Molecular evolution
Global, semi-global and local
Affine gap penalty
How sequences evolve

• point mutations (single base changes)
• deletion (loss of residues within the sequence)
• insertion (gain of residue within the sequence)
• truncation (loss of either end)
• extension (gain of residues at either end)

Mechanisms of insertion or extension:
  • duplication or whole gene or domain
  • polymerase "stutter"
  • transposable element
  • more??
How evolution is scored

• point mutations ................................ substitution matrix
• deletion ...................................... gap penalty
• insertion ..................................... gap penalty
• truncation ................................... end gap penalty
• extension ...................................... end gap penalty
Yes, an **alignment algorithm** is really
A Model for Sequence Evolution!

- The way we do alignment should model how sequences evolve.

How much evolutionary time is represented by....

- point mutations .......... relatively frequent, usually bad
- deletion ..................... infrequent, always bad, location dependent
- insertion ..................... infrequent, always bad, location dependent
- truncation ..................... frequent, not so bad
- extension ..................... frequent, not so bad
Example: here is an alignment of mouse nitric oxide synthase (thick black line). It has multiple domains which are homologous to several shorter proteins. If we penalize end gaps, what happens to the score of the true alignment? Did "end gaps" evolve the same way as internal gaps? (no!)

Unless the two proteins are known to be single domains, it makes more sense NOT to penalize end gaps.
A structure-based multiple sequence alignment: fluorescent proteins.

Note the...
- ragged N-terminal edge,
- small number of indels,
- large number of point substitutions distributed unevenly over the sequence.

Note conserved regions, vs. hot spots.
To make a ragged edge: don't penalize end gaps

First: To ignore *starting* gap penalties, set gap rows to zero (keep the traceback arrows).
How to NOT penalize end gaps

Second: To ignore *ending* gap penalty, start the traceback with the MAX score at the end of either sequence.

(𝑖.𝑒. use last row or column)

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>T</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If we penalize end gaps in neither sequence, we are asking for the best alignment that contains at least two of the 4 termini. Good for identifying *terminal domains* in two multi-domain proteins.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>T</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Semi-global: no end gaps**
If we penalize end gaps in sequence 2 but not in sequence 1, we are asking for an alignment that contains all of sequence 2 within sequence 1.

<table>
<thead>
<tr>
<th>G</th>
<th>T</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Semi-global: Penalize end gaps on one side
If we penalize end gaps in sequence 1 but not in sequence 2, we are asking for an alignment that *contains all of sequence 1 within sequence 2.*
Local Alignment

A local alignment can start anywhere and end anywhere in the alignment matrix.

\[
A(i,j) = \max \begin{cases} 
A(i-1,j-1) + S(i,j) \\
A(i,j-1) + \text{gap} \\
A(i-1,j) + \text{gap} \\
0 + S(i,j)
\end{cases}
\]

\textbf{end} is the maximum score anywhere in the matrix.
Local Alignment

- Asks for largest domain, sub-domain, or set of contiguous domains that are in common between two sequences.
- Worst score is always zero (0) for “no alignment”
- It's possible to have multiple local alignments between two proteins
- More appropriate than global or semi-global when there are no assumptions about the sequence relationship.
- Used for database searches.
- Required to obtain e-values (stay tuned)
Global, semi-global, and local alignment

The choice of alignment method makes a statement about how the sequences are related. Was one sequence inserted into the other?

• **Global alignment** (end gaps) requires that all 4 termini are counted. In general, the two sequences are about the same length.

• **Semi-global** (no end gaps in 1 or both seqs) requires that one of the two sequences be completely contained in the other or that 2 or the 4 the termini be included.

