3. SEQUENCE ANALYSIS

BIOINFORMATICS COURSE
MTAT.03.239

25.09.2012
SEQUENCE ANALYSIS IS IMPORTANT FOR ...

- Prediction of function
- Gene finding – the process of identifying the regions of genomic DNA that encode genes.
- Protein structure prediction
- Sequence assembly
- Database searching
HOW GOOGLE SEARCH WORKS

Have you ever wondered what happens when you type in a query in Google’s search field?

*Tham Yuen-C and Quek Hong Shin* go behind the scenes of the search engine

1. Google sends out Web crawlers, known as Googlebots, that go through all the Web pages on the World Wide Web.

2. These bots are made up of millions of computers that surf the Web much like how people do it, except they do it much more quickly. To make sure that they have the most current sites, these Googlebots are working all the time, crawling and re-crawling websites.

3. How often a site is crawled depends on how quickly its content changes. For example, a newspaper site that has frequent updates will get crawled more often than a static company site. Webmasters can also indicate how often they want their sites to be crawled.

4. The bots also detect all links on the site, which are then put into a queue system for crawling later on. These links are also important in deciding the search ranking of a site.

5. Copies of all these pages are stored in Google’s index database. The index works like the contents page of a book and contains all the possible search terms.

6. When a user types in a query in the search engine, the index will match the query with all the webpages in which the query terms appear and then grab the pages for the search results.

7. At the same time, a short description of the pages is generated for the search results page.

8. Web pages are ranked in terms of relevance before they are displayed on the results page. Google considers some 200 factors when ranking sites. One of them is PageRank, which takes into consideration how many sites are linked to a webpage and the quality of the linking sites.
BIOLOGICAL SEQUENCES

DNA

RNA
http://www.uic.edu/classes/phys/phys461/phys450/ANJUM04/

TELPQPQQTSGG
http://en.wikipedia.org/wiki/Protein
EVOLUTION OF SEQUENCES

- All living organisms are **related to each** other through evolution.
- This means: any pair of organism, no matter how different, have a common ancestor sometime in the past, from which they evolved.
- **Mutations** and **selection** over long periods of time can result in considerable difference between present-day sequences derived from the same ancestral sequences.
- The base pair composition of the sequences can change due to point mutation (**substitutions**), and the sequence lengths can vary due to indels (**insertions/deletions**).
MUTATIONS / SUBSTITUTIONS

Adenine (A) and Guanine (G) are examples of purines, while Cytosine (C) and Thymine (T) are examples of pyrimidines.

Transitions occur between purines (A to G or G to A) and Transversions occur between pyrimidines (C to T or T to C).

http://en.wikipedia.org/wiki/Transition_(genetics)
# POINT MUTATIONS

<table>
<thead>
<tr>
<th></th>
<th>Silent</th>
<th>Nonsense</th>
<th>Missense</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DNA level</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTC</td>
<td>TTT</td>
<td>ATC</td>
<td>TCC</td>
</tr>
<tr>
<td>mRNA level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AAG</td>
<td>AAA</td>
<td>UAG</td>
<td>AGG</td>
</tr>
<tr>
<td>protein level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lys</td>
<td>Lys</td>
<td>STOP</td>
<td>Arg</td>
</tr>
</tbody>
</table>

http://en.wikipedia.org/wiki/Point_mutation
DNA SEQUENCE EVOLUTION

GGCTA

Substitution
GCGTA

Deletion
GGTA
DNA SEQUENCE EVOLUTION

```
GGCTA
```

- **Substitution**
  - GCGTA
  - GTA
  - ?
  - ?

- **Deletion**
  - GGTA
  - CGTA
  - ?
  - ?

