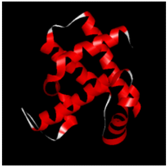
Pairwise sequence alignment

Christoph Dieterich

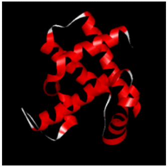
Department of Evolutionary Biology
Max Planck Institute for Developmental Biology



Example

Alignment between very similar human alpha- and beta globins:

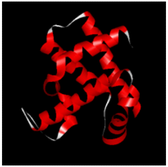
```
GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
G+ +VK+HGKKV  A+++++AH+D++ +++++LS+LH  KL
GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL
```



Example

Plausible alignment to leghaemoglobin from yellow lupin:

```
GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
++ ++++H+ KV   + +A  ++          +L+ L+++H+ K
NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG
```



Example

A spurious high-scoring alignment of human alpha globin to a nematode glutathione *S*-transferase homologue:

```
GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSD----LHAHKL
GS+ + G +   +D L  ++ H+ D+  A +AL D    ++AH+
GSGYLVGDSLTFVDLLVAQHTADLL--AANAALLDEFPQFKAHQE
```

Assessing the quality of an alignment

The goal is to use similarity-based alignments to uncover *homology*, while avoiding *homoplasy*
Homoplasy: random mutations that appear in parallel or convergently in two different lineages.

The scoring model

Computation of an alignment critically depend on the choice of parameters. Generally no existing scoring model can be applied to all situations.

- Evolutionary relationships between the sequences are reconstructed. Here scoring matrices based on mutation rates are usually applied.
- Protein domains are compared. Then the scoring matrices should be based on composition of domains and their substitution frequency.

The scoring model

Computation of an alignment critically depend on the choice of parameters. Generally no existing scoring model can be applied to all situations.

- Evolutionary relationships between the sequences are reconstructed. Here scoring matrices based on mutation rates are usually applied.
- Protein domains are compared. Then the scoring matrices should be based on composition of domains and their substitution frequency.

Substitution matrices

To be able to score an alignment, we need to determine score terms for each aligned residue pair.

Definition

A substitution matrix S over an alphabet $\Sigma = \{a_1, \ldots, a_\kappa\}$ has $\kappa \times \kappa$ entries, where each entry (i, j) assigns a score for a substitution of the letter a_i by the letter a_j in an alignment.

Substitution matrices

Basic idea: Follow scheme of statistical hypothesis testing.

$$Score\left(\begin{array}{c} a \\ b \end{array} \right) = \frac{f(a,b)}{f(a) \cdot f(b)}$$

Frequencies of the letters $f(a)$ as well as substitution frequencies $f(a,b)$ stem from a representative data set.

Null hypothesis / Random model

Given a pair of aligned sequences (without gaps), the null hypothesis states that the two sequences are unrelated (not homologous). The alignment is then random with a probability described by the model R. The unrelated or *random* model R assumes that in each aligned pairs of residues the two residues occur independently of each other. Then the probability of the two sequences is:

$$\mathbb{P}(X, Y \mid R) = \mathbb{P}(X \mid R)\mathbb{P}(Y \mid R) = \prod_i p_{x_i} \prod_i p_{y_i}.$$

Match model

In the *match* model M, describing the alternative hypothesis, aligned pairs of residues occur with a joint probability p_{ab}, which is the probability that a and b have each evolved from some unknown original residue c as their common ancestor. Thus, the probability for the whole alignment is:

$$\mathbb{P}(X, Y \mid M) = \prod_i p_{x_i y_i}.$$

Odds ratio

The ratio of the two gives a measure of the relative likelihood that the sequences are related (model M) as opposed to being unrelated (model R). This ratio is called *odds ratio*:

$$\frac{\mathbb{P}(X, Y \mid M)}{\mathbb{P}(X, Y \mid R)} = \frac{\prod_i p_{x_i y_i}}{\prod_i p_{x_i} \prod_i p_{y_i}} = \prod_i \frac{p_{x_i y_i}}{p_{x_i} p_{y_i}}$$

Log-odds ratio

To obtain an additive scoring scheme, we take the logarithm (base 2 is usually chosen) to get the *log-odds ratio:*

$$S = \log(\frac{\mathbb{P}(X, Y \mid M)}{\mathbb{P}(X, Y \mid R)}) = \log(\prod_i \frac{p_{x_i y_i}}{p_{x_i} p_{y_i}}) = \sum_i s(x_i, y_i),$$

with

$$s(a, b) := \log\left(\frac{p_{ab}}{p_a p_b}\right).$$

PAM matrices

Definition (PAM)

One point accepted mutation (1 PAM) is defined as an expected number of substitutions per site of 0.01. A 1 PAM substitution matrix is thus derived from any evolutionary model by setting the row sum of off-diagonal terms to 0.01 and adjusting the diagonal terms to keep the row sum equal to 1.

