Non-Approximability of Weighted Multiple Sequence Alignment for Arbitrary Metrics

Bodo Manthey [1]

Universität zu Lübeck, Institut für Theoretische Informatik
Ratzeburger Allee 160, 23538 Lübeck, Germany

Abstract

We prove that the multiple sequence alignment problem with weighted sum-of-pairs score is APX-hard for arbitrary metric scoring functions over the binary alphabet. This holds even when the weights are restricted to zero and one.

Key words: Computational biology, multiple sequence alignment, approximation hardness, sum-of-pairs score

1 Introduction

The multiple sequence alignment problem (MSA) is one of the most fundamental problems in computational biology [10]. One of the most widely used measures for scoring multiple sequence alignments is the *sum-of-pairs score* (SP-score), which is the sum of pairwise distances of the sequences in this alignment. MSA is the problem of finding an alignment with minimum SP-score. Elias [3] proved MSA to be NP-hard for all metric scoring functions over binary alphabets. The currently best approximation algorithm for MSA with SP-score achieves an approximation ratio of $2 - \frac{r}{n}$ for any metric scoring function [2]. Here, n is the number of sequences, and r is an arbitrary fixed constant. It is unknown whether MSA admits a polynomial time approximation scheme [6].

Although widely used, the SP-score is no longer an appropriate measure for multiple alignments if the evolutionary distances between the sequences are

Email address: manthey@tcs.uni-luebeck.de (Bodo Manthey).
URL: www.tcs.uni-luebeck.de/pages/manthey/ (Bodo Manthey).

[1] Supported by DFG research grant RE 672/3.

not evenly distributed. In this case, several highly correlated sequences may dominate the whole alignment. This problem can be solved by using the *weighted SP-score* [5], where we have a non-negative weight for each pair of sequences. The weighted SP-score of an alignment is the sum of all pairwise distances, each multiplied with the corresponding weight. We call the problem of finding an alignment with minimum weighted SP-score *weighted multiple sequence alignment* (WMSA). A restriction of WMSA is *binary weighted multiple sequence alignment* (BMSA), where the weights are restricted to zero and one. BMSA is equivalent to *generalised SP alignment* [7]: In addition to the sequences, we have a subset of pairs of sequences whose pairwise alignments are especially critical. The aim is to find an alignment that minimises the sum of all pairwise alignments of pairs in this subset. Manthey [8] proved that BMSA and WMSA are APX-hard for a three-letter alphabet and one specific metric scoring function. WMSA can be approximated with factor $O(\log n)$, where n is the number of sequences, due to work by Wu et al. [11] and Fakcharoenphol et al. [4].

We will prove the following theorem.

Theorem 1 *For every metric scoring function and every alphabet that contains at least two letters,* WMSA *and* BMSA *are* APX*-hard.*

Throughout the paper, we restrict ourselves to considering WMSA. For any fixed scoring function, the weights used in our proofs are at most linear in the number of sequences. Thus, the APX-hardness holds for BMSA as well, since WMSA with polynomially bounded weights and BMSA are equivalent with respect to their approximability [8].

2 Preliminaries

Let Σ be an alphabet and $\Sigma' = \Sigma \cup \{\text{-}\}$, where - $\notin \Sigma$ denotes the gap. Let S be a sequence of length ℓ over Σ, then $S = S[1]S[2]\ldots S[\ell]$ with $S[k] \in \Sigma$. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a multiset of sequences. An *alignment* of $\mathcal{S}$ is a multiset $\mathcal{A} = \{\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_n\}$ of sequences over Σ', such that all $\tilde{S}_i$ are of equal length $\ell_{\mathcal{A}}$ and $\tilde{S}_i$ is obtained from S_i only by inserting gaps.

Let $d : \Sigma' \times \Sigma' \to \mathbb{R}_0^+$ be a *scoring function.* We allow arbitrary metrics as scoring functions, i.e. for all $x, y, z \in \Sigma'$, we have $d(x, y) \geq 0$ with $d(x, y) = 0$ if and only if $x = y$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$. Given an alignment $\mathcal{A}$ of $\mathcal{S}$, the *cost* of two sequences S_i and S_j is

$$D_{\mathcal{A}}(S_i, S_j) = \sum_{k=1}^{\ell_{\mathcal{A}}} d(\tilde{S}_i[k], \tilde{S}_j[k]) .$$

We omit the index $\mathcal{A}$ when the alignment we are speaking of is clear. Furthermore, we have non-negative integer weights $W = (W_{S_i,S_j})_{S_i,S_j \in \mathcal{S}}$. The *weighted SP-score* of the alignment $\mathcal{A}$ is

$$D_W(\mathcal{A}) = \sum_{1 \le i < j \le n} W_{S_i,S_j} \cdot D_{\mathcal{A}}(S_i, S_j) \ .$$

We omit the index W if the weight matrix is clear. WMSA is the optimisation problem of finding an alignment with minimum weighted SP-score. If we restrict the weights to zero and one, we obtain BMSA. By setting all weights to one, we obtain MSA.

Let us now fix some terms that we will frequently use in the next section. Let again $\mathcal{A} = \{\tilde{S}_1, \dots, \tilde{S}_n\}$ be an alignment of $\mathcal{S} = \{S_1, \dots, S_n\}$. Let $\tilde{k}$ be the position where $S_i[k]$ occurs in $\tilde{S}_i$. We say that $S_i[k]$ *matches* $S_j[k']$ if $\tilde{k}$ is also the position where $S_j[k']$ occurs in $\tilde{S}_j$. We say that $S_i[k]$ *matches a gap* in S_j if $\tilde{S}_j[\tilde{k}] = \text{-}$. If no letter of S_i matches a gap in S_j and no letter of S_j matches a gap of S_i, we say that S_i and S_j are *identically aligned.* When some letter matches a different letter (but not a gap) in some other sequence, we call this a *mismatch.* In the following alignment, the N matching the O is a mismatch, and the other N matches a gap.

```
A L I G N M E N T - -
A L - G O R I - T H M
```

3 Proof

We will now give the proof of Theorem 1, which was stated in the introduction. Throughout the paper, we consider the alphabet $\{0, 1\}$. The scoring function d will be given as $\delta_0 = d(0, \text{-})$, $\delta_1 = d(1, \text{-})$, and $\alpha = d(0, 1)$. Without loss of generality we assume $1 \le \delta_1 \le \delta_0$ and $1 \le \alpha$. We start by considering scoring functions with $\alpha < \delta_0 + \delta_1$ and postpone the case $\alpha = \delta_0 + \delta_1$ to Section 3.2.

We reduce from Max-Cut, which is APX-complete [9]. Max-Cut is the following optimisation problem: Given an undirected graph $G = (V, E)$, we ask for a subset $\tilde{V} \subseteq V$ that maximises the number of edges connecting $\tilde{V}$ to $V \setminus \tilde{V}$. Throughout this work, $G = (V, E)$ is a graph with node set $V = \{v_1, \dots, v_n\}$ and edge set E of cardinality m. Node v_i has degree γ_i and is incident with the edges $e_{i,1}, e_{i,2}, \dots, e_{i,\gamma_i}$ (in arbitrary order).

3.1 The case $\alpha < \delta_0 + \delta_1$

Let $\eta = \min\{1, \delta_0 + \delta_1 - \alpha\} > 0$. We construct a set of sequences that depend on a parameter κ. This parameter depends on the scoring function, and we will set its value later on.

- We have one control sequence $Z = \mathtt{000}\ldots\mathtt{000}$ of length $4\kappa + 4$.
- For each node $v_i \in V$, we have a sequence $X_i = \mathtt{1}\,\mathtt{000}\ldots\mathtt{000}\,\mathtt{1}$ of length $4\kappa + 5$ containing $4\kappa + 3$ 0s.
- Let $e_{i,j} = e_{i',j'} = \{v_i, v_{i'}\}$ be any edge of G. (Without loss of generality we assume $i < i'$.) We represent this edge by two sequences

$$\begin{aligned} Y_{i,j} &= \mathtt{1}\,\underbrace{\mathtt{00}\ldots\mathtt{00}}_{(\kappa+1)\text{ 0s}}\,\underbrace{\mathtt{01010}\ldots\mathtt{1010}}_{\kappa\text{ 1s and }(\kappa+1)\text{ 0s}}\,\underbrace{\mathtt{00}\ldots\mathtt{00}}_{(\kappa+1)\text{ 0s}}\,\mathtt{1}\text{ and} \\ Y_{i',j'} &= \mathtt{1}\,\underbrace{\mathtt{00}\ldots\mathtt{00}}_{(\kappa+1)\text{ 0s}}\,\underbrace{\mathtt{10101}\ldots\mathtt{0101}}_{(\kappa+1)\text{ 1s and }\kappa\text{ 0s}}\,\underbrace{\mathtt{00}\ldots\mathtt{00}}_{(\kappa+1)\text{ 0s}}\,\mathtt{1}\,. \end{aligned}$$