**GGCTA**

“Sequence analysis”
Bioinformatics Course
DNA SEQUENCE EVOLUTION

GGCTA

Substitution

Deletion

Deletion

GTA

GCTA

GGGA

Deletion

Insertion

CGTA

GGGTA

“Sequence analysis"
Bioinformatics Course
SEARCHING THE DATABASE

exact matches

```
Query: 1 MK   Query: 2 MKV   Query: 1 MKVR   Query: 1 MKVRA
    ->          ->          ->          
Sbjct: 1 MK   Sbjct: 1 MKV  Sbjct: 1 MKVR  Sbjct: 1 MKVRA

Query: 1 MKVRASVKKLKASGHCAHIJCALIKACIKOPRLACPOKACLKKLWWQ
.... ->

Sbjct: 1 MKVRASVKKLKASGHCAHIJCALIKACIKOPRLACPOKACLKKLWWQ
```

"Sequence analysis"
Bioinformatics Course
SEARCHING THE DATABASE

exact matches

Query: 1 MK  Query: 2 MKV  Query: 1 MKVR  Query: 1 MKVRA
    ->  ->  ->
Sbjct: 1 MK  Sbjct: 1 MKV  Sbjct: 1 MKVR  Sbjct: 1 MKVRA

Query: 1 MKVRASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ
.... ->
Sbjct: 1 MKVRASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ

Query1: 1 MKVRASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ
Query2: 1 KVRASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ
Query3: 1 VRASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ
Query4: 1 RASVKKLKASGHCAHIJCALKACIKOPRLACPOKACLKKLKWWQ
## SEARCHING THE DATABASE

**exact matches**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Occurrences in the database</th>
</tr>
</thead>
<tbody>
<tr>
<td>KV</td>
<td>488,559</td>
</tr>
<tr>
<td>KVR</td>
<td>28,592</td>
</tr>
<tr>
<td>KVRA</td>
<td>2,077</td>
</tr>
<tr>
<td>KVRAS</td>
<td>124</td>
</tr>
<tr>
<td>KVRASV</td>
<td>23</td>
</tr>
<tr>
<td>KVRASVK</td>
<td>8</td>
</tr>
<tr>
<td>KVRASVKK</td>
<td>4</td>
</tr>
<tr>
<td>KVRASVKKL</td>
<td>1</td>
</tr>
<tr>
<td>KVRASVKKLC</td>
<td>1</td>
</tr>
</tbody>
</table>

“Sequence analysis"  
Bioinformatics Course
SEARCHING THE DATABASE

- Searching databases is an important task in molecular biology.
- Searching databases for sequences tries to find similar sequences to the query sequence in the database.
- Such search amounts to aligning the query sequence to sequences in the database and returning results with “good” alignment score.
- Exact match approaches can be computationally very expensive which yields to usage of heuristic approaches.
SEQUENCE SIMILARITY

- **Homology**: deriving from a common ancestor-gene.
- **Orthologous**: homologous genes in different organisms.
- **Paralogous**: homologous genes in one organism that derive from **gene duplication**.
- **Gene duplication**: one gene is duplicated in multiple copies that can each evolve separately and assume new functions.
Alignment is the task of locating “equivalent” regions of two or more sequences to maximize their similarity.

Mismatches (correspond to mutations)

gaps (correspond to indels: insertions/deletions)
Alignment can reveal homology between sequences

- **Similarity** is a descriptive term that tells about the degree of match between the two sequences.
- Sequence similarity does not always imply a common function.
- Conserved function does not always imply similarity at the sequence level.
PRINCIPLES OF SEQUENCE ALIGNMENT

- It is easier to detect homology when comparing protein sequences then when comparing nucleic acid sequences.
- The probability of a “match by chance” is much higher in DNA sequences then in protein sequences.
- The genetic code is redundant: identical amino acids can be coded by different codons.
- The complex 3D structure of a protein, and hence its function, is determined by the amino acid sequence. Hence, conserving function leads to fewer changes in the amino acids than in the nucleotide sequence.
TYPES OF ALIGNMENT

➢ Pairwise Alignment

used to find best-matching piecewise local or global alignment of **two query sequences**.

