Multiple Alignment

Anders Gorm Pedersen
Molecular Evolution Group
Center for Biological Sequence Analysis
gorm@cbs.dtu.dk
Refresher: pairwise alignments

43.2% identity;  
Global alignment score: 374

alpha  V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
beta   VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDSLSTPDAVMGNP

10  20  30  40  50

60  70  80  90  100  110

alpha  QVKGHGKKVADALTNAHAVHDDMPNALSAALSALHLAHKLKLRVDPVFKNLHSLHCLLVTLAHL
beta   KVKAHGKKVLGAFSDGLAHLDNLKTFATLSELHCDKLVDPENFRLLGNVLVCVALAHHF

120  130  140

alpha  PAEFTPAVHASLDFLASSVSTVLTYSKR
beta   GKEFTPPVQAAYQKVAGVANALAHKYH

120  130  140
Refresher: pairwise alignments

- Alignment score is calculated from substitution matrix
- Identities on diagonal have high scores
- Similar amino acids have high scores
- Dissimilar amino acids have low (negative) scores
- Gaps penalized by gap-opening + gap elongation

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>R</th>
<th>N</th>
<th>D</th>
<th>C</th>
<th>Q</th>
<th>E</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>-2</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>-1</td>
<td>-1</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>-4</td>
<td>-2</td>
<td>-4</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-3</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-3</td>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>8</td>
</tr>
</tbody>
</table>

KLAA SVILSDAL
KLAA--SSDAL

-10 + 3 x (-1) = -13
Refresher: pairwise alignments

The number of possible pairwise alignments increases explosively with the length of the sequences:

Two protein sequences of length 100 amino acids can be aligned in approximately $10^{60}$ different ways

$10^{60}$ bottles of beer would fill up our entire galaxy
Refresher: pairwise alignments

- **Solution:**
  dynamic programming

- **Essentially:**
  the best path through any grid point in the alignment matrix must originate from one of three previous points

- **Far fewer computations**

- **Best alignment guaranteed to be found**
Refresher: pairwise alignments

- Most used substitution matrices are themselves derived empirically from simple multiple alignments

\[
\text{Score}(A/C) = \log \frac{\text{Freq}(A/C)_{\text{observed}}}{\text{Freq}(A/C)_{\text{expected}}}
\]

A/A  2.15%
A/C  0.03%
A/D  0.07%
\ldots
Multiple alignment
Multiple alignments: what use are they?

- Starting point for studies of molecular evolution
Multiple alignments: what use are they?

- Characterization of protein families:
  - Identification of conserved (functionally important) sequence regions
  - Construction of profiles for further database searching
  - Prediction of structural features (disulfide bonds, amphipathic alpha-helices, surface loops, etc.)
Scoring a multiple alignment: the “sum of pairs” score

One column from alignment

\[ \begin{align*}
  \text{A} & : 4, \text{A} & : 1, \text{A} & : 0 \\
  \text{S} & : 1, \text{T} & : 0 \\
  \text{ST} & : 1 \\
\end{align*} \]

SP-score: \(4 + 1 + 0 + 1 + 0 + 1 = 7\)

**Weighted sum of pairs**: each SP-score is multiplied by a weight reflecting the evolutionary distance (avoids undue influence on score by sets of very similar sequences)

=> In theory, it is possible to define an alignment score for multiple alignments (there are several alternative scoring systems)
Multiple alignment: dynamic programming is only feasible for very small data sets

- In theory, optimal multiple alignment can be found by dynamic programming using a matrix with more dimensions (one dimension per sequence)

- BUT even with dynamic programming finding the optimal alignment very quickly becomes impossible due to the astronomical number of computations

- Full dynamic programming only possible for up to about 4-5 protein sequences of average length

- Even with heuristics, not feasible for more than 7-8 protein sequences

- Never used in practice

For 3 sequences, optimal path must come from one of 7 previous points
Multiple alignment: an approximate solution

- Progressive alignment (ClustalX and other programs):

