Supplementary Text for Murlet: A practical multiple alignment tool for structural RNA sequences

Hisanori Kiryu, Yasuo Tabei, Taishin Kin, and Kiyoshi Asai

January 20, 2007

1 Algorithm for Reducing DP Region

We use the convention that the i-th base of the first sequence that is inserted between $j-1$ and j-th base of the second sequence is emitted at position (i, j) in the DP matrix. In this convention, the size of the DP matrix is $(L^{(1)}+1) \times (L^{(2)}+1)$ for sequences of lengths $L^{(1)}$ and $L^{(2)}$. and the ranges of row and column indexes are $1 \leq i \leq (L^{(1)}+1)$ and $1 \leq j \leq (L^{(2)}+1)$ respectively, in order to account for the $3'$-terminal gap insertions. We represent the DP region by two arrays of left and right column boundaries $j_l[i]$ and $j_r[i]$ in the DP matrix. Using these arrays, the DP region is represented by the set $\{(i,j)|1 \leq i \leq (L^{(1)}+1), j_l[i] \leq j \leq j_r[i]\}$.

Algorithm 1 shows the algorithm for reducing the DP region. The initial DP region is represented as $j_l[i]$ and $j_r[i]$. These boundaries are modified to represent the reduced DP region after the computation. The algorithm requires as input the initial DP region $j_l[i]$ and $j_r[i]$, the match probability matrix $p^{(a)}$, the threshold value ϵ and the minimum DP region that enclose the initial DP path, which is represented by $j_{l0}[i]$ and $j_{r0}[i]$.

The reduced DP region has several properties.

- The region is simply connected. In other words, the region has no holes. This is obvious since each slice of the region by rows is represented by only one segment $j_l[i] \leq j \leq j_r[i]$.
- The region includes the initial alignment path $j_l[i] \leq j_{l0}[i] \leq j_{r0}[i] \leq j_r[i]$.
- $j_r[i] \leq j_r[i+1]$ and $j_l[i] \leq j_l[i+1]$.
- for each position (i, j) that has match probability $p^{(a)} > \epsilon$ and is right of the initial path $j > j_{r0}[i]$, the lower left region $\{(i', j')|i' = i, j_{l0}[i'] < j' \leq j\} \cup \{(i', j')|i' > i, j_{r0}[i'] < j' \leq (j+1)\}$ is contained in the reduced DP region.

Algorithm 1 Algorithm for reducing the dynamic programming region. $j_l[i]$ and $j_r[i]$ are the left and right column boundaries of the DP region at row i. On input, $j_l[i]$ and $j_r[i]$ represent the strip region around the initial DP alignment path. On output, $j_l[i]$ and $j_r[i]$ represent the reduced DP region. $j_{l0}[i]$ and $j_{r0}[i]$ are the boundaries of the minimum DP region that enclose the initial DP path. $p^{(a)}(i,j)$ is assumed to returns an element of the match probability matrix at position (i,j) if $1 \leq i \leq L^{(1)}$ and $1 \leq j \leq L^{(2)}$, and returns 0 otherwise.

Input: $j_l[\cdot]$, $j_r[\cdot]$, $j_{l0}[\cdot]$, $j_{r0}[\cdot]$, ϵ, $p^{(a)}(\cdot,\cdot)$
Output: $j_l[\cdot]$, $j_r[\cdot]$

$j_0 \leftarrow 1$
for $i \leftarrow 1 \cdots (L^{(1)}+1)$ **do**
  $j_0 \leftarrow \max(j_0, j_{r0}[i])$
  $j \leftarrow j_r[i]$
  $j_r[i] \leftarrow j_0$
  while $j \geq j_0$ **do**
    if $\epsilon \leq p^{(a)}(i,j)$ **then**
      $j_r[i] \leftarrow j$
      $j_0 \leftarrow (j+1)$
      break
    end if
    $j \leftarrow (j-1)$
  end while
end for
$j_0 \leftarrow L^{(2)}$
for $i \leftarrow (L^{(1)}+1) \cdots 1$ **do**
  $j_0 \leftarrow \min(j_0, j_{l0}[i])$
  $j \leftarrow j_l[i]$
  $j_l[i] \leftarrow j_0$
  while $j \leq j_0$ **do**
    if $\epsilon \leq p^{(a)}(i-1,j-1)$ **then**
      $j_l[i] \leftarrow j$
      $j_0 \leftarrow (j-1)$
      break
    end if
    $j \leftarrow (j+1)$
  end while
end for

- for each position (i, j) that has match probability $p^{(a)} > \epsilon$ and is left of the initial path $j < j_{l0}[i]$, the upper right region $\{(i', j')|i' = (i+1), (j+1) \leq j' < j_{l0}[i']\} \cup \{(i', j')|i' \leq i, j \leq j' < j_{l0}[i']\}$ is contained in the reduced DP region.

From the last two properties, it follows that for any position pair (i, j) and (i', j') that have match probabilities $p^{(a)}(i, j), p^{(a)}(i', j') > \epsilon$ and can coexist in an alignment (i.e. ($i < i'$ and $j < j'$) or ($i' < i$ and $j' < j$)), there exists at least one alignment path in the reduced DP region that connect these positions. In fact, the reduced DP region is given by the union of the region corresponding to $j_{l0}[i]$ and $j_{r0}[i]$ and all the upper-left and lower-right regions that are described in the last two properties.

2 Proof of $0 \leq q_x^{(b)\mathbf{PCT}}(i) \leq 1$

In this section, we prove the formula

$$0 \leq q_x^{(b)\mathrm{PCT}}(i) \leq 1 \tag{1}$$

where $q_x^{(b)\mathrm{PCT}}(i)$ is defined by

$$q_x^{(b)\mathrm{PCT}}(i) = 1 - \frac{1}{N} \sum_{w \in X} \left[\sum_{1 \leq j < i} t_{xw}(j, i) + \sum_{i < j \leq L_x} t_{xw}(i, j) \right]$$
$$t_{xw}(i, j) = \sum_{1 \leq k < l \leq L_w} p_{x,w}^{(a)}(i, k) p_{x,w}^{(a)}(j, l) p_w^{(b)}(k, l)$$

The proof proceeds as follows. Since $t_{xw}(i, j) \geq 0$, the inequality $q_x^{(b)\mathrm{new}}(i) \leq 1$ is obviously satisfied. Hence, we prove only the inequality,

$$\sum_{1 \leq j < i} t_{xw}(j, i) + \sum_{i < j \leq L_x} t_{xw}(i, j) \leq 1 \tag{2}$$

for fixed i. The first term of the above formula can be bounded from above as follows:

$$\sum_{1 \leq j < i} t_{xw}(j, i) = \sum_{1 \leq k < l \leq L_w} \left[\sum_{1 \leq j < i} p_{x,w}^{(a)}(j, k) \right] p_{x,w}^{(a)}(i, l) p_w^{(b)}(k, l)$$
$$\leq \sum_{1 \leq k < l \leq L_w} p_{x,w}^{(a)}(i, l) p_w^{(b)}(k, l)$$

The expression in the square bracket is not greater than one since it is the probability that the position k of sequence w is aligned to the range between 1 and $i - 1$ of sequence x. Similarly, the second term satisfies the inequality.

$$\sum_{i < j \leq L_x} t_{xw}(i, j) \leq \sum_{1 \leq k < l \leq L_w} p_{x,w}^{(a)}(i, k) p_w^{(b)}(k, l)$$

Hence, the left-hand-side lhs of Equation 2 satisfies the formula:

$$\begin{aligned} lhs &\leq \sum_{1 \leq k \leq L_w} p_{x,w}^{(a)}(i,k) \left[\sum_{1 \leq l < k} p_w^{(b)}(l,k) + \sum_{k < l \leq L_w} p_w^{(b)}(k,l) \right] \\ &\leq \sum_{1 \leq k \leq L_w} p_{x,w}^{(a)}(i,k) \\ &\leq 1 \end{aligned}$$

In the above formula, the expression inside the square bracket is not greater than one because it is the probability that the position k forms any base pair with other positions. Further, since the right-hand-side of the second inequality represents the probability that the position i of sequence x is aligned to any position of sequence w, the last inequality follows. Thus, the formula 1 is proved.

