Video tutorials

BLAST E-values: https://www.youtube.com/watch?v=nO0wJgZRZJs
https://www.youtube.com/watch?v=Z7ek7UoP7Bg

Phylogenetic trees: https://www.youtube.com/watch?v=09eD4A_HxVQ
Basic bioinformatics concepts, databases and tools

Module 2

Searching for similar sequences
Thanks to annotations we can use text searches to fetch sequences from sequence databases (module 1)

In module 2 we will use sequence searches to fetch similar sequences from sequence databases
Sequence searches are based on sequence alignments

Arrange sequences so that similar regions are placed directly under each other

| indicates that both sequences contain the same nucleotide in that position

- indicates a gap in one of the sequences

```
tcctctgcctctgccatcat---caaccccaaaagtccg
 ||   ||             ||   ||   ||   ||   ||   ||   ||   ||   ||   ||
tcctgtgcatctgcaatcatggaagtgga
```

Also done for protein sequences

Shows how similar sequences are
How is alignment done?

Computational approaches:

**global**: force the alignment to span entire length of all sequences

**local**: identify regions of similarity in sequences

Local alignments are preferable but more difficult to calculate

<table>
<thead>
<tr>
<th>Global</th>
<th>FTFTALILLAVAV</th>
<th>Comparing two papers, start to end</th>
</tr>
</thead>
<tbody>
<tr>
<td>F--TAL--LLA--AV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Local</th>
<th>FTFTALILLAVAV--F</th>
<th>Find the two most similar sentences between two papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>--FTAL--LLAAV--</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
There are many ways to align 2 sequences

tcctctgctgctat---atcatca
|||   |||   |||   |||   |||
tcctgtgcatatgcaatcata

OR

tcctctgtgctatatat--ca--tca
||  |||   |||   |||   |||   |||
tc--ctgtgcatatgcaatcata

Which is the correct alignment?
Scores to determine the best alignment

Each individual comparison has a score:

- **match**: +5
- **mismatch**: -2
- **insertion**: -6
- **deletion**: -6

**Similarity score** of an alignment

= sum of the scores of individual comparisons

```
t c c c t c t - t
| | | | | | | |
t c c - t g t g t
```

Score = 5 + 5 + 5 - 6 + 5 - 2 + 5 - 6 + 5 = 16
Which computational approaches are used?

Variety of algorithms have been applied to sequence alignment:

**Dynamic programming:** slow but correct

- sure that the alignment is the best possible

**Heuristic algorithms:** faster but less accurate

- good alignment but no guarantee that it is the best possible

**Combination:** start with heuristic, proceed with dynamic programming
Dynamic programming divides a problem into simple subproblems

To solve a problem: break it down into subproblems

If many of these subproblems are the same

Dynamic programming:

- solves each subproblem only once
- stores the solutions to the subproblems
- next time the same subproblem pops up the solution is retrieved

Not applicable to all problems: same subproblems popping up repeatedly

Dynamic programming applied to sequence alignment

Problem: Align two sequences of length L1 and L2

Subproblems: Align sequences of smaller lengths

You compare the first letter of each sequence:

**match/mismatch:**
- A TC
  - proceed alignment with TCG (L1-1)
- A GC
  - proceed alignment with GCGG (L2-1)

**insertion:**
- ATCG
  - proceed alignment with ATCG (L1)
- C ATCG
  - ATCG (L2-1)

**deletion:**
- ATCG
  - proceed alignment with TCG (L1-1)
- TCGGG
  - TCGGG (L2)

Program calculates similarity score for each subalignment

Finds the sequence of subalignments leading to the highest total score
Dynamic programming is very accurate but slow

Too slow for searching a database of billions of sequences

For database searches dynamic programming is combined with heuristic methods

For aligning two sequences dynamic programming is used
Heuristics: well considered assumptions, educated guesses...

Not based on solid calculations

Fast but not always accurate

Heuristic will quickly produce a solution that is good enough for solving the problem not find all exact solutions

Some heuristics have a strong underlying theory:

- assumptions derived from theory
- inferred from experimental data

Others are just rules of thumb learned by experience without any theory
Heuristics applied to sequence alignment

Based on the assumption that:

we \textbf{expect} to find a few regions that are completely the same in the two sequences

The algorithm looks for such short identical words (= seeds)

The seeds are then used to initiate further alignment by dynamic programming

We will see examples of this approach (BLAST) later in this module
Calculating how similar sequences are

DNA sequences: scores

Protein sequences: scoring matrixes

Gap penalties
Amino acids have properties that make them more or less similar

Protein sequences: more complex than just match / mismatch
20 amino acids: some are more similar than others because of similar properties

- **Aromatic R groups**
  - Phenylalanine
  - Tyrosine
  - Tryptophan

- **Positively charged R groups**
  - Lysine
  - Arginine
  - Histidine

- **Negatively charged R groups**
  - Aspartate
  - Glutamate
For proteins, scores for matches are stored in a scoring matrix.

e.g. BLOSUM62 scoring matrix

<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>-1</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Arg</td>
<td>-1</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>9</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>Asn</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>7</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>Asp</td>
<td>-2</td>
<td>-2</td>
<td>6</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>Cys</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>9</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>-4</td>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>-2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Glu</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>2</td>
<td>-2</td>
<td>-4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>-4</td>
<td>4</td>
<td>4</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>-4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>-4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>His</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</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>-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>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>-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>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>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</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>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</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>-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>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>-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>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>Ser</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>0</td>
<td>-1</td>
<td>-2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Thr</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</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>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</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>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Val</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>
<td>-3</td>
</tr>
</tbody>
</table>

Likely changes: positive score

Unlikely changes: negative score
Many scoring matrices have been developed

Derived from analysis of alignments of related sequences

PAM Matrices (Point Accepted Mutations) obtained by making global alignments of families of very similar proteins

Not many families used: many mutations were not observed!

Protein family: group of homologous proteins (from one or multiple organisms)
PAM1 matrix is the basis for the PAMn matrices

PAM1 matrix:
if 1% of amino acids in the family changed how many times was A replaced by V ?
and these replacement counts are calculated for all amino acid combinations

PAMn matrices are calculated based on PAM1 by assuming that:
next mutations follow the same pattern as those in PAM1
multiple substitutions can occur at the same site

PAM30 and PAM120 are most used
Many scoring matrices have been developed

BLOSUM (BLOck SUbstitution Matrices) obtained by:

- making alignments of families of **less similar proteins**
- selecting **blocks** of highly similar regions
- in a block merging sequences with similarity above a **threshold**
- counting replacements of amino acids in resulting blocks
BLOSUMn matrices have different sequence similarity thresholds

E.g. for the BLOSUM62 matrix the **threshold** was set at 62%

-> sequences in blocks that are > 62% similar are merged and counted once

-> sequences < 62% similar are kept

BLOSUM62 is most frequently used
Which scoring matrix do you use?

Global alignment: PAM matrices

Lower PAMs for short alignments of similar sequences
Higher PAMs for longer alignments of similar sequences

Local alignment: BLOSUM matrices

higher BLOSUMs for similar sequences
lower BLOSUMs for dissimilar sequences

Matrices do not deal with insertions/deletions: gap penalties
Gap penalties strongly influence the outcome of the search

<table>
<thead>
<tr>
<th>Gap penalty</th>
<th>Alignment</th>
<th>Identity</th>
<th>Gaps</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="alignment0.png" alt="" /></td>
<td>27/50</td>
<td>23/50</td>
<td>S=135</td>
</tr>
<tr>
<td>5</td>
<td><img src="alignment5.png" alt="" /></td>
<td>26/44</td>
<td>11/44</td>
<td>S=67</td>
</tr>
<tr>
<td>10</td>
<td><img src="alignment10.png" alt="" /></td>
<td>10/67</td>
<td>57/67</td>
<td>S=50</td>
</tr>
</tbody>
</table>

Gap penalty too large => gaps are avoided, sequences cannot be properly aligned

Gap penalty too low => gaps are inserted everywhere to prevent mismatches
Multiple gap penalties are defined

In most alignments multiple gap penalties are used:

- Penalty for creating a new gap: strongly penalized e.g. -4
- Penalty for extending an existing gap: not so hard penalized e.g. -1
Why make alignments?

Similarity is a prerequisite of homology between sequences

**Homologs** = sequences that share a common ancestor

Since they evolved from the same ancestral gene they have similar sequence

Same sequence $\rightarrow$ same structure $\rightarrow$ same function

Used to predict function of proteins

*Understanding evolution: [http://evolution.berkeley.edu/evolibrary/article/similarity_hs_06](http://evolution.berkeley.edu/evolibrary/article/similarity_hs_06)*
Different types of homologs

**Orthologs**: most homologs from different species

Result of *speciation* (evolution of species from common ancestor)

**Paralogs**: homologs from same species + some homologs from different species

Result of gene *duplication*

All orthologs and paralogs are homologs!