Let $\mathcal{S}$ be the set of all sequences thus constructed.

The weights between the sequences are as follows (we will specify w later):

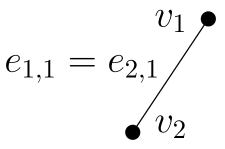
- For $i \in \{1, \ldots, n\}$, we set $W_{Z,X_i} = \gamma_i w$.
- For $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, \gamma_i\}$, we set $W_{X_i,Y_{i,j}} = w$.
- For all edges $e_{i,j} = e_{i',j'}$ of G, we set $W_{Y_{i,j},Y_{i',j'}} = 1$.
- All pairs not mentioned have weight 0.

We call an alignment $\mathcal{A}$ of $\mathcal{S}$ *consistent with* $e_{i,j} = e_{i',j'}$ if

- X_i and $Y_{i,j}$ are identically aligned,
- $X_{i'}$ and $Y_{i',j'}$ are identically aligned,
- only either the first or the last character of X_i matches a gap in Z, and
- only either the first or the last letter of $X_{i'}$ matches a gap in Z.

We call an alignment *consistent* if it is consistent with all edges in E.

If an alignment is consistent with $e_{i,j} = e_{i',j'}$, then $Y_{i,j}$ and $Y_{i',j'}$ are either identically aligned or they are displaced by one position (as $Y_{1,1}$ and $Y_{2,1}$ in Figure 1 are). We obtain a subset $\tilde{V} \subseteq V$ from a consistent alignment by considering the X_i: If the first letter of an X_i matches a gap in Z, then we have $v_i \in \tilde{V}$. If the last letter of the X_i matches a gap in Z, we have $v_i \notin \tilde{V}$. See Figure 1 for an example. If, for an edge $e_{i,j} = e_{i',j'}$, $Y_{i,j}$ and $Y_{i',j'}$ are identically aligned (meaning that either $v_i, v_{i'} \in \tilde{V}$ or $v_i, v_{i'} \notin \tilde{V}$), they cost $(2\kappa + 1) \cdot \alpha$. If they are displaced by one position (meaning that either $v_i \in \tilde{V}$ or $v_{i'} \in \tilde{V}$), they cost $3\alpha + 2\delta_1$. Let $\Delta_\kappa = (2\kappa + 1) \cdot \alpha - (3\alpha + 2\delta_1)$. We choose κ sufficiently large such that Δ_κ becomes positive. Then having exactly one



$$\begin{array}{ll}
Z & = \texttt{- 0 0 0 0 0 0 0 0 0 0 0 0 -} \\
X_1 & = \texttt{1 0 0 0 0 0 0 0 0 0 0 0 1 -} \\
Y_{1,1} & = \texttt{1 0 0 0 0 1 0 1 0 0 0 0 1 -} \\
Y_{2,1} & = \texttt{- 1 0 0 0 1 0 1 0 1 0 0 0 1} \\
X_2 & = \texttt{- 1 0 0 0 0 0 0 0 0 0 0 0 1}
\end{array}$$

Fig. 1. A simple graph and a consistent alignment for $\kappa = 2$ representing $\tilde{V} = \{v_1\}$.

of v_i and $v_{i'}$ in $\tilde{V}$ is cheaper than having both or none of them in $\tilde{V}$.

For all edges $e = e_{i,j} = e_{i',j'}$ of G, we define

$$D_e = D(X_i, Z) + D(X_i, Y_{i,j}) + D(X_{i'}, Z) + D(X_{i'}, Y_{i',j'}) \,.$$

Then

$$D(\mathcal{A}) = \sum_{e=e_{i,j}=e_{i',j'} \in E} w \cdot D_e + D(Y_{i,j}, Y_{i',j'}) \,.$$

The costs $\gamma_i w \cdot D(X_i, Z)$ of X_i with Z are equally distributed among the γ_i edges incident with v_i. We define $K_\kappa = (2\kappa + 3) \cdot \alpha + 2\delta_1$.

