (5.38).*

The proof of this theorem is detailed in many informatics books, for example, [23,88], etc. Therefore, we omit it here and refer the reader to other literature sources.

5.2.5 The Similarity Rate and the Rate of Virtual Symbols

Problems of the SP-Penalty (or Scoring) Function and the Information-Based Penalty Function

In previous sections, we defined two important penalty functions: the SP-penalty function and the information-based penalty function, which are frequently used to study MA. We also discussed their roles in the optimal analysis. However, these discussions were not in-depth enough for further study. We must study these two functions with respect to the following:

1. The comparability of the minimum penalty solution must be solved. In other words, we are unable to show a difference between the optimal solution and the minimum penalty solution based on these two functions.
2. The rate of virtual symbols proportional to the length of a sequence. Based on the results of MA, the optimization index for MA often involves the rate of virtual symbols, which will be defined later. The value of the SP-penalty function or the information-based function increases as the rate of virtual symbols increases. Conversely, the value of the SP-penalty function or information-based function decreases as the rate of the virtual symbols decreases. Determining the exact relationship between the rate of virtual symbols and the value of the penalty function is the problem to be discussed.
3. These two functions are unable to construct an optimally fast alignment. Therefore, the optimization criteria of MA similar to the fast MA still need to be discussed further. In this subsection, we focus on finding more optimization indices of MA besides the SP-penalty (or scoring) function and the information-based penalty function.

Similarity Rate

Let $\mathcal{A}$ be a given multiple sequence, so that we may obtain the minimum penalty matrix $\mathcal{B} = (B_{s,t})$ based on $\mathcal{A}$, and the output $\mathcal{C} = \{C_1, C_2, \cdots, C_m\}$. Based on these three elements, we have the following results:

1. A scoring matrix $W = (w_{s,t})_{s,t=1,2,\cdots,m}$ is induced by the matrix $\mathcal{B} = (B_{s,t})$ in the natural way: $w_{s,t} = w(B_{s,t}, B_{t,s})$.
2. A scoring matrix of MA $W' = (w'_{s,t})_{s,t=1,2,\cdots,m}$ is induced by result $\mathcal{C}$ in the natural way: $w'_{s,t} = w(C_s, C_t)$.

We then define the similarity rate as follows:

$$R(\mathcal{C}) = \frac{1}{m(m-1)} \sum_{s=1}^{m} \sum_{t \neq s} \frac{w'_{s,t}}{w_{s,t}} . \tag{5.52}$$

Since $w_{s,t}$ is the score of the minimum penalty alignment based on A_s, A_t, we have that $w'_{s,t} \leq w_{s,t}$ always holds. Hence, $R(\mathcal{C}) \leq 1$ holds. We define $\mathcal{C}$ as the optimal (or suboptimal) alignment of $\mathcal{A}$ if $R(\mathcal{C}) = 1$ (or $R(\mathcal{C}) \sim 1$). Therefore, the similarity rate describes the closeness between the optimal alignment and the minimum penalty alignment.

Rate of Virtual Symbols

The so-called rate of virtual symbols is the proportion of all virtual symbols "$-$" (or 4) in $\mathcal{C}$, namely,

$$P(\mathcal{C}) = \frac{\text{the total of virtual symbols “}-\text{” in } \mathcal{C}}{m \times n}\,, \tag{5.53}$$

where m is the multiplicity of $\mathcal{C}$, and n is the length of each sequence in $\mathcal{C}$.

In conclusion, the challenge of the optimization problem of MA is how to make the value $w_{\text{SP}}(\mathcal{C})$ and the similarity ratio $R(\mathcal{C})$ as large as possible while making the rate of virtual symbols $P(\mathcal{C})$ as small as possible. Or, how to make the rate of virtual symbols as small as possible while making the value $w_{\text{SP}}(\mathcal{C})$ and the similarity rate $R(\mathcal{C})$ as large as possible.

5.3 Super Multiple Alignment

With the above principle in mind, we developed a fast algorithm for MA known as the super multiple alignment (SMA). The associated software package was also developed by the Nankai University group, and is freely available to the public on the website (see Table 5.1). Next, we introduce the relevant materials of SMA.

