CS 466
Introduction to Bioinformatics
Lecture 7

Mohammed El-Kebir
September 16, 2020
Course Announcements

Instructor:
• Mohammed El-Kebir (melkebir)
• Office hours: Wednesdays, 3:15-4:15pm

TA:
• Sarah Christensen (sac2) – Mondays, 3-4pm
• WesleyWei Qian (weiqian3) – Fridays, 9-10am

Homework 1 due 9/17 by 11:59pm
Outline

• Multiple sequence alignment
• Carrillo-Lipman algorithm
• Progressive alignment

Reading:
• Jones and Pevzner. Chapter 6.10
• Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
• Lecture notes
A **multiple sequence alignment** $\mathcal{M}$ between $k$ strings $v_1, ..., v_k$ is a $k \times q$ matrix, where $q = \max\{|v_i| : i \in [k]\}, ..., \sum_{i=1}^{k} |v_i|$ such that the $i$-th row contains the characters of $v_i$ in order with spaces ‘-’ interspersed and no column contains $k$ spaces.

**MSA-SP problem:** Given strings $v_1, ..., v_k$ and scoring function $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \rightarrow \mathbb{R}$ find multiple sequence alignment $\mathcal{M}^*$ with **maximum** value of $\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(v_i, v_j)$ where $S(v_i, v_j)$ is the score of the induced pairwise alignment of $(v_i, v_j)$ in $\mathcal{M}^*$ using $\delta$.
Sum-of-Pairs (SP) Score

\[ S(\mathbf{v}_i, \mathbf{v}_j) \] is score of induced pairwise alignment of sequences \((\mathbf{v}_i, \mathbf{v}_j)\)

Multiple sequence alignment \(\mathcal{M}\)

\[
\text{SP-score}(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)
\]
Question: Can we construct a multiple alignment that induces the above three pairwise alignments?
Question: Can we construct a multiple alignment that induces the above three pairwise alignments?
Compatibility

**Compatible**: Pairwise alignments can be combined into multiple alignment

**Incompatible**: Pairwise alignments *cannot* be combined into multiple alignment
**Compatibility**

**Compatible**: Pairwise alignments can be combined into multiple alignment

**Incompatible**: Pairwise alignments *cannot* be combined into multiple alignment
From Compatible Pairwise to Multiple Alignment

Optimal multiple alignment

(Sub)optimal multiple alignment

Pairwise alignments between all pairs of sequences, but they are not necessarily optimal

Good (or optimal) compatible pairwise alignments between all sequences
Outline

• Multiple sequence alignment
• Carrillo-Lipman algorithm
• Progressive alignment

Reading:
• Jones and Pevzner. Chapter 6.10
• Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
• Lecture notes
Multiple Sequence Alignment Problem w/ SP-Score

**MSA-SP problem:** Given strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and scoring function
\[ \delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \to \mathbb{R}, \]
find multiple sequence alignment $\mathcal{M}^*$ with **maximum** value of $\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$
where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$ using $\delta$

**Weighted SP-Edit Distance problem:** Given strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and scoring function
\[ \delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \to \mathbb{R}, \]
find multiple sequence alignment $\mathcal{M}^*$ with **minimum** value of
$\text{SP-score}(\mathcal{M}^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$ using $\delta$
Recall: Banded Alignment

Alignment is a path from source (0, 0) to target (m, n) in edit graph

Constraint path to band of width $k$ around diagonal

Running time: $O(nk)$

Question: Alternative ways of constraining search space?

Constrain traceback to band of DP matrix (penalize big gaps)

Figure source: [http://jinome.stanford.edu/stat366/pdfs/stat366_win0607_lecture04.pdf](http://jinome.stanford.edu/stat366/pdfs/stat366_win0607_lecture04.pdf)
Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

Alternatively: Stop computing when remaining alignment will be suboptimal
Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

Alternatively: Stop computing when remaining alignment will be suboptimal

Forward dynamic programming – think of Dijkstra’s algorithm:
• Queue of unvisited vertices
• Maintain $p[i, j, k]$ shortest distance yet found from (0,0,0) to (i,j,k).
• For each directed edge (i, j, k) to (i', j', k') with cost w, set $p[i', j', k'] = \min\{p[i', j', k'], p[i, j, k] + w\}$
Forward Dynamic Programming