➢ Multiple Alignment

an extension to pairwise alignment to incorporate **more than two sequences** at a time – it tries to align all of the sequences in a given query set.
GLOBAL VS LOCAL ALIGNMENT

- Global alignment tries to align the entire sequence, using as many characters as possible, up to both ends of each sequence.

- In local alignment, stretches of sequences with the highest density of matches are aligned, generating one or more subalignments in the aligned sequences.

```
LGPSSKQTGKGS---SRIWDN
\|  |  |  |  |  |  |
LN-ITKSAGKGAIMRLGDA
```

Global alignment

```
-------TGKG---------
  | | |
-------AGKG---------
```

Local alignment
PAIRWISE SEQUENCE ALIGNMENT

- **Dot matrix** analysis
- The **dynamic programming** (or DP) algorithms
  - Needleman-Wunch (1970) – global alignment
  - Smith-Waterman (1981) – local alignment
- **Word or** \(k\)-**tuple methods**
  - FASTA (Wilbur and Lipman, 1983)
  - BLAST
**DOT PLOT**

Red rectangles are true matching of identical residue-pairs and green rectangles represent noise.
DOT PLOT

- First described by Gibbs and McIntyre (1970)
- Sequence “A” is listed from **left to right**
- Sequence “B” is listed from **up to down**
- Starting from the first character of “B”, one moves across all the characters in “A” and places a dot whenever the character in “A” is the same as the character in “B”
- The process is continued until all characters from both sequences are compared against each other
- Similar regions are revealed by diagonal rows of dots
- Isolated dots that are not on the diagonal represent random noise
There is a lot of background noise, so we need to add a filter.
library(seqinr)
seq1 = unlist( strsplit( "THISISADNASEQUENCE", split = "" ) )
seq2 = unlist( strsplit( "THISISNOTRNASEQUENCEWTH", split = "" ) )
dotPlot( seq1, seq2, wsize=4, wstep=1, nmatch=3, col=c("white","black"))
SCORING ALIGNMENTS

- Alignment of related sequences should give good scores compared with non-related alignments.
- Genuine matches do not have to be identical.
  - The more similar the physiochemical properties of two residues, the greater the chance that the substitution is harmless to protein function.
  - Such substitution should be penalized less than the one where physiochemical properties are more different.
SCORING ALIGNMENTS

- **Identity** is the extent to which two sequences are invariant to each other.
- **Percent Identity** is obtained by taking the percentage of identical matches from the total length of sequence alignment.
- Taking into consideration the similarity between the amino acids, one can replace percent identity with **percent similarity**.
SUBSTITUTION MATRICES

- Describes the rate at which one character in a sequence changes to other character states over time.
- Provide scores for matches based on their occurrences in aligned protein families.
- A dot is placed in the matrix only if a minimum similarity score is found.
- Can be used with the sliding window option that averages the score within the window and prints a dot only above a certain average score.
SUBSTITUTION MATRICES

PAM – Point Accepted Mutation (Margaret Dayhoff)

- Depicts the likelihood of change from one amino acid to another based on observed mutations in 71 families of closely related proteins (85% identity) depicting homologous protein sequences during evolution.

<table>
<thead>
<tr>
<th></th>
<th>Cys</th>
<th>Gly</th>
<th>Pro</th>
<th>Ser</th>
<th>Ala</th>
<th>Thr</th>
<th>Asp</th>
<th>Glu</th>
<th>Asn</th>
<th>Gln</th>
<th>His</th>
<th>Lys</th>
<th>Arg</th>
<th>Val</th>
<th>Met</th>
<th>Ile</th>
<th>Leu</th>
<th>Phe</th>
<th>Tyr</th>
<th>Trp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>-5</td>
<td>-5</td>
<td>-4</td>
<td>-5</td>
<td>-3</td>
<td>-5</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-4</td>
<td>0</td>
<td>-8</td>
</tr>
<tr>
<td>Cys</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gly</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ala</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gln</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>His</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lys</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Val</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Met</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tyr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PAM250** corresponds to 250 amino acid replacements per 100 residues
SUBSTITUTION MATRICES

PAM – Point Accepted Mutation (Margaret Dayhoff)

- Depicts the likelihood of change from one amino acid to another based on observed mutations in 71 families of closely related proteins (85% identity) depicting homologous protein sequences during evolution.

