5.5 NP-Completeness of Core Bioinformatics Problems

5.5.1 Multiple Alignment

The MULTIPLE ALIGNMENT problem under sum-of-pairs scoring (MA) is the problem of finding a multiple alignment $T_1, \ldots, T_k$ having *maximum* SP-score for given strings $S_1, \ldots, S_k$ and scoring function σ. Formulated as a decision problem it is to be decided whether a multiple alignment $T_1, \ldots, T_k$ with SP-score *at least* M exists for given strings $S_1, \ldots, S_k$, scoring function σ and lower bound M. Using the specific scoring function σ shown in Fig. 5.18 and strings over the 4-letter alphabet $\Sigma = \{0, 1, \mathrm{a}, \mathrm{b}\}$, we show that SSSEQ can be polynomially reduced to this specialization of SP-Align. Thus SP-Align with this specific scoring function, as well as SP-Align in general is NP-complete (see [77]). Values of the used scoring function σ are chosen in such a way that reduction of SSSEQ to MA and numerical computations are as easy as possible.

σ	0	1	a	b	-
0	-4	-4	-1	-2	-2
1	-4	-4	-2	-1	-2
a	-1	-2	0	-∞	0
b	-2	-1	-∞	0	0
-	-2	-2	0	0	0

Fig. 5.18. Scoring matrix used in the reduction of SSSEQ to MA

Consider now an instance of SSSEQ consisting of binary strings $S_1, \ldots, S_k$ and upper bound m for the length of a common super-sequence. Define

$$s = |S_1| + \ldots + |S_k| \ . \tag{5.1}$$

For both directions of the reduction defined below we require the following lemma.

Lemma 5.30.
Each multiple alignment $T_1, \ldots, T_k$ of $S_1, \ldots, S_k$ has the same SP-score $-2s(k-1)$.

Proof. Consider a fixed column of an alignment $T_1, \ldots, T_k$. Assume that it contains x bits 0 or 1, and $k - x$ spacing symbols. Comparisons between bits and spacing symbols contribute to SP-score the following value:

$$x(k-x)(-2) = -2xk + 2x^2 \ .$$

Comparisons among any two bits contribute to SP-score the following value:

$$\frac{1}{2}(x-1)x(-4) = -2x^2 + 2x \; .$$

Comparisons among any two spacing symbols contribute 0 to the SP-score. Thus, any column containing x bits contributes $-2x(k-1)$ to SP-score. Summarizing over all columns leads to SP-score $-2s(k-1)$ as s was defined to be the number of bits 0 and 1 occurring in $T_1, \ldots, T_k$. □

We now *truth-table reduce* (see Sect. 5.1 for an explanation of this notion) the considered instance consisting of binary strings $S_1, \ldots, S_k$ and upper bound m to $m+1$ different instances of SP-Align, $\Im(i)$, for $i = 0, 1, \ldots, m$, and lower bound M as follows, with $\mathrm{a}^i, \mathrm{b}^{m-i}$ denoting strings that consist of i resp. $m-i$ repetitions of characters 'a' resp. 'b':

$$\Im(i) = S_1, \ldots, S_k, \mathrm{a}^i, \mathrm{b}^{m-i}$$
$$M = -2s(k-1) - 3s \; .$$

Lemma 5.31.
Assume that strings $S_1, \ldots, S_k$ have a super-sequence T of length m consisting of i bits 0 and $j = m - i$ bits 1. Then from T a multiple alignment of strings $S_1, \ldots, S_k, a^i, b^{m-i}$ with SP-score M can be obtained.

Proof.

- Write the characters of S_p below identical characters of T corresponding to an embedding of S_p into super-sequence T, for $p = 1, \ldots, k$.
- Write the characters of string a^i below the i bits 0 of T.
- Write the characters of string b^{m-i} below the $m - i$ bits 1 of T.
- At all positions not filled so far with a character, write a spacing symbol.
- Call the resulting strings $T_1, \ldots, T_k, A, B$. These form a multiple alignment of $S_1, \ldots, S_k, \mathrm{a}^i, \mathrm{b}^{m-i}$.

As an example, Fig. 5.19 shows the multiple alignment constructed this way for strings $S_1 = 01011001$, $S_2 = 101111$, $S_3 = 00000011$, and super-sequence $T = 00010110111011$. Here, there are $i = 6$ bits 0, and $j = 8$ bits 1 in T.

We compute the score of the constructed multiple alignment. The contribution of $T_1, \ldots, T_k$ to SP-score was shown above to be $-2s(k-1)$. Since every 0 in $T_1, \ldots, T_k$ is aligned in A and B to 'a' and -, and every 1 is aligned to - and 'b', the contribution to SP-score that results from comparisons between all of $T_1, \ldots, T_k$ and all of A, B is $-s - 2s = -3s$. Comparison of A and B contributes value 0 to SP-score. Summarizing all contributions leads to a contribution of

$$M = -2s(k-1) - 3s \; .$$

□

T	0	0	0	1	0	1	1	0	1	1	1	0	1	1
T_1	-	0	-	1	0	1	1	0	-	-	-	0	1	-
T_2	-	-	-	1	0	1	-	-	1	1	-	-	1	-
T_3	0	0	0	-	0	-	-	0	-	-	-	0	1	1
A	a	a	a	-	a	-	-	a	-	-	-	a	-	-
B	-	-	-	b	-	b	b	-	b	b	b	-	b	b

Fig. 5.19. Example multiple alignment

Lemma 5.32.
Let i be any number between 0 and m. From a multiple alignment of strings $S_1, \ldots, S_k, \mathrm{a}^i, \mathrm{b}^{m-i}$ with SP-score $\Gamma \geq M$, a super-sequence T of length m for $S_1, \ldots, S_k$ can be obtained.

Proof. Let $T_1, \ldots, T_k, A, B$ be a multiple alignment of $S_1, \ldots, S_k, \mathrm{a}^i, \mathrm{b}^{m-i}$ with SP-score

$$\Gamma \geq M = -2s(k-1) - 3s .$$

Let n be the length of each of the strings $T_1, \ldots, T_k, A, B$. As was already computed in the lemma above, the contribution of strings $T_1, \ldots, T_k$ to Γ is

$$= -2s(k-1) .$$

In order for $\Gamma \geq M$ to hold, comparison of all of strings A, B to all of strings $T_1, \ldots, T_k$ must further contribute to Γ a value

$$\geq -3s .$$

As a consequence, there cannot be any alignment of 'a' with 'b' since this would contribute value $-\infty$ to the SP-score. Thus we conclude that $n \geq m$ holds. Note that every alignment of a bit within one of $T_1, \ldots, T_k$ and a character or spacing symbol within A, B contributes a value as shown in Fig. 5.20 to the overall SP-score. In order for all contributions to sum up to some value $\geq -3s$, each of the s bits of $T_1, \ldots, T_k$ must contribute exactly -3 to Γ. Thus each bit 0 must be aligned with 'a' and -, and each bit 1 must be aligned with - and 'b'. As a consequence, there is no pairing of - with - in the alignment of A and B, thus $n \leq m$. We have now shown that the alignment exactly looks like the alignment constructed in the proof of the lemma before. Taking as string T the string that has bits 0 at all positions where string A has a character 'a', and bits 1 at all positions where string B has a character 'b', defines a string of length m into which every string S_i is embedded. □

Thus we have shown the following theorem.