Claim 2 *If $\mathcal{A}$ is consistent with e, then $D_e = K_\kappa$. Otherwise, $D_e \geq K_\kappa + \eta$.*

PROOF. Let $e = e_{i,j} = e_{i',j'}$. If $\mathcal{A}$ is consistent with e, we have $D(X_i, Y_{i,j}) = \kappa\alpha$, $D(X_{i'}, Y_{i',j'}) = (\kappa + 1) \cdot \alpha$, and $D(X_i, Z) = D(X_{i'}, Z) = \alpha + \delta_1$.

If $\mathcal{A}$ is not consistent with $e_{i,j}$, we have four possibilities: X_i and $Y_{i,j}$, X_i and Z, $X_{i'}$ and $Y_{i',j'}$, or $X_{i'}$ and Z are inconsistently aligned. Due to symmetry, we only consider the first two cases.

We start with the first case. There is at least one letter of X_i matching a gap in $Y_{i,j}$ and one letter in $Y_{i,j}$ matching a gap in X_i. We call all 1s except for the first and the last of each sequence *internal* 1*s*. If all internal 1s in $Y_{i,j}$ match 0s in X_i, we are done: The internal 1s cost at least $\kappa\alpha$, and additionally we have costs of at least $2\delta_1$ for the two gaps. If an internal 1 in $Y_{i,j}$ matches a 1 in X_i, then we have at least κ 0s and one 1 in $Y_{i,j}$ matching gaps in X_i, which costs at least $\kappa\delta_0 + \delta_1$, and at least $\kappa + 1$ letters in X_i match gaps in $Y_{i,j}$, which costs at least $(\kappa + 1) \cdot \delta_1$.

The case that remains to be considered is that an internal 1 of $Y_{i,j}$ matches a gap in X_i. For every such 1, there is also one letter in X_i matching a gap in $Y_{i,j}$. If that letter is a 0, we are done, since $\delta_0 + \delta_1 \geq \alpha + \eta$. If that letter is a 1, then the first or last 1 of $Y_{i,j}$ matches a 0 in X_i (if it matches a gap again, there must be another letter in X_i matching a gap in $Y_{i,j}$). Thus, every such 1 results in costs of at least $\alpha + 2\delta_1$.

Now we turn to the case that X_i and Z are not consistently aligned. Then either both 1s of X_i match a 0 in Z (then still at least one 0 of X_i matches a gap in Z) or there is a 0 in Z that matches a gap in X_i. In the former case, we have $D(X_i, Z) \geq 2\alpha + \delta_0$. In the latter case, we have $D(X_i, Z) \geq \delta_1 + \min\{\delta_1, \alpha\} + \delta_0$. □

Claim 3 *Let $\mathcal{A}$ be an arbitrary alignment. We can construct a consistent alignment $\tilde{\mathcal{A}}$ with $D(\tilde{\mathcal{A}}) \leq D(\mathcal{A})$ in polynomial time.*

PROOF. Let $I \subseteq E$ be the set of edges e such that $\mathcal{A}$ is not consistent with e. Let $e = e_{i,j} = e_{i',j'}$ be any edge. Due to Claim 2, we have $D_e = K_\kappa$ for $e \notin I$ and $D_e \geq K_\kappa + \eta$ for $e \in I$. If $\mathcal{A}$ is consistent with e, then $D(Y_{i,j}, Y_{i',j'}) \leq (2\kappa+1)\cdot\alpha$.