1. Perform all *pairwise* alignments; keep track of sequence similarities between all pairs of sequences (construct “distance matrix”)

2. Align the most similar pair of sequences

3. Progressively add sequences to the (constantly growing) multiple alignment in order of decreasing similarity.
Progressive alignment: details

1) Perform all pairwise alignments, note pairwise distances (construct “distance matrix”)

   S1  S2  S3  S4
   S1  3   1   3
   S2  3   2   3
   S3  3   2   3

2) Construct pseudo-phylogenetic tree from pairwise distances
3) Use tree as guide for multiple alignment:
   a) Align most similar pair of sequences using dynamic programming

   \[ \begin{align*}
   S1 & \quad \quad \quad \quad \quad \quad \quad \\
   S3 & \quad \quad \quad \quad \quad \quad \\
   \end{align*} \]

   b) Align next most similar pair

   \[ \begin{align*}
   S2 & \quad \quad \quad \quad \quad \quad \quad \quad \quad \\
   S4 & \quad \quad \quad \quad \quad \quad \\
   \end{align*} \]

c) Align alignments using dynamic programming - preserve gaps

   \[ \begin{align*}
   S1 & \quad \quad \quad \quad \quad \quad \\
   S3 & \quad \quad \quad \quad \quad \quad \\
   S2 & \quad \quad \quad \quad \quad \quad \\
   S4 & \quad \quad \quad \quad \quad \\
   \end{align*} \]

   New gap to optimize alignment of \((S2,S4)\) with \((S1,S3)\)
Scoring profile alignments

Compare each residue in one profile to all residues in second profile. Score is average of all comparisons.

\[ \text{Score: } \frac{1+0+4+1}{4} = 1.5 \]
Additional ClustalX heuristics

• **Sequence weighting:**
  – scores from similar groups of sequences are down-weighted

• **Variable substitution matrices:**
  – during alignment ClustalX uses different substitution matrices depending on how similar the sequences/profiles are

• **Variable gap penalties:**
  • gap penalties depend on substitution matrix
  • gap penalties depend on similarity of sequences
  • reduced gap penalties at existing gaps
  • increased gap penalties CLOSE to existing gaps
  • reduced gap penalties in hydrophilic stretches (presumed surface loop)
  • residue-specific gap penalties
  • and more...
Other multiple alignment programs

pileup
multalign
multal
saga
hmmt
MUSCLE
ProbCons

DIALIGN
SBpima
MLpima
T-Coffee
mafft
poa
prank
...
...
Quantifying the Performance of Protein Sequence Multiple Alignment Programs

• Compare to alignment that is known (or strongly believed) to be correct

• Quantify by counting e.g. fraction of correctly paired residues

• Option 1: Compare performance to benchmark data sets for which 3D structures and structural alignments are available (BALiBASE, PREfab, SABmark, SMART).
  – Advantage: real, biological data with real characteristics
  – Problem: we only have good benchmark data for core regions, no good knowledge of how gappy regions really look

• Option 2: Construct synthetic alignments by letting a computer simulate evolution of a sequence along a phylogenetic tree
  – Advantage: we know the real alignment including where the gaps are
  – Problem: Simulated data may miss important aspects of real biological data
Performance on BALiBASE benchmark

Fig. 1. Color coded matrix showing which method performed best for each pair-combination of conditions: average sequence length (x-axis) and average evolutionary distance (y-axis). The methods are Poa (green), Dialign (yellow), T-Coffee (blue) and ClustalW (red).
Performance on simulated data, few gaps
Performance on simulated data, many gaps
So which method should I choose?

