Introduction to Sequence Analysis

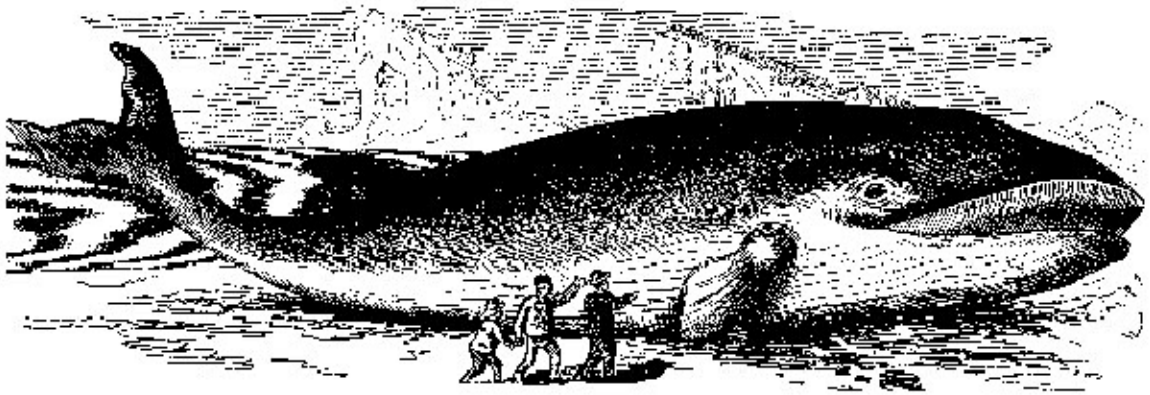
I. Pairwise Alignment

What is sequence analysis and why

- “To organize, classify and parse” sequence data
- To infer evolutionary paths
- To infer structures and functions of RNA or proteins
  - Although the most reliable way to determine structure of function of biological molecules (DNA,RNA,proteins) is by direct experimentation, sequencing is faster, easier and cheaper
  - New species or new proteins are not “invented” from scratch, but rather from adapted from preexisting sequences

Therefore, finding **homology** (similarity) between sequences is fruitful.

How different are we?



WHALE GTGTGGTCTCGTGATCAAAGGCGAAAGGTGGCTCTAGAGAATCCC

HUMAN GTGTGGTCTCGCGATCAGAGGCGCAAGATGGCTCTAGAGAATCCC

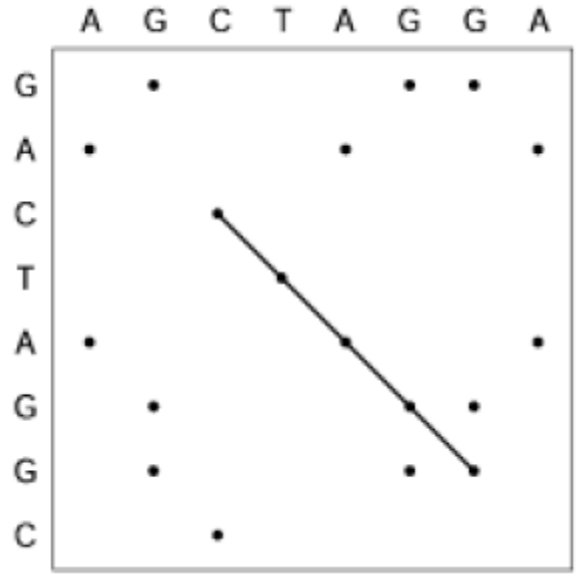
- Define a metric so we can have a measure of similarity or distance –a score for alignments
- Given a metric one can always find the best alignment(s)
- Choose a metric that give the biologically likely alignment the highest score
- Evaluate the significance of alignment scores