Definition (Jukes-Cantor Model)

The basic assumption is equality of substitution frequency for any nucleotide at any site. Thus, changing a nucleotide to each of the three remaining nucleotides has probability α per time unit. The rate of nucleotide substitution per site per time unit is then $r = 3\alpha$.

PAM matrices

Let's build a PAM 1 matrix under a Jukes-Cantor model of sequence evolution.

$$\begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix}$$

PAM matrices

We scale matrix entries such that the expected number of substitutions per site is 0.01 $= 3\alpha$ and obtain a probabilty matrix:

$$\begin{pmatrix} 0.99 & 0.003 & 0.003 & 0.003 \\ 0.003 & 0.99 & 0.003 & 0.003 \\ 0.003 & 0.003 & 0.99 & 0.003 \\ 0.003 & 0.003 & 0.003 & 0.99 \end{pmatrix}$$

PAM matrices

A scoring matrix is then obtained by computing the log-odds ratios:

$$s(a, b) := \log \left(\frac{p_{ab}}{p_a p_b} \right) .$$

with $p_A = p_C = p_G = p_T = 0.25$ and joint probabilities as given by the PAM probability matrix.

PAM matrices

This leads to the following substitution score matrix:

$$\begin{pmatrix} 398 & -438 & -438 & -438 \\ -438 & 398 & -438 & -438 \\ -438 & -438 & 398 & -438 \\ -438 & -438 & -438 & 398 \end{pmatrix}$$

MPI for Developmental Biology, Tübingen

BLOCKS and BLOSUM matrices

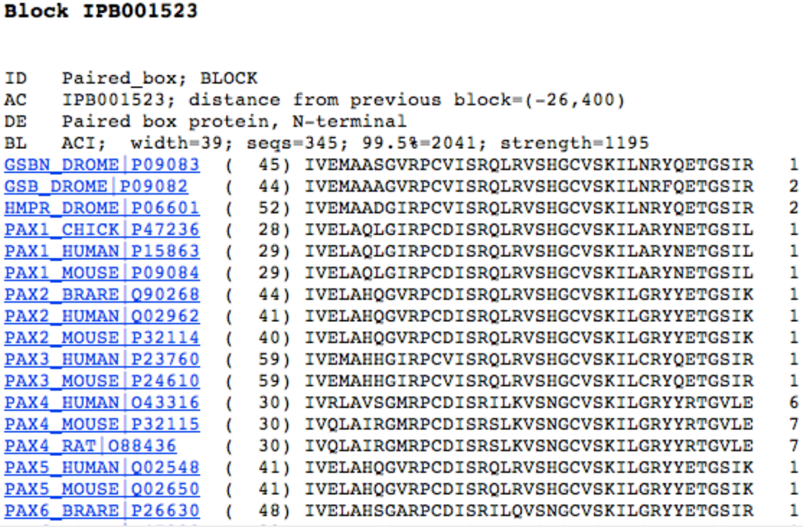
The BLOSUM matrices were derived from the database BLOCKS[1] Blocks are multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins.

[1] Henikoff, S and Henikoff, JG (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A. 89(22):10915-9. BLOCKS database server: http://blocks.fhcrc.org/

BLOCKS and BLOSUM matrices

For the scoring matrices of the BLOSUM (=BLOcks SUbstitution Matrix) family all blocks of the database are evaluated columnwise. For each possible pair of amino acids the frequency $f(a_i, a_j)$ of common pairs (a_i, a_j) in all columns is determined.