3 Novel Accuracy Measures: SQS, SSS, and PCS

To define SQS, SSS and PCS mathematically, We first give a few definitions. Let $\iota_{\mathcal{A}}^{(h)}$ be the mapping from the position $i \in \mathcal{C}^{(h)}$ of sequence $x^{(h)}$ to the corresponding alignment column $I \in \mathcal{C}_{\mathcal{A}}$ in the alignment $\mathcal{A}$

$$\begin{gathered} \iota_{\mathcal{A}}^{(h)} : \mathcal{C}^{(h)} \longrightarrow \mathcal{C}_{\mathcal{A}} \\ h = 1, \cdots, N \end{gathered}$$

For each consensus secondary structure $\mathcal{S} = \{\mathcal{L}, \mathcal{P}\}$ of the alignment $\mathcal{A}$, the secondary structure $\mathcal{S}^{(h)}$ of sequence $x^{(h)}$ associated to $\mathcal{S}$ is defined by,

$$\begin{aligned} \mathcal{S}^{(h)} &= \{\mathcal{L}^{(h)}, \mathcal{P}^{(h)}\} \\ \mathcal{P}^{(h)} &= \{(i,j) \in \mathcal{PC}^{(h)} | \exists (I,J) \in \mathcal{P}, I = \iota^{(h)}(i), J = \iota^{(h)}(j)\} \\ \mathcal{L}^{(h)} &= \{i \in \mathcal{C}^{(h)} | \forall (i',j') \in \mathcal{P}^{(h)}, i \neq i', j'\} \end{aligned}$$

For each alignment column I in the alignment $\mathcal{A}$, the column vector $c_{\mathcal{A},I}$ is defined as follows,

$$\begin{gathered} c_{\mathcal{A},I}(h) = \begin{cases} \text{'-' if the column } I \text{ is a gap position for sequence } x^{(h)} \\ \iota_{\mathcal{A}}^{(h)-1}(I) \end{cases} \\ h = 1, \cdots, N \end{gathered}$$

where $i = \iota_{\mathcal{A}}^{(h)-1}(I)$ is the position of sequence $x^{(h)}$ aligned at the column I.

To compute SQS, the number of quadruples $((i,j),(k,l)) \in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')}$ satisfying the following constraint is computed for each pair $(x^{(h)}, x^{(h')})$ of se-

quences.

$$\begin{aligned}((i,j),(k,l)) &\in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')} \\ \iota^{(h)}_{\mathrm{ref}}(i) &= \iota^{(h)}_{\mathrm{ref}}(k) \\ \iota^{(h)}_{\mathrm{ref}}(j) &= \iota^{(h)}_{\mathrm{ref}}(l) \\ \iota^{(h)}_{\mathrm{sbj}}(i) &= \iota^{(h)}_{\mathrm{sbj}}(k) \\ \iota^{(h)}_{\mathrm{sbj}}(j) &= \iota^{(h)}_{\mathrm{sbj}}(l)\end{aligned}$$

where the subscripts ref and sbj indicate the reference alignment and the subject alignment being evaluated, respectively. Then the count is summed over all the pairs of sequences. The SQS is obtained by taking the ratio of the count of the subject alignment to that of the ideal alignment that is identical to the reference alignment. To compute SSS, the number of quadruples that satisfies the constraint

$$\begin{aligned}((i,j),(k,l)) &\in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')} \\ \iota^{(h)}_{\mathrm{sbj}}(i) &= \iota^{(h)}_{\mathrm{sbj}}(k) \\ \iota^{(h)}_{\mathrm{sbj}}(j) &= \iota^{(h)}_{\mathrm{sbj}}(l)\end{aligned}$$

is computed. The SSS value is obtained by taking the ratio between the count of the subject alignment and that of the ideal alignment. To compute PCS, the number of pair columns (I, J) that satisfies the constraint

$$\begin{aligned}(I,J) &\in \mathcal{PC}_{\mathrm{sbj}} \\ \exists(K,L) &\in \mathcal{P}_{\mathrm{ref}} \\ c_{\mathrm{ref},K} &= c_{\mathrm{sbj},I} \\ c_{\mathrm{ref},L} &= c_{\mathrm{sbj},J}\end{aligned}$$

is calculated. The PCS value is obtained by taking the ratio between the count of the subject alignment and that of the ideal alignment.