Visualizing pairwise alignment

Dot Matrix Alignment

GACTAGGC

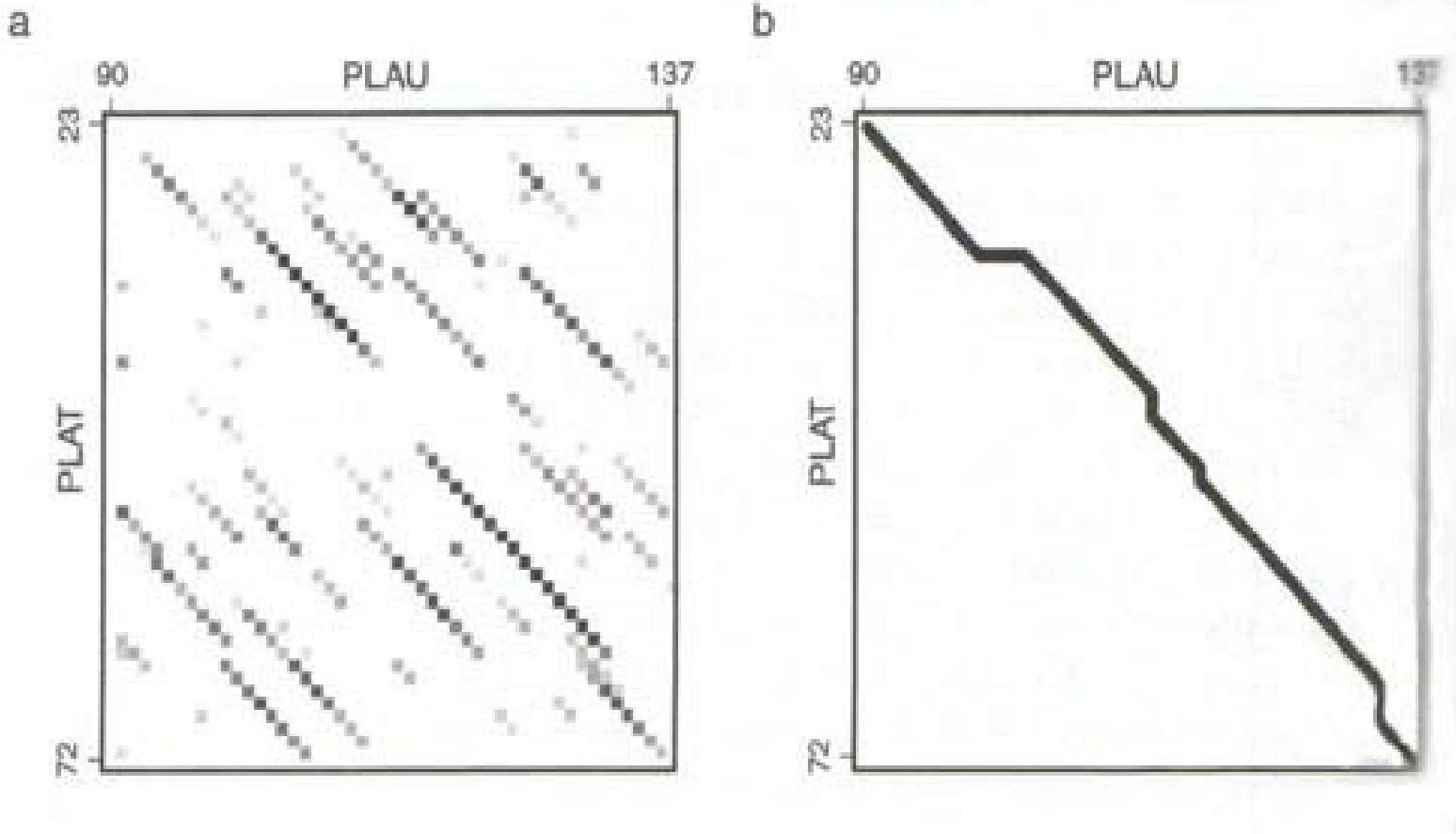
AGCTAGGA



Gibbs & McIntyre

(1970)

Dot Matrix Alignment



Dot Matrix Alignment

- Has many variations
- Can be used to find sequence repeats
- Find self-complimentary subsequences of RNA to predict secondary structure

Alignment scores: an Example

- GCGCATGGATTGAGCGA
- TGCGCCATTGATGACCA

A possible alignment:

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

Alignments

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

Three elements:

- Perfect matches 
- Mismatches 
- Gaps (insertion or deletion) 

We look for evidence that two sequences descent from a common ancestor by mutation or selection.

Choosing Alignments

There are many possible alignments

For example, compare:

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

to

```
------GCGCATGGATTGAGCGA
TGCGCC----ATTGATGACCA--
```

Which one is better?