PAM1 gives the probability that a given amino acid will be replaced by any other particular amino acid after a given evolutionary interval, in this case 1 accepted point mutation per 100 amino acids.
SUBSTITUTION MATRICES

BLOSUM – Blocks Amino Acid Substitution Matrix (Henikoff and Henikoff)

- Based on the observed amino acid substitutions in a large set (≈ 2000) of conserved amino acid patterns, called blocks.
- More than 500 protein families used to create the matrix.
- BLOSUM62 means that sequences clustered in the block were at least 62% identical.
- Allows for detection of more distantly related sequences.
## SUBSTITUTION MATRICES

**BLOSUM – Blocks Amino Acid Substitution Matrix** (Henikoff and Henikoff)

<table>
<thead>
<tr>
<th></th>
<th>Ala</th>
<th>Arg</th>
<th>Asn</th>
<th>Asp</th>
<th>Cys</th>
<th>Gln</th>
<th>Glu</th>
<th>Gly</th>
<th>His</th>
<th>Ile</th>
<th>Leu</th>
<th>Lys</th>
<th>Met</th>
<th>Phe</th>
<th>Pro</th>
<th>Ser</th>
<th>Thr</th>
<th>Trp</th>
<th>Tyr</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ala</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>-3</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>-3</td>
<td>-1</td>
<td>-2</td>
<td>-4</td>
<td>0</td>
<td>-4</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-3</td>
</tr>
<tr>
<td>Arg</td>
<td>-1</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Asn</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-5</td>
<td>-3</td>
<td>-4</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>Asp</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
</tr>
<tr>
<td>Cys</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>Gln</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-3</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>5</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>Glu</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-4</td>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>-4</td>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>-4</td>
<td>2</td>
<td>-5</td>
<td>2</td>
<td>-4</td>
<td>2</td>
</tr>
<tr>
<td>Gly</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>His</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>Ile</td>
<td>-1</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>4</td>
<td>-3</td>
<td>4</td>
</tr>
<tr>
<td>Leu</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>2</td>
<td>-4</td>
<td>3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>3</td>
</tr>
<tr>
<td>Lys</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>1</td>
<td>1</td>
<td>-2</td>
<td>-1</td>
<td>-3</td>
<td>2</td>
<td>-1</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>2</td>
<td>-2</td>
</tr>
<tr>
<td>Met</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>5</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>2</td>
<td>-2</td>
</tr>
<tr>
<td>Phe</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>Pro</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>Ser</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-4</td>
<td>-3</td>
</tr>
<tr>
<td>Thr</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Trp</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-4</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
</tr>
<tr>
<td>Tyr</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>Val</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>3</td>
<td>-3</td>
<td>3</td>
<td>1</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>-3</td>
</tr>
</tbody>
</table>

Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro Ser Thr Trp Tyr Val

"Sequence analysis"
Bioinformatics Course
WHICH MATRIX TO USE?

- For **global alignment** use **PAM** matrices
  - Lower PAM matrices tend to find short alignments of highly similar regions.
  - Higher PAM matrices will find weaker, longer alignments.

- For **local alignments** use **BLOSUM** matrices
  - Higher number BLOSUM matrices are better for similar sequences.
  - Low number BLOSUM matrices are better for distant sequences.

[http://benedick.rutgers.edu/homology/scoringmatrices06.pdf](http://benedick.rutgers.edu/homology/scoringmatrices06.pdf)
DYNAMIC PROGRAMMING

- Breaks down the alignment of sequences into small parts, considering all possible changes when moving from one pair of characters to the next.
- Finds the **best or optimal alignments** given an additive alignment score.
- May produce **more than one optimal alignment**.
- Global alignment uses **Needleman-Wunch** algorithm.
- Local alignment uses **Smith-Waterman** algorithm.
DYNAMIC PROGRAMMING

- Score measurement is determined by “match award”, “mismatch penalty” and “gap penalty”. The higher the score the better the alignment.