• Performance depends on way of measuring and on nature of data set

• No single method performs best under all conditions (although mafft and ProbCons look quite good)

• To be on the safe side, you ought to check that results are robust to alignment uncertainty (try a number of methods, check conclusions on each alignment)

• Future perspectives: Bayesian techniques, alignment inferred along with rest of analysis, conclusions based on probability distribution over possible alignments.
Special purpose alignment programs

- **RevTrans**: alignment of coding DNA based on information at protein level
- Codon-codon boundaries maintained

Figure 1. Multiple alignment of coding DNA. (A) How alignment at the DNA level may lead to incorrectly aligned codon-codon boundaries. (B) How alignment of coding DNA at the amino acid level yields an alignment where analogous codon positions are properly lined up. The encoded amino acids are indicated at the bottom of (B).
Phylogenetic Reconstruction: Distance Matrix Methods

Anders Gorm Pedersen
Molecular Evolution Group
Center for Biological Sequence Analysis
Technical University of Denmark
gorm@cbs.dtu.dk
Trees: terminology

Terminal node (leaf)

Internal node (hypothetical ancestor)

Branch (edge)

Root
Trees: terminology

- Star tree
- Partially resolved
- Fully resolved

- Monophyletic
- Non-monophyletic

Polytomy
Trees: representations

Three different representations of the same tree
Trees: representation in computer files

Newick format:
- Leafs: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon
Newick format: named for seafood restaurant where standard was decided upon
Trees: representation in computer files

Newick format:
- Leafs: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon
Trees: representation in computer files

Newick format:

- Leaf nodes: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon

\[(A, B), (C, D)\]
Trees: representation in computer files

Newick format:

- Leafs: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon
Trees: representation in computer files

Newick format:
• Leafs: represented by taxon name
• Internal nodes: represented by pair of matching parentheses
• Descendants of internal node given as comma-delimited list.
• Tree string terminated by semicolon

( (A, B), (C, D) );
Trees: representation in computer files

Newick format:

- Leaf nodes: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon

Example:

```
( (A, B), (C, D) );
```
Trees: representation in computer files

Newick format:
• Leafs: represented by taxon name
• Internal nodes: represented by pair of matching parentheses
• Descendants of internal node given as comma-delimited list.
• Tree string terminated by semicolon

(A, B), (C, D);
Trees: representation in computer files

Newick format:

- Leafs: represented by taxon name
- Internal nodes: represented by pair of matching parentheses
- Descendants of internal node given as comma-delimited list.
- Tree string terminated by semicolon

```plaintext
( (A:1, B:0.5):2, (C:0.7, D:0.3):2.5 );
```
Trees: rooted vs. unrooted

- A rooted tree has a single node (the root) that represents a point in time that is earlier than any other node in the tree.

- A rooted tree has directionality (nodes can be ordered in terms of “earlier” or “later”).

- In the rooted tree, distance between two nodes is represented along the time-axis only (the second axis just helps spread out the leaves).
Trees: rooted vs. unrooted

• A rooted tree has a single node (the root) that represents a point in time that is earlier than any other node in the tree.

• A rooted tree has directionality (nodes can be ordered in terms of “earlier” or “later”).

• In the rooted tree, distance between two nodes is represented along the time-axis only (the second axis just helps spread out the leaves).
Trees: rooted vs. unrooted

- A rooted tree has a single node (the root) that represents a point in time that is earlier than any other node in the tree.

- A rooted tree has directionality (nodes can be ordered in terms of “earlier” or “later”).

- In the rooted tree, distance between two nodes is represented along the time-axis only (the second axis just helps spread out the leaves).
Trees: rooted vs. unrooted

- In unrooted trees there is no directionality: we do not know if a node is earlier or later than another node

- Distance along branches directly represents node distance
Trees: rooted vs. unrooted

- In unrooted trees there is no directionality: we do not know if a node is earlier or later than another node.
- Distance along branches directly represents node distance.
Reconstructing a tree using non-contemporaneous data
Reconstructing a tree using present-day data
Data: molecular phylogeny

- DNA sequences
  - genomic DNA
  - mitochondrial DNA
  - chloroplast DNA
- Protein sequences
- Restriction site polymorphisms
- DNA/DNA hybridization
- Immunological cross-reaction
Morphology vs. molecular data