Banded alignment: constraint path to polyhedron around diagonal

Alternatively: Stop computing when remaining alignment will be suboptimal

Forward dynamic programming – think of Dijkstra’s algorithm:
- Queue of unvisited vertices
- Maintain \( p[i, j, k] \) shortest distance yet found from \((0,0,0)\) to \((i,j,k)\).
- For each directed edge \((i, j, k)\) to \((i', j', k')\) with cost \(w\), set \( p[i', j', k'] = \min\{p[i', j', k'], p[i, j, k] + w\} \)

**Question:** Can we remove vertices from consideration based on alignment score of prefix?
Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length $n$

- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length $n$

- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$

$$d_{p,q}(i, j) \geq D_{p,q}(i, j)$$
Alignment Projection and SP-score

Sequences $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ each of length $n$

- $D(i, j, k)$ is min SP-cost of aligning $\mathbf{v}_1[1..i], \mathbf{v}_2[1..j], \mathbf{v}_3[1..k]$
- $d_{p,q}(i, j)$ is cost of induced alignment of $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$
- $D_{p,q}(i, j)$ is min cost of aligning $\mathbf{v}_p[1..i], \mathbf{v}_q[1..j]$

\[
d_{p,q}(i, j) \geq D_{p,q}(i, j)
\]

\[
D(i, j, k) = d_{1,2}(i, j) + d_{1,3}(i, k) + d_{2,3}(j, k)
\geq D_{1,2}(i, j) + D_{1,3}(i, k) + D_{2,3}(j, k)
\]
Carrillo-Lipman Method

- $D^+(i, j, k)$ is min SP-cost of alignment of suffix $v_1[i..n], v_2[j..n], v_3[k..n]$
- $d^+_{p,q}(i, j)$ is cost of induced alignment of suffix $v_p[i..n], v_q[j..n]$
- $D^+_{p,q}(i, j)$ is min cost of alignment of suffix $v_p[i..n], v_q[j..n]$

\[
d^+_{p,q}(i, j) \geq D^+_{p,q}(i, j)
\]

\[
D^+(i, j, k) = d^+_{1,2}(i, j) + d^+_{1,3}(i, k) + d^+_{2,3}(j, k) \\
\geq D^+_{1,2}(i, j) + D^+_{1,3}(i, k) + D^+_{2,3}(j, k)
\]
Carrillo-Lipman Method

\[ D^+(i, j, k) = d^+_{1,2}(i, j) + d^+_{1,3}(i, k) + d^+_{2,3}(j, k) \geq D^+_{1,2}(i, j) + D^+_{1,3}(i, k) + D^+_{2,3}(j, k) \]

\[ D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D^+_{1,2}(i, j) + D^+_{1,3}(i, k) + D^+_{2,3}(j, k) \]
Carrillo-Lipman Method

\[ D^+(i, j, k) = d^+_{1, 2}(i, j) + d^+_{1, 3}(i, k) + d^+_{2, 3}(j, k) \geq D^+_{1, 2}(i, j) + D^+_{1, 3}(i, k) + D^+_{2, 3}(j, k) \]

\[ D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D^+_{1, 2}(i, j) + D^+_{1, 3}(i, k) + D^+_{2, 3}(j, k) \]

**Question:** What if we have an alignment with cost \( z \)?
Carrillo-Lipman Method

\[ D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \]

\[ D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \]

**Question:** What if we have an alignment with cost \( z \)?

If \( z < D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \) then \((i, j, k)\) not on optimal path => **Prune!**
Carrillo-Lipman Method

\[ D^+(i, j, k) = d_{1,2}^+(i, j) + d_{1,3}^+(i, k) + d_{2,3}^+(j, k) \geq D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \]

\[ D(i, j, k) + D^+(i, j, k) \geq D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \]

**Question:** What if we have an alignment with cost \( z \)?

**Question:** How to find this alignment?

If \( z < D(i, j, k) + D_{1,2}^+(i, j) + D_{1,3}^+(i, k) + D_{2,3}^+(j, k) \) then \( (i, j, k) \) not on optimal path => **Prune**!
Outline

• Multiple sequence alignment
• Carrillo-Lipman algorithm
• Progressive alignment

Reading:
• Jones and Pevzner. Chapter 6.10
• Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
• Lecture notes
Heuristic: Iterative/Progressive Alignment

Iteratively add strings (or alignments) to existing alignment(s).