- Optimal alignment has the highest possible score given a substitution matrix and a set of gap penalties.


<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-2</td>
</tr>
<tr>
<td>A</td>
<td>-3</td>
</tr>
</tbody>
</table>

match is 1
mismatch is -1
insertion deletion (gap) is -1
GAP PENALTY

- Used to **score insertions and deletions**.
- Optimal alignment maximizes the number of matches and minimizes the number of gaps.
- Adding gaps reduces mismatches.
- "**Affine**" gap penalties give a big penalty for each new gap, but a much smaller penalty for "**gap elongation**".

```
ACTCTTACCGGCATATTGCTAGCATTGGCTAGCCTCA
```

```
ACTCTT-----CATATT-CTAGCA---GCTAGCCTCA
```

| penalties | 18 | 10 | 14 |

**gap penalty:** 10  
**gap elongation:** 2
DYNAMIC PROGRAMMING
BASIC PRINCIPLES

- Creation of an alignment path matrix
- Stepwise calculation of score values
- Backtracking (evaluation of the optimal path)
NEEDLEMAN-WUNSCH

1. INITIALIZATION

- assign values for the first row and column
- the score of each cell is set to gap score multiplied by the distance from the origin

match is 1
mismatch is -1
insertion deletion (gap) is -1
NEEDLEMAN-WUNSCH

2. FILL

- the entire matrix is filled with scores and pointers
- compute match score, vertical gap score and horizontal gap score
- assign the maximal value to the cell

\[ D(i, j) = \max \left\{ \begin{array}{l}
D(i - 1, j - 1) + s(x_i, y_j) \quad \text{score} \\
D(i - 1, j) + g \quad \text{gap} \\
D(i, j - 1) + g \quad \text{gap}
\end{array} \right. \]
NEEDLEMAN-WUNSCH

2. FILL

- the entire matrix is filled with scores and pointers
- compute match score, vertical gap score and horizontal gap score
- assign the maximal value to the cell

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>I</th>
<th>O</th>
<th>I</th>
<th>N</th>
<th>F</th>
<th>O</th>
<th>R</th>
<th>M</th>
<th>A</th>
<th>T</th>
<th>I</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
<td>-10</td>
<td>-11</td>
<td>-12</td>
<td>-13</td>
<td>-14</td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
<td>-10</td>
<td>-11</td>
</tr>
<tr>
<td>I</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
</tr>
<tr>
<td>O</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
</tr>
<tr>
<td>M</td>
<td>-4</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
<td>-6</td>
</tr>
<tr>
<td>A</td>
<td>-5</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-6</td>
<td>-4</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>I</td>
<td>-7</td>
<td>-5</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>-8</td>
<td>-6</td>
<td>-4</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>S</td>
<td>-9</td>
<td>-7</td>
<td>-5</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
NEEDLEMAN-WUNSCH

3. TRACEBACK

- traceback recovers the alignment from the matrix
- start at the bottom right and move in the direction of arrows until you arrive at the top left corner
SMITH-WATERMAN

- Similar to Needleman-Wunsch where negative scoring matrix cells are set to 0.
- Backtracking starts with the highest scoring cell and continues until a cell with score zero or D(0,0) is encountered, yielding the highest scoring local alignment.