New and old world vultures seem to be closely related based on morphology. Molecular data indicates that old world vultures are related to birds of prey (falcons, hawks, etc.) while new world vultures are more closely related to storks. Similar features presumably the result of convergent evolution.
Molecular data useful for analyzing single-celled organisms (which have only few prominent morphological features).
Distance Matrix Methods

1. Construct multiple alignment of sequences

Gorilla : ACGT CGTA
Human : ACGTT CCT
Chimpanzee: ACGTT TCG

2. Construct table listing all pairwise differences (distance matrix)

<table>
<thead>
<tr>
<th></th>
<th>Go</th>
<th>Hu</th>
<th>Ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>-</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Hu</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Ch</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

3. Construct tree from pairwise distances
Finding Optimal Branch Lengths

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>-</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>$S_2$</td>
<td></td>
<td>-</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>$S_3$</td>
<td></td>
<td></td>
<td>-</td>
<td>34</td>
</tr>
<tr>
<td>$S_4$</td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

Observed distance

Distance along tree

Goal:

$D_{12} \approx d_{12} = a + b + c$
$D_{13} \approx d_{13} = a + d$
$D_{14} \approx d_{14} = a + b + e$
$D_{23} \approx d_{23} = d + b + c$
$D_{24} \approx d_{24} = c + e$
$D_{34} \approx d_{34} = d + b + e$
Optimal Branch Lengths: Least Squares

- Fit between given tree and observed distances can be expressed as “sum of squared differences”:

\[ Q = \sum_{j>i} (D_{ij} - d_{ij})^2 \]

- Find branch lengths that minimize Q - this is the optimal set of branch lengths for this tree.

**Goal:**

\[
\begin{align*}
D_{12} &\approx d_{12} = a + b + c \\
D_{13} &\approx d_{13} = a + d \\
D_{14} &\approx d_{14} = a + b + e \\
D_{23} &\approx d_{23} = d + b + c \\
D_{24} &\approx d_{24} = c + e \\
D_{34} &\approx d_{34} = d + b + e
\end{align*}
\]
Least Squares Optimality Criterion

- Search through all (or many) tree topologies
- For each investigated tree, find best branch lengths using least squares criterion
- Among all investigated trees, the best tree is the one with the smallest sum of squared errors.
Exhaustive search impossible for large data sets

<table>
<thead>
<tr>
<th>No. taxa</th>
<th>No. trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>945</td>
</tr>
<tr>
<td>8</td>
<td>10,395</td>
</tr>
<tr>
<td>9</td>
<td>135,135</td>
</tr>
<tr>
<td>10</td>
<td>2,027,025</td>
</tr>
<tr>
<td>11</td>
<td>34,459,425</td>
</tr>
<tr>
<td>12</td>
<td>654,729,075</td>
</tr>
<tr>
<td>13</td>
<td>13,749,310,575</td>
</tr>
<tr>
<td>14</td>
<td>316,234,143,225</td>
</tr>
<tr>
<td>15</td>
<td>7,905,853,580,625</td>
</tr>
</tbody>
</table>
Heuristic search

1. Construct initial tree; determine sum of squares

2. Construct set of “neighboring trees” by making small rearrangements of initial tree; determine sum of squares for each neighbor

3. If any of the neighboring trees are better than the initial tree, then select it/them and use as starting point for new round of rearrangements. (Possibly several neighbors are equally good)

4. Repeat steps 2+3 until you have found a tree that is better than all of its neighbors.

5. This tree is a “local optimum” (not necessarily a global optimum!)
Clustering Algorithms

- Starting point: Distance matrix
- Cluster least different pair of sequences:
- Repeat until all nodes are linked
- Results in only one tree, there is no measure of tree-goodness.
Neighbor Joining Algorithm

- For each tip compute $u_i = \sum_j D_{ij} / (n-2)$
  (this is essentially the average distance to all other tips, except the denominator is n-2 instead of n)

- Find the pair of tips, i and j, where $D_{ij} - u_i - u_j$ is smallest

- Connect the tips i and j, forming a new ancestral node. The branch lengths from the ancestral node to i and j are:
  \[
  v_i = 0.5 D_{ij} + 0.5 (u_i - u_j) \\
  v_j = 0.5 D_{ij} + 0.5 (u_j - u_i)
  \]