Notation

- $X=x_1x_2...x_i...x_n$
- $X=y_1y_2...y_i...y_n$

- $x_i\ y_i$ from {A,T,G,C}

Ungapped pairwise alignments

- Null (Random) model: two sequences are unrelated, therefore alignment is coincidence
- Simple Alternative: two sequences are related so matches have higher frequency

Alignment scores

$$P(x, y \mid H_0) = \prod_i q_{x_i} \prod_j q_{y_j}$$

$$P(x, y \mid H_1) = \prod_i p_{x_i y_i}$$

Log odds ratio:

$$\log \frac{P(x, y \mid H_1)}{P(x, y \mid H_0)} = \log \frac{\prod_i p_{x_i y_i}}{\prod_i p_{x_i} \prod_i p_{y_i}} = \log \prod_i \frac{p_{x_i y_i}}{p_{x_i} p_{y_i}} = \sum_i \log \frac{p_{x_i y_i}}{p_{x_i} p_{y_i}}$$

$$s(a, b) = \log \frac{P_{ab}}{P_a P_b}$$

Gap penalty

- P(gap with seq x|H_1)=P(having a gap with length g)
  x P(the gap seq is x|gap g)
  - f(g)
  - Assuming independent residues

$$\prod_{k=1}^{g} q_{x_k}$$ for both H1 and H0

- Under the assumption of same distributions of residues in gap regions and aligned regions, Log likelihood is log[f(g)] since $\prod_{k=1}^{g} q_{x_k}$ cancels out
  - Exceptions: for example, polar AAs are more likely to appear in gaps because gaps are morelikely to be in loops on surface

Scoring Rule example

- Example Score =

  (# matches) – (# mismatches) – (# gaps) x 2
- Log[P(related)/P(unrelated)]
- An additive scoring rule assumes independence among locations (often reasonable for DNA and proteins, not for structural RNAs)
- Penalty on gaps are heavier than mismatches

Example

```
-GCGC-ATGGATTGAGCGA
TGCGCCATTGAT-GACC-A
```

Score: (+1x13) + (-1x2) + (-2x4) = 3

```
------GCGCATGGATTGAGCGA
TGCGCC----ATTGATGACCA--
```

Score: (+1x5) + (-1x6) + (-2x11) = -23

Alignment using Dynamic Programming

Optimal Alignment

- Optimal alignment is achieved at best similarity score *d*, thus is determined by the scoring rule

$$\mathrm{d}(s,t) = \max_{\text{alignment of } s\&t} \text{score(alignment)}$$

Finding the Best Alignment Score

- The additive form of the score allows to perform dynamic programming to find the best score efficiently
- Guaranteed to find the best alignment

Assume that an Optimal Score Exists

- d(s,t) – Optimal score for globally aligning “s” and “t”

The Idea

- The best alignment that ends at a given pair of bases: the best among best alignments of the sequences up to that point, plus the score for aligning the two additional bases.

Dynamic Programming

Consider the best alignment score of two sequences *s, t* at base/residue *i+1, j+1,* respectively:

$$\ldots \quad - \quad s[i] \quad - \quad s[i+1] \quad - \quad \ldots$$

$$\ldots \quad - \quad t[j] \quad - \quad t[j+1] \quad - \quad \ldots$$

$d(s[i+1], t[j+1])$ is the * best * alignment score

up to $s[i+1], t[j+1]$

$\sigma(s[i+1], t[j+1])$ is the score for aligning the two bases

Dynamic Programming

The best alignment must be in one of three cases:

1. Last position is (*s[i+1], t[j +1]*)
2. Last position is (-, *t[j +1]*)
3. Last position is (*s[i +1]*, -)

$$d(s[1..i+1], t[1..j+1]) = d(s[1.., i], t[1..j]) + \sigma(s[i+1], t[\,j+1])$$

Dynamic Programming

The best alignment must be in one of three cases:

1. Last position is (*s[i+1], t[j +1]*)
2. Last position is (-, *t[j +1]*)
3. Last position is (*s[i +1]*, -)

$$d(s[1..i+1], t[1..j+1]) = d(s[1.., i+1], t[1..j]) + \sigma(-, t[j+1])$$

Dynamic Programming

The best alignment must be in one of three cases:

1. Last position is (*s[i+1], t[j +1]*)
2. Last position is (-, *t[j +1]*)
3. Last position is (*s[i +1]*, -)

$$d(s[1..i+1], t[1..j+1]) = d(s[1..,i], t[1..j+1]) + \sigma(s[i+1], -)$$

Dynamic Programming

$$d(s[i+1],t[j+1]) = \max\begin{pmatrix} d(s[i],t[j]) + \sigma(s[i+1],t[j+1]) \\ d(s[i],t[j+1]) + \sigma(s[i+1],-) \\ d(s[i+1],t[j]) + \sigma(-,t[j+1]) \end{pmatrix}$$

Dynamic Programming

- Of course, we first need to handle the base cases in the recursion:

$$d(s[0], t[0]) = 0$$

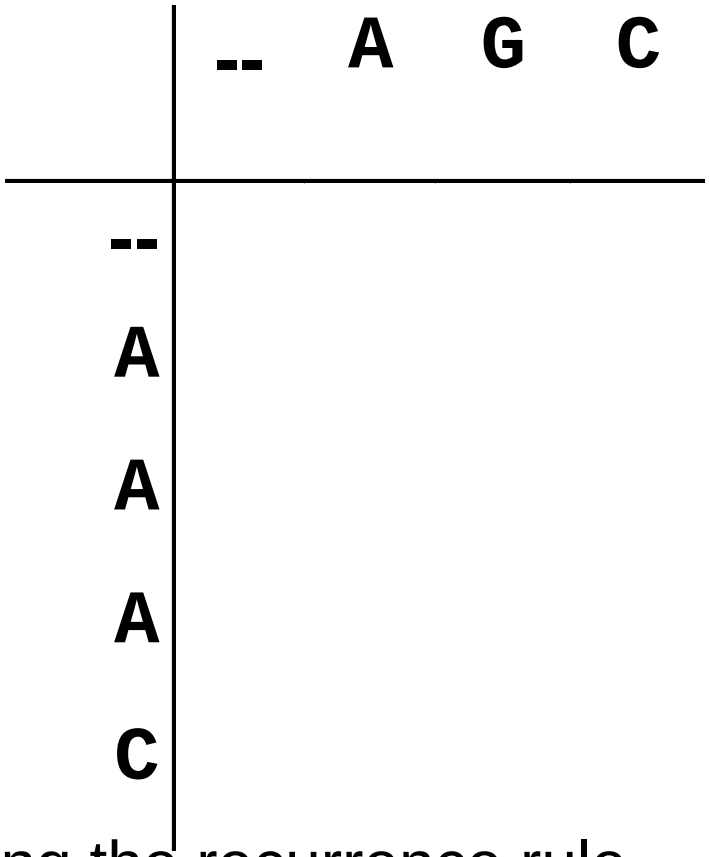
$$d(s[i+1], t[0]) = d(s[i], t[0]) + \sigma(s[i+1], -)$$

$$d(s[0], t[j+1]) = d(s[0], t[j]) + \sigma(-, t[j+1])$$

Dynamic Programming

– A G C –

– A A A C –



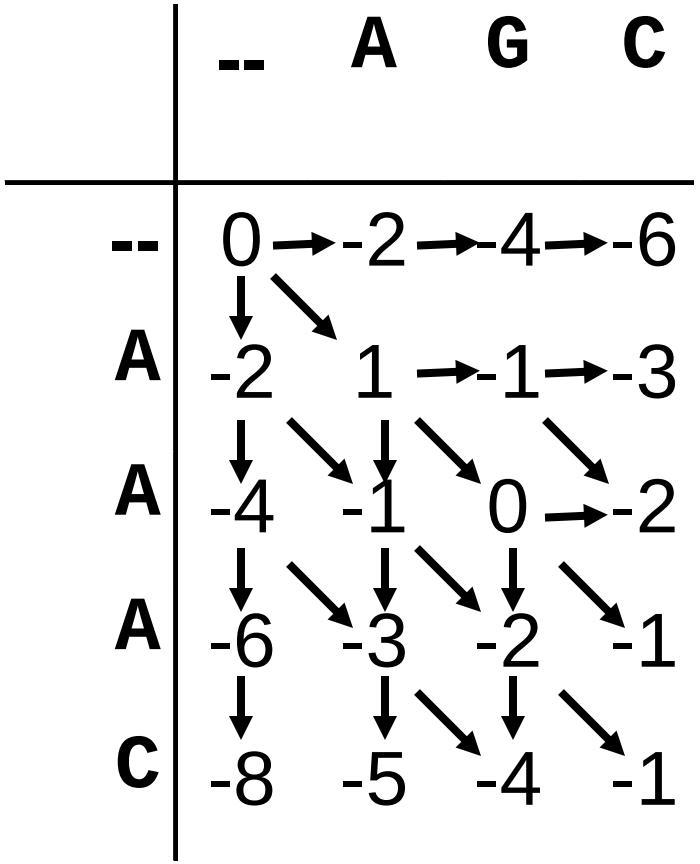
We fill the matrix using the recurrence rule