\[
D(i, j) = \max \begin{cases} 
D(i-1, j-1) + s(x_i, y_j) \\
D(i-1, j) + g \\
D(i, j-1) + g 
\end{cases}
\]
• Begin with **maximal scoring element**
• Follow pointers that gave max score for each element
• Continue until you **reach an element with zero score**
• Construct alignment from traceback path

```
B I O I N F O R M A T I C S
0 0 0 0 0 0 0 0 0 0 0 0 0
B 0 1 0 0 0 0 0 0 0 0 0 0 0
I 0 0 2 1 1 0 0 0 0 0 0 1 0 0
O 0 0 1 3 2 1 0 1 0 0 0 0 0 0
M 0 0 0 2 1 0 0 0 0 1 0 0 0 0
A 0 0 0 1 1 0 0 0 0 2 1 0 0 0
T 0 0 0 0 0 0 0 0 0 1 3 2 1 0
I 0 0 1 0 1 0 0 0 0 2 4 3 2
C 0 0 0 0 0 0 0 0 0 1 3 5 4
S 0 0 0 0 0 0 0 0 0 2 4 6
```

\[
D(i, j) = \max \begin{cases} 
0 \\
D(i-1, j-1) + s(x_i, y_j) \\
D(i-1, j) + g \\
D(i, j-1) + g 
\end{cases}
\]
### EXTENDED SMITH-WATERMAN

- Delete regions around best path
- Repeat backtracking

```
<table>
<thead>
<tr>
<th>BIOINFO MAT ICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>B 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>I 0 0 2 1 1 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>O 0 0 1 3 2 1 0 1 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>M 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>A 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>T 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>S 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
```

START HERE

**BIOINFORMATICS**

||| Local alignment show
**BIOMATICS** in red

“Sequence analysis”
Bioinformatics Course
K-TUPLE METHODS
FASTA & BLAST

Instead of comparing individual characters, k-tuple methods compare sequence patterns of words, called k-tuples.

These patterns comprise of k consecutive matches in both sequences.

Such methods are much faster than dynamic programming methods, but are also less sensitive.
For query **BIOINFORMATICS**, for \( k = 8 \), the set of \( k \)-tuples for query is:

**BIOINFOR**

**IOINFORM**

**OINFORMA**

**INFORMAT**

**FORMATI**

**FORMATIC**

**ORMATICS**

How many \( k \)-tuples are there in a string of length \( n \)?
K-TUPLE METHODS
FASTA & BLAST

For query BIOINFORMATICS, for \( k = 8 \), the set of \( k \)-tuples for query is:

BIOINFOR
IOINFORM
OINFORMA
INFORMAT
FORMATI
FORMATIC
ORMATICS

How many \( k \)-tuples are there in a string of length \( n \)?
The answer is: \( n - k + 1 \)
For query \textit{BIOINFORMATICS}, for \( k = 8 \), the set of \( k \)-tuples for query is:

\begin{verbatim}
BIOINFORM 
IOINFORMAT 
OINFORMATI 
FORMATICA 
FORMATIC 
ORMATICS
\end{verbatim}

If not one query \( k \)-tuple is found from the target sequence, we can deduce that these two sequences are different from each other.

How many \( k \)-tuples are there in a string of length \( n \)?

The answer is: \( n - k + 1 \)
FASTA

- Program for rapid alignment of pairs of protein and DNA sequences.
- Uses local sequence alignment to find matches of similar database sequences.
- Main idea:
  - Choose regions of the two sequences that look promising (have some degree of similarity).
  - Compute local alignment using dynamic programming in these regions.
FASTA

1. Identify common k-words between two sequences.
2. Score diagonals with k-word matches, identify 10 best diagonals.
3. Rescore initial regions with a substitution matrix.
4. Join initial regions using gaps, penalize gaps.
5. Perform dynamic programming to find initial alignments.
FASTA

Identify all exact matches of length $k$ or greater between the two sequences.

I: GCATCGGC
J: CCATCGCCCATCG

Look up table

<table>
<thead>
<tr>
<th>k-word</th>
<th>I Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>3</td>
</tr>
<tr>
<td>CA</td>
<td>2</td>
</tr>
<tr>
<td>CG</td>
<td>5</td>
</tr>
<tr>
<td>GC</td>
<td>1,7</td>
</tr>
<tr>
<td>GG</td>
<td>6</td>
</tr>
<tr>
<td>TC</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>k-word</th>
<th>J Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>3,9</td>
</tr>
<tr>
<td>CA</td>
<td>2,8</td>
</tr>
<tr>
<td>CC</td>
<td>1,7</td>
</tr>
<tr>
<td>CG</td>
<td>5,11</td>
</tr>
<tr>
<td>GC</td>
<td>6</td>
</tr>
<tr>
<td>TC</td>
<td>4,10</td>
</tr>
</tbody>
</table>
FASTA

k-tup matches can be depicted in a matrix; diagonals indicate matches that have the highest density of common words.

Top ten matches are selected (initial regions).

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
FASTA

Rescore top 10 diagonals using a substitution matrices.

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
FASTA

Check if the initial regions can be joined to form an approximate alignment with gaps.

Calculate the similarity score, penalize with gaps.

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
FASTA

Uses Smith-Waterman algorithm to find an optimal score for alignment.

Use dynamic programming to optimise the alignment in a narrow band that encompasses the top scoring segments.

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
BLAST

**BASIC LOCAL ALIGNMENT SEARCH TOOL**

- Retrieves homologous sequences from the database.
- Used to find best local alignment to a query sequences against the sequences in the database (both DNA and protein).
- Heuristic approach based on Smith-Waterman algorithm.
- Calculates the statistical significance of matches.
BLAST

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, delta-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

"Sequence analysis"
Bioinformatics Course
BLAST

1. Compile a list of high-scoring words
2. Scan the database for instances of these words, called hits
3. Extend hits to differentiate random hits from meaningful hits
BLAST

Create **neighborhood words** for each query word

Compile a list of **high scoring words** of length \( w \).

For each word from the query sequence find the list of words that will score at least \( T \) when scored using a pairscore matrix (e.g. PAM 250). For typical parameters there are around 50 words per residue of the query.

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
BLAST

Create **neighborhood words** for each query word.

Compile a list of **high scoring words** of length w.

---

Query word $W = 3$

```
GSVEDTTGSQSLAALLNKCKTPQGRLVNQWIKQPQPLMDKRNIEERLNLVEAFVEDAEL
```

**Neighborhood words**

<table>
<thead>
<tr>
<th>Neighborhood words</th>
<th>Threshold for neighborhood words</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQG 18</td>
<td>$T = 13$</td>
</tr>
<tr>
<td>PEQ 15</td>
<td></td>
</tr>
<tr>
<td>PRG 14</td>
<td></td>
</tr>
<tr>
<td>PKG 14</td>
<td></td>
</tr>
<tr>
<td>PNG 13</td>
<td></td>
</tr>
<tr>
<td>PDG 13</td>
<td></td>
</tr>
<tr>
<td>PHG 14</td>
<td></td>
</tr>
<tr>
<td>PMG 13</td>
<td></td>
</tr>
<tr>
<td>PSG 13</td>
<td></td>
</tr>
<tr>
<td>PQA 12</td>
<td></td>
</tr>
<tr>
<td>PQN 12</td>
<td></td>
</tr>
<tr>
<td>etc...</td>
<td></td>
</tr>
</tbody>
</table>
BLAST

Compare the word list to a database and identify exact matches

http://www.compbio.dundee.ac.uk/ftp/preprints/review93/review93.pdf
BLAST

Extend hits by summing residue pairs from both sides of the word boundary.

Extension stops when score drops below a threshold of the best score yet observed.

All extended hits above the minimum score are reported.

Maximal Segment Pairs (MSPs)

http://www.compbio.dundee.ac.uk/ftp/preprints/review93(review93.pdf
BLAST

**Alignments**

- **Select All**
- **Get selected sequences**
- **Distance tree of results**
- **Multiple alignment**

**gb|AAV66967.1|**

**gb|ABG49490.1|**

**gb|AAB59568.1|**
WHY OR WHEN TO COMPARE TWO SEQUENCES

- Are they homologous / share common ancestor
- Do they share similar domains
- Identify exact locations to see common feature-active sites
- Compare a gene and its product to other genes and products