- Update the distance matrix: Compute distance between new node and each remaining tip as follows:
  \[
  D_{ij,k} = (D_{ik} + D_{jk} - D_{ij}) / 2
  \]

- Replace tips i and j by the new node which is now treated as a tip

- Repeat until only two nodes remain.
Superimposed Substitutions

- Actual number of evolutionary events: 5
- Observed number of differences: 2
- Distance is (almost) always underestimated
Model-based correction for superimposed substitutions

- Goal: try to infer the real number of evolutionary events (the real distance) based on

1. Observed data (sequence alignment)

2. A model of how evolution occurs
Jukes and Cantor Model

- Four nucleotides assumed to be equally frequent (\(f=0.25\))
- All 12 substitution rates assumed to be equal
- Under this model the corrected distance is:

\[
D_{JC} = -0.75 \times \ln(1-1.33 \times D_{OBS})
\]

- For instance:

\[
D_{OBS}=0.43 \Rightarrow D_{JC}=0.64
\]
### Table 3.1 Models of nucleotide substitution

<table>
<thead>
<tr>
<th>S</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Two-parameter model (Kimura 1980)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>$1-\alpha-2\beta$</td>
<td>$\beta$</td>
<td>$\beta$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\beta$</td>
<td>$1-\alpha-2\beta$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>$\beta$</td>
<td>$\alpha$</td>
<td>$1-\alpha-2\beta$</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>$\alpha$</td>
<td>$\beta$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>b. Four-parameter model (Blaisdell 1985)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>$1-\alpha-2\gamma$</td>
<td>$\gamma$</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\delta$</td>
<td>$1-\alpha-2\delta$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>$\beta$</td>
<td>$1-\beta-2\delta$</td>
<td>$\delta$</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>$\beta$</td>
<td>$\gamma$</td>
<td>$1-\beta-2\gamma$</td>
</tr>
<tr>
<td>c. Six-parameter model (Kimura 1981a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>$1-2\alpha-\gamma$</td>
<td>$\gamma$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\delta$</td>
<td>$1-2\alpha-\delta$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>$\beta$</td>
<td>$\beta$</td>
<td>$1-2\beta-\xi$</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>$\beta$</td>
<td>$\beta$</td>
<td>$1-2\beta-\xi$</td>
</tr>
<tr>
<td>d. Nine-parameter model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>$1-\gamma_1-\gamma_2-\delta_1-\delta_2-\alpha_1$</td>
<td>$\gamma_1\beta_1$</td>
<td>$\gamma_1\alpha_1$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\delta_1$</td>
<td>$1-\gamma_2-\delta_2-\delta_2-\alpha_2$</td>
<td>$\gamma_2\alpha_2$</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>$\gamma_1\alpha_1$</td>
<td>$\gamma_2\alpha_2$</td>
<td>$1-\gamma_1-\gamma_2-\delta_1-\delta_2-\beta_2$</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>$\gamma_1\alpha_1$</td>
<td>$\gamma_2\beta_2$</td>
<td>$1-\gamma_1-\gamma_2-\delta_1-\delta_2-\beta_2$</td>
</tr>
<tr>
<td>e. General model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>$1-\alpha_1-\alpha_2-\alpha_3-\alpha_4$</td>
<td>$\alpha_{12}$</td>
<td>$\alpha_{13}$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\alpha_{21}$</td>
<td>$1-\alpha_2-\alpha_3-\alpha_4$</td>
<td>$\alpha_{23}$</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>$\alpha_{31}$</td>
<td>$\alpha_{32}$</td>
<td>$1-\alpha_1-\alpha_2-\alpha_3$</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>$\alpha_{41}$</td>
<td>$\alpha_{42}$</td>
<td>$\alpha_{43}$</td>
</tr>
</tbody>
</table>

*O, Original nucleotide; S, substitute nucleotide.