# Pairwise Alignment 

Anders Gorm Pedersen<br>Henrik Nielsen<br>Center for Biological Sequence Analysis

## Sequences are related

- Darwin: all organisms are related through descent with modification
- => Sequences are related through descent with modification
- => Similar molecules have similar functions in different organisms

BACTERIA


Phylogenetic tree based on ribosomal RNA:
three domains of life

## Sequences are related, II



Phylogenetic tree of globin-type proteins found in humans

## Why compare sequences?



Protein 1: binds oxygen

Sequence similarity

Protein 2: binds oxygen ?

- Determination of evolutionary relationships
- Prediction of protein function and structure (database searches).


## Dotplots: visual sequence comparison



1. Place two sequences along axes of plot
2. Place dot at grid points where two sequences have identical residues
3. Diagonals correspond to conserved regions

## Pairwise alignments



## Global versus local alignments

Global alignment: align full length of both sequences.


Local alignment: find best partial alignment of two sequences


## Pairwise alignment



Percent identity is not a good measure of alignment quality

## Pairwise alignments: alignment score



## Alignment scores: match vs. mismatch

Simple scoring scheme (too simple in fact...):

Matching amino acids: 5
Mismatch: 0

Scoring example:


## Pairwise alignments: conservative substitutions



## Amino acid properties



Serine (S) and Threonine ( $T$ ) have similar physicochemical properties


Aspartic acid (D) and Glutamic acid (E) have similar properties
=> Substitution of S/T or E/D occurs relatively often during evolution
=> Substitution of S/T or E/D should result in scores that are only moderately lower than identities

## Protein substitution matrices



## Pairwise alignments: insertions/deletions



## Alignment scores: insertions/deletions



Affine gap penalties:
Multiple insertions/deletions may be one evolutionary event => Separate penalties for gap opening and gap elongation

## Handout

Compute 4 alignment scores: two different alignments using two different alignment matrices (and the same gap penalty system)

Score 1: Alignment 1 + BLOSUM-50 matrix + gaps
Score 2: Alignment 1 + ID-6,3 matrix + gaps
Score 3: Alignment 2 + BLOSUM-50 matrix + gaps
Score 4: Alignment 2 + ID-6,3 matrix + gaps

Handout: summary of results

|  | Alignment 1 | Alignment 2 |
| :--- | :--- | :--- |
| BLOSUM-50 |  |  |
| ID-6,3 |  |  |

## Protein substitution matrices



## Protein substitution matrices: different types

- Identity matrix
(match vs. mismatch)
- Genetic code matrix
(how similar are the codons?)
- Chemical properties matrix
(use knowledge of physicochemical properties to design matrix)
- Empirical matrices
(based on observed pair-frequencies in hand-made alignments)
- PAM series
- BLOSUM series
- Gonnet


## Estimation of the PAM1 matrix

$607080 \quad 90 \quad 100$
alpha QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
beta KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF

| 60 | 70 | 80 | 90 | 100 | 110 |
| :--- | :--- | :--- | :--- | :--- | :--- |

- Start from given alignments of closely related proteins
- Count the aligned amino acid pairs (e.g., A aligned with A makes up $1.5 \%$ of all pairs. A aligned with C makes up $0.01 \%$ of all pairs, etc.)
- Expected pair frequencies are computed from single amino acid frequencies. (e.g, $f_{A, C}=f_{A} \times f_{C}=7 \% \times 3 \%=0.21 \%$ ).
- For each amino acid pair the substitution scores are essentially computed as:

$$
\log \frac{\text { Pair-freq(observed) }}{\text { Pair-freq(expected) }} \quad S_{\mathrm{A}, \mathrm{C}}=\log \frac{0.01 \%}{0.21 \%}=-1.3
$$

- To obtain the PAM1 (1 Percent Accepted Mutations) matrix, normalize pair frequencies to $1 \%$ difference before applying the logarithm
- To obtain higher number PAM matrices, extrapolate the PAM1 matrix via matrix multiplication


## Percent Accepted Mutations (PAM)

PAM (Percent Accepted Mutations) can be used as a measure of evolutionary distance.
Note: 100PAM does NOT mean that sequences are 100\% different!

The Limits of Sequence Similarity


In the "Twilight Zone", it becomes difficult to see whether sequences are related

## Estimation of the BLOSUM 50 matrix

- Use the BLOCKS database (ungapped alignments of especially conserved regions of multiple alignments)
- For each alignment in the BLOCKS database the sequences are grouped into clusters with at least $50 \%$ identical residues (for BLOSUM 50)
- All pairs of sequences are compared between clusters, and the observed pair frequencies are noted
- Substitution scores are calculated as for the PAM matrix

ID FIBRONECTIN_2; BLOCK
COG9_CANFA GNSAGEPCVFPFIFLGKQYSTCTREGRGDGHLWCATT COG9_RABIT GNADGAPCHFPFTFEGRSYTACTTDGRSDGMAWCSTT FA12_HUMAN LTVTGEPCHFPFQYHRQLYHKCTHKGRPGPQPWCATT HGFA_HUMAN LTEDGRPCRFPFRYGGRMLHACTSEGSAHRKWCATTH MANR_HUMAN GNANGATCAFPFKFENKWYADCTSAGRSDGWLWCGTT MPRI_MOUSE ETDDGEPCVFPFIYKGKSYDECVLEGRAKLWCSKTAN PB1 PIG
SFP1_BOVIN SFP3_BOVIN SFP4_BOVIN SP1_HORSE COG2_CHICK COG2_HUMAN COG2_MOUSE COG2_RABIT COG2_RAT COG9_BOVIN COG9_HUMAN COG9 MOUSE COG9_RAT FINC_BOVIN FINC_HUMAN FINC_RAT MPRI_BOVIN MPRI_HUMAN PA2R_BOVIN PA2R_RABIT

AITSDDKCVFPFIYKGNLYFDCTLHDSTYYWCSVTTY ELPEDEECVFPFVYRNRKHFDCTVHGSLFPWCSLDAD AETKDNKCVFPFIYGNKKYFDCTLHGSLFLWCSLDAD AVFEGPACAFPFTYKGKKYYMCTRKNSVLLWCSLDTE AATDYAKCAFPFVYRGQTYDRCTTDGSLFRISWCSVT GNSEGAPCVFPFIFLGNKYDSCTSAGRNDGKLWCAST GNSEGAPCVFPFTFLGNKYESCTSAGRSDGKMWCATT GNSEGAPCVFPFTFLGNKYESCTSAGRNDGKVWCATT GNSEGAPCVFPFTFLGNKYESCTSAGRSDGKMWCATS GNSEGAPCVFPFTFLGNKYESCTSAGRNDGKVWCATT GNADGKPCVFPFTFQGRTYSACTSDGRSDGYRWCATT GNADGKPCQFPFIFQGQSYSACTTDGRSDGYRWCATT GNGEGKPCVFPFIFEGRSYSACTTKGRSDGYRWCATT GNGDGKPCVFPFIFEGHSYSACTTKGRSDGYRWCATT GNSNGALCHFPFLYNNHNYTDCTSEGRRDNMKWCGTT GNSNGALCHFPFLYNNHNYTDCTSEGRRDNMKWCGTT GNSNGALCHFPFLYSNRNYSDCTSEGRRDNMKWCGTT ETEDGEPCVFPFVFNGKSYEECVVESRARLWCATTAN ETDDGVPCVFPFIFNGKSYEECIIESRAKLWCSTTAD GNAHGTPCMFPFQYNQQWHHECTREGREDNLLWCATT GNAHGTPCMFPFQYNHQWHHECTREGRQDDSLWCATT

## Substitution matrices and sequence similarity

Substitution matrices come as series of matrices calculated for different degrees of sequence similarity (different evolutionary distances).

| "Hard" matrices | "Soft" matrices |
| :--- | :--- |
| Designed for very similar <br> sequences | Designed for less similar sequences |
| High numbers in the BLOSUM <br> series (e.g., BLOSUM90) | Low numbers in the BLOSUM <br> series (e.g., BLOSUM30) |
| Low numbers in the PAM series <br> (e.g. PAM30) | High numbers in the PAM series <br> (e.g. PAM250) |
| Severe mismatch penalties | Less severe mismatch penalties |
| Yield short alignments with high <br> \%identity | Yield longer alignments with lower <br> \%identity |

## Pairwise alignment

Optimal alignment:
alignment having the highest possible score given a substitution matrix and a set of gap penalties

So:
best alignment can be found by exhaustively searching all possible alignments, scoring each of them and choosing the one with the highest score?

## The problem: <br> How many possible alignments are there?

Consider two sequences of two letters each: $A B$ and $X Y$. How many ways are there to align them?

```
Insert no gaps:
```

AB
XY

Insert one gap in each sequence:

| $\mathrm{A}-\mathrm{B}$ | $\mathrm{AB}-$ | $\mathrm{A}-\mathrm{B}$ | -AB | $\mathrm{AB}-$ | -AB |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $\mathrm{XY}-$ | $\mathrm{X}-\mathrm{Y}$ | -XY | $\mathrm{X}-\mathrm{Y}$ | -XY | $\mathrm{XY}-$ |

Insert two gaps in each sequence:

| $\mathrm{AB}--$ | --AB | $\mathrm{A}-\mathrm{B}-$ | $-\mathrm{A}-\mathrm{B}$ | $\mathrm{A}--\mathrm{B}$ | $-\mathrm{AB}-$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| --XY | $\mathrm{XY}--$ | $-\mathrm{X}-\mathrm{Y}$ | $\mathrm{X}-\mathrm{Y}-$ | $-\mathrm{XY}-$ | $\mathrm{X}--\mathrm{Y}$ |

In total: 13 ways!

## The problem:

## How many possible alignments are there?

Consider two sequences of length $n 1$ and $n 2$. How many ways are there to align them?

| $n 1 \backslash n 2$ | 0 | 1 | 2 | 3 | 4 | 5 |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 3 | 5 | 7 | 9 | 11 |
| 2 | 1 | 5 | 13 | 25 | 41 | 61 |
| 3 | 1 | 7 | 25 | 63 | 129 | 231 |
| 4 | 1 | 9 | 41 | 129 | 321 | 681 |
| 5 | 1 | 11 | 61 | 231 | 681 | 1683 |

## The problem: <br> How many possible alignments are there?

The number of possible pairwise alignments increases explosively with the length of the sequences:

$$
f\left(n_{1}, n_{2}\right)=\sum_{i=0}^{n_{1}}\binom{n_{1}}{i}\binom{n_{2}+i}{n_{1}} .
$$

Two protein sequences of length 100 amino acids can be aligned in approximately $10^{60}$ different ways

Time needed to test all possibilities is same order of magnitude as the entire lifetime of the universe.

## Pairwise alignment: the solution

"Dynamic programming"
(the Needleman-Wunsch algorithm)


## Alignment depicted as path in matrix



## Alignment depicted as path in matrix




Meaning of point in matrix: all residues up to this point have been aligned (but there are many different possible paths).

Position labeled " $x$ ": TC aligned with TC

```
--TC
TC--
```

$-T C$
T-C

## Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible previous positions (you cannot "align backwards").
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

## Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$
\operatorname{score}(x, y)=\max \left\{\begin{array}{l}
\operatorname{score}(x, y-1)-\text { gap-penalty } \\
\end{array}\right.
$$

## Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$
\operatorname{score}(x, y)=\max \left\{\begin{array}{l}
\operatorname{score}(x, y-1)-\text { gap-penalty } \\
\operatorname{score}(x-1, y-1)+\text { substitution-score }(x, y)
\end{array}\right.
$$

## Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$
\operatorname{score}(x, y)=\max \left\{\begin{array}{l}
\operatorname{score}(x, y-1)-\text { gap-penalty } \\
\operatorname{score}(x-1, y-1)+\text { substitution-score }(x, y) \\
\operatorname{score}(x-1, y)-\text { gap-penalty }
\end{array}\right.
$$

## Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Each new score is found by choosing the maximum of three possibilities.
For each square in matrix: keep track of where best score came from.
Fill in scores one row at a time, starting in upper left corner of matrix, ending in lower right corner.

$$
\operatorname{score}(x, y)=\max \left\{\begin{array}{l}
\operatorname{score}(x, y-1)-\text { gap-penalty } \\
\operatorname{score}(x-1, y-1)+\text { substitution-score }(x, y) \\
\operatorname{score}(x-1, y)-\text { gap-penalty }
\end{array}\right.
$$

## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Dynamic programming: example



## Global versus local alignments

Global alignment: align full length of both sequences.
(The "Needleman-Wunsch" algorithm).


Local alignment: find best partial alignment of two sequences (the "Smith-Waterman" algorithm).


## Local alignment overview

- The recursive formula is changed by adding a fourth possibility: zero. This means local alignment scores are never negative.

- Trace-back is started at the highest value rather than in lower right corner
- Trace-back is stopped as soon as a zero is encountered


## Local alignment: example

|  |  | H |  |  | E | A |  | G |  | A | W | W |  | G |  | H | E | E |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 0 | 0 |  |  | 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 | 0 | 0 |
| P | 0 | 0 |  |  | 0 | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 | 0 | 0 |
| A | 0 | 0 |  |  | 0 | 5 |  | 0 |  | 5 |  | 0 |  | 0 |  | 0 | 0 | 0 |
| W | 0 | 0 |  |  | 0 | 0 | * | 2 |  | 0 |  | 20 |  | 12 |  | 4 | 0 | 0 |
| H | 0 |  | 1 |  | 2 | 0 |  | 0 |  | 0 |  | 4 12 4 |  | $18$ |  | 22 | $14 \leftarrow$ | 6 |
| E | 0 | 2 |  |  | $16$ | 8 |  | 0 |  | 0 |  | 4 |  | 10 |  | $18$ | $28$ | 20 |
| A E | 0 0 | 0 | K |  | 8 6 |  | \% | 13 18 |  |  |  | 0 4 |  |  |  |  | ${ }_{16}{ }^{2}$ | 27 26 |
| AWGHE <br> AW-HE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Alignments: things to keep in mind

"Optimal alignment" means "having the highest possible score, given substitution matrix and set of gap penalties".

This is NOT necessarily the biologically most meaningful alignment.

Specifically, the underlying assumptions are often wrong: substitutions are not equally frequent at all positions, affine gap penalties do not model insertion/deletion well, etc.

Pairwise alignment programs always produce an alignment even when it does not make sense to align sequences.

