Bioinformatics

Richard Bonneau
Lecture 2: sequence alignment I.
Associated reading.

- Durbin, Eddy, et al. Ch. 2

- Papers for discussion section:
Pairwise Alignments I

- **What do you need:** i. definition of alignment ii. score to rank alignments iii. algorithm to search for alignments iv. score significance of match.

- This lecture
  1. additive scoring scheme, substitution matrices (ii above)
  2. alignment algorithms
  3. judging match, calculating significance
  4. heuristic tools, BLAST, gapped-BLAST, FASTA, PSI-BLAST
  5. Useful links
• Bayes Rule:

\[ P(y | x) = \frac{P(x | y)P(y)}{P(x)} \]

• Small amount of probability:

\[ P(x | \tilde{Y}) = P(x | y_1)P(y_1) + ... + P(x | y_n)P(y_n) \]

• Estimating P given 1 sample and a distribution, example with normal distribution, t-test. We’ll use different distribution today.
(i.) alignment def

- sequence i: ABAACABA
- sequence j: ABAAAABA
- alignment: abaa-caba
  \[
  \begin{array}{ccccccc}
  & & & a & a & a & a \\
  & & & & & & b \\
  & & & & & & a \\
  & & & & & & b \\
  \end{array}
  \]
  \[
  \begin{array}{ccccccc}
  & & & c & a & b & a \\
  & & & & & & a \\
  & & & & & & b \\
  & & & & & & a \\
  \end{array}
  \]
  abaaa-aba
(ii.) ranking alignments

alignment : abaa-caba

|||...|||
abaaca-aba

One simple score:
alignment to gap = mismatch

\[ S(x_i, y_j) = \begin{cases} 
2 & \text{if match} \\
-1 & \text{mismatch} 
\end{cases} \]

Sum over S
Better score for matches approximates \( p(x,y|\text{Match}) \):

\[
P(x,y|\text{Random}) = \prod_i q_{x_i} \prod_j q_{y_j}
\]

\[
P(x,y|\text{Match}) = \prod_i p_{x_i,y_i}
\]

\[
\frac{P(x,y|M)}{P(x,y|R)} = \prod_i p_{x_i,y_i} \quad \frac{\prod_i q_{x_i} \prod_j q_{y_j}}
\]

\[
\log \left( \frac{P(x,y|M)}{P(x,y|R)} \right) = S(x,y)
\]
Better score for matches approximates $p(x,y|\text{Match})$:

Where do we get the "$|M|$" in $P(x,y|M)$?!!

**BLOSUM 50,62:**
From high quality alignments with no redundancy
> 50, 62% identity

**PAM, others**
Better score for matches approximates $p(x,y|\text{Match})$:

Scoring gaps, gap penalties:

- $g$ - length of gap; $d$ - gap open
- $e$ - gap extend

$$\gamma(g) = -eg \quad \text{linear}$$
$$\gamma(g) = -e(g - 1) - d \quad \text{affine}$$
(iii.) Algorithm for finding optimal matches

Cost of trying every possible alignment exhaustively

\[ \text{if : } n_i = n_j = n \]

\[
\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2^n}}{\sqrt{\pi n}}
\]

Brute force wont cut it…
Dynamic Programming:

- global = Needleman-Wunsch 1970
  Gotoh 1982
- local = Smith Waterman
(iii.) Algorithm for finding optimal matches (global)

We base larger global alignments on smaller preceding alignments.

Best alignment for x - up to $x_i$ => y - up to $y_j$
based on alignment up to $x_{i-1}, y_{j-1}$
(iii.) Algorithm for finding optimal matches (global)

1. Set $F(0,0) = 0$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>c</th>
<th>a</th>
<th>d</th>
<th>b</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(iii.) Algorithm for finding optimal matches
(global)

\[ F(i,j) : \]
3 possible outcomes:
\[ x_i \rightarrow y_i \quad F(i,j) = F(i-1, j-1) + s(x_i, y_j) \]
\[ x_i \rightarrow \text{gap} \quad F(i,j) = F(i-1, j) - d \]
\[ \text{gap} \rightarrow y_i \quad F(i,j) = F(i, j-1) - d \]

\[ F(i,j) = \text{max of three above} \]