Homology is not the same as similarity!

Homology is either TRUE or FALSE

You cannot make the following statements:

“Two sequences are 50% homologous”

“Two sequences show significant homology”

“Two sequences show high homology”

Just like you cannot say:

“This woman is 50% my mother” -> She either is your mother or not

So two sequences either are homologs or they are not

You can say: “Two sequences are 50% similar”, “significant similarity”...
Different algorithms for different types of alignments

One to many: BLAST
Local alignments
Combinations of heuristics and dynamic programming

One to one: pairwise alignment
Global or local alignments
Dynamic programming

Many to many: multiple alignment
Dynamic programming: based on pairwise alignments
Heuristic: progressive alignment
Why searching for similar sequences in a database?

Looking for species
You want to identify the species your sequence comes from or homologous species

Mapping DNA to a known chromosome
You can find the position of your sequence in the genome

Annotation
You can map annotations from one organism to another

...
Sequence searches on NCBI nucleotide databases

1. **BLAST removes repeats from the query sequence**

   Repeats would confuse the program so they’re filtered out

   Default for nucleotide queries

   Not for protein queries

2. **Divides the query into short words (= seeds)**

   protein: words are min 2 amino acids, default = 6

   DNA/RNA: words are min 7 nucleotides, default = 11
3. Searches for highest scoring words

All possible 6-letter combinations are compared to seeds

Scores are calculated between seed – combination using a scoring matrix

All high-scoring combinations are retained

Scores are calculated using BLOSUM62 matrix

Query: SEARCHESFR

<table>
<thead>
<tr>
<th>Use words for db search</th>
<th>Scores</th>
<th>Threshold for scores to be retained</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHES</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>AKCHES</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>AQCHES</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>AHCHES</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>ASCHES</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

...
e.g. BLOSUM62 scoring matrix

<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>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>0</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>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Arg</td>
<td>-1</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>0</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>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Asn</td>
<td>-2</td>
<td>-1</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>-1</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>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Asp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</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>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Cys</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Gln</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Glu</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>Gly</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>His</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>Ile</td>
<td>-1</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
<td>-1</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>Leu</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>4</td>
<td>-1</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>Lys</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>4</td>
<td>-1</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>Met</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</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>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Pro</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Thr</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Trp</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>
<td>-3</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>-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>
</tbody>
</table>
4. Search database for exact matches to seeds

5. When match is found, BLAST performs local alignment

Neighborhood regions are included into alignment by dynamic programming

Query: SEARCHESFRHI

Hit: FEACKCHESYREA seed is found in database sequence

Score: -2 5 4 2 9 8 5 4 3 5 0 -1 extend alignment

HSP high scoring pair

Similarity score $S$: 5+4+2+9+8+5+4+3+5+0=45
6. Calculate bit score for each HSP based on similarity score

\[ S' = \frac{\lambda S - \ln(K)}{\ln(2)} \]

*\( S \): total similarity score

\( \lambda \) and \( K \) depend on the scoring matrix and on the gap penalties

Magnitude of search space you have to look through before you find a similarity score as good as or better than yours by chance:

\( S' = 30 \): you have to score \( 2^{30} \) sequence pairs to find a score this high by chance

Log rescaled version of similarity score independent of the size of the db

7. List all HSPs with bit scores above threshold

Threshold score is called cutoff

Determined by BLAST by comparison of scores to these of random sequences
8. Calculate significance of alignment scores

\[ E = mn \cdot P_{val} \]
\[ = Kmn e^{-\lambda S} \]
\[ = NK e^{-\lambda S} \]
\[ = N / 2^S \]

N = size of search space = m*n  (n = length of query; m = length of database)

E = number of alignments with score >= S expected to occur by chance

Low E value -> ? High/low bit score -> ? Biologically relevant / irrelevant hit

Repeat BLAST search next month – same E ? Same S’ ?

Video tutorial: https://www.youtube.com/watch?v=nO0wJgZRZJs
9. Combine multiple HSPs in same hit

Multiple matches in same hit -> Combined bit score is calculated

10. Create output

Show local alignments of query and hits

Report every HSP whose E is lower than threshold

Threshold is the Expect value (one of the parameters in the BLAST form)
Which BLAST do you choose?


Depends on sequence type of query and sequence type of database sequences

<table>
<thead>
<tr>
<th>Query</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blastn</td>
<td>DNA</td>
</tr>
<tr>
<td>Blastp</td>
<td>Protein</td>
</tr>
<tr>
<td>Blastx</td>
<td>Translated DNA</td>
</tr>
<tr>
<td>Tblastn</td>
<td>Protein</td>
</tr>
<tr>
<td>Tblastx</td>
<td>Translated DNA</td>
</tr>
</tbody>
</table>

**Translated DNA**: DNA is translated in 6 different ways

- 3 reading frames
- 2 strands

Nucleotide BLAST can be done by several algorithms

**nucleotide blast** | Search a **nucleotide** database using a **nucleotide** query

*Algorithms:* blastn, megablast, discontiguous megablast

**Megablast**

- to find sequences nearly identical to query (min. 95%)
- long seeds (16-256nt)
- fast

To identify a sequence

- check if a sequence is already present in a database
- find transcript variants
- **check primer specificity**
- map oligo’s, cDNAs and PCR products to the genome

Nucleotide BLAST can be done by several algorithms

**Discontiguous megablast** to find sequences similar but not identical to query

- 11-12nt seeds that allow mismatches in 3\(^{rd}\) codon position
- more sensitive and faster than blastn with same word size

To find similar sequences in other species

BLAST using a coding sequence as an input

**Blastn** to find more dissimilar sequences

- 7nt seeds for very sensitive searches
- slow

Used for more distant cross-species comparison
Nucleotide blast is not ideal for finding homologs!

Nucleotide BLAST searches are not the best method for finding homologs!

Protein BLAST or translated BLAST are better for this because:

- Codon degeneracy
- Greater information available in amino acid sequence (20 letters versus 4)
- More sophisticated algorithm and scoring matrix used in protein BLAST
Protein BLAST can be done by several algorithms

**Blastp**  general tool for protein similarity searches
to find sequences both identical and similar to query

Other protein blast tools have different functionality: see later
Translated blast searches

**BLASTX:**

Use when reading frame of sequence is not known e.g. finding CDS in genome

Most used with first time sequenced sequences

BLASTX translates **nt query** in all reading frames and compares to **protein database**

More sensitive than blastn since comparison is done at protein level!

**TBLASTN:**

Compares a **protein sequence** to translated **nucleotide database**

Find homologs in unannotated nucleotide sequences such as ESTs and draft genomes

(ESTs are too short to contain a full CDS so no protein translations available)

**TBLASTX:**

Finding novel genes and proteins encoded by ESTs
Specialized BLAST methods for specific applications

**Primer Blast**
- search primers with Primer3 + check specificity with BLAST

**Vecscreen**
- searches the UniVec database
to identify vector contamination in your sequences

**IgBlast**
- searches a database of annotated immunoglobulin sequences

---

**Specialized searches**

- **SmartBLAST**
  - Find proteins highly similar to your query

- **Primer-BLAST**
  - Design primers specific to your PCR template

- **Global Align**
  - Compare two sequences across their entire span (Needleman-Wunsch)

- **CD-search**
  - Find conserved domains in your sequence

- **GEO**
  - Find matches to gene expression profiles

- **IgBLAST**
  - Search immunoglobulins and T cell receptor sequences

- **VecScreen**
  - Search sequences for vector contamination

- **CDART**
  - Find sequences with similar conserved domain architecture
The BLAST interface: specify the **input** sequence

Input sequence is called **Query**

If you want to use only a part of the query sequence

From: amino acid at this position is included

To: amino acid at this position is included

Only works if you have a single query

If you have many query sequences you can upload a txt file from your computer

Upload a list of:
- accession numbers
- GI numbers
- sequences in FASTA format
Specify the **database** you want to search in

**Database**
- Non-redundant protein sequences (nr)

**Organism**
- Enter organism name or id—completions will be suggested
- Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.

**Exclude**
- Models (XM/XP)
- Uncultured/environmental sample sequences

**Entrez Query**
- Enter an Entrez query to limit search

Select database you want to search in

You can select or exclude specific organisms from this database

You can exclude predicted (model) sequences

You can limit the database to the results of an Entrez query

Select or exclude sequences from the database

Taxid appears automatically once you start typing organism name

Exclude excludes the organism from the database

To select more than one organism use +

What will you search in?

You can restrict the database by an Entrez query

Entrez query = Gquery: question to search all NCBI databases simultaneously

Restrict the database to sequences that satisfy the query

Terms accepted by Entrez queries are also accepted here (see module 1)

Use Boolean operators, parentheses, quotes and search fields

What will you search in?

Video tutorial on using an Entrez query to create a custom database for BLAST
https://www.youtube.com/watch?v=t8fKz9rvuOk
Selecting the BLAST program you want to use

**Program Selection**

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastp (protein-protein BLAST)</td>
</tr>
<tr>
<td>PSI-BLAST (Position-Specific Iterated BLAST)</td>
</tr>
<tr>
<td>PHI-BLAST (Pattern Hit Initiated BLAST)</td>
</tr>
<tr>
<td>DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)</td>
</tr>
</tbody>
</table>

Select the BLAST algorithm you want to use

Start the BLAST search

Show parameter settings

**BLAST**

Search database Non-redundant protein sequences (nr) using Blastp (protein-protein BLAST)

Show results in a new window

**Algorithm parameters**

General BLAST parameters

For short (< 30 aa) query sequences: word size: 6 -> 2

- expect threshold: 10 -> 20000
- score matrix: BLOSUM62 -> PAM30

Why?

Seed length

High word size -> ? more / less similar hits

Video tutorial: https://www.youtube.com/watch?v=nO0wJgZRZJs
Parts of the query sequence can be masked for alignment

Mask segments of the query with **low compositional complexity**

- e.g. PPCDPPPPPKDKKKKDDGPP or AAATAAAAAAAATAAAAA
- e.g. common acidic-, basic- or proline-rich regions

Filtering is only applied to query not to database sequences

By default masking is done on DNA sequences, not on proteins  

*Why not?*
The default BLAST parameters work for most cases

Default BLAST parameters are quite optimal and well tested

Change the default parameters when:

• Query contains many identical residues -&gt; Change filter

• BLAST doesn’t report results
  -&gt; Change scoring system / gap penalties / database / word length

• Hit has a borderline E-value -&gt; Change scoring system / gap penalties

• Too many matches are reported
  -&gt; Change database / filter database by Entrez search / increase number reported matches / ? lower / higher Expect threshold
### Overview table of BLASTP results