5.3.1 The Situation for MA

In Sect. 1.1, we introduced the general situation for the algorithms of MA, and we discuss this issue in more detail at this point.

Definition of the MA

In 1982, the pairwise alignment problem had been primarily solved as the Smith–Waterman algorithm was validated. Since then, interest has turned to the question of how to get MA and how to improve the existing pairwise alignment. Almost all bioinformatics literature such as [64] involve MA.

MA is widely used in various fields. For example, to study biological evolution, researchers analyze structural changes based on the MA of special DNA sequences or protein sequences (such as mitochondrial DNA, cytochrome, *C. intestinalis*, etc.). To study the virus genome, MA is also used to get the evolution processes of specific viruses (such as SARS, HIV-1, and various tumors) [101]. As a result, *Paguma larvata* is identified as the source of the SARS virus based on the MA of 63 SARS genome sequences. In contrast, the article [101] used pairwise alignment rather than MA, and as a result, too much information was lost.

Another feature of MA is that the sizes of both the multiplicity m and the lengths of sequences are growing rapidly as work on this problem progresses.

It is common for a MA to involve hundreds of sequences which are hundreds of million base pairs in length. For example, there are 706 HIV-1 sequences in the GenBank 2004 edition (release 43); hopefully, the total number of HIV-1 sequences in all databases combined will exceed 1000. Therefore, there is great demand for fast algorithms of MA for the analysis of these large-scale data.

Progress of MA

The earliest MA algorithm is the MA software package [56], which extended the dynamic programming-based algorithm for pairwise alignment to the multiple cases by changing the penalty matrix to the multiple penalty tensor. The computational complexity of this algorithm is $O(n^m)$, so it is hard to compute as m, n increase. As a result, the scale of this algorithm is only $(m, n) = (7, 300)$. Progress on the improvement of MA is very slow, so it does not keep pace with the exponential speed of the data growth.

After this phase, the study of MA has been developing along two directions. One is to discuss the computational complexity of the solution with minimum penalty, which many publications consider to be a very difficult problem. It was called the first open problem in biological computing in [46], while refs. [15,36,106] call it the NP-hard and Max-Snp hard problem. Hence, it is difficult to achieve MA with minimum penalty theoretically. The MA problems become problems of computational complexity, as described in these publications.

On the other hand, interest in this problem is ongoing because of the importance of MA. Many algorithms, software packages and alignment results appear in the literature one after another. For example, BLAST and FASTA are both able to perform MA. Several specialized software packages, such as CLUSTAL-W/X, BioEdit, MulAlin, GCG, Match-Box, BCM, and CINEMA, etc. are all specific algorithms for MA. The common feature of these algorithms is that they are not concerned with minimum penalty solutions, but result in an increased scale of alignment. These algorithms achieve the suboptimal solutions to some degree, and get a large return for increasing the alignment scale. The alignment scale and the performance indices are shown in Table 5.2.

With MA emerging, the question of how to judge the quality of an algorithm becomes increasingly important. The four indices given in Sect. 1.1.3, namely, the utility range, alignment size, computational speed, and optimization index, are useful when judging the quality of an algorithm. In addition, the SP-penalty function, information-based penalty function, similarity rate and the rate of virtual symbols defined in (1.9), (5.37), (5.38), (5.52), and (5.53), respectively, should also be comprehensively considered if we want to judge the quality of a MA.

Features of the SMA

The purpose of this section is to present a fast algorithm, the so-called super MA (SMA) to fit large-scale MA. Several specific features of the algorithm can be summarized here:

1. **Wide applicability**. This algorithm may still lead to good results if the homology (similarity) between the multiple sequences is only slightly larger than 50%. For instance, we may get good alignment of the DNA sequences of the mitochondria of Primates, although the sequence homology for these sequences ranges from 55 to 90%. In fact, the homology ratio approaches 1, which exceeds our expectations.
2. **Large-scale**. Generally, the computational scale of the SMA is without limitation if a super computer is used. Even running this algorithm on a PC, the size limit of $n \times m$ is beyond 20 Mbp. We may get better results if the size $m \times n$ is less than 20 Mbp and if the homology for these sequences is larger than 80%.
3. **Fast**. On a PC with a 2.8 GHz processor, the alignment of $118 \times 30{,}000$ SARS sequences, takes 21 min; while the alignment of 706×8000 bp HIV-1 sequences takes 34 min. This is much faster than other algorithms.
4. **Highly superior to other algorithms based on three indices**. We compare this algorithm with others based on the following three optimization indices: the SP-scoring function, similarity ratio and ratio of virtual symbols. This algorithm is superior to the other algorithms in all three cases.

The SMA has been published on the Nankai University website [99], and computational service is also offered there. In addition, the alignments for the SARS sequences and HIV-1 sequences are also included on the website. [1]

5.3.2 Algorithm of SMA

For a given multiple sequence $\mathcal{A}$, in order to get its MA, we must first construct an algorithm. To construct an algorithm, we begin by formulating the computational principles.

Principles of MA

Principles of MA include the following:

1. Pairwise alignment. The most popular pairwise alignment include dynamic programming-based algorithms (i.e., the Smith–Waterman algorithm) and the statistical decision-based algorithm (i.e., SPA) [69,90,95].

[1] http://mathbio.nankai.edu.cn/database/exe/sma/PerformanceofSMA/SarsPredictbySMA.txt;
http://mathbio.nankai.edu.cn/database/exe/sma/PerformanceofSMA/HivGeneMatchCompare/

These two kinds of algorithms are easy to compute. Using a dynamic programming-based algorithm, we get the minimum penalty alignment with computational complexity $O(n^2)$, while we may get the suboptimal alignment with the computational complexity $O(n)$ if we use statistical decision-based algorithms. Therefore, we may use the dynamic programming-based algorithms if the lengths of the sequences are less than 10 kbp.

2. Modulus structure. Let (C_s, C_t) be the alignment of (A_s, A_t); then we describe all the virtual symbols in the sequence (C_s, C_t) by a mathematical formula referred to as the modulus structure or alignment mode. The modulus structure is a set of transformations and operations detailed in [89].
3. Clustering analysis of multiple sequences $\mathcal{A}$. Using the characteristics of $\mathcal{A}$ such as length function $n_s = ||A_s||$, $s = 1, 2, \cdots, m$, the scoring matrix of pairwise alignment of $\mathcal{A}$, etc., we construct the phylogenetic tree or the minimum distance tree. Both the phylogenetic and minimum distance trees are typical clustering methods in statistics and combination graph theory [35].

Algorithm of MA

Using the principles of MA, we construct the MA as follows:

Step 5.3.1 Preprocess the relevant parameters and data:

1. Let $M' = \{\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_{2m-1}\}$ be the set of nodes in the clustering tree, where each node $\mathbf{A}_s \in M'$ is a subset of $\mathcal{A} = \{A_1, A_2, \cdots, A_m\}$. Specifically, $\mathbf{A}_s$ is a single-point set, namely, $\mathbf{A}_s = \{A_s\}$ if $s = 1, 2, \cdots, m$, and $\mathbf{A}_s$ is a set with at least two sequences if $s > m$. In some cases, we may simply use the following form:
$$M = \{1, 2, \cdots, m\}, \quad M' = \{1, 2, \cdots, m'\}, \quad m' = 2m - 1 .$$
2. Let $G' = \{M', V'\}$ denote the graph associated with the clustering tree, in which V' is the set of edges in the clustering tree, which will be defined later.
3. Let $w(s,t)$, $s, t \in M$ be the clustering function that may be chosen in many ways, as follows:
   (a) If C_s, C_t is the minimum penalty alignment of A_s, A_t, then choose $w(s,t) = w(C_s, C_t)$.
   (b) Let C_s, C_t be the minimum penalty alignment of A_s, A_t, and let $n(C_s, C_t)$ be the total number of the virtual symbols in C_s, C_t. We choose $w(s,t) = n(C_s, C_t)$.
   (c) If the sequences A_s, A_t are not the same length, we choose $w(s,t) = |n_a - n_t|$.

We now only show the algorithm based on the choice of Step 5.3.1, procedure 3a, leaving analysis of the remaining cases up to the reader.

Step 5.3.2 With the notations defined in Step 5.3.1, we plant the clustering tree based on the multiple sequence $\mathcal{A} = \{A_1, A_2, \cdots, A_m\}$ as follows:

1. Let $M^{(k)} = \{s_1, s_2, \cdots, s_{m-k+1}\} \subset M'$ be the set of states at the kth clustering. It then satisfies the following conditions:
   (a) Each node s_i in $M^{(k)}$ corresponds to a subset of M, denoted by $\mathbf{A}_{s_i}^{(k)}$, here $s_{m-k+1} = m + k$.
   (b) $M^{(1)} = M = \{1, 2, \cdots, m\}$ is the set of states at the initial clustering. Thus, each node s corresponds to a single-point set $\{A_s\}$ if $s \le m$; and it corresponds to a set $\mathbf{A}_s$ with at least two points if $s > m$.
   (c) All the points of $M^{(k)}$ comprise a division of M. In other words, these subsets are mutually disjoint, and the union of them is M.
2. If the $M^{(k)}$ is found, we calculate
$$w_{s,t}^{(k)} = \min\left\{w(s', t'),\ s' \in \mathbf{A}_s^{(k)},\ t' \in \mathbf{A}_t^{(k)}\right\}, \quad s \ne t \in M^{(k)} . \tag{5.54}$$
Let $s'_0 \in \mathbf{A}_s^{(k)}$, $t'_0 \in \mathbf{A}_t^{(k)}$ be the pair of points satisfying $w(s'_0, t'_0) = w_{s,t}^{(k)}$, and let the pair s'_0, t'_0 be the closest nodes within $\mathbf{A}_s^{(k)}$ and $\mathbf{A}_t^{(k)}$. If there is a pair $s_0, t_0 \in M^{(k)}$ such that
$$w_{s_0,t_0}^{(k)} = \min\left\{w^{(k)}(s,t),\ s,t \in M^{(k)}\right\}, \tag{5.55}$$
then the set $M^{(k+1)}$ at the $(k+1)$th clustering is defined by: Let $\mathbf{A}_{m+k}$ denote the union of $\mathbf{A}_{s_0}^{(k)}$ and $\mathbf{A}_{t_0}^{(k)}$, and keep the rest of the nodes invariant. Then, $(s_0, m+k), (t_0, m+k)$ are two edges on the clustering tree G', and $m+k$ is the clustering point of s_0, t_0.
3. Continuing this procedure, we may get the structure for each point of M' defined in Step 5.3.1, and we may also get all the edges in graph G' defined by Step 5.3.2, procedure 2. Finally, we may find the graph of clustering tree G'.

Step 5.3.3 Based on the clustering tree $G' = \{M', V'\}$ obtained by Steps 5.3.1 and 5.3.2, we construct the MA of $\mathcal{A}$ as follows. If r is the clustering point of s, t, then s, t correspond to the union of sets
$$\mathbf{A}_s = \{A_{s,1}, A_{s,2}, \cdots, A_{s,p_s}\}, \quad \mathbf{A}_t = \{A_{t,1}, A_{t,2}, \cdots, A_{t,p_t}\}, \tag{5.56}$$
in which $\mathbf{A}_r = \mathbf{A}_s \cup \mathbf{A}_t$, $\mathbf{A}_s \cap \mathbf{A}_t = \emptyset$, and $\mathbf{A}_s, \mathbf{A}_t$ both are subsets of $\mathcal{A}$. If we found the MA for $\mathbf{A}_s$ and $\mathbf{A}_t$, respectively, then we construct the MA for $\mathbf{A}_r$ in the following way:

1. Let
$$C_s = \{C_{s,1}, C_{s,2}, \cdots, C_{s,p_s}\}, \quad C_t = \{C_{t,1}, C_{t,2}, \cdots, C_{t,p_t}\} \tag{5.57}$$
be the MA for A_s and A_t, respectively, and let
$$H_s = \{H_{s,1}, H_{s,2}, \cdots, H_{s,p_s}\}, \quad H_t = \{H_{t,1}, H_{t,2}, \cdots, H_{t,p_t}\} \tag{5.58}$$
be the expanded modes that A_s, A_t mutates to C_s, C_t, respectively.