Issues:
1. How to merge alignments?
2. What order to use in merging strings/alignments?
Heuristic Approach: Merge Pairwise Alignments

Question: Can we align two alignments?

Need a way to summarize an alignment and score merged alignments
A profile $P = [p_{i,j}]$ is a $(|\Sigma| + 1) \times l$ matrix, where $p_{i,j}$ is the frequency of $i$-th letter in $j$-th position of alignment.
We know how to align sequence against sequence.

Question: Can we align sequence against profile?

Question: Can we align profile against profile?
Aligning String to Profile

A profile \( P = [p_{i,j}] \) is a \((|\Sigma| + 1) \times n\) matrix, where \( p_{i,j} \) is the frequency of \( i \)-th letter in \( j \)-th position of alignment.

**Given:** Sequences \( v = v_1, \ldots, v_m \) and profile \( P \) with \( n \) columns

- \( s[i,j] \) is optimal alignment of \( v_1, \ldots, v_i \) and first \( j \) columns of \( P \)
- \( \delta(x, y) \) is score for aligning characters \( x \) and \( y \)
- \( \tau(x, j) \) is score for aligning character \( x \) and column \( j \) of \( P \)
Aligning String to Profile

\[ \tau(x, j) = \sum_{y \in \Sigma \cup \{-\}} p_{y,j} \cdot \delta(x, y) \]

\[ s[i, j] = \max \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0, \\ s[i - 1, j] + \delta(v_i, -), & \text{if } i > 0, \text{ Insert space in profile} \\ s[i, j - 1] + \tau(-, j), & \text{if } j > 0, \text{ Insert space in string} \\ s[i - 1, j - 1] + \tau(v_i, j), & \text{if } i > 0 \text{ and } j > 0. \end{cases} \]

- \( s[i, j] \) is optimal alignment of \( v_1, \ldots, v_i \) and first \( j \) columns of \( P \)
- \( \delta(x, y) \) is score for aligning characters \( x \) and \( y \)
- \( \tau(x, j) \) is score for aligning character \( x \) and column \( j \) of \( P \)
Progressive Multiple Alignment: Greedy Algorithm

Choose most similar pair among $k$ input strings, combine into a profile. This reduces the original problem to alignment of $k-1$ sequences to a profile. Repeat.
Example

Score of +1 for matches, -1 otherwise.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_2$</td>
<td>GTCTGA</td>
<td>$s_1$</td>
<td>GATTCA--</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_4$</td>
<td>GTCAGC (score = 2)</td>
<td>$s_4$</td>
<td>G–T–CAGC(score = 0)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_1$</td>
<td>GAT–TCA</td>
<td>$s_2$</td>
<td>G–TCTGA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td>G–TCTGA (score = 1)</td>
<td>$s_3$</td>
<td>GATAT–T (score = -1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_1$</td>
<td>GAT–TCA</td>
<td>$s_3$</td>
<td>GAT–ATT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td>GATAT–T (score = 1)</td>
<td>$s_4$</td>
<td>G–TCAGC (score = -1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Score of +1 for matches, -1 otherwise.

| s2    | GTCTGA          |
| s4    | GTCAGC (score = 2) |
| s1    | GAT-TCA         |
| s2    | G-TCTGA (score = 1) |
| s1    | GAT-TCA         |
| s3    | GATAT-T (score = 1) |
| s4    | G-T-CAGC (score = 0) |

Question: Any theoretical guarantees on optimality?

No guarantees!
Outline

• Multiple sequence alignment
• Carrillo-Lipman algorithm
• Progressive alignment

Reading:

• Jones and Pevzner. Chapter 6.10
• Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
• Lecture notes