CS342: Bioinformatics
Multiple Alignments

Section 6.10
Multiple Alignment versus Pairwise Alignment

- Up until now we have only tried to align two sequences.
- What about more than two? And what for?
- A faint similarity between two sequences becomes significant if present in many.
- Multiple alignments can reveal subtle similarities that pairwise alignments do not reveal.
Generalizing Pairwise 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

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

- Score: more conserved columns, better alignment
Alignment Paths

• Align 3 sequences: ATGC, AATC, ATGC

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>--</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>T</td>
<td>--</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>--</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- x coordinate
- y coordinate
- z coordinate

• Resulting path in \((x,y,z)\) space:

\[(0,0,0) \rightarrow (1,1,0) \rightarrow (1,2,1) \rightarrow (2,3,2) \rightarrow (3,3,3) \rightarrow (4,4,4)\]
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
2-D vs 3-D Alignment Grid

2-D edit graph

3-D edit graph
2-D cell versus 3-D Alignment Cell

In 2-D, 3 edges lead to each interior vertex

In 3-D, 7 edges lead to each interior vertex

- 2-D \([(i-1,j-1), (i-1,j), (i,j-1)] \rightarrow (i,j)]
- 3-D \([(i-1,j-1,k-1), (i-1,j,k), (i,j,k-1), (i,j,k-1), (i-1,j,k-1), (i-1,j,k-1), (i-1,k,k-1)] \rightarrow (i,j,k)\)
Architecture of 3-D Alignment Cell

1 - Match/Mismatch Path

6 – indels Paths
3 in one seq
3 in two seqs
Multiple Alignment: Dynamic Programming

- \( s_{i,j,k} = \max \) \[
\begin{align*}
  s_{i-1,j-1,k-1} &+ \delta(v_i, w_j, u_k) \\
  s_{i-1,j,k} &+ \delta(v_i, w_j, _) \\
  s_{i-1,j,k-1} &+ \delta(v_i, _, u_k) \\
  s_{i,j-1,k} &+ \delta(_, w_j, u_k) \\
  s_{i-1,j,k} &+ \delta(v_i, _, _) \\
  s_{i,j,k} &+ \delta(_, w_j, u_k) \\
  s_{i,j,k-1} &+ \delta(_, _, u_k)
\end{align*}
\]

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

- cube diagonal: no indels
- face diagonal: one indel
- Lattice edge: two indels
Multiple Alignment: Running Time

• For 3 sequences of length \( n \), the run time is \( 7n^3; \) \( O(n^3) \)

• For \( k \) sequences, build a \( k \)-dimensional table, with run time \( (2^k-1)(n^k); \) \( O(2^k n^k) \)

• Conclusion: dynamic programming approach for alignment between two sequences is easily extended to \( k \) sequences but it is impractical due to exponential running time
Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments

\[ x: \text{AC-GCGG-C} \]
\[ y: \text{AC-GC-GAG} \]
\[ z: \text{GCCGC-GAG} \]

Induces:

\[ x: \text{ACGCGG-C}; \quad x: \text{AC-GCGG-C}; \quad y: \text{AC-GCGAG} \]
\[ y: \text{ACGC-GAC}; \quad z: \text{GCCGC-GAG}; \quad z: \text{GCCGCGAG} \]
Inverse Problem: Do Pairwise Alignments imply a Multiple Alignment?

Given 3 arbitrary pairwise alignments:

\[
\begin{align*}
x & : \text{ACGCTGG--C;} & x & : \text{AC-GCTGG--C;} & y & : \text{AC-GC-GAG} \\
y & : \text{ACGC--GAC;} & z & : \text{GCCGCA-GAG;} & z & : \text{GCCGCAGAG}
\end{align*}
\]

Can we construct a multiple alignment that induces them?

**NOT ALWAYS**

Why? Because pairwise alignments may be arbitrarily inconsistent
Combining Optimal Pairwise Alignments into Multiple Alignment

Can combine pairwise alignments into multiple alignment

Can *not* combine pairwise alignments into multiple alignment
Inferring Multiple Alignment from Pairwise Alignments

- From an optimal multiple alignment, we can infer pairwise alignments between all pairs of sequences, but they are not necessarily optimal.
- It is difficult to infer a “good” multiple alignment from optimal pairwise alignments between all sequences.
- Are we stuck, or is there some other trick?
Multiple Alignment using Profile Scores

- A   G   G   C   T   A   T   C   A   C   C   T   G
T   A   G   -   C   T   A   C   C   A   -   -   -   G
C   A   G   -   C   T   A   C   C   A   -   -   -   G
C   A   G   -   C   T   A   T   C   A   C   -   G   G
C   A   G   -   C   T   A   T   C   G   C   -   G   G

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>0</th>
<th>5</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>5</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- Thus far we have aligned a sequence against other sequences
- Can we align a sequence against a profile?
- Can we align a profile against a profile?
Aligning alignments

• Given two alignments, can we align them?