BLOCKS and BLOSUM matrices



Block IPB001523

```
ID   Paired_box; BLOCK
AC   IPB001523; distance from previous block=(-26,400)
DE   Paired box protein, N-terminal
BL   ACI;  width=39; seqs=345; 99.5%=2041; strength=1195
GSBN_DROME|P09083  (  45) IVEMAASGVRPCVISRQLRVSHGCVSKILNRYQETGSIR   1
GSB_DROME|P09082   (  44) IVEMAAAGVRPCVISRQLRVSHGCVSKILNRFQETGSIR   2
HMPR_DROME|P06601  (  52) IVEMAADGIRPCVISRQLRVSHGCVSKILNRYQETGSIR   2
PAX1_CHICK|P47236  (  28) IVELAQLGIRPCDISRQLRVSHGCVSKILARYNETGSIL   1
PAX1_HUMAN|P15863  (  29) IVELAQLGIRPCDISRQLRVSHGCVSKILARYNETGSIL   1
PAX1_MOUSE|P09084  (  29) IVELAQLGIRPCDISRQLRVSHGCVSKILARYNETGSIL   1
PAX2_BRARE|Q90268  (  44) IVELAHQGVRPCDISRQLRVSHGCVSKILGRYYETGSIK   1
PAX2_HUMAN|Q02962  (  41) IVELAHQGVRPCDISRQLRVSHGCVSKILGRYYETGSIK   1
PAX2_MOUSE|P32114  (  40) IVELAHQGVRPCDISRQLRVSHGCVSKILGRYYETGSIK   1
PAX3_HUMAN|P23760  (  59) IVEMAHHGIRPCVISRQLRVSHGCVSKILCRYQETGSIR   1
PAX3_MOUSE|P24610  (  59) IVEMAHHGIRPCVISRQLRVSHGCVSKILCRYQETGSIR   1
PAX4_HUMAN|O43316  (  30) IVRLAVSGMRPCDISRILKVSNGCVSKILGRYYRTGVLE   6
PAX4_MOUSE|P32115  (  30) IVQLAIRGMRPCDISRSLKVSNGCVSKILGRYYRTGVLE   7
PAX4_RAT|O88436    (  30) IVQLAIRGMRPCDISRSLKVSNGCVSKILGRYYRTGVLE   7
PAX5_HUMAN|Q02548  (  41) IVELAHQGVRPCDISRQLRVSHGCVSKILGRYYETGSIK   1
PAX5_MOUSE|Q02650  (  41) IVELAHQGVRPCDISRQLRVSHGCVSKILGRYYETGSIK   1
PAX6_BRARE|P26630  (  48) IVELAHSGARPCDISRILQVSNGCVSKILGRYYETGSIR   1
```

BLOCKS and BLOSUM matrices

Altogether there are $\binom{n}{2}$ possible pairs that we can draw from this alignment. We now assume that the observed frequencies are equal to the frequencies in the population. Then

$$p_{aa} = observed / \binom{n}{2}$$

The observed frequency of a single amino acid is generally computed as $p_a = p_{aa} + \sum_{b \neq a} p_{ab}/2$. For this example we then get $p_A = 0.8 + 0.2/2 = 0.9$ and $p_C = 0.1$.

BLOCKS and BLOSUM matrices

Different levels of the BLOSUM matrix can be created by differentially weighting the degree of similarity between sequences. For example, a BLOSUM62 matrix is calculated from protein blocks such that if two sequences are more than 62% identical, then the contribution of these sequences is weighted to sum to one.

BLOCKS and BLOSUM matrices

BLOSUM62 is scaled so that its values are in half-bits, ie. the log-odds were multiplied by $2/\log_2 2$ and then rounded to the nearest integer value.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	
A	4	−1	−2	−2	0	−1	−1	0	−2	−1	−1	−1	−1	−2	−1	1	
R	−1	5	0	−2	−3	1	0	−2	0	−3	−2	2	−1	−3	−2	−1	−
N	−2	0	6	1	−3	0	0	0	1	−3	−3	0	−2	−3	−2	1	
D	−2	−2	1	6	−3	0	2	−1	−1	−3	−4	−1	−3	−3	−1	0	−

....

MPI for Developmental Biology, Tübingen

Gap penalties

Gaps are undesirable and thus penalized. The standard cost associated with a gap of length g is given either by a *linear* score

$$\gamma(g) = -gd$$

or an *affine* score

$$\gamma(g) = -d - (g-1)e,$$

where d is the *gap open* penalty and e is the *gap extension* penalty.

MPI for Developmental Biology, Tübingen

Gap penalties

Usually, $e < d$, with the result that less isolated gaps are produced, as shown in the following comparison:

Linear gap penalty:

```
GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
GSAQVKGHGKK--------VA--D----A-SALSDLHAHKL
```

Affine gap penalty:

```
GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
GSAQVKGHGKKVADA---------------SALSDLHAHKL
```

Alignment algorithms

Given a scoring scheme, we need to have an algorithm that computes the highest-scoring alignment of two sequences.
As for the edit distance-based alignments we will discuss alignment algorithms based on *dynamic programming*. They are guaranteed to find the optimal scoring alignment.
Note of caution: Optimal Pairwise alignment algorithms are of complexity $O(n \cdot m)$

Global alignment: Needleman-Wunsch algorithm

Problem

Consider the problem of obtaining the best global *alignment of two sequences. The Needleman-Wunsch algorithm is a dynamic program that solves this problem.*

Idea: Build up an optimal alignment using previous solutions for optimal alignments of smaller substrings.

Global alignment: Needleman-Wunsch algorithm

$$F : \{1, 2, \ldots, n\} \times \{1, 2, \ldots, m\} \to \mathbb{R}$$

in which $F(i, j)$ equals the best score of the alignment of the two prefixes $(x_1, x_2, \ldots, x_i)$ and $(y_1, y_2, \ldots, y_j)$.

Global alignment: Needleman-Wunsch algorithm

	0	x_1	x_2	x_3	$\ldots\ldots$	x_{i-1}		x_i	$\ldots\ldots$
0	$F(0,0)$								
y_1									
y_2									
y_3									
$\ldots$									
y_{j-1}						$F(i-1,j-1)$		$F(i,j-1)$	
							$\searrow$	$\downarrow$	
y_j	$-$	$-$	$-$	$-$	$-$	$F(i-1,j)$	$\rightarrow$	$\boxed{F(i,j)}$	
$\ldots$									
y_m									

We obtain $F(i,j)$ as the largest score arising from these three options:

$$F(i,j) := \max \begin{cases} F(i-1,j-1) + s(x_i,y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d. \end{cases}$$

This is applied repeatedly until the whole matrix $F(i,j)$ is filled with values.

Recursion

To complete the description of the recursion, we need to set the values of $F(i,0)$ and $F(0,j)$ for $i \neq 0$ and $j \neq 0$:
We set $F(i,0) =$ ________ for $i = 0, 1, \ldots, n$ and
we set $F(0,j) =$ ________ for $j = 0, 1, \ldots, m$.
The final value $F(n,m)$ contains the score of the best global alignment between X and Y.

Example of a global alignment matrix

D	0	G	A	T	T	A	G
0	0	-2	-4	-6	-8	-10	-12
A	-2	-1	-1	-3	-5	-7	-9
T	-4	-3	-2	0	-2	-4	-6
T	-6	-5	-4	-1	1	-1	-3
A	-8	-7	-4	-3	0	2	0
C	-10	-9	-6	-5	-2	0	1

Pseudo code of Needleman-Wunsch

```
Input: two sequences $X$ and $Y$
Output: optimal alignment and score $\alpha$
Initialization:
Set $F(i,0) := -i \cdot d$ for all $i = 0, 1, 2, \ldots, n$
Set $F(0,j) := -j \cdot d$ for all $j = 0, 1, 2, \ldots, m$
For $i = 1, 2, \ldots, n$ do:
    For $j = 1, 2, \ldots, m$ do:
        Set $F(i,j) := \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$
        Set backtrace $T(i,j)$ to the maximizing pair $(i', j')$
The score is $\alpha := F(n,m)$
Set $(i,j) := (n,m)$

repeat
    if $T(i,j) = (i-1, j-1)$ print $\binom{x_i}{y_j}$
    else if $T(i,j) = (i-1, j)$ print $\binom{x_i}{-}$ else print $\binom{-}{y_j}$
    Set $(i,j) := T(i,j)$
until $(i,j) = (0,0)$.
```

MPI for Developmental Biology, Tübingen

Complexity of Needleman-Wunsch

We need to store $(n+1) \times (m+1)$ numbers. Each number takes a constant number of calculations to compute: three sums and a max.

Hence, for filling the matrix, the algorithm requires $O(nm)$ time and memory. Given the filled matrix, the construction of the alignment is done in time $O(n+m)$.

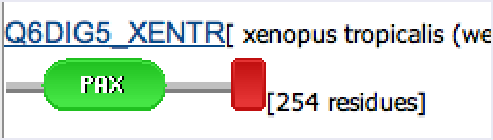
Local alignment: Smith-Waterman algorithm

Global alignment is applicable when we have two similar sequences that we want to align from end-to-end, e.g. two homologous genes from related species.

Local alignment: Smith-Waterman algorithm

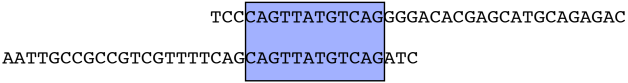
Problem

Global alignment is inapplicable to modular sequence.



Here we would like to find the best match between substrings *of two sequence.*

Local alignment: Smith-Waterman algorithm



```
                  TCCCAGTTATGTCAGGGACACGAGCATGCAGAGAC
AATTGCCGCCGTCGTTTTCAGCAGTTATGTCAGATC
```

Here the score of an alignment between two substrings would be larger than the score of an alignment between the full lengths strings.

Local alignment: Smith-Waterman algorithm

Definition

Let $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$ be two sequences over an alphabet Σ. Let δ be a score function for an alignment. A *local alignment* of X and Y is a global alignment of substrings $X' = x_{i_1} \ldots x_{i_2}$ and $Y' = y_{j_1} \ldots y_{j_2}$. An alignment $A = (X', Y')$ of substrings X' and Y' is an *optimal local alignment* of X and Y with respect to δ if

$$\delta(A) = \max_{A'}\{\delta(X', Y') | X' \text{ is a substring of } X, Y' \text{ is a substring of } Y\}$$

Example

Let $X = \texttt{AAAAACTCTCTCT}$ and $Y = \texttt{GCGCGCGCAAAAA}$. Let $s(a, a) = +1$, $s(a, b) = -1$ and $s(a, -) = s(-, a) = -2$ be a scoring function. Then an optimal local alignment

```
          AAAAA(CTCTCTCT)
          |||||
(GCGCGCGC)AAAAA
```

in this case has a score 5 whereas the optimal global alignment

```
AAAAACTCTCTCT
     | |
GCGCGCGCAAAAA
```

has score -11.

Local alignment: Smith-Waterman algorithm

The Smith-Waterman (Smith, T. and Waterman, M. Identification of common molecular subsequences. J. Mol. Biol. 147:195-197, 1981)local alignment algorithm is a modification of the global alignment algorithm.

Modification in main recursion

In the main recursion, we set the value of $F(i,j)$ to zero, if all attainable values at position (i,j) are negative:

$$F(i,j) = \max \begin{cases} 0, \\ F(i-1,j-1) + s(x_i,y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

The value $F(i,j) = 0$ indicates that we should start a new alignment at (i,j). This is because, if the best alignment up to (i,j) has a negative score, then it is better to start a new one, rather than to extend the old one.

Base conditions

For local alignments we need to set $F(i, 0) =$ ____ and $F(0, j) =$ ____ for all $i = 0, 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, m$.

Modification in traceback

Instead of starting the traceback at (n, m), we start it at the cell with the highest score: $\texttt{argmax}\ F(i, j)$. The traceback ends upon arrival at a cell with score 0, with corresponds to the start of the alignment.

Traceback via recursion

```
**Input:** Similarity matrix $M$ of two strings $s = s_1 \dots s_m$ and $t = t_1 \dots t_n$
**Output:** Optimal local alignment (s',t') of s and t
**Procedure Align(i,j):**
if M(i,j) 0 then
$s' := \epsilon$
$t' := \epsilon$
else
    if $(M(i,j) = M(i-1,j) + g$ then
    $(\bar{s}, \bar{t}) := \texttt{Align}(i-1, j)$
    $s' := \texttt{concat}(\bar{s}, s_i)$
    $t' := \texttt{concat}(\bar{t},' -')$
    else if $(M(i,j) = M(i,j-1) + g$ then
    $(\bar{s}, \bar{t}) := \texttt{Align}(i, j-1)$
    $s' := \texttt{concat}(\bar{s},' -')$
    $t' := \texttt{concat}(\bar{t}, t_j)$
    else
    $(\bar{s}, \bar{t}) := \texttt{Align}(i-1, j-1)$
    $s' := \texttt{concat}(\bar{s}, s_i)$
    $t' := \texttt{concat}(\bar{t}, t_j)$
return(s',t')
```

MPI for Developmental Biology, Tübingen

Local alignment: Smith-Waterman algorithm

For this algorithm to work, we require that the expected score for a random match is negative, i.e. that

$$\sum_{a,b \in \Sigma} p_a \cdot p_b \cdot s(a,b) < 0,$$

where p_a and p_b are the probabilities for seeing the symbol a or b respectively, at any given position. Otherwise, matrix entries will tend to be positive, producing long matches between random sequences.

Local vs. Global Alignment

The Global Alignment Problem tries to find the optimal path between vertices $(0, 0)$ and (n, m) in the matrix graph.
The Local Alignment Problem tries to find the optimal path among paths between arbitrary vertices (i, j) and (i', j') in the matrix graph such that $i < i'$ and $j < j'$.

Example

Smith-Waterman matrix of the sequences `GATTAG` and `ATTAC` with $s(a, a) = 1$, $s(a, b) = -1$ and $s(a, -) = s(-, a) = -2$:

F	0	G	A	T	T	A	G
0							
A							
T							
T							
A							
C							

Score: ____;

Alignment =