Figure 1 shows examples of the alignments. (a) is the reference alignment, (b) is the subject alignment and the alignment (c) is a copy of the reference alignment used for the comparison. The secondary structures of sequences in the three alignments are derived from the structure annotated to the reference, which are shown in the bottom part of the figure, where the aligned bases are replaced with the corresponding sequence positions. Figure 2 shows examples of the computation of SQS and SSS values for the multiple alignment of Figure 1. For the SQS computation, three quadruples $((1,7),(1,9))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$, $((2,8),(1,7))$ and $((3,7),(2,6))$ in $\mathcal{P}^{(2)} \times \mathcal{P}^{(3)}$ contribute to the count for the 'subject' alignment (Figure 2(a)), while five quadruples $((1,7),(1,9)$ and $((2,6),(2,8))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$, $((2,6),(1,7))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$, $((2,8),(1,7))$ and $((3,7),(2,6))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$ contribute to the count for the 'subject0' alignment (Figure 2(b)). The SQS value is given by the ratio $3/5 = 0.6$. The count

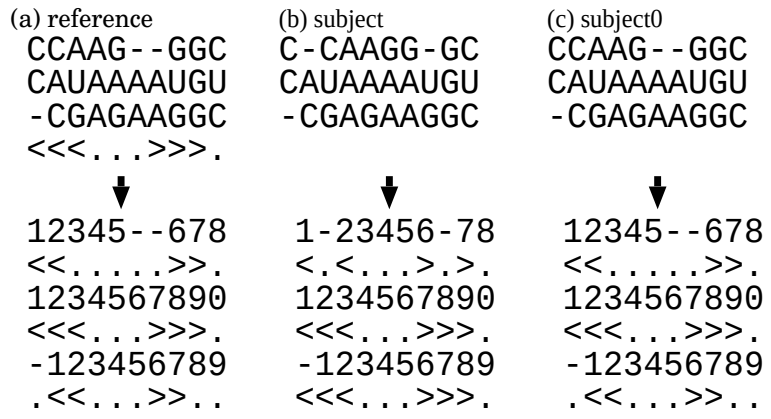

Figure 1: Derivation of the secondary structures of sequences from the consensus secondary structure of the alignment.

that contributes to SQS also contributes to the count for SSS. However the quadruples $((2,6),(3,7))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$ and $((2,6),(2,6))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$ also contribute to the SSS count (Figure 2(c)). The count for the 'subject0' alignment is unchanged from that of SQS (Figure 2(d)). Therefore, SSS is given by $(3+2)/5 = 1$. Figure 3 shows an example of the PCS computation. Since the pair of column vectors $((1,1,-),(7,9,8))$ exists both in the reference and 'subject' alignments and these columns are annotated to form a base pair in the reference alignment, The pair column $(I,J) = (1,9)$ in $\mathcal{PC}_{\rm sbj}$ contributes to the count of PCS (Figure 3(b)). Similarly the three pair columns $(1,9)$, $(2,8)$ and $(3,7)$ in $\mathcal{PC}_{\rm sbj0}$, whose pair column vectors are $((1,1,-),(7,9,8))$, $((-,2,1),(-,8,7))$, and $((2,3,2),(6,7,6))$, respectively, contribute to the count for the 'subject0' alignment (Figure 3(c)). The PCS value is then given by the ratio $1/3 \approx 0.33$.

4 Consensus Structure Prediction By Stemloc, PMMulti, and RNAcast+RNAforester

Table 1 shows the MCC values of the Pfold predictions to the Stemloc, PMMMulti, and RNAcast+RNAforester alignments and the original consensus structure predictions made by these programs for the dataset of Table 2 in the main text. The table shows that the accuracies of the original predictions made by Stemloc and PMMulti are almost 10% lower than those of Pfold predictions. For RNAcast+RNAforester, the Pfold predictions are slightly better than the original predictions.

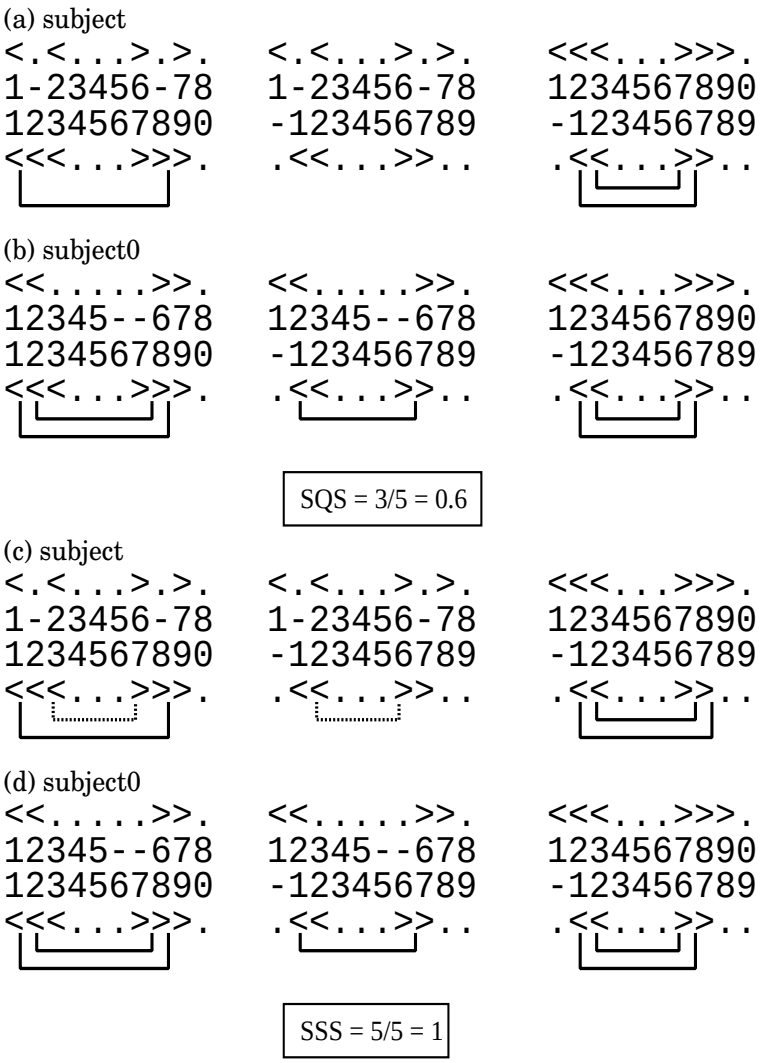

Figure 2: Examples of the computation of SQS and SSS. The left, center, and right alignments correspond to the sequence pairs $(h, h') = (1, 2), (1, 3)$, and $(2, 3)$, respectively.

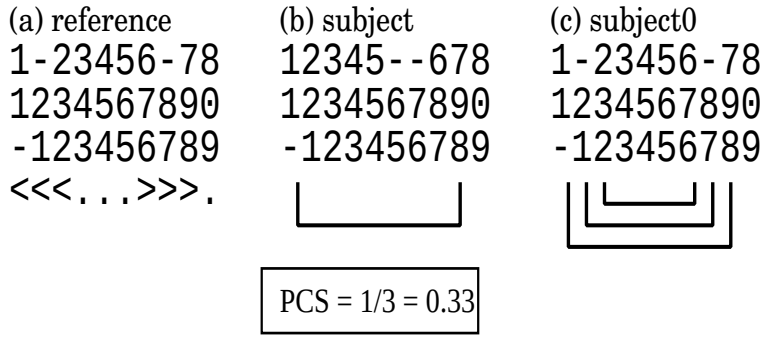

Figure 3: An example of the computation of PCS.

Table 1: Comparison of MCC values between the predictions made by Pfold and those made by Stemloc, PMMulti, and RNAcast+RNAforester. The first columns "Average(Stemloc)," "Average(PMMulti)," etc. have the same meanings as those in the main text. "original" indicates the MCC values for the original predictions made by the alignment programs.

	Stemloc Pfold / original	PMMulti Pfold / original	RNAcast Pfold / original
Average (Stemloc)	0.67 / 0.58	– / –	– / –
Average (PMMulti)	– / –	0.54 / 0.42	– / –
Average (RNAcast)	– / –	– / –	0.55 / 0.54
Average (common)	0.74 / 0.65	0.59 / 0.48	0.62 / 0.61