x GGGCACTGCAT
y GGTTACGTC-- Alignment 1
z GGGAAACTGCAG

w GGACGTACC-- Alignment 2
v GGACCT------
Aligning alignments

• Given two alignments, can we align them?
• Hint: don’t use the sequences…
  align their profiles

x GGGCAC=TGCAT
y GGTTAC=GTC--
z GGGAAC=TGCAG
   ||   ||   |   |   | Combined Alignment
w GG==ACGTACC--
v GG==ACCT------
Multiple Alignment: Greedy Approach

- Choose most similar pair of strings and combine into a profile, thereby reducing alignment of $k$ sequences to an alignment of of $k-1$ sequences/profiles. **Repeat**
- This is a heuristic *greedy* method

\[ \begin{align*}
    u_1 &= \text{ACGTACGTACGT} \ldots \\
    u_2 &= \text{TTAATTAATTTAA} \ldots \\
    u_3 &= \text{ACTACTACTACT} \ldots \\
    \vdots \\
    u_k &= \text{CCGGCCCGGCCCGG} \\
    u_1 &= \text{ACg/tTACg/tTACg/cT} \ldots \\
    u_2 &= \text{TTAATTAATTTAA} \ldots \\
    \vdots \\
    u_k &= \text{CCGGCCCGGCCCGG} \\
    k-1
\end{align*} \]
Greedy Approach: Example

• Consider these 4 sequences

S1: GATTCA
S2: GTCTGA
S3: GATATT
S4: GTCAGC

Scoring Matrix:
Match = 1
Mismatch = -1
Indel = -1
Greedy Approach: Example

• There are $\binom{4}{2} = 6$ possible alignments

\begin{align*}
s2 & \quad \text{GTCTGA} & s1 & \quad \text{GATTCA--} \\
s4 & \quad \text{GTCAGC (score = 2)} & s4 & \quad \text{G-T-CAGC (score = 0)} \\
\text{s1 GAT-TCA} & & & \\
\text{s2 G-TCTGA (score = 1)} & & &\
\text{s1 GAT-TCA} & & & \\
\text{s2 G-TCTGA} & & &\
\text{s3 GATAT-T (score = -1)} & & & \\
\text{s3 GAT-ATT} & & & \\
\text{s4 G-TCAGC (score = -1)} & & & \\
\end{align*}
Greedy Approach: Example

$s_2$ and $s_4$ are closest; combine:

$s_2$ \hspace{0.5cm} GTCTGA
$s_4$ \hspace{0.5cm} GTCAGC \hspace{0.5cm} s_{2,4} \hspace{0.5cm} GTCT/aGa/c

(profile)

new set of 3 sequences:

$S_1$ \hspace{0.5cm} GATTCA
$S_3$ \hspace{0.5cm} GATATT
$S_{2,4}$ \hspace{0.5cm} GTCT/aGa/c

Repeat
Greedy Approach: Example

Repeat for $\binom{3}{2} = 3$ possible alignments

$s_1 : \text{GAT-TCA}$
$s_3 : \text{GATAT-T}$
$(\text{score} = 1 + 1 + 1 - 1 + 1 - 1 - 1 = 1)$

$s_1 : \text{GAT-TCA}$
$s_{2,4} : \text{G-TCtGa}$
$(\text{score} = 2 - 2 + 2 - 2 + 1 - 1 + 1 = 1)$

$s_3 : \text{GATAT-T}$
$s_{2,4} : \text{G-TCtGa}$
$(\text{score} = 2 - 2 + 2 - 2 + 1 - 1 - 1 = -1)$
Progressive Alignment

- *Progressive alignment* is a variation of greedy algorithm with a somewhat more intelligent strategy for choosing the order of alignments.
- Progressive alignment works well for close sequences, but deteriorates for distant sequences
  - Gaps in consensus string are permanent
  - Use profiles to compare sequences

- CLUSTAL
ClustalW (Clustal Omega)

- Popular multiple alignment tool commonly used today
- ‘W’ stands for ‘weighted’ (different parts of alignment are weighted differently).
- Three-step process
  1.) Construct pairwise alignments
  2.) Build Guide Tree
  3.) Progressive Alignment guided by the tree
Step 1: Pairwise Alignment

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

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

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

- Create Guide Tree using the similarity matrix
  
  - ClustalW uses the neighbor-joining method (we will discuss this later in the course, in the section on clustering)
  
- Guide tree roughly reflects evolutionary relations
## Step 2: Guide Tree (cont’d)

### Calculate:

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.17</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.87</td>
<td>.28</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>.59</td>
<td>.33</td>
<td>.62</td>
<td>-</td>
</tr>
</tbody>
</table>

Calculate:

- $v_{1,3} = \text{alignment} (v_1, v_3)$
- $v_{1,3,4} = \text{alignment}((v_{1,3}), v_4)$
- $v_{1,2,3,4} = \text{alignment}((v_{1,3,4}), v_2)$
Step 3: Progressive Alignment

- Start by aligning the two most similar sequences
- Following the guide tree, add in the next sequences, aligning to the existing alignment
- Insert gaps as necessary

Dots and stars show how well-conserved a column is.