Dynamic Programming

```
----AGC   -8
AAAC---

AGC-      -1
AAAC

  AGC----
  ---AAAC
```

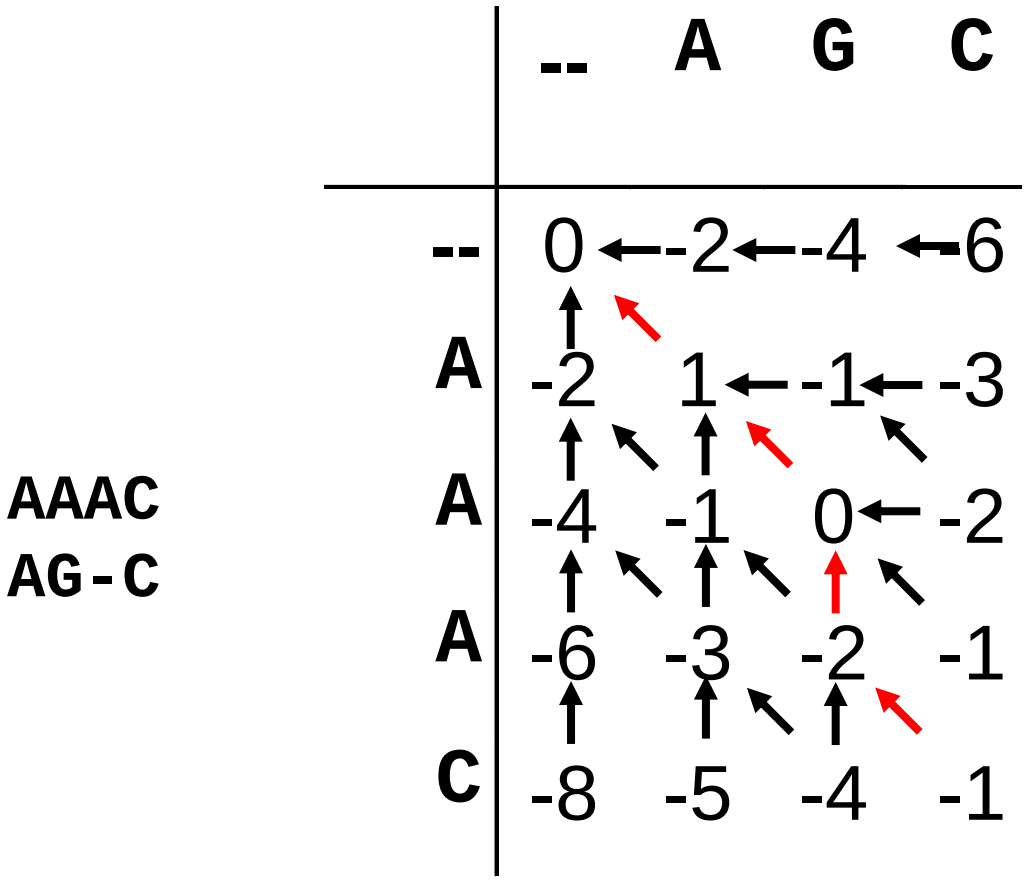


Dynamic Programming

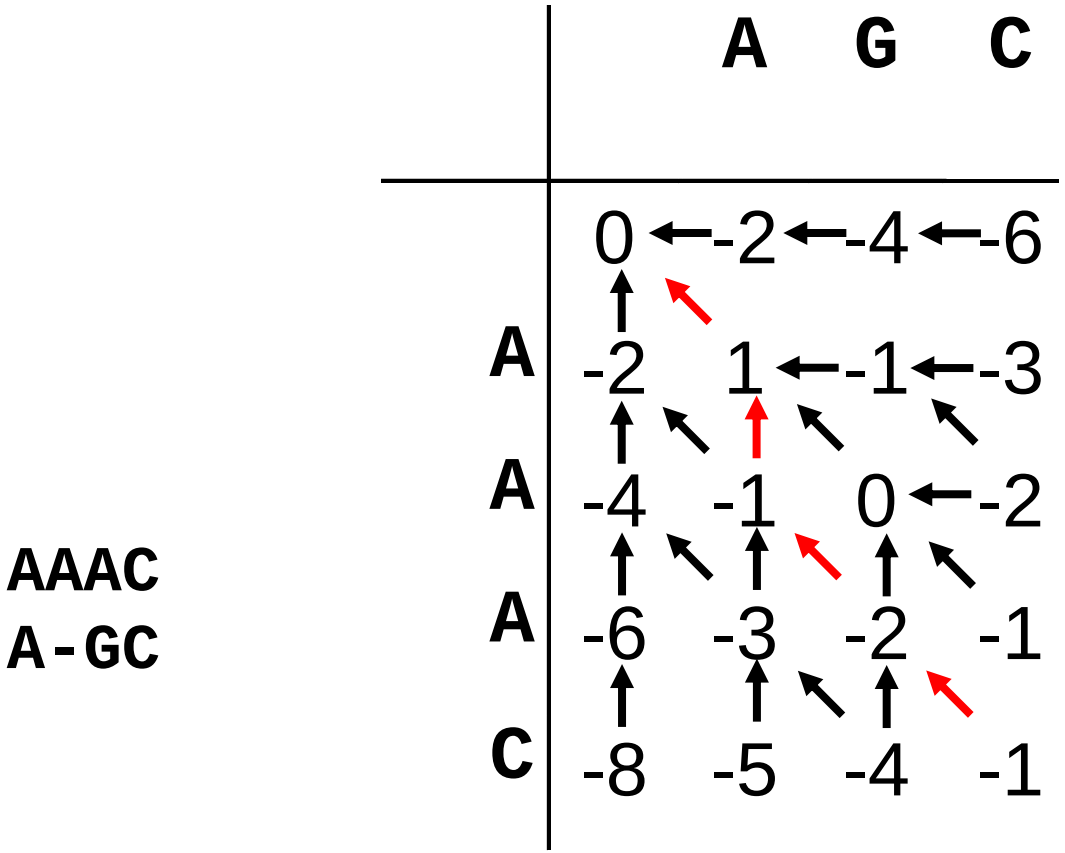
	--	A	G	C
--	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

Conclusion:
d(AAAC,AGC) = -1

Reconstructing the Best Alignment



More than one best alignment



Complexity

Space: $O(mn)$

Time: $O(mn)$

- Filling the matrix $O(mn)$
- Backtrace $O(m+n)$

Needleman & Wunsch (1970)

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

J. Mol. Biol. 48: 443-453

Local Alignment

- We have introduced global alignment
  - Align two sequences from start to end
- local alignment: Often we are searching for homologous regions but don't know where they are
  - A local Alignment between sequence *s* and sequence *t* is an alignment with maximum similarity between a substring of s and a substring of *t*.

Smith and Waterman (1981)

"Identification of Common Molecular Subsequences"

J. Mol. Biol., 147:195-197

Best-aligned Subsequences

To continue or not to continue?

extend previous alignment or start over

$$d(s[i+1],t[j+1]) = \max \begin{pmatrix} d(s[i],t[j]) + \sigma(s[i+1],t[j+1]) \\ d(s[i],t[j+1]) + \sigma(s[i+1],-) \\ d(s[i+1],t[j]) + \sigma(-,t[j+1]) \\ 0 \end{pmatrix}$$

Match = 5, Mismatch = -4, Gap w= -7

		j=0	1	2	3	4	5	6	7	8	9	10	11	12
i=		-	G	C	T	G	G	A	A	G	G	C	A	T
0	-	0	0	0	0	0	0	0	0	0	0	0	0	0
1	G	0												
2	C	0												
3	A	0												
4	G	0												
5	A	0												
6	G	0												
7	C	0												
8	A	0												
9	C	0												
10	T	0												