Simple score for this example:
Mismatch = -1
Match = 2
(iii.) Algorithm for finding optimal matches (global)

1. Set $F(0,0) = 0$
2. $F(i,0) = -id$
3. $F(0,j) = -jd$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>c</th>
<th>a</th>
<th>d</th>
<th>b</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(iii.) Algorithm for finding optimal matches
(global)

1. Set $F(0,0) = 0$
2. $F(i,0) = -id$
3. $F(0,j) = -jd$
4. Fill in $F(i,j)$ from top-left to bottom-right (remember pointers)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>c</th>
<th>a</th>
<th>d</th>
<th>b</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>-2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(iii.) Algorithm for finding optimal matches (global)

1. Set \( F(0,0) = 0 \)
2. \( F(i,0) = -id \)
3. \( F(0,j) = -jd \)
4. Fill in \( F(i,j) \) from top-left to bottom-right (remember pointers)
5. Traverse to read alignment

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>c</th>
<th>a</th>
<th>d</th>
<th>b</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>a</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>c</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>b</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>-4</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>-5</td>
<td>-2</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Score = 2

**cadb-d-**

**.|.|.|.|.**

**-acbcdb**
(iii.) Algorithm for finding optimal matches (global)

\[ F(i,j) : \]
3 possible outcomes:
\[ x_i \rightarrow y_i , \quad F(i,j) = F(i-1, j-1) + s(x_i, y_j) \]
\[ x_i \rightarrow \text{gap}, \quad F(i,j) = F(i-1, j) - d \]
\[ \text{gap} \rightarrow y_i , \quad F(i,j) = F(i, j-1) - d \]
No outcome > 0, \( F(i,j) = 0 \)
\[ F(i,j) = \max \text{ of three above OR } 0 \]
0 effectively starts new alignment

Both these algorithms work in \( O(nm) \) time and \( O(nm) \) space

Should I explain big O?
Algorithm for finding optimal matches (local - smith-waterman)

\[ F(i,j) : \]
3 possible outcomes:
\[ x_i \rightarrow y_i \quad F(i,j) = F(i-1, j-1) + s(x_i, y_j) \]
\[ x_i \rightarrow \text{gap} \quad F(i,j) = F(i-1, j) - d \]
\[ \text{gap} \rightarrow y_i \quad F(i,j) = F(i, j-1) - d \]
**No outcome > 0, \( F(i,j) = 0 \)**

\[ F(i,j) = \text{max of three above OR } 0 \]
**0 effectively starts new alignment**
(iv.) calculating/estimating alignment significance (bayes)

We have: \( P(x, y \mid M) \)

We want: \( P(M \mid x, y) \)

Bayes rule: \[ P(M \mid x, y) = \frac{P(x, y \mid M)P(M)}{P(x, y)} \]

\[ P(x, y) = P(x, y \mid R)P(R) + P(x, y \mid M)P(M) \]

\( P(M) = \text{prior} \)

\( P(R) = 1 - P(M) \)

\[ S' = \log \left( \frac{P(M)}{P(R)} \right) + \log \left( \frac{P(x, y \mid M)}{P(x, y \mid R)} \right) \]

\[ P(M \mid x, y) = \sigma(S') = \frac{e^{S'}}{1 + e^{S'}} \]

I take it back, i'm skipping some steps here … we don’t want to be here all day
(iv.) calculating/estimating alignment significance 
(extreme value distribution)

\[ P(M_N \leq x) \approx \exp\left(-KNe^{\lambda(x-\mu)}\right) \]

Extreme value distribution
Free parameters:
Ungapped: S Karlin + Altschul analytical
Gapped : - Believed to follow analytical
         solution from Karlin (Mott 1992)
         - Altschul + Gish 1996 - fit from
           large generated dataset.
Next week’s reading

- Durbin, Eddy, et al. Ch. 3
- Papers for discussion section: