Lecture 2: Information Retrieval. *Entrez*

What is Entrez?

- An integrated Information Retrieval System
- A system of 31 linked databases
- A text search engine
- A tool for finding biologically linked data
- A retrieval engine
- A virtual workspace for manipulating large datasets
- NOT a database!

### Hard links

Cross-references between elements in different databases.
- CDS -> Protein -> Structure
- Gene -> SNP -> Disease

### Neighboring

Relationship between elements of the same database

- **Weighted key terms**: Relevance pairs model of retrieval. Based on word proximity and frequency in database.
- **BLAST**: DNA and Protein sequence similarities.
- **VAST**: Vector Alignment Search Tool. Protein structure similarities.
**The Entrez System: Text Searches**

### Boolean operators

- **AND, OR, NOT** *(Capital letters)*
  - Can use truncation *
  - Can use (): a **AND** (b **OR** c)

### General syntax

- **search term** [tag] **Boolean operator** **search term** [tag] ...
  - [**ORGN**] taxa
  - [**AU**] Author
  - [**TITL**] title
  - [**LA**] language
  - [**PROT**] protein
  - [**JOUR**] journal
  - [**EDAT**] date
  - [**GENE**] gene
Searching Entrez  


**Search** individual databases or across databases

**Limit searches**

- **Saving Searches** *My NCBI* (ex Cubby)
  - Register in *My NCBI*
  - Click the Save search link
  - Ask for Email updates
  - Retrieve previous searches
  - Combine searches from *History*
    - #1 AND #2
VAST: Structure Neighbors

**Vector Alignment Search Tool**


For each 3D domain,

locate SSEs (secondary structure elements),

and represent them as individual vectors.

**VAST uses 3D Domains only! Whole polypeptides are assigned 3D domain 0 (zero).**

Human IL-4
Sequence similarities

- Sequence alignments search for matches between sequences

- Two broad classes of sequence alignments
  - **Global**: entire length. For highly similar seq. of $\approx$ length
  - **Local**: find the most similar regions

- Alignment can be performed between two or more sequences
  - **Pairwise** (Chp. 11)
  - **Multiple Seq. Alignment** (Chp. 12)

Sequence alignment is the cornerstone of bioinformatics
The biological importance of sequence alignment

- Sequence alignments assess the degree of similarity between sequences
- Most common measurement: percent identity
- Similar sequences suggest:
  - **Similar structure:** Proteins with similar sequences are likely to have similar structures
  - **Similar function:** Proteins with similar structures are likely to have similar functions and play similar biochemical roles. Conserved amino acids likely have an important function
  - **Common evolutionary history**
    - Fewer differences mean more recent divergence
    - Similarity ≠ **homology** (applies to 2 types of relationships)
      - **Orthologous:** direct descendant from a common ancestor by speciation (similarity and colinearity).
      - **Paralogous:** Separated by an event of gene duplication
Pairwise alignment: protein sequences can be more informative than DNA

- protein is more informative (20 vs 4 characters); many amino acids share related biophysical properties

- codons are degenerate: changes in the third position often do not alter the amino acid that is specified

- protein sequences offer a longer “look-back” time

- DNA sequences can be translated into protein, and then used in pairwise alignments
<table>
<thead>
<tr>
<th>First letter</th>
<th>Second letter</th>
<th>Third letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>UUU</td>
<td>Phe</td>
</tr>
<tr>
<td></td>
<td>UUC</td>
<td>Ser</td>
</tr>
<tr>
<td></td>
<td>UUA</td>
<td>Leu</td>
</tr>
<tr>
<td></td>
<td>UUG</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CUU</td>
<td>Leu</td>
</tr>
<tr>
<td></td>
<td>CUC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CUA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CUG</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUG</td>
<td>Met</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td>GUU</td>
<td>Val</td>
</tr>
<tr>
<td></td>
<td>GUC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GUA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GUG</td>
<td></td>
</tr>
</tbody>
</table>
Pairwise alignment: protein sequences can be more informative than DNA

- Many times, DNA alignments are appropriate
  -- to confirm the identity of a cDNA
  -- to study noncoding regions of DNA
  -- to study DNA polymorphisms
  -- example: Neanderthal vs modern human DNA
**Definition: pairwise alignment**

**Pairwise alignment**
The process of lining up two sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology.
Homology
Similarity attributed to descent from a common ancestor.

Two types:

Orthologs
Homologous sequences in different species that arose from a common ancestral gene during speciation; may or may not be responsible for a similar function.

Paralogs
Homologous sequences within a single species that arose by gene duplication.
Beta globin (NP_000509) 2HHB

myoglobin (NP_005359) 2MM1
Orthologs: members of a gene (protein) family in various organisms. This tree shows globin orthologs.
Paralogs: members of a gene (protein) family within a species. This tree shows human globin paralogs.
Orthologs and paralogs are often viewed in a single tree

Source: NCBI
The algorithmic problem of aligning sequences

- Comparison of similar sequences of similar length is straightforward.
- How does one deal with insertions and gaps that may hide true similarity?
- How does one interpret minimal similarity?
  - Are sequences actually related?
  - Is alignment by chance?
Methods of sequence alignment

- **Graphical methods: dot plots**
  - **Dotter**: [http://sonnhammer.sbc.su.se/Dotter.html](http://sonnhammer.sbc.su.se/Dotter.html)
  - **Dotlet**: [http://myhits.isb-sib.ch/cgi-bin/dotlet](http://myhits.isb-sib.ch/cgi-bin/dotlet)
  - **Dottup**: [http://emboss.bioinformatics.nl/cgi-bin/emboss/dottup](http://emboss.bioinformatics.nl/cgi-bin/emboss/dottup)

- **Dynamic-programming (DP) methods**
  - **Global**: Needleman-Wunsch algorithm (1970)
  - **Local**: Smith-Waterman algorithm (1981)

- **Word methods (heuristic)**
  - **BLAST**
  - **FASTA**
Dot matrix analysis

- A graphical method
- Shows all possible alignments
- Caveats
  - Some guesswork in picking parameters
    - Window size
    - Stringency
  - Not as rigorous or quantitative as other methods
Dot matrix analysis: a real example

Detecting repeats with dotter

Window size: 1
Stringency: 1

MITE2.txt vs MITE2.txt

Window size: 23
Stringency: 15

WIS.txt vs WIS.txt

Greyamp tool

Alignment tool
Devising a scoring system

- **Dotplots problem:** no statistical measure of the quality of the alignment

- **Measurement of sequence similarity:** implies a metric - a statement of quantitatively how similar the two sequences are

  Match = 1  
  Mismatch = -1

  Match = 1  
  Open Gap = -2
  Mismatch = -1  
  Extend gap = -1

<table>
<thead>
<tr>
<th>Sequence 1</th>
<th>Sequence 2</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATCGCTTTTACA</td>
<td>ATCGCGGTTTGCA</td>
<td>6-5 = 1</td>
</tr>
<tr>
<td>AGCGCATCGGA</td>
<td>AGCGC--TTCGGA</td>
<td>8-3-2-1 = 2</td>
</tr>
<tr>
<td>AGCGC-TT-CGGA</td>
<td>ATCGCGGTTTGCA</td>
<td>7-4-2-2 = -1</td>
</tr>
</tbody>
</table>

**Affine gap penalties:** a fix deduction G is made for introducing the gap and then an additional smaller deduction L is made that is proportional to the length (n) of the gap

\[
\text{Gap penalty} = G + (L \times n)
\]
Scoring system for proteins

- **Problem**: alignment of different types of amino acids have different effects

- **Scoring matrices**: Most protein scoring matrices based on
  - **Conservation**: some amino acid changes more drastic than others
  - **Frequency**: if changes by chance, they depend on frequency
  - **Evolution**: depending on set of initial proteins, adjusted to different evolutionary distances

- **Some popular scoring matrices**
  - PAM (Percent Accepted Mutations),
  - BLOSUM (BLOcks amino acid SUbstitution Matrix)
Emile Zuckerkandl and Linus Pauling (1965) considered substitution frequencies in 18 globins (myoglobins and hemoglobins from human to lamprey).

<table>
<thead>
<tr>
<th>Substituent residue</th>
<th>(Percentage of total residue sites at which the substituent occurs)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>R</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lys found at 58% of arg sites.

Black: identity
Gray: very conservative substitutions (>40% occurrence)
White: fairly conservative substitutions (>21% occurrence)
Red: no substitutions observed
PAM Matrices (Percent Accepted mutations)

• Development (Margaret Dayhoff et al. 1978)
  - Based on observed replacement frequencies of aa for one another in a set of 71 groups of proteins >85% similar, (1572 aa changes)
  - Short evolutionary distance, so mutations likely not change protein function
  - PAM1 matrix: the overall probability of each aa changing to another is 1%
  - Based on the previous assumption it is possible to multiply a PAM matrix by itself to obtain higher order PAM matrices. PAM250 multiplied 250 times

• Limitations
  - Based on limited set of proteins available in 1978
  - PAM matrices over different evolutionary distances are extrapolations
  - Assume similar evolutionary forces over short and long evolutionary time
  - No consideration of block motifs
Dayhoff’s approach to assigning scores for any two aligned amino acid residues

Dayhoff et al. defined the score of two aligned residues \(i,j\) as 10 times the log of how likely it is to observe these two residues (based on the empirical observation of how often they are aligned in nature) divided by the background probability of finding these amino acids by chance. This provides a score for each pair of residues.

\[ s_{i,j} = 10 \times \log \left( \frac{q_{i,j}}{p_i} \right) \]
Why do we go from a mutation probability matrix to a log odds matrix?

• We want a scoring matrix so that when we do a pairwise alignment (or a BLAST search) we know what score to assign to two aligned amino acid residues.

• Logarithms are easier to use for a scoring system. They allow us to sum the scores of aligned residues (rather than having to multiply them).
How do we go from a mutation probability matrix to a log odds matrix?

- The cells in a log odds matrix consist of an “odds ratio”:
  
  the probability that an alignment is authentic
  the probability that the alignment was random

The score $S$ for an alignment of residues $a,b$ is given by:

$$S(a,b) = 10 \log_{10} \left( \frac{M_{ab}}{p_b} \right)$$

As an example, for tryptophan,

$$S(a,\text{tryptophan}) = 10 \log_{10} \left( \frac{0.55}{0.010} \right) = 17.4$$
# Normalized frequencies of amino acids

|   | A  | R  | N  | D  | C  | Q  | E  | G  | H  | I  | L  | K  | M  | F  | P  | S  | T  | W  | Y  | V  |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | 13 | 6  | 9  | 9  | 5  | 8  | 9  | 12 | 6  | 8  | 6  | 7  | 7  | 4  | 11 | 11 | 11 | 2  | 4  | 9  |
| R | 3  | 17 | 4  | 3  | 2  | 5  | 3  | 2  | 6  | 3  | 2  | 9  | 4  | 1  | 4  | 4  | 4  | 3  | 7  | 2  | 2  |
| N | 4  | 4  | 6  | 7  | 2  | 5  | 6  | 4  | 6  | 3  | 2  | 5  | 3  | 2  | 4  | 5  | 4  | 2  | 3  | 3  |
| D | 5  | 4  | 8  | 11 | 1  | 7  | 10 | 5  | 6  | 3  | 2  | 5  | 3  | 1  | 4  | 5  | 5  | 1  | 2  | 3  |
| C | 2  | 1  | 1  | 1  | 52 | 1  | 1  | 2  | 2  | 2  | 1  | 1  | 1  | 1  | 2  | 3  | 2  | 1  | 4  | 2  |
| Q | 3  | 5  | 5  | 6  | 1  | 10 | 7  | 3  | 7  | 2  | 3  | 5  | 3  | 1  | 4  | 3  | 3  | 1  | 2  | 3  |
| E | 5  | 4  | 7  | 11 | 1  | 9  | 12 | 5  | 6  | 3  | 2  | 5  | 3  | 1  | 4  | 5  | 5  | 1  | 2  | 3  |
| G | 12 | 5  | 10 | 10 | 4  | 7  | 9  | 27 | 5  | 5  | 4  | 6  | 5  | 3  | 8  | 11 | 9  | 2  | 3  | 7  |
| H | 2  | 5  | 5  | 4  | 2  | 7  | 4  | 2  | 15 | 2  | 2  | 3  | 2  | 2  | 3  | 3  | 2  | 2  | 3  | 2  |
| I | 3  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 10 | 6  | 2  | 6  | 5  | 2  | 3  | 4  | 1  | 3  | 9  |
| L | 6  | 4  | 4  | 3  | 2  | 6  | 4  | 3  | 5  | 15 | 34 | 4  | 20 | 13 | 5  | 4  | 6  | 6  | 7  | 13 |
| K | 6  | 18 | 10 | 8  | 2  | 10 | 8  | 5  | 8  | 5  | 4  | 24 | 9  | 2  | 6  | 8  | 8  | 4  | 3  | 5  |
| M | 1  | 1  | 1  | 0  | 1  | 1  | 1  | 1  | 2  | 3  | 2  | 6  | 2  | 1  | 1  | 1  | 1  | 1  | 2  |
| F | 2  | 1  | 2  | 1  | 1  | 1  | 1  | 1  | 3  | 5  | 6  | 1  | 4  | 32 | 1  | 2  | 2  | 4  | 20 | 3  |
| P | 7  | 5  | 5  | 4  | 3  | 5  | 4  | 5  | 5  | 3  | 3  | 4  | 3  | 2  | 20 | 6  | 5  | 1  | 2  | 4  |
| S | 9  | 6  | 8  | 7  | 7  | 6  | 7  | 9  | 6  | 5  | 4  | 7  | 5  | 3  | 9  | 10 | 9  | 4  | 4  | 6  |
| T | 8  | 5  | 6  | 6  | 4  | 5  | 5  | 6  | 4  | 6  | 4  | 6  | 5  | 3  | 6  | 8  | 15 | 2  | 2  | 6  |
| W | 0  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 1  | 55 | 1  |
| Y | 1  | 1  | 2  | 1  | 3  | 1  | 1  | 1  | 3  | 2  | 2  | 1  | 2  | 15 | 1  | 2  | 2  | 3  | 1  | 2  |
| V | 7  | 4  | 4  | 4  | 4  | 4  | 4  | 5  | 4  | 15 | 10 | 4  | 10 | 5  | 5  | 5  | 7  | 2  | 4  | 17 |

**Normalized frequencies of amino acids:**

- Arg: 4.1%
- Asn: 4.0%
- Phe: 4.0%
- Gln: 3.8%
- Ile: 3.7%
- His: 3.4%
- Cys: 3.3%
- Tyr: 3.0%
- Met: 1.5%
- Trp: 1.0%
What do the numbers mean in a log odds matrix?

\[ S(a, \text{tryptophan}) = 10 \log_{10} \left( \frac{0.55}{0.010} \right) = 17.4 \]

A score of +17 for tryptophan means that this alignment is 50 times more likely than a chance alignment of two Trp residues.

\[ S(a,b) = 17 \]
\[ \text{Probability of replacement } \left( \frac{M_{ab}}{p_b} \right) = x \]
Then
\[ 17 = 10 \log_{10} x \]
\[ 1.7 = \log_{10} x \]
\[ 10^{1.7} = x = 50 \]
What do the numbers mean in a log odds matrix?

A score of +2 indicates that the amino acid replacement occurs 1.6 times as frequently as expected by chance.

A score of 0 is neutral.

A score of –10 indicates that the correspondence of two amino acids in an alignment that accurately represents homology (evolutionary descent) is one tenth as frequent as the chance alignment of these amino acids.
Rat versus mouse globin

Rat versus bacterial globin
A substitution matrix contains values proportional to the probability that amino acid $i$ mutates into amino acid $j$ for all pairs of amino acids.

Substitution matrices are constructed by assembling a large and diverse sample of verified pairwise alignments (or multiple sequence alignments) of amino acids.

Substitution matrices should reflect the true probabilities of mutations occurring through a period of evolution.

The two major types of substitution matrices are PAM and BLOSUM.
PAM matrices: Point-accepted mutations

PAM matrices are based on global alignments of closely related proteins.

The PAM1 is the matrix calculated from comparisons of sequences with no more than 1% divergence. At an evolutionary interval of PAM1, one change has occurred over a length of 100 amino acids.

Other PAM matrices are extrapolated from PAM1. For PAM250, 250 changes have occurred for two proteins over a length of 100 amino acids.

All the PAM data come from closely related proteins (>85% amino acid identity).
PAM Scoring Matrices or Mutation Data Matrix (MDM)

**Side chains:**
- **C:** Sulfhydryl (Cys)
- **STPAG:** small hydrophilic
- **NDEQ:** acid, acid amide, and hydrophilic
- **HRK:** basic (His, Arg, Lys)
- **MILV:** small hydrophobic
- **FYW:** aromatic (F=Phe, Y=Tyr, W=Trp)

The PAM30 substitution matrix. The numbers indicate the substitution scores for each replacement. The greater the number, the lesser the penalty for the given substitution. Note the high penalty for replacing Cys and aromatic amino acids (Phe, Tyr, and Trp) with any other residues and, accordingly, the high reward for conservation of these residues (see the diagonal elements).
BLOSUM matrices

- Created by Henikoff & Henikoff (1992) based on local multiple alignments of more distantly related sequences (based on 2000 conserved motifs called blocks, 500 groups of pr.).
- Multiple alignments of short regions (without gaps) of related sequences were gathered.
- In each alignment, sequences equal to threshold value of percent identity were clustered into groups and averaged. Calculations across ≠ evolutionary distances, no extrapolation.
- Substitution frequencies for all pairs of amino acids were calculated between the groups, this was used to create the log-odds BLOSUM (Block Substitution Matrix).
- Thus, BLOSUM62 means that the sequences clustered in this block are no more than 62% identical. Perform better than PAM matrices.
- This allows detection of more distantly related sequences, as it downplays the role of the more related sequences in the block when building the matrix.

Score = \( \log \left( \frac{q_{ij}}{p_i p_j} \right) \)

- \( q_{ij} \) = How often aa i and j align
- \( p_i \) = pb. of aa i
- \( p_j \) = pb. of aa j
An example of scoring

A sequence comparison

<table>
<thead>
<tr>
<th>A</th>
<th>D</th>
<th>D</th>
<th>R</th>
<th>Q</th>
<th>C</th>
<th>E</th>
<th>R</th>
<th>A</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Q</td>
<td>E</td>
<td>R</td>
<td>Q</td>
<td>E</td>
<td>C</td>
<td>Q</td>
<td>A</td>
<td>Q</td>
</tr>
</tbody>
</table>

4 0 2 5 5 -4 -4 1 4 0

Total score: 13

Comparison PAM vs. BLOSUM

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Best use</th>
<th>Similarity %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAM140</td>
<td>Short alignment, high similarity</td>
<td>70-90</td>
</tr>
<tr>
<td><strong>PAM160</strong></td>
<td>Members of a protein family</td>
<td>50-60</td>
</tr>
<tr>
<td>PAM250</td>
<td>Longer alignments, more divergent sequence</td>
<td>≈30</td>
</tr>
<tr>
<td>BLOSUM90</td>
<td>Short alignment, high similarity</td>
<td>70-90</td>
</tr>
<tr>
<td>BLOSUM80</td>
<td>Members of a protein family</td>
<td>50-60</td>
</tr>
<tr>
<td><strong>BLOSUM62</strong></td>
<td>Most effective in finding all potential similarities</td>
<td>30-40</td>
</tr>
<tr>
<td>BLOSUM30</td>
<td>Longer alignments, more divergent sequence</td>
<td>&lt;30</td>
</tr>
</tbody>
</table>
Calculation of an alignment score

\[ S = \sum \text{(identities, mismatches)} - \sum \text{(gap penalties)} \]

\[ \text{Score} = \text{Max}(S) \]

Find BLAST from the home page of NCBI and select protein BLAST…

BLAST Assembled Genomes

Choose a species genome to search, or list all genomic BLAST databases.

- Human
- Mouse
- Rat
- Arabidopsis thaliana
- Oryza sativa
- Bos taurus
- Danio rerio
- Drosophila melanogaster
- Gallus gallus
- Pan troglodytes
- Microbes
- Apis mellifera

Basic BLAST

Choose a BLAST program to run.

- nucleotide blast: Search a nucleotide database using a nucleotide query
  Algorithms: blastn, megablast, discontinuous megablast
- protein blast: Search protein database using a protein query
  Algorithms: blastp, psi-blast, phi-blast
- blastx: Search protein database using a translated nucleotide query
- tblastn: Search translated nucleotide database using a protein query
- tblastx: Search translated nucleotide database using a translated nucleotide query
Choose align two or more sequences...
Enter the two sequences (as accession numbers or in the fasta format) and click BLAST.

 Optionally select “Algorithm parameters” and note the matrix option.
Pairwise alignment result of human beta globin and myoglobin

Myoglobin RefSeq

Information about this alignment: score, expect value, identities, positives, gaps...

Query = HBB
Subject = MB

Middle row displays identities; + sign for similar matches
Pairwise alignment result of human beta globin and myoglobin: the score is a sum of match, mismatch, gap creation, and gap extension scores

Score = 18.1 bits (35),  Expect = 0.015, Method: Composition-based stats.  
Identities = 11/24 (45%),  Positives = 12/24 (50%),  Gaps = 2/24 (8%)

<table>
<thead>
<tr>
<th></th>
<th>Query 12</th>
<th>Sbjct 11</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VTALWGKVNVND--EVGGEALGRLL</td>
<td>V +WGKV D G E L RL</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>VLNVWGKVEADIPGHGQEVLIRLF</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>match</td>
<td>4 11 5 6 4 6 5 4 4 5</td>
<td>sum of matches: +60</td>
<td></td>
</tr>
<tr>
<td>mismatch</td>
<td>-1 1 0 -2 -2 -4 0</td>
<td>sum of mismatches: -13</td>
<td></td>
</tr>
<tr>
<td>gap open</td>
<td>-11 0 0</td>
<td>sum of gap penalties: -12</td>
<td></td>
</tr>
<tr>
<td>gap extend</td>
<td>-1 -1 -3 0</td>
<td>total raw score: 60 - 13 - 12 = 35</td>
<td></td>
</tr>
</tbody>
</table>
Pairwise alignment result of human beta globin and myoglobin: the score is a sum of match, mismatch, gap creation, and gap extension scores

Score = 18.1 bits (35),  Expect = 0.015, Method: Composition-based stats.  
Identities = 11/24 (45%), Positives = 12/24 (50%), Gaps = 2/24 (8%)

<table>
<thead>
<tr>
<th></th>
<th>Query 12</th>
<th>V-TAL-WGK-VNVD--E-VGGEALGRLL</th>
<th>33</th>
<th>V</th>
<th>+WGKV</th>
<th>D</th>
<th>G</th>
<th>E</th>
<th>L</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sbjct 11</td>
<td>VLNVGKVEADIPGHGQEVLIRLF</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**match**

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>11</th>
<th>5</th>
<th>6</th>
<th>4</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum of matches:</td>
<td>+60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**mismatch**

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>1</th>
<th>0</th>
<th>-2</th>
<th>-2</th>
<th>-4</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum of mismatches:</td>
<td>-13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**gap open**

<table>
<thead>
<tr>
<th></th>
<th>-11</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum of gap penalties:</td>
<td>-12</td>
</tr>
</tbody>
</table>

**gap extend**

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
</tr>
</thead>
</table>

Total raw score: 60 - 13 - 12 = 35

V matching V earns +4  
T matching L earns -1

These scores come from a “scoring matrix”
Mind the gaps

Score = 18.1 bits (35), Expect = 0.015, Method: Composition-based stats.
Identities = 11/24 (45%), Positives = 12/24 (50%), Gaps = 2/24 (8%)

<table>
<thead>
<tr>
<th>Query 12</th>
<th>VTALWGBKVNVD--EVGGGEALGRLL 33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sbjct 11</td>
<td>VLNWGBKXEADIPGHGQEVLRFL 34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>match</th>
<th>mismatch</th>
<th>gap open</th>
<th>gap extend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 11 5</td>
<td>-1 1 -2</td>
<td>-11</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>6 4</td>
<td>0 0 -3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 5 4 5</td>
<td>-2 -2 -4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

sum of matches: +60  
sum of mismatches: -13  
sum of gap penalties: -12  

total raw score: 60 - 13 - 12 = 35

First gap position scores -11  
Second gap position scores -1  

gap creation tends to have a large negative score;  
gap extension involves a small penalty
Gaps

- Positions at which a letter is paired with a null are called gaps.

- Gap scores are typically negative.

- Since a single mutational event may cause the insertion or deletion of more than one residue, the presence of a gap is ascribed more significance than the length of the gap. Thus there are separate penalties for: gap creation and gap extension.

- In BLAST, it is rarely necessary to change gap values from the default.
Pairwise alignment of retinol-binding protein and β-lactoglobulin: Example of an alignment with internal, terminal gaps

1 MKWVWALLLLAAWAAAERDCRVSFRVKFNFDKARFSGTWYAMAKKDPEG 50 RBP  
   . ||| | . . . | :: ||||:| :  
1 ...MKCLLLALALTCGAQALIVT..QTMEGLD1QKVAGTWYSLAMAASD. 44 lactoglobulin

51 LFLQDNIVAEFSVDETGQMSATAKGRVR.LLNNWD..VCADMVGTFTDTE 97 RBP  
: | | | | : | | | | | | | | : | | | |  
45 ISLLDAQSAPLRV.YVEELKPTPEGDLEILLQKWENGECQAQQKIIAEKTK 93 lactoglobulin

98 DPAKFKMKYWGASFLQKGNDDHWIVDTHYTYAV..........QYSC 136 RBP  
|| || . | ::||| | .  
94 IPAVFKIDALNENKVLD....VLDTDEYYYYLLFCMENSAEPEQSLAC 135 lactoglobulin

137 RLLNLGTCADSYSFVFSRDPNGLPPEAQKIVRQRQ.EELCLRQYRLIV 185 RBP  
. | | | | : | | • | |  
136 QCLVRTPEVDDEALEKFDKALPMHIRLSFNPTQLEEQCHI...... 178 lactoglobulin
Pairwise alignment of retinol-binding protein from human (top) and rainbow trout (*O. mykiss*): Example of an alignment with few gaps

```
1 MKWVWALLLLA.AWAAAERDCRVSSFRVKENFDKARFSGTWYAMAKKDP 48
  ::   ||   ||   ||   .||.|| . | :|||:.|:.| ||| .|||||
1 MLRICVALCALATCWAA...QDCQVSNIQVMQNFDRSRYTGRWYAVAKKDP 47

49 EGLFLQDNIVAEFSVDETGQMSATAKGRVRLNNWDVCADMVGTFTDTED 98
  |||| ||::||:||| || | || | ::|||:.|| | || |||||
48 VGLFLLDNVVAQFSVDESGKMTATAHGRVIILNNWEMCANDMFGTFEDTPD 97

99 PAKFKMKYWGVASFLQKGNDDHWIVDTDYAVQYSCRLLNLDGTCADS 148
  |||| ||::|| | ::|| | ||||||:: || || |
98 PAKFKMRWGAASYLQTGNDDHWVIDTDYDNAYIAHYSCREVDLDGTCLDG 147

149 YSFVFSRDPNLPPPEAQKIVRQRQEEELCLARQYRLIVHNGYCDGRSERNLL 199
  ||::|| | | | | || || ::||:|: |::|:
148 YSFIFSRHPTGLRPEDQKIVTDKKKEICFLGKYRRVGVHTGFCESS...... 192
```
Pairwise sequence alignment allows us to look back billions of years ago.

When you do a pairwise alignment of homologous human and plant proteins, you are studying sequences that last shared a common ancestor 1.5 billion years ago!
Multiple sequence alignment of glyceraldehyde 3-phosphate dehydrogenases: example of extremely high conservation

<table>
<thead>
<tr>
<th></th>
<th>fly</th>
<th>human</th>
<th>plant</th>
<th>bacterium</th>
<th>yeast</th>
<th>archaeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAKKVIISAP</td>
<td>SAD.APM...F</td>
<td>VCGVNLDAKY</td>
<td>PDMKVVSNAS</td>
<td>CTTNCLAPLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAKRVIISAP</td>
<td>SAD.APM...F</td>
<td>VMGVNHEKYD</td>
<td>NSLKIISNAS</td>
<td>CTTNCLAPLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAKKVIISAP</td>
<td>SAD.APM...F</td>
<td>VVGVNEHTYQ</td>
<td>PNMDIVSNAS</td>
<td>CTTNCLAPLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAKKVVMGP</td>
<td>SKDNTPM...F</td>
<td>VKGANGFDKY.</td>
<td>AGQDIVSNAS</td>
<td>CTTNCLAPLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAKKVVIAP</td>
<td>SS.TAPM...F</td>
<td>VMGVNEEKYT</td>
<td>SDLKIVSNAS</td>
<td>CTTNCLAPLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GADKVLISAP</td>
<td>PKGDEPVKQL</td>
<td>VYGVNHDEYD</td>
<td>GE.DVVSNAS</td>
<td>CTTNSITPVA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAKAVGKVI</td>
<td>PALNGKLTGM</td>
<td>AFRVPTPNVS</td>
<td>VVDLTVRLGK</td>
<td>GASYEIKAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAKAVGKVI</td>
<td>PELNGKLTGM</td>
<td>AFRVPTANVS</td>
<td>VVDLTCRLEK</td>
<td>PAKYDIKKV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAKAVGKVL</td>
<td>PELNGKLTGM</td>
<td>AFRVPTSNVS</td>
<td>VVDLTCRLEK</td>
<td>GASYEDVKA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAKAVGKVL</td>
<td>PELNGKLTGM</td>
<td>AFRVPTPNVS</td>
<td>VVDLTVRLKE</td>
<td>AATYEQIKAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAKAVGKVL</td>
<td>PELQGKLTGM</td>
<td>AFRVPTVDFS</td>
<td>VVDLTVKLKN</td>
<td>ETTYDEIKKV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAAQAATEVL</td>
<td>PELEGKLDGM</td>
<td>AIRVPVPNGS</td>
<td>ITEFVVDLDD</td>
<td>DVTESDVNA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multiple sequence alignment of human lipocalin paralogs: example of extremely low conservation

lipocalin 1
odorant-binding protein 2a
progestagen-assoc. endo.
apolipoprotein D
retinol-binding protein
neutrophil gelatinase-ass.
prostaglandin D2 synthase
alpha-1-microglobulin
complement component 8
Dynamic programming

• Given a scoring matrix: What is the best alignment?

• Cannot compare all: possibility of gaps (or insertions) makes number of possible sequence alignments astronomical (Total No of alignments > n^n/n!)

• Dynamic programming makes sequence alignment possible by abandoning low scoring alignments among subsequences as the algorithm progresses

• Mathematically proven to provide optimal alignments

• DP algorithms for sequence alignment
  - **Global alignments**: Needleman-Wunsch algorithm
  - **Local alignments**: Smith-Waterman algorithm

• DP alignment algorithms still too slow for searching an entire sequence database
Dynamic programming (Needleman-Wunsch)

Represent an alignment as a path between cells of a matrix

1. Compile all pair-wise scores from scoring matrix

2. Select highest score in the last column row

3. Highest score in next col.-row is the highest values we can get from adding a value from previous col.-row, from a cell that could be part of an alignment path

4. Highest score in next col.-row is the highest values we can get from adding a value from previous col.-row, from a cell that could be part of an alignment path

5. The highest possible score is found after the matrix is filled. Then backtrack, and determine which cells contributed to the highest possible score

http://www.ebi.ac.uk/Tools/emboss/align/index.html
Global alignment with the algorithm of Needleman and Wunsch (1970)

• Two sequences can be compared in a matrix along x- and y-axes.

• If they are identical, a path along a diagonal can be drawn

• Find the optimal subpaths, and add them up to achieve the best score. This involves
  --adding gaps when needed
  --allowing for conservative substitutions
  --choosing a scoring system (simple or complicated)

• N-W is guaranteed to find optimal alignment(s)
Three steps to global alignment with the Needleman-Wunsch algorithm

[1] set up a matrix
[2] score the matrix
[3] identify the optimal alignment(s)
Four possible outcomes in aligning two sequences

[1] identity (stay along a diagonal)
[2] mismatch (stay along a diagonal)
[3] gap in one sequence (move vertically)
[4] gap in the other sequence (move horizontally)
Start Needleman-Wunsch with an identity matrix

Sequence 2
(from honeybee globin)

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Start Needleman-Wunsch with an identity matrix

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>6</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-4</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>K</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>H</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
<td>5</td>
<td>-3</td>
<td>-1</td>
<td>-2</td>
<td>2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>E</td>
<td>-3</td>
<td>-2</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>-3</td>
<td>-3</td>
<td>6</td>
<td>-1</td>
<td>-1</td>
<td>-4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>P</td>
<td>-4</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>7</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>2</td>
<td>-4</td>
<td>-1</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>-3</td>
<td>-2</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
Fill in the matrix using “dynamic programming”
Fill in the matrix using “dynamic programming”

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
<td>-12</td>
<td>-14</td>
<td>-16</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fill in the matrix using “dynamic programming”

(b)

Score = Max \[\begin{align*}
& F(i - 1, j - 1) + s(x_i, y_j) \\
& F(i - 1, j) - \text{gap penalty} \\
& F(i, j - 1) - \text{gap penalty}
\end{align*}\]

Score (this example) = +1 (match)  
-2 (mismatch)  
-2 (gap penalty)
Fill in the matrix using “dynamic programming”
Fill in the matrix using “dynamic programming”
Fill in the matrix using “dynamic programming”

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
<td>-12</td>
<td>-14</td>
</tr>
<tr>
<td>K</td>
<td>-4</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>7</td>
<td>-9</td>
<td>11</td>
</tr>
<tr>
<td>H</td>
<td>-6</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-5</td>
<td>7</td>
<td>-9</td>
<td>11</td>
</tr>
<tr>
<td>M</td>
<td>-8</td>
<td>-5</td>
<td>-2</td>
<td>-4</td>
<td>-5</td>
<td>7</td>
<td>-9</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>-10</td>
<td>-7</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
<td>-7</td>
<td>-9</td>
<td>11</td>
</tr>
<tr>
<td>D</td>
<td>-12</td>
<td>-9</td>
<td>-6</td>
<td>-3</td>
<td>-5</td>
<td>7</td>
<td>-9</td>
<td>11</td>
</tr>
<tr>
<td>P</td>
<td>-14</td>
<td>-11</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>L</td>
<td>-16</td>
<td>-13</td>
<td>-10</td>
<td>-7</td>
<td>-7</td>
<td>-6</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>E</td>
<td>-18</td>
<td>-15</td>
<td>-12</td>
<td>-9</td>
<td>-9</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
</tr>
</tbody>
</table>
**Traceback to find the optimal (best) pairwise alignment**

(a) Sequence 2

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
<td>-12</td>
<td>-14</td>
</tr>
<tr>
<td>K</td>
<td>-4</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>H</td>
<td>-6</td>
<td>-3</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
<td>-13</td>
</tr>
<tr>
<td>M</td>
<td>-8</td>
<td>-5</td>
<td>-2</td>
<td>-4</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>E</td>
<td>-10</td>
<td>-7</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>D</td>
<td>-12</td>
<td>-9</td>
<td>-6</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>P</td>
<td>-14</td>
<td>-11</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
</tr>
<tr>
<td>L</td>
<td>-16</td>
<td>-13</td>
<td>-10</td>
<td>-7</td>
<td>-7</td>
<td>-6</td>
<td>-3</td>
<td>-5</td>
</tr>
<tr>
<td>E</td>
<td>-18</td>
<td>-15</td>
<td>-12</td>
<td>-9</td>
<td>-9</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
</tr>
</tbody>
</table>

(b) Sequence 2

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
<td>-12</td>
<td>-14</td>
</tr>
<tr>
<td>K</td>
<td>-4</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>H</td>
<td>-6</td>
<td>-3</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
<td>-13</td>
</tr>
<tr>
<td>M</td>
<td>-8</td>
<td>-5</td>
<td>-2</td>
<td>-4</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>E</td>
<td>-10</td>
<td>-7</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>D</td>
<td>-12</td>
<td>-9</td>
<td>-6</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
<td>-9</td>
<td>-11</td>
</tr>
<tr>
<td>P</td>
<td>-14</td>
<td>-11</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
</tr>
<tr>
<td>L</td>
<td>-16</td>
<td>-13</td>
<td>-10</td>
<td>-7</td>
<td>-7</td>
<td>-6</td>
<td>-3</td>
<td>-5</td>
</tr>
<tr>
<td>E</td>
<td>-18</td>
<td>-15</td>
<td>-12</td>
<td>-9</td>
<td>-9</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
</tr>
</tbody>
</table>

(c) Sequence 1

|   | +1 | -1 | -3 | -2 | -4 | -3 | -5 | -4 |

Sequence 1: F K H M E D P L E
Sequence 2: F - - M - D T P L N E
N-W is guaranteed to find optimal alignments, although the algorithm does not search all possible alignments.

It is an example of a dynamic programming algorithm: an optimal path (alignment) is identified by incrementally extending optimal subpaths. Thus, a series of decisions is made at each step of the alignment to find the pair of residues with the best score.
EMBOSS | Pairwise Alignment Algorithms

This tool is used to compare 2 sequences. When you want an alignment that covers the whole length of both sequences, use needle. When you are trying to find the best region of similarity between two sequences, use water.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gap Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMBOSS::needle (global)</td>
<td>10.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gap Extend</th>
<th>Molecule</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>Protein</td>
<td>Blosum62</td>
</tr>
</tbody>
</table>

Sequence 1: paste Sequence in any format OR upload a file:

```
MVHLTPEKSAVTALWGKVNVDEVGEALGRLLVYYPWTQRFFESFDLSAFSDGLAHLDNLKGTATLSELHDCKLHVDPEFNFRLLGNVLVCVLAAHFFGALAHKYH
```

Queries:
- beta globin (NP_000509)
- alpha globin (NP_000549)
# Program: needle
# Rundate: Tue Aug 22 16:29:58 2006
# Align_format: srspair
# Report_file: /ebi/extserv/old-work/needle-20060822-16295743003385.output

#=========================================================================

# Aligned_sequences: 2
# 1: EMBoss_001
# 2: EMBoss_001
# Matrix: EbloSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 149
# Identity: 65/149 (43.6%)
# Similarity: 90/149 (60.4%)
# Gaps: 9/149 (6.0%)
# Score: 292.5
#
#=========================================================================

EMBoss_001
1  MWHLTPEEKSAVTALWGKV--NVDENVGEALGRLLVVPWQWRFFFESFGD
   || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || || |
Global alignment versus local alignment

Global alignment (Needleman-Wunsch) extends from one end of each sequence to the other.

Local alignment finds optimally matching regions within two sequences (“subsequences”).

Local alignment is almost always used for database searches such as BLAST. It is useful to find domains (or limited regions of homology) within sequences.

Smith and Waterman (1981) solved the problem of performing optimal local sequence alignment. Other methods (BLAST, FASTA) are faster but less thorough.
How the Smith-Waterman algorithm works

Set up a matrix between two proteins (size m+1, n+1).

No values in the scoring matrix can be negative! S > 0.

The score in each cell is the maximum of four values:

1. \( s(i-1, j-1) + \) the new score at \([i,j]\) (a match or mismatch)
2. \( s(i, j-1) - \) gap penalty
3. \( s(i-1, j) - \) gap penalty
4. zero
Smith-Waterman algorithm allows the alignment of subsets of sequences

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>C</th>
<th>U</th>
<th>C</th>
<th>G</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td>1.3</td>
<td>0.3</td>
<td>1.0</td>
<td>0.3</td>
<td>2.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.0</td>
<td>1.0</td>
<td>3.0</td>
<td>1.7</td>
<td>1.3</td>
<td>1.0</td>
<td>1.3</td>
<td>1.7</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td>0.7</td>
<td>0.3</td>
<td>1.7</td>
<td>2.7</td>
<td>1.3</td>
<td>1.0</td>
<td>0.7</td>
<td>1.0</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.7</td>
<td>0.3</td>
<td>1.3</td>
<td>2.7</td>
<td>2.3</td>
<td>1.0</td>
<td>0.7</td>
<td>1.7</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>U</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>1.0</td>
<td>2.3</td>
<td>2.3</td>
<td>2.0</td>
<td>0.7</td>
<td>1.7</td>
<td>2.7</td>
<td>1.7</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.3</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>3.3</td>
<td>2.0</td>
<td>1.7</td>
<td>1.3</td>
<td>2.3</td>
</tr>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.7</td>
<td>0.7</td>
<td>2.0</td>
<td>3.0</td>
<td>1.7</td>
<td>1.3</td>
<td>2.3</td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>2.0</td>
<td>0.7</td>
<td>1.7</td>
<td>1.7</td>
<td>3.0</td>
<td>2.7</td>
<td>1.3</td>
<td>1.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>1.0</td>
<td>0.3</td>
<td>0.7</td>
<td>1.7</td>
<td>0.3</td>
<td>2.7</td>
<td>1.7</td>
<td>2.7</td>
<td>2.3</td>
<td>1.0</td>
</tr>
<tr>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.7</td>
<td>0.7</td>
<td>0.3</td>
<td>0.3</td>
<td>1.3</td>
<td>1.3</td>
<td>2.3</td>
<td>1.3</td>
<td>2.3</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Queries:
beta globin (NP_000509)
alpha globin (NP_000549)

Choose (A) program and (B, C) sequences to compare:

(A) Program: FASTA: protein:protein

(B) Enter first (query) sequence: FASTA format

(C) Enter second (library) sequence: FASTA format

Other options:
Ktup: ktup = 2
Scoring matrix: Blossum50
open: -10
ext: -2

Entrez protein sequence browser
Entrez DNA sequence browser
Query library @ vs /tmp/FAWW_4bMsB library searching /tmp/FAWW_4bMsB library

1>>>gi|4504349|ref|NP_000509.1| beta globin [Homo 147 aa - 147 aa
vs /tmp/FAWW_4bMsB library

142 residues in 1 sequences
Altschul/Gish params: nC: 147 Lambda: 0.158 K: 0.019 K: 0.100

FASTA (3.49 May 2006) function [optimized, BLOSUM matrix (15:-5)] ktup: 2
join: 36, opt: 24, open/ext: -10/-2, width: 16
Scan time: 0.000

The best scores are:

opt bits E(1) %_id %_sim alen
QUERY (142) 381 94.1 9.8e-25 0.434 0.745 145 align

147 aa vs /tmp/FAWW_4bMsB library

>>>QUERY
init: 325 init1: 273 opt: 381 2-score: 476.6 bits: 94.1 E(): 9.8e-25
Smith-Waterman score: 381: 43.448% identity (74.483% similar) in 145 aa overlap (4-146:3-141)

Entrez Lookup Re-search database General re-search
10 20 30 40 50 60 70

Entrez Lookup Re-search database General re-search
10 20 30 40 50 60 70

gi|450 MVHLTEEPSAHTALGKAYNVEIVGELSGKLLVYLYWDTIDFGDLSPTAVISSNPVKAHGGMKVGLGSDGLAH
::: :: :: :: :: :: :: :: :: :: :: :: :: :: :: !!!
QUERY MVLSPAKNTVKAAGWKGAYAGYAEALEMFLSPKATKTPHPF DLS---HGSACVQKNGKVIDALTRVANAH
10 20 30 40 50 60 70

80 90 100 110 120 130 140

gi|450 LDNLKGTFTALSELHCDEFLHVDPENFRRLLGNVLVCLAHJFHGRFPTPVPQAVYQKVAVGANALAHKYH
::: :: :: :: :: :: :: :: :: :: :: :: :: :: !!!
QUERY VDDKPMALSALDLHAIHLRDPVNFELLSHCLLVTOLLHLPDEFTPVAVHASLDDKFLASVSTVLTSKYF
80 90 100 110 120 130 140

147 residues in 1 query sequences
142 residues in 1 library sequences

Tcmlpib [34t26] (2 proc)
Total Scan time: 0.000 Total Display time: 0.000

Function used was FASTA [version 3.4t26 July 7, 2006]
Rapid, heuristic versions of Smith-Waterman: FASTA and BLAST

Smith-Waterman is very rigorous and it is guaranteed to find an optimal alignment.

But Smith-Waterman is slow. It requires computer space and time proportional to the product of the two sequences being aligned (or the product of a query against an entire database).

Gotoh (1982) and Myers and Miller (1988) improved the algorithms so both global and local alignment require less time and space.

FASTA and BLAST provide rapid alternatives to S-W.
Heuristic methods with $k$-tuples

- Examples: BLAST & FASTA
  
  - Faster than dynamic programming. Good for database searches.
  
  - Using query sequence, derive a list of words of length $w$ (e.g., 3 for protein, 11 for DNA)
  
  - High-scoring words are compared with database sequences or 2nd sequence
  
  - Sequences with many matches to high-scoring words are used for final alignments