Multiple Sequence Alignment (MSA)

Yuzhen Ye (yze@indiana.edu)
School of Informatics & Computing, IUB

Outline

- Multiple sequence alignment (MSA)
- Generalize DP to 3 sequence alignment
- Heuristic approaches to MSA
  - Progressive alignment – ClustalW (using substitution matrix based scoring function)
  - Consistency-based approach – T-Coffee (consistency-based scoring function)
  - MUSCLE (MUSCLE-fast, MUSCLE-prog): reduces time and space complexity

From pairwise to multiple alignment

- Alignment of 2 sequences is represented as a 2-row matrix
- In a similar way, we represent alignment of 3 sequences as a 3-row matrix

\[
\begin{align*}
\text{A} & \quad \text{T} & \quad \text{G} & \quad \text{C} & \quad \text{G} & \quad \underline{\text{_}} \\
\text{A} & \quad \underline{\text{_}} & \quad \text{C} & \quad \text{G} & \quad \text{T} & \quad \text{_} & \quad \text{A} \\
\text{A} & \quad \text{T} & \quad \text{C} & \quad \text{A} & \quad \text{C} & \quad \underline{\text{_}} & \quad \text{A}
\end{align*}
\]

- Score: more conserved columns, better alignment

What’s multiple sequence alignment (MSA)

- A model
- Indicates relationship between residues of different sequences
- Reveals similarity/disimilarity

Why we need MSA

- MSA is central to many bioinformatics applications
- Phylogenetic tree
- Motifs
- Patterns
- Structure prediction (RNA, protein)

Aligning three sequences

- Same strategy as aligning two sequences
- Use a 3-D “Manhattan Cube”, with each axis representing a sequence to align
- For global alignments, go from source to sink
**2D vs 3D alignment grid**

2D table

3D graph

**Architecture of 3D alignment cell**

\( (i-1,j-1,k-1) \)

\( (i,j-1,k-1) \)

\( (i,j-1,k) \)

\( (i-1,j,k) \)

\( (i-1,j,k-1) \)

\( (i,j,k-1) \)

\( (i,j,k) \)

**Multiple alignment: dynamic programming**

\[ S_{i,j,k} = \max \begin{cases} 
S_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) & \text{cube diagonal: no indels} \\
S_{i-1,j,k} + \delta(v_i, w_j, -) & \text{face diagonal: one indel} \\
S_{i,j-1,k} + \delta(-, w_j, u_k) & \text{edge diagonal: two indels} 
\end{cases} \]

\( \delta(x, y, z) \) is an entry in the 3D scoring matrix

**MSA: running time**

- For 3 sequences of length \( n \), the run time is \( 7n^3; O(n^3) \)
- For \( k \) sequences, build a \( k \)-dimensional Manhattan, with run time \( (2^k-1)n^k; O(2^k n^k) \)

**Profile representation of multiple alignment**

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>A</td>
<td>G</td>
<td>-</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>G</td>
<td>-</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>G</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>G</td>
<td>-</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>.6</td>
<td>1</td>
<td>.4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>.2</td>
<td>1</td>
<td>.6</td>
<td>.4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>.2</td>
<td>.8</td>
<td></td>
<td>.4</td>
<td>4</td>
<td>.4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Aligning alignments/profiles

- Given two alignments, can we align them?

<table>
<thead>
<tr>
<th>Alignments</th>
<th>Alignment 1</th>
<th>Alignment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>GGGACTGCA</td>
<td>z</td>
</tr>
<tr>
<td>y</td>
<td>GTTACGTC--</td>
<td>GGACTGCA</td>
</tr>
<tr>
<td>w</td>
<td>GGGACTGCA</td>
<td>w</td>
</tr>
<tr>
<td>v</td>
<td>GGACGTACC--</td>
<td>GGACCT-----</td>
</tr>
</tbody>
</table>

Progressive alignment

- Progressive alignment uses guide tree
- Sequence weighting & scoring scheme and gap penalties
- Progressive alignment works well for close sequences, but deteriorates for distant sequences
  - Gaps in consensus string are permanent
  - Use profiles to compare sequences

ClustalW

- Popular multiple alignment tool today
- ‘W’ stands for ‘weighted’ (sequences are weighted differently).
- Three-step process
  1. Construct pairwise alignments
  2. Build guide tree
  3. Progressive alignment guided by the tree

ClustalW algorithm

Step 1: Pairwise alignment

- Aligns each sequence again each other giving a similarity matrix
- Similarity = exact matches / sequence length (percent identity)

<table>
<thead>
<tr>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.17</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>v2</td>
<td>.87</td>
<td>.28</td>
<td>-</td>
</tr>
<tr>
<td>v3</td>
<td>.59</td>
<td>.33</td>
<td>.62</td>
</tr>
</tbody>
</table>

(.17 means 17 % identical)
Step 2: Guide tree

Calculate:
\[ v_{1,3} = \text{alignment}(v_1, v_3) \]
\[ v_{1,3,4} = \text{alignment}(v_{1,3}, v_4) \]
\[ v_{1,2,3,4} = \text{alignment}(v_{1,3,4}, v_2) \]

ClustalW uses NJ to build guide tree; guide tree roughly reflects evolutionary relations.

Step 3: Tree based recursion

Progressive alignment: Scoring scheme
- Scoring scheme is arguably the most influential component of the progressive algorithm
- Matrix-based algorithms
  - ClustalW, MUSCLE, Kalign
  - Use a substitution matrix to assess the cost of matching two symbols or two profiled columns
  - Once a gap, always a gap
- Consistency-based schemes
  - T-Coffee, Dialign
  - Compile a collection of pairwise global and local alignments (primary library) and to use this collection as a position-specific substitution matrix

Substitution matrix based scoring
- Sum of pairs (SP score)
- Tree based scoring
- Entropy score

Sum of pairs score (SP score)

Tree-based scoring

Problem: over-estimation of the mutation costs (assuming each sequence is the ancestor of itself; requires a weighting scheme)
Entropy-based scoring

In information theory, entropy is a measure of the uncertainty associated with a random variable (it means to quantify information using some kind of currency, usually bits). The entropy $H$ of a discrete random variable $X$ with possible values $x_1, ..., x_n$ is $H(X) = -\sum p(x_i) \log p(x_i)$, where $p(x_i)$ is the information content of $x_i$.

If $p$ denotes the probability mass function of $X$ then the entropy is

$$H(X) = -\sum p(x_i) \log p(x_i).$$

Assume a genome has the following frequencies in its DNA:

- $p(A) = 0.2, p(T) = 0.2, p(C) = 0.3, p(G) = 0.3$.
- Then its entropy is $-\left(-0.0 \log_2(0.2) + 0.2 \log_2(0.2) + 0.3 \log_2(0.3) + 0.3 \log_2(0.3)\right) = 1.97$.

Entropy: Example

<table>
<thead>
<tr>
<th>$A$</th>
<th>$T$</th>
<th>$G$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$T$</td>
<td>0</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>$G$</td>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$C$</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Given a DNA sequence, what is its maximum entropy?

Alignment entropy

- Define frequencies for the occurrence of each letter in each column of multiple alignment
  - $p_A = 1, p_T = p_G = p_C = 0$ (1st column)
  - $p_A = 0.75, p_T = 0.25, p_G = p_C = 0$ (2nd column)
  - $p_A = 0.50, p_T = 0.25, p_C = 0.25, p_G = 0$ (3rd column)

- Compute entropy of each column

```
  A
  A
  A
  C
  C
  A
  C
  G
  A
  C
  T
```

An alignment with 3 columns

Alignment entropy: 2.811

Consistency-based approaches

- T-Coffee
  - M-Coffee & 3D-Coffee (Expresso)
- Principle
  - Primary library
  - Library extension

T-Coffee: Primary library

Input sequences:

- Sequences: GARFIELD THE LAST FAT CAT
- Sequences: GARFIELD THE FAST CAT
- Sequences: GARFIELD THE VERY FAST CAT
- Sequences: THE FAT CAT

Primary library: collection of global/local pairwise alignments

- Sequences: GARFIELD THE LAST FAT CAT
- Sequences: GARFIELD THE FAST CAT
- Sequences: GARFIELD THE VERY FAST CAT
- Sequences: THE FAT CAT

T-Coffee: Library extension

Extended library: new pairwise alignment (AAH, HCA, ADC, DCA, BFD, and SCD)
T-Coffee uses progressive strategy to derive multiple alignment
- Guide tree
- First align the closest two sequences (DP using the weights derived from the extended library)
- Align two “alignments” (using the weights from the extended library – average over each column)
- No additional parameters (gaps etc)
  - The substitution values (weights) are derived from extended library which already considered gaps
  - High-scoring segments (consistent segments) enhanced by the data set to the point that they are insensitive to the gap penalties

MUSCLE: a tool for fast MSA
- Initial progressive alignment followed by horizontal refinement (stochastic search for a maximum objective score)
  - Step 1: draft progressive (using k-mer counting for fast computation of pairwise distance; tree building using UPGMA or NJ)
  - Step 2: Improved progressive to improve the tree and builds a new progressive alignment according to this tree (can be iterated).
  - Step 3: Refinement using tree-dependent restricted partitioning (each edge is deleted from the tree to divide the sequences into two disjoint subsets, from each a profile is built; the profile-profile alignment is computed, and if the score improves, retain the new alignment).
- Ref: MUSCLE: a multiple sequence alignment method with reduced time and space complexity; BMC Bioinformatics 2004, 5:113

Multiple alignment: History
1975 Sankoff
Formulated multiple alignment problem and gave DP solution
1988 Carrillo-Lipman
Branch and Bound approach for MSA
1990 Feng-Doolittle
Progressive alignment
1994 Thompson-Higgins-Gibson-ClustalW
Most popular multiple alignment program
1998 DIALIGN (Segment-based multiple alignment)
2000 T-coffee (consensus-based)
2004 MUSCLE
2005 ProbCons (uses Bayesian consistency)
2006 M-Coffee (consensus meta-approach)
2006 Expresso (3D-Coffee; use structural template)
2007 PROMALS (profile-profile alignment)

Summary & references
- “A majority of studies indicate that consistency-based methods are more accurate than their matrix-based counterparts, although they typically require an amount of CPU time N times higher than simpler methods (N being the number of sequences)”
- http://tcoffee.vital-it.ch/cgi-bin/Tcoffee/tcoffee_cgi/index.cgi
- Cedric Notredame. Recent evolutions of multiple sequence alignment algorithms. 2007, 3(8):e123