<table>
<thead>
<tr>
<th>Description</th>
<th>Max score</th>
<th>Total score</th>
<th>Query cover</th>
<th>E value</th>
<th>Max ident</th>
<th>Accession</th>
</tr>
</thead>
<tbody>
<tr>
<td>defensin-like protein 16 [Arabidopsis thaliana] &gt; sp</td>
<td>Q9FI23.1</td>
<td>DEF1</td>
<td>100%</td>
<td>5e-51</td>
<td>100%</td>
<td>NP_199255.1</td>
</tr>
<tr>
<td>defensin [Arabidopsis halleri subsp. halleri]</td>
<td>159</td>
<td>159</td>
<td>100%</td>
<td>1e-48</td>
<td>96%</td>
<td>CCN78788.1</td>
</tr>
<tr>
<td>putative plant defensin PDF1.3 [Arabidopsis halleri] &gt; emb</td>
<td>CCD74541.1</td>
<td>defensin [Arabidopsis halleri subsp. halleri]</td>
<td>158</td>
<td>158</td>
<td>100%</td>
<td>2e-43</td>
</tr>
<tr>
<td>putative defensin-like protein 15 [Arabidopsis thaliana] &gt; sp</td>
<td>Q80994.1</td>
<td>DEF15</td>
<td>ARATH RecName: Full=Putative defensin-like protein 15; AltName: Full=Putative peptide 4 [Heliophila coronipitifolia]</td>
<td>157</td>
<td>157</td>
<td>100%</td>
</tr>
<tr>
<td>antifungal protein [Sinapis alba]</td>
<td>157</td>
<td>157</td>
<td>100%</td>
<td>6e-48</td>
<td>91%</td>
<td>AAY15221.1</td>
</tr>
<tr>
<td>defensin [Arabidopsis thaliana] &gt; sp</td>
<td>Q9FI22.1</td>
<td>DEF17</td>
<td>ARATH RecName: Full=Defensin-like protein 17; AltName: Full=Plant defensin 1.2;</td>
<td>155</td>
<td>155</td>
<td>100%</td>
</tr>
<tr>
<td>putative plant defensin PDF1.2 [Arabidopsis halleri]</td>
<td>154</td>
<td>154</td>
<td>100%</td>
<td>5e-47</td>
<td>90%</td>
<td>CCN78786.1</td>
</tr>
<tr>
<td>hypothetical protein ARALYDRAFT_95457 [Arabidopsis lyrata subsp. lyrata] &gt; qb</td>
<td>EFH63873.1</td>
<td>hypothetical protein ARALYDRAFT_95457 [Arabidopsis lyrata]</td>
<td>152</td>
<td>152</td>
<td>100%</td>
<td>3e-46</td>
</tr>
<tr>
<td>defensin-like protein 14 [Arabidopsis thaliana] &gt; sp</td>
<td>Q80955.1</td>
<td>DEF14</td>
<td>ARATH RecName: Full=Defensin-like protein 14; AltName: Full=Cysteine-rich antifungal protein 4; Short=AFP4; Flags: Precursor &gt; emb</td>
<td>CA65963.1</td>
<td>defensin [Orychophragmus violaceus]</td>
<td>152</td>
</tr>
<tr>
<td>defensin [Brassica juncea] &gt; gb</td>
<td>AF859548.1</td>
<td>defensin [Brassica juncea]</td>
<td>149</td>
<td>149</td>
<td>100%</td>
<td>5e-45</td>
</tr>
<tr>
<td>downy mildew resistance-related protein DMR1 [Brassica oleracea var. italica]</td>
<td>149</td>
<td>149</td>
<td>100%</td>
<td>1e-44</td>
<td>86%</td>
<td>AB859548.1</td>
</tr>
<tr>
<td>gamma-thionin1 [Eutrema wasabii]</td>
<td>149</td>
<td>149</td>
<td>100%</td>
<td>1e-44</td>
<td>85%</td>
<td>ADV3301.1</td>
</tr>
<tr>
<td>defensin [Brassica oleracea var. italica]</td>
<td>148</td>
<td>148</td>
<td>100%</td>
<td>2e-44</td>
<td>84%</td>
<td>AB859548.1</td>
</tr>
<tr>
<td>antifungal peptide 4 [Heliophila coronipitifolia]</td>
<td>148</td>
<td>148</td>
<td>100%</td>
<td>2e-44</td>
<td>85%</td>
<td>AER4542.1</td>
</tr>
<tr>
<td>putative plant defensin PDF1.3 [Arabidopsis halleri] &gt; emb</td>
<td>CCN78777.1</td>
<td>defensin [Arabidopsis halleri subsp. halleri]</td>
<td>148</td>
<td>148</td>
<td>100%</td>
<td>2e-44</td>
</tr>
<tr>
<td>defensin [Brassica rapa subsp. pekinensis]</td>
<td>147</td>
<td>147</td>
<td>100%</td>
<td>3e-44</td>
<td>86%</td>
<td>AAN23105.1</td>
</tr>
</tbody>
</table>

**Descriptions + links to alignments**

**Video tutorial on interpreting the BLAST report**: [https://www.youtube.com/watch?v=yfITRIk34Js](https://www.youtube.com/watch?v=yfITRIk34Js)
Overview table of BLASTP results

<table>
<thead>
<tr>
<th>Description</th>
<th>Max score</th>
<th>Total score</th>
<th>Query cover</th>
<th>E value</th>
<th>Ident</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecName: Full=Outer membrane protein lom; AltName: Full=ORF 206A; Flags: Precursor</td>
<td>71.0</td>
<td>71.0</td>
<td>100%</td>
<td>4e-15</td>
<td>74%</td>
</tr>
<tr>
<td>RecName: Full=Copper homeostasis protein cutC homolog</td>
<td>32.0</td>
<td>32.0</td>
<td>81%</td>
<td>0.17</td>
<td>52%</td>
</tr>
<tr>
<td>RecName: Full=Virulence membrane protein PagC; Flags: Precursor</td>
<td>31.6</td>
<td>31.6</td>
<td>81%</td>
<td>0.24</td>
<td>50%</td>
</tr>
<tr>
<td>RecName: Full=Glutamate–tRNA ligase; AltName: Full=Glutamyl–tRNA synthetase; Short=GluRS</td>
<td>28.6</td>
<td>28.6</td>
<td>62%</td>
<td>2.6</td>
<td>67%</td>
</tr>
<tr>
<td>RecName: Full=50S ribosomal protein L6</td>
<td>27.8</td>
<td>27.8</td>
<td>37%</td>
<td>4.9</td>
<td>60%</td>
</tr>
<tr>
<td>RecName: Full=Glutamate–tRNA ligase; AltName: Full=Glutamyl–tRNA synthetase; Short=GluRS</td>
<td>27.8</td>
<td>27.8</td>
<td>48%</td>
<td>5.1</td>
<td>69%</td>
</tr>
<tr>
<td>RecName: Full=MORN repeat-containing protein 3</td>
<td>26.9</td>
<td>92.1</td>
<td>48%</td>
<td>9.8</td>
<td>62%</td>
</tr>
<tr>
<td>RecName: Full=D-alanine–D-alanine ligase; AltName: Full=D-Ala-D-Ala ligase; AltName: Full=D-alanylalanine synthetase</td>
<td>26.5</td>
<td>26.5</td>
<td>62%</td>
<td>14</td>
<td>48%</td>
</tr>
</tbody>
</table>