• **Local alignment** finds subsequences in both. Does not require that the termini be included in the alignment.
The **optimal** alignment *may be no alignment*

If the maximum score in the alignment matrix is < 0., then the optimal local alignment has score = 0 and looks like this:

```
ATSFM~~~~~~~
~~~~~PGTSFEP
```
A structure-based multiple sequence alignment: fluorescent proteins.

Note the ...
- lengths of gaps

Longer gaps are not so much less frequent than shorter ones. But very long gaps are rare.
Affine gap penalty-- theory

• Each gap represents an evolutionary event (duplication, polymerase stutter, deletion/ligation, etc.)

• If the alignment has "evolutionary distance" meaning, then the gap penalty score should be proportional to the number of gaps.

Q: Are long gaps proportionally less likely?
A: Less likely, yes, but proportionally, no.
Which alignment is intuitively better?

AGGCTACT\textasciitilde T\textasciitilde TCA
GGCTACTATATCA

AGGCTACTTTT\textasciitilde\textasciitilde CA
GGCTACTATATCA
Linear versus Affine gap penalty. Making alignments agree with biological intuition.

Gap penalty for the whole sequence is just the total number of gap characters times a constant.

Gap penalty for the whole sequence is the function.
\[ N \times (\text{gap initiation penalty}) + E \times (\text{gap extension penalty}) \]

where \( N \) is the number of gap initiation characters, \( E \) is the number of gap extension characters.
Example: affine gap

gap initiation $= -5$  
gap extension $= -1$

-5  -5

AGGCTACT~T~TCA
GGCTACTATATCA

-10

-5 -1

AGGCTACTTTT~CA
GGCTACTATATATCA

-6
Affine Gap DP, two ways.
Either this...

♫ You can have 5 types of arrows, instead of just three.

(1) Match
(2) Open a gap in first sequence.
(3) Open a gap in second sequence.
(4) Extend a gap in first sequence.
(5) Extend a gap in second sequence.

—or this…—

♫ Variable length arrows. no fixed number.
Arrows carry length info.
Affine gap DP algorithm using variable length arrows

\[ S_{i,j} = \max_n \{ S_{i-1,j-1} + s(i,j), \]
\[ S_{i-1-n,j-1} + s(i,j) - g_{\text{init}} - (n-1) g_{\text{ext}}, \]
\[ S_{i-1,j-1-n} + s(i,j) - g_{\text{init}} - (n-1) g_{\text{ext}} \} \]

...where \( s(i,j) \) is the substitution score, \( n \) is the length of the gap, \( g_{\text{init}} \) is the gap initiation penalty, and \( g_{\text{ext}} \) is the gap extension penalty.

Notes: All arrows end in match. Gap-to-gap not possible. Local or semi-global only. End-gaps not scored. Arrows still translate to an alignment. Still optimal.
Traceback for linear gap penalty

Each arrow advances either one sequence or both, by 1. Each column has one arrow.
Traceback for affine gap DP with variable length arrows

Each arrow advances one sequence by 1, the other sequence by $n$.

Output of one arrow is $n$ columns. $n =$ arrowlength.

Number of arrows is $\leq$ number of columns.

End gaps are ignored in local alignment.
Does deletion-to-insertion make sense???

Special rules may apply for going from $I$ to $D$ and $D$ to $I$.

If you think this alignment does not make sense, then $D$ to $I$ and $I$ to $D$ can simply be disallowed in the DP algorithm. Most programs do this.

[Exception: For a global alignment, D-to-I or I-to-D arrows are allowed at the ends of alignments because there is no other way to complete the matrix.]
If it doesn’t make biological sense, your program shouldn’t do it!
Q: How do I know what the gap penalties should be?
A: Optimize the gap penalties against "gold standard" alignments.
The gold standard for alignments: BAliBase

- A database of curated multiple sequence alignments derived from structure-based alignments.

A structure-based alignment. Based on D structures.

Two similar structures may be superimposed 3D. The parts that overlay well are the matches (purple and green), and the parts that do not overlay well are the insertions (yellow) and deletions (red).

*Aligned positions have similar chemical 3D environment.*
*Unaligned positions deviate in 3D, therefore their environment is different.*
Jargon word: *indel*

- Since deletions are just the flip-side of insertions, often we want a word that lumps both insertions and deletions together -- *indel*. Say it when you don't care whether it's an *insertion* or a *deletion*. It's another word for gap.
Example of a "gold standard" alignment

A structure-based alignment is a sequence alignment that comes from a protein structure superposition.

What do you see?
Lots of mismatches (%id=29),