Note: different scoring rule

Match = 5, Mismatch = -4, Gap w= -7

		j=0	1	2	3	4	5	6	7	8	9	10	11	12
i=		-	G	C	T	G	G	A	A	G	G	C	A	T
0	-	0	0	0	0	0	0	0	0	0	0	0	0	0
1	G	0	$S_{1,1}$											
2	C	0												
3	A	0												
4	G	0												
5	A	0												
6	G	0												
7	C	0												
8	A	0												
9	C	0												
10	T	0												

$$S_{1,1} = \max \begin{Bmatrix} S_{0,0} + s_{G,G} = 0 + 5 = 5 \\ S_{1,0} + w = 0 - 7 = -7 \\ S_{0,1} + w = 0 - 7 = -7 \\ 0 \end{Bmatrix} = 5$$

Match = 5, Mismatch = -4, Gap w= -7

		j=0	1	2	3	4	5	6	7	8	9	10	11	12
i=		-	G	C	T	G	G	A	A	G	G	C	A	T
0	-	0	0	0	0	0	0	0	0	0	0	0	0	0
1	G	0	5	$S_{1,2}$										
2	C	0												
3	A	0												
4	G	0												
5	A	0												
6	G	0												
7	C	0												
8	A	0												
9	C	0												
10	T	0												

$$S_{1,2} = \max \begin{Bmatrix} S_{0,1} + s_{G,C} = 0 - 4 = -4 \\ S_{1,2} + w = 5 - 7 = -2 \\ S_{0,2} + w = 0 - 7 = -7 \\ 0 \end{Bmatrix} = 0$$

Match = 5, Mismatch = -4, Gap w= -7

		j=0	1	2	3	4	5	6	7	8	9	10	11	12
i=		-	G	C	T	G	G	A	A	G	G	C	A	T
0	-	0	0	0	0	0	0	0	0	0	0	0	0	0
1	G	0	5	0										
2	C	0	0	$S_{2,2}$										
3	A	0												
4	G	0												
5	A	0												
6	G	0												
7	C	0												
8	A	0												
9	C	0												
10	T	0												

$$S_{2,2} = \max \begin{Bmatrix} S_{1,1} + s_{C,C} = 5 + 5 = 10 \\ S_{2,1} + w = 0 - 7 = -7 \\ S_{1,2} + w = 0 - 7 = -7 \\ 0 \end{Bmatrix} = 10$$

Match = 5, Mismatch = -4, Gap w= -7

	0	G	C	T	G	G	A	A	G	G	C	A	T
0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	0	0	5	5	0	0	5	5	0	0	0
C	0	0	10	3	0	1	1	0	0	1	10	3	0
A	0	0	3	6	0	0	6	6	0	0	3	15	8
G	0	5	0	0	11	5	0	2	11	5	0	8	11
A	0	0	1	0	4	7	10	5	4	7	1	5	4
G	0	5	0	0	5	9	3	6	10	9	3	0	1
C	0	0	10	3	0	2	5	0	3	6	14	7	0
A	0	0	3	6	0	0	7	10	3	0	7	19	12
C	0	0	5	0	2	0	0	3	6	0	5	12	15
T	0	0	0	10	3	0	0	0	0	2	0	5	17

Match = 5, Mismatch = -4, Gap w= -7

	0	G	C	T	G	G	A	A	G	G	C	A	T
0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	0	0	5	5	0	0	5	5	0	0	0
C	0	0	10	3	0	1	1	0	0	1	10	3	0
A	0	0	3	6	0	0	6	6	0	0	3	15	8
G	0	5	0	0	11	5	0	2	11	5	0	8	11
A	0	0	1	0	4	7	10	5	4	7	1	5	4
G	0	5	0	0	5	9	3	6	10	9	3	0	1
C	0	0	10	3	0	2	5	0	3	6	14	7	0
A	0	0	3	6	0	0	7	10	3	0	7	19	12
C	0	0	5	0	2	0	0	3	6	0	5	12	15
T	0	0	0	10	3	0	0	0	0	2	0	5	17

Best aligned subsequences

```
G   A   A   G   -   G   C   A
|       |   |       |   |   |
G   C   A   G   A   G   C   A
```

6 matches: 6 * 5 = 30

1 mismatch: -4

1 gaps: 1 * -7 = -7

Total: 19

- http://blast.wustl.edu/doc/infotheory.html