2. To cluster, let s', t' be the closest nodes within sets $\mathbf{A}_s$ and $\mathbf{A}_t$, then $A_{s'}, A_{t'} \in \mathcal{A}$. Let $(C_{s'}, C_{t'})$ be the pairwise alignment of $(A_{s'}, A_{t'})$, and let $(H_{s'}, H_{t'})$ be the corresponding expanded mode such that $(A_{s'}, A_{t'})$ mutates to $(C_{s'}, C_{t'})$.
3. Constructing the union modes based on H_s, H_t defined in (5.58) and $(H_{s'}, H_{t'})$ defined in Step 3.5.3, procedure 2, we have two modes as follows:

$$\begin{cases} H_s \vee H_{s'} = \{H_{s,1} \vee H_{s'}, H_{s,2} \vee H_{s'}, \cdots, H_{s,p_s} \vee H_{s'}\}, \\ H_t \vee H_{t'} = \{H_{t,1} \vee H_{t'}, H_{t,2} \vee H_{t'}, \cdots, H_{t,p_t} \vee H_{t'}\}. \end{cases} \tag{5.59}$$

Furthermore, we construct the new mode

$$H_r = H_s \vee H_{s'} \cup H_t \vee H_{t'}. \tag{5.60}$$

This H_r is then the expanded mode by which multiple sequences $\mathbf{A}_r$ mutate to C_r.

Step 5.3.4 Repeating Step 5.3.3 for each clustering point on the tree G' defined by Steps 5.3.1 and 5.3.2, we calculate the MA of each $\mathbf{A}_r$, and finally find the alignment $\mathcal{C}$ of the multiple sequence $\mathcal{A}$.

Step 5.3.5 Generally, the MA $\mathcal{C}$ obtained by Steps 5.3.1–5.3.4 is a suboptimal solution. In order to improve the optimization index of MA, we continue to align $\mathcal{C}$ through the following steps:

1. For each given $s' \in \{1, 2, \cdots, m\}$, let

$$\mathcal{C}_{s'} = \{C_1, C_2, \cdots, C_{s'-1}, C_{s'+1}, \cdots, C_m\}. \tag{5.61}$$

This is a sequence with multiplicity $(m-1)$, where the general form of the component is represented as follows:

$$C_s = (c_{s,1}, c_{s,2}, \cdots, c_{s,n_c}), \tag{5.62}$$

where n_c is the common length for all components. Next, let $M_{s'} = \{1, 2, \cdots, s'-1, s'+1, \cdots, m\}$ denote the set of subscripts of $\mathcal{C}_{s'}$, so that it is a $(m-1)$-ary set.

2. For each column in $\mathcal{C}_{s'}$, calculate its frequency distribution: $\bar{f}_j = (f_{j,c},\ c \in V_{q+1})$, in which, $f_{j,c}$ is the number of the elements of $\bar{c}_{s',j}$ whose value is c. Then, the transpose of this column $\bar{c}_{s',j}$ is

$$\bar{c}_{s',j}^T = (c_{1,j}, c_{2,j}, \cdots, c_{s'-1,j}, c_{s'+1,j}, \cdots, c_{m,j}). \tag{5.63}$$

The SP-penalty function of $\mathcal{C}_{s'}$ is

$$w_{\mathrm{SP}}(\mathcal{C}_{s'}) = \sum_{s<t \in M_{s'}} \sum_{j=1}^{n_c} w(c_{s,j}, c_{t,j}) = \frac{1}{2} \sum_{j=1}^{n_c} \sum_{c \neq c' \in V_{q+1}} f_{j,c} f_{j,c'} w(c, c') \tag{5.64}$$

and the SP-penalty functions of $\mathcal{C}_{s'}$ and $\mathcal{C}$ satisfy the following relationship:

$$w_{\mathrm{SP}}(\mathcal{C}) = \sum_{s<t\in M} \sum_{j=1}^{n_c} w(c_{s,j}, c_{t,j}) = w_{\mathrm{SP}}(\mathcal{C}_{s'}) + \sum_{t=1}^{m} \sum_{j=1}^{n_c} w(c_{s',j}, c_{t,j}) \ . \tag{5.65}$$

Let $w_{\mathrm{SP}}(C_{s'}, \mathcal{C}_{s'}) = \sum_{t=1}^{m} \sum_{j=1}^{n_c} w(c_{s',j}, c_{t,j})$ and choose the $s_0 \in M$ such that

$$w_{\mathrm{SP}}(C_{s_0}, \mathcal{C}_{s_0}) = \max\{w_{\mathrm{SP}}(C_{s'}, \mathcal{C}_{s'}) \, , \ s' \in M\} \ . \tag{5.66}$$

3. Delete these columns of $\mathcal{C}_{s'}$ if they are purely "–" and let $\mathcal{C}'_{s'}$ denote the rest of the multiple sequence. If $C'_{s'} = (c'_{s',1}, c'_{s',2}, \cdots, c'_{s',n'})$ is the expansion of $A_{s'}$, we define the penalty function of $C_{s'}$ and $\mathcal{C}'_{s'}$ as follows:
$$w(C'_{s'}, \mathcal{C}'_{s'}) = \sum_{t=1}^{m} \sum_{j=1}^{n'_c} w\left(c'_{s',j}, c'_{t,j}\right) \ , \tag{5.67}$$
in which $n'_c = \max\{n', n_c\}$.
4. Compute the alignment of A_{s_0} and $\mathcal{C}'_{s_0}$ under the penalty function in (5.67) with the dynamic programming-based algorithm. Let $\mathcal{C}''$ be the output, then $\mathcal{C}''$ is united by C''_{s_0} and $\mathcal{C}''_{s_0}$, where C''_{s_0} is the expansion of A_{s_0}, and $\mathcal{C}''_{s_0}$ is the expansion of $\mathcal{C}'_{s_0}$ by inserting an $(m-1)$-dimensional vector consisting of "–". According to (5.67), we can get the corresponding penalty matrix:
$$w(c, \bar{c}) = \begin{cases} \sum_{c'=0}^{4} f_{c'} w(c, c') \, , & \text{if } \bar{c}' \text{ is a column vector in } \mathcal{C}'_{s_0} \, , \\ m-1 \, , & \text{if } \bar{c}' \text{ is an } (m-1)\text{-dimensional vector} \\ & \text{filled by virtual symbols, and } c' \neq 4 \, , \\ 0 \, , & \text{if } c' = 4, \text{ and } \bar{c}' \text{ is an } (m-1)\text{-dimen-} \\ & \text{sional vector filled by virtual symbols} \, . \end{cases} \tag{5.68}$$
Under this penalty matrix, we may prove that $\mathcal{C}''$ is the optimal alignment of sequence A_{s_0} and $\mathcal{C}'_{s_0}$, and
$$w_{\mathrm{SP}}(\mathcal{C}) \geq w_{\mathrm{SP}}(\mathcal{C}'') \ . \tag{5.69}$$
5. Repeating Step 5.3.5, we continue until the SP-penalty score can no longer be reduced.

Remark 3. The above steps form just the outline for the SMA. It still needs to be adjusted according to specific cases of multiple sequences if we are constructing a program.

Table 5.1. Comparison of the size of multiple alignment

Software package or name of algorithm	Multiplicity restriction	Length restriction	Web page
SMA	No	No	http://mathbio.nankai.edu.cn /eversion/align-query.php
HMMER	No	No	http://hmmer.janelia.org/
POA	No	< 1 kbp	http://www.bioinformatics.ucla.edu/poa
MLAGAN	< 31	Unrestricted	http://genome.lbl.gov/vista /lagan/submit.shtml
ClustalW 1.8	< 500	Unrestricted	http://www.ebi.ac.uk/clustalw/
MuAlin	< 80	< 20 kbp	http://bioinfo.genopole-toulouse.prd.fr /multalin/multalin.html
MSA	< 8	< 800 bp	http://searchlauncher.bcm.tmc.edu /multi-align/multi-align.html
Match-Box	< 50	< 2 kbp	http://searchlauncher.bcm.tmc.edu /multi-align/multi-align.html

Table 5.2. Comparison of the optimization indices

Name of sequence	Scale of alignment	Software package or algorithm	CPU time (min)	SP-score	Similarity rate (%)	Rate of virtual symbols (%)
SARS	118×30 kbp	ClustalW 1.8	4740	9.7×10^7	99.97	0.40
SARS	Same	HMMER 2.2	381	1.0×10^8	99.93	0.47
SARS	Same	SMA	21	1.0×10^8	99.99	0.53
HIV1	706×10 kbp	HMMER 2.2	256	1.65×10^9	98.03	49.13
HIV1	Same	SMA	34	1.68×10^9	98.58	31.23

5.3.3 Comparison Among Several Algorithms

To show how well the SMA performs, we compare it with some popular MA with respect to the indices listed in Tables 5.1 and 5.2.

Remark 4. CPU time is defined as the time required for a PC with a 2.8 GHz processor to compute. The results in Tables 5.1 and 5.2 were obtained in 2004.

Following from Tables 5.1 and 5.2, we draw the following conclusions:

1. For size, SMA is the same as the HMMER 2.2 algorithm, but is far superior to other algorithms.

2. For speed, SMA is 8–18 times faster than the HMMER 2.2 algorithm, as well as 230 times faster than the ClustalW 1.8 algorithm.
3. For the SP-score index and similarity ratio index, SMA is slightly better than the HMMER 2.2 and ClustalW 1.8 algorithms.
4. For the ratio of virtual symbols index, SMA is far superior to the HMMER 2.2 algorithm if we consider the case of HIV1 because its rate of virtual symbols is less 18%. SMA is slightly inferior to HMMER 2.2 algorithm and ClustalW 1.8 if we use SARS as the benchmark set.

As a result we conclude that SMA is generally superior to other MA in terms of size, CPU time, similarity rate and rate of virtual symbols since HMMER 2.2 and ClustalW 1.8 are both the best among existing MA.

5.4 Exercises, Analyses, and Computation

Exercise 21. The metric relations distance, measurement (or probability), and uncertainty are frequently used in mathematics. Compare them, focusing on the aspects of content, definition and difference. For example:

1. Write down the objects they act upon.
2. Construct the basic requirements (axiom system) for these metrics.
3. Write down the expressions of these metrics (i.e., the formula).
4. Write down the definitions of these metrics and indicate in which fields they tend to be applied.

Exercise 22. Check whether or not the SP-penalty functions satisfy Conditions 5.2.1–5.2.7.

Exercise 23. Check whether or not the criterion of similarity rate satisfies Conditions 5.2.1–5.2.7.

Exercise 24. Download the data sets of SARS and HIV-1 from the Web [99], obtain the pairwise alignment using a dynamic programming-based algorithm and the SPA algorithm, respectively, and then analyze the results based on CPU time for pairwise alignment. Compute the matrix consisting of similarity rates based on the Hamming matrix.

Exercise 25. Download the ClustalW algorithm for MA from the Web [22]. Input the SARS and HIV-1 sequences, and compute the alignment output.

Exercise 26. According to the steps in Sect. 5.3, develop a program to obtain the SMA algorithm, and align the SARS and HIV-1 sequences.

Exercise 27. Prove that the expansion C_r obtained by Step 5.3.3 is just the alignment of the multiple sequence A_r.

Exercise 28. Continue Exercises 22 and 23 to analyze MA outputs for SARS and HIV-1 according to the following indices:

1. CPU time and rate of virtual symbols.
2. The SP-penalty function, the information-based function.

Hints

For SMA, we suggest that the reader write a program satisfying the steps presented in Sect. 5.2. If this proves too difficult, the reader may use the algorithm given on the Nankai University website [99], and then try to develop a program independently.