A few indels (8),
Indels are usually > 1 residue.
Gap penalties should be numbers that consistently reproduce the gold standard alignments.
How we do alignment research

1. Hypothesize a scoring function that makes better alignments.
2. Implement it.
3. Train it.
4. Publish it iff it works.
How do we "train it"?

ANSWER: “machine learning”.

Ingredients for machine learning:
1) A training set
2) A test set
3) An objective function
4) A parameter space
5) A learning algorithm
Machine learning for alignment

- Create a database of sequence alignments (BAliBase)
  - Divide database into Training set and Test set.
  - Alignments must be non-redundant. No duplicates.
  - Alignments must be representative. Enough of them that adding more won’t change the results.
  - Test set must have no overlap with Training set.
- Define an objective function, $\varepsilon$
  - $\varepsilon$ is defined as a function of alignments, given parameters.
  - $\varepsilon$ reaches an optimum as alignments approach the truth for members of the Training set.
- Explore parameter space (try all reasonable values)
  - may use exhaustive search, gradient descent, Monte Carlo, bracketing, etc.
- Cross-validate when all training is done.
  - Report the accuracy on a Test set not on the Training set.
Possible objective functions for **machine learning** an alignment function

"*number of gaps*" metric  
(minimized)

\[ \varepsilon = \sum |N_{\text{gap}}(\text{test}) - N_{\text{gap}}(\text{gold std})| \]

"*gold standard matches*" metric  
(maximized)

\[ \varepsilon = \sum N(\text{gold std aligned positions found in test alignments}) \]
Bioinformatics research story:

*Bioinformatics* 22(4):413-420, 2006

**Improved pairwise alignments of proteins in the Twilight Zone using local structure predictions**

**Yao-ming Huang and Christopher Bystroff**

Center for Bioinformatics, Dept of Biology, Rensselaer Polytechnic Institute, Troy, New York 12180 USA

Objective function $\varepsilon$ used in Huang & Bystroff:

$$\varepsilon = \text{total number of correct matches}, \text{ based on bAliBase.}$$

Different substitution matrices (left) gave different alignments when sequence similarity is in the “Twilight zone” (very difficult alignments).
Ex 4: optimizing gap penalty

- In a browser, goto to NCBI. Search Protein database for 1DRF. Save. Then 2DRC. Save.
- Select both. Save as Alignment.
- Select alignment. Align using Kalign. Gop=50, Gep=8, Free end gaps (terminal gap=0). No bonus. Get...
  - G = | # of indels - 8 |. (8 indels in gold std, see Slide 30)
  - ID = percent identity. (mouse-over-R-clk/statistics/distance, identity, percent)
- Optimize $\varepsilon = ID - 5 \ G$  (number of gaps should count for more than %id)
- What is the optimal Gop, Gep? Write the Gop, Gep, G and ID on a piece of paper with your name on it and turn it in next time.
Kalign parameter window in UGENE

Causes fewer indels

Causes shorter indels

Forces/ignores alignment of termini

Causes more matches*

*Increasing the bonus is essentially the same as decreasing the combined gap penalties.
**Ex 4: exhaustive approach**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5</strong></td>
<td>+20, 35%</td>
<td></td>
<td></td>
<td>+50, 40%</td>
</tr>
<tr>
<td><strong>10</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>20</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>50</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>70</strong></td>
<td>-8, 5%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>90</strong></td>
<td>-8, 5%</td>
<td></td>
<td></td>
<td>-8, 5%</td>
</tr>
</tbody>
</table>

In each box record: $G = \# \text{ of gaps} - 8$, $ID = \% \text{ Identity}$
Ex 4: bracketing

- Increase Gop by 10. Calculate $\varepsilon$.
- Increase Gop by another 10. Calculate $\varepsilon$.
- Plot it. If it is going up, go the other way. If it is going down, keep going. If it goes down then up, you have bracketed the solution.

![Graph showing bracketing process](#)
Some things to ponder

• How does scoring approximate the evolutionary distance?
• When should you use “semi-global” alignment?
• What is an evolutionary “hot spot”?
• How is dynamic programming different for local alignment?
• Is the affine gap penalty more biologically relevant than a linear gap penalty? Why?
• Why are structure-based alignments considered the gold standard of sequence alignment?
• What does it mean for a deletion to follow immediately after an insertion, evolutionarily? Structurally?