For all $e \in I$, we realign X_i, $X_{i'}$, $Y_{i,j}$, and $Y_{i',j'}$ to obtain a consistent alignment $\tilde{\mathcal{A}}$. (For both v_i and $v_{i'}$, we choose arbitrarily whether to put them into $\tilde{V}$ or not.) This decreases D_e by at least η due to Claim 2. On the other hand, no $D(Y_{i,j}, Y_{i'j'})$ increases by more than $(2\kappa + 1) \cdot \alpha - (\delta_0 + \delta_1)$. For $w = \left\lceil \frac{(2\kappa+1)\cdot\alpha - \delta_0 - \delta_1}{\eta} \right\rceil$, no $w \cdot D_e + D(Y_{i,j}, Y_{i',j'})$ increases, which completes the proof. □

We have a consistent alignment with cost

$$\begin{aligned} & wmK_\kappa + (2\kappa + 1) \cdot \alpha \cdot (m - c) + (2\delta_1 + 3\alpha) \cdot c \\ = & \, (wK_\kappa + (2\kappa + 1) \cdot \alpha) \cdot m - \Delta_\kappa \cdot c \, , \end{aligned}$$

if and only if the graph G has a cut of size c.

Lemma 4 WMSA *is* APX*-hard for the binary alphabet and all scoring functions d fulfilling $d(0,1) < d(0,\text{-}) + d(\text{-},1)$.*

PROOF. We show that the reduction presented above is an L-reduction [9] (see also Ausiello et al. [1, Def. 8.4]). Let $\mathrm{opt}(\mathcal{S})$ be the cost of an optimal alignment and $\mathrm{opt}(G)$ be the size of a maximum cut. We have $\mathrm{opt}(\mathcal{S}) \leq (wK_\kappa + (2\kappa + 1) \cdot \alpha) \cdot m$ by the choice of κ and $\mathrm{opt}(G) \geq \frac{m}{2}$, since any graph with m edges has a cut of size at least $\frac{m}{2}$. Thus, $\mathrm{opt}(\mathcal{S}) \leq 2 \cdot (wK_\kappa + (2\kappa + 1) \cdot \alpha) \cdot \mathrm{opt}(G)$.

On the other hand, let $\mathcal{A}$ be any alignment with cost $D(\mathcal{A})$. We can construct a consistent alignment $\tilde{\mathcal{A}}$ with $D(\tilde{\mathcal{A}}) \leq D(\mathcal{A})$ in polynomial time. This alignment yields a subset $\tilde{V}$ of the nodes, which yields a cut of size c. Then we have $|\,\mathrm{opt}(G) - c| = \frac{1}{\Delta_\kappa} \cdot |D(\tilde{\mathcal{A}}) - \mathrm{opt}(\mathcal{S})| \leq \frac{1}{\Delta_\kappa} \cdot |D(\mathcal{A}) - \mathrm{opt}(\mathcal{S})|$. □

3.2 *The case* $\alpha = \delta_{\mathsf{0}} + \delta_{\mathsf{1}}$

Now we turn to scoring functions with $\alpha = \delta_{\mathsf{0}} + \delta_{\mathsf{1}}$. The difficulty is that a substitution can be explained by an insertion plus a deletion. The result is that we cannot guarantee consistency when applying the reduction presented in the previous section. Thus, we present a slightly different reduction from Max-Cut.

Given a graph as in the previous sections, we create sequences as follows:

- We have three control sequences

$$
\begin{aligned}
Z_{\text{short}} &= \quad\quad\ \mathsf{1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1}\ ,\\
Z_{\text{med}} &= \quad\ \mathsf{00\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,00}\ , \text{ and}\\
Z_{\text{long}} &= \mathsf{1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1}\ .
\end{aligned}
$$

- For each node $v_i \in V$, we have a sequence

$$X_i = \mathsf{1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1\,0000\,1}\ .$$

- Let $e_{i,j} = e_{i',j'} = \{v_i, v_{i'}\}$ be any edge of G and $i < i'$. We represent this edge by two sequences (the spaces are only for readability):

$$
\begin{aligned}
Y_{i,j} &= \mathsf{1\,0000\,1\,0000\,1} \qquad \mathsf{1\,0000\,1} \qquad \mathsf{1\,0000\,1\,0000\,1}\ ,\\
Y_{i',j'} &= \mathsf{1\,0000\,1} \qquad \mathsf{1\,0000\,1} \qquad \mathsf{1\,0000\,1} \qquad \mathsf{1\,0000\,1}\ .
\end{aligned}
$$

The weights between the sequences are as follows (we will again specify w later on):

- We set $W_{Z_{\text{short}},Z_{\text{med}}} = W_{Z_{\text{med}},Z_{\text{long}}} = mw$.
- For $i \in \{1, \ldots, n\}$, we set $W_{Z_{\text{short}},X_i} = W_{Z_{\text{long}},X_i} = \gamma_i w$.
- For $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, \gamma_i\}$, we set $W_{X_i,Y_{i,j}} = w$.
- For all edges $e_{i,j} = e_{i',j'}$ of G, we set $W_{Y_{i,j},Y_{i',j'}} = 1$.
- All pairs not mentioned have weight 0.

We need a slightly different notion of consistency. An alignment is now called *consistent with* $e_{i,j} = e_{i',j'}$ if the following properties hold:

- All $\mathsf{0}$s and $\mathsf{1}$s in Z_{short} match $\mathsf{0}$s and $\mathsf{1}$s, respectively, in Z_{med}. All $\mathsf{0}$s and $\mathsf{1}$s in Z_{med} match $\mathsf{0}$s and $\mathsf{1}$s, respectively, in Z_{long}.
- One of the following two cases holds for X_i:
  - · The first five letters of X_i match gaps in Z_{short}, and the last five letters of Z_{long} match gaps in X_i. This corresponds to $v_i \in \tilde{V}$.

Z_{short} = -----1000010000100001000010000100001-----

Z_{med} = ---00100001000010000100001000010000100---

Z_{long} = 10000100001000010000100001000010000100001

X_1 = 100001000010000100001000010000100001-----

$Y_{1,1}$ = 10000100001----100001----10000100001-----

$Y_{2,1}$ = -----100001----100001----100001----100001

X_2 = -----100001000010000100001000010000100001

Fig. 2. A consistent alignment representing $\tilde{V} = \{v_1\}$ for the graph shown in Figure 1.

- · The last five letters of X_i match gaps in Z_{short}, and the first five letters of Z_{long} match gaps in X_i. This corresponds to $v_i \notin \tilde{V}$.

  All other letters in X_i, Z_{short}, and Z_{long} match equal letters in the other two sequences.

  The same condition holds for $X_{i'}$.
- All letters in $Y_{i,j}$ match equal letters in X_i. The same holds for $Y_{i',j'}$ and $X_{i'}$.

We call an alignment *consistent* if it is consistent with all edges. See Figure 2 for an example.

Let $e = e_{i,j} = e_{i',j'}$ be any edge and $\mathcal{A}$ be any alignment. We define

$$\begin{aligned} D_e = {} & D(Z_{\text{short}}, Z_{\text{med}}) + D(Z_{\text{med}}, Z_{\text{long}}) + D(X_i, Y_{i,j}) + D(X_{i'}, Y_{i',j'}) \\ & + D(X_i, Z_{\text{short}}) + D(X_i, Z_{\text{long}}) + D(X_{i'}, Z_{\text{short}}) + D(X_{i'}, Z_{\text{long}}) \, . \end{aligned}$$

Then we have

$$D(\mathcal{A}) = \sum_{e=e_{i,j}=e_{i',j'} \in E} w \cdot D_e + D(Y_{i,j}, Y_{i',j'}) \, .$$

The costs $\gamma_i w \cdot (D(X_i, Z_{\text{short}}) + D(X_i, Z_{\text{long}}))$ are equally distributed among the γ_i edges incident with v_i. The costs $mw \cdot (D(Z_{\text{short}}, Z_{\text{med}}) + D(Z_{\text{med}}, Z_{\text{long}}))$ are equally distributed among all m edges.

Claim 5 *If Z_{short}, Z_{med}, and Z_{long} are consistently aligned, then we have $D(Z_{\text{short}}, Z_{\text{med}}) + D(Z_{\text{med}}, Z_{\text{long}}) = 8\delta_0 + 2\delta_1$. Otherwise, $D(Z_{\text{short}}, Z_{\text{med}}) + D(Z_{\text{med}}, Z_{\text{long}}) \geq 8\delta_0 + 3\delta_1$.*

PROOF. If Z_{short}, Z_{med}, and Z_{long} are consistently aligned, then we have $D(Z_{\text{short}}, Z_{\text{med}}) = 4\delta_0$ and $D(Z_{\text{med}}, Z_{\text{long}}) = 4\delta_0 + 2\delta_1$. In every alignment, we have $D(Z_{\text{short}}, Z_{\text{med}}) \geq 4\delta_0$ and $D(Z_{\text{med}}, Z_{\text{long}}) \geq 4\delta_0 + 2\delta_1$.

Assume that Z_{short} and Z_{med} are not consistently aligned. We prove that then $D(Z_{\text{short}}, Z_{\text{med}}) \geq 4\delta_0 + \delta_1$. Assume that there is a mismatch, which costs $\alpha = \delta_0 + \delta_1$. Additionally, at least three 0s in Z_{med} cannot match 0s in Z_{short}, which costs at least $3\delta_0$. If there is no mismatch, at least one letter in Z_{short} matches a gap in Z_{med}, which costs at least δ_1. Additionally, at least four 0s in Z_{med} cannot match 0s in Z_{short}, which costs at least $4\delta_0$.

The proof that Z_{med} and Z_{long} cost at least $4\delta_0+3\delta_1$, if they are not consistently aligned, is very similar, and we therefore omit it. □

Claim 6 *Assume that Z_{short} and Z_{long} are consistently aligned. If X_i is consistently aligned with Z_{short} and Z_{long}, then $D(X_i, Z_{\text{short}}) + D(X_i, Z_{\text{long}}) = 8\delta_0 + 2\delta_1$. If X_i is not consistently aligned with Z_{short} and Z_{long}, then the cost is at least δ_1 higher.*

PROOF. If X_i is consistently aligned with both Z_{short} and Z_{long}, then we have $D(X_i, Z_{\text{short}}) = D(X_i, Z_{\text{long}}) = 4\delta_0 + \delta_1$. In every alignment, we have $D(X_i, Z_{\text{short}}) \geq 4\delta_0 + \delta_1$ and $D(X_i, Z_{\text{long}}) \geq 4\delta_0 + \delta_1$.

Assume that X_i is not consistently aligned with Z_{short} and Z_{long}. Since Z_{short} and Z_{long} are assumed to be consistently aligned, any mismatch of X_i with Z_{short} results in a mismatch of X_i with Z_{long}.

Assume that there is a mismatch between X_i and Z_{long}, which costs α. Additionally, five letters of Z_{long}, at least three of them 0s, cannot match equal letters in X_i, which costs at least $3\delta_0 + 2\delta_1$. Overall, $D(X_i, Z_{\text{long}}) \geq 4\delta_0 + 3\delta_1$. If there is no mismatch, some letter in Z_{short} matches a gap in X_i or some letter in X_i matches a gap in Z_{long}, which costs at least δ_1. In the first case, there are at least six letters in X_i that cannot match equal letters in Z_{short}. At least four of them are 0s. We obtain $D(X_i, Z_{\text{short}}) \geq 4\delta_0 + 3\delta_1$. In the second case, there are at least six letters in Z_{long} that cannot match equal letters in X_i. At least four of them are 0s. We obtain $D(X_i, Z_{\text{long}}) \geq 4\delta_0 + 3\delta_1$. □

The proof of the following claim is obvious and therefore omitted.

Claim 7 *Let $e_{i,j} = e_{i',j'} \in E$ with $i < i'$. If X_i and $Y_{i,j}$ are consistently aligned, then $D(X_i, Y_{i,j}) = 8\delta_0$. Otherwise, $D(X_i, Y_{i,j}) \geq 8\delta_0 + \delta_1$. If $X_{i'}$ and $Y_{i',j'}$ are consistently aligned, then $D(X_{i'}, Y_{i',j'}) = 12\delta_0$. Otherwise, $D(X_{i'}, Y_{i',j'}) \geq 12\delta_0 + \delta_1$.*

In any consistent alignment and for any edge $e = e_{i,j} = e_{i',j'}$, we have

$$D_e = \underbrace{8\delta_0 + 2\delta_1}_{Z_{\text{short}}, Z_{\text{med}}, Z_{\text{long}}} + \underbrace{2 \cdot (8\delta_0 + 2\delta_1)}_{X_i,\ X_{i'} \text{ with } Z_{\text{short}},\ Z_{\text{long}}} + \underbrace{20\delta_0}_{D(X_i, Y_{i,j}) + D(X_{i'}, Y_{i',j'})} = 44\delta_0 + 6\delta_1 \,.$$

Furthermore, we have $D(Y_{i,j}, Y_{i',j'}) = 20\delta_0$ if either $v_i, v_{i'} \in \tilde{V}$ or $v_i, v_{i'} \notin \tilde{V}$, and $D(Y_{i,j}, Y_{i',j'}) = 12\delta_0 + 2\delta_1$ if exactly one of v_i and $v_{i'}$ is in $\tilde{V}$. (We have $12\delta_0 + 2\delta_1 < 20\delta_0$, since $\delta_1 \leq \delta_0$.)

Claim 8 *Let $\mathcal{A}$ be an arbitrary alignment. We can construct a consistent alignment $\tilde{\mathcal{A}}$ with $D(\tilde{\mathcal{A}}) \leq D(\mathcal{A})$ in polynomial time.*

PROOF. Let $I \subseteq E$ be the set of edges e such that $\mathcal{A}$ is not consistent with e. Due to Claims 5, 6, and 7, we have $D_e = 44\delta_0 + 6\delta_1$ for $e \notin I$ and $D_e \geq 44\delta_0 + 7\delta_1$ for $e \in I$.

Let $e = e_{i,j} = e_{i',j'} \in I$. If $\mathcal{A}$ is consistent with e, we have $D(Y_{i,j}, Y_{i',j'}) \leq 20\delta_0$. We now realign X_i, $X_{i'}$, $Y_{i',j'}$, and $Y_{i',j'}$. If necessary, we realign Z_{short}, Z_{med}, Z_{long} as well (this can be done without increasing any edge costs for edges that $\mathcal{A}$ is consistent with). This is done in such a manner that we obtain a consistent alignment. (For both v_i and $v_{i'}$, we choose arbitrarily whether to put them into $\tilde{V}$ or not.)

For $e \in I$, the transformations that are made decrease D_e by at least $w\delta_1$ while $D(Y_{i,j}, Y_{i',j'})$ increases by at most $20\delta_0$. Setting $w = \left\lceil \frac{20\delta_0}{\delta_1} \right\rceil$ completes the proof. □

The reduction presented in this section turns out again to be an L-reduction. Thus, we obtain the following lemma, which completes the proof of Theorem 1.

Lemma 9 WMSA *and* BMSA *are* APX*-hard for the binary alphabet and all scoring functions d with $d(0,1) = d(0,\text{-}) + d(\text{-},1)$.*

Acknowledgement

I thank Jan Arpe and Martin Böhme for valuable discussions and careful proofreading.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, Springer, 1999.

[2] V. Bafna, E. L. Lawler, P. A. Pevzner, Approximation algorithms for multiple sequence alignment, Theoretical Computer Science 182 (1–2) (1997) 233–244.

[3] I. Elias, Settling the intractability of multiple alignment, in: T. Ibaraki, N. Katoh, H. Ono (Eds.), Proc. of the 14th Ann. Int. Symp. on Algorithms and Computation (ISAAC), Vol. 2906 of Lecture Notes in Computer Science, Springer, 2003, pp. 352–363.

[4] J. Fakcharoenphol, S. Rao, K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, Journal of Computer and System Sciences 69 (3) (2004) 485–497.

[5] O. Gotoh, A weighting system and algorithm for aligning many phylogenetically related sequences, Computer Applications in the Biosciences (CABIOS) 11 (5) (1995) 543–551.

[6] T. Jiang, P. E. Kearney, M. Li, Some open problems in computational molecular biology, Journal of Algorithms 34 (1) (2000) 194–201.

[7] G. Lancia, Optimization problems in computational molecular biology, Ph.D. thesis, Graduate School of Industrial Administration, Carnegie Mellon University (May 1997).

[8] B. Manthey, Non-approximability of weighted multiple sequence alignment, Theoretical Computer Science 296 (1) (2003) 179–192.

[9] C. H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, Journal of Computer and System Sciences 43 (3) (1991) 425–440.

[10] P. A. Pevzner, Computational Molecular Biology: An Algorithmic Approach, MIT Press, 2000.

[11] B. Y. Wu, G. Lancia, V. Bafna, K.-M. Chao, R. Ravi, C. Y. Tang, A polynomial-time approximation scheme for minimum routing cost spanning trees, SIAM Journal on Computing 29 (3) (1999) 761–778.