Scores = bit scores

Ident: % of identical amino acids in alignment

Why difference between max score and total score?
Why does fourth hit have a lower bit score than third?
Do we find an exact match? Why (not)?

Video tutorial on interpreting the BLAST report: [https://www.youtube.com/watch?v=yfIbRIk34Js](https://www.youtube.com/watch?v=yfIbRIk34Js)
The actual alignments of the query protein to a hit

Hit =

Positions in the sequence

Does the alignment stretch over the full length of the hit?

Link to sequence record of hit

Scores: bit score + E-value

Number of gaps in alignment

Video tutorial on interpreting the BLAST report: https://www.youtube.com/watch?v=yfITRIk34Js
Important issue when doing a translated search

Specify the codon table used in the translation of the input nucleotide query

*Candida albicans*: Alternative Yeast Nuclear (12)


Nucleotide BLAST interface is very similar to protein BLAST. Databases are of course different.

<table>
<thead>
<tr>
<th>Database</th>
<th>Content Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr ¹</td>
<td>All GenBank + EMBL + DDBJ + PDB sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences; No longer &quot;non-redundant&quot; due to computational cost.</td>
</tr>
<tr>
<td>refseq_mrna</td>
<td>mRNA sequences from NCBI Reference Sequence Project.</td>
</tr>
<tr>
<td>refseq_genomic</td>
<td>Genomic sequences from NCBI Reference Sequence Project.</td>
</tr>
<tr>
<td>est</td>
<td>Database of GenBank + EMBL + DDBJ sequences from EST division.</td>
</tr>
<tr>
<td>gss</td>
<td>Genome Survey Sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences.</td>
</tr>
<tr>
<td>htgs</td>
<td>Unfinished High Throughput Genomic Sequences: phases 0, 1 and 2. Finished, phase 3 HTG sequences are in nr.</td>
</tr>
<tr>
<td>pat</td>
<td>Nucleotides from the Patent division of GenBank.</td>
</tr>
<tr>
<td>pdb</td>
<td>Sequences derived from the 3-dimensional structure records from Protein Data Bank. They are NOT the coding sequences for the corresponding proteins found in the same PDB record.</td>
</tr>
<tr>
<td>alu_repeats</td>
<td>Select Alu repeats from REPBASE, suitable for masking Alu repeats from query sequences. See &quot;Alu alert&quot; by Claverie and Makalowski, Nature 371: 752 (1994).</td>
</tr>
<tr>
<td>dbsts</td>
<td>Database of Sequence Tag Site entries from the STS division of GenBank + EMBL + DDBJ.</td>
</tr>
</tbody>
</table>
Same parameters, different settings

Why are words much longer than in protein blast?
You can install BLAST on your own computer and run it locally

1. Download the blast+ command line tool:

2. Download or create a database

On local BLAST you can use your own databases for sequence searches

Just create a FASTA file of the sequences you want to search in and use this as db

  e.g. (Linux) Create a BLAST database from a fasta file: `makeblastdb -in mydb.fasta`

  Run BLAST: `blastp -query seq.fasta -db mydb.fasta -out output.xml`

You can also do this in CLC and UGene


Holy grail of biology:  
Infer biological function based on sequences

Most proteins: sequence is known but not function

Relation between sequence and function:

**Sequence** determines **structure** which amino acids can interact and which not?  
**Structure** determines **function** amino acids that do an action must be accessible

Structure of functionally important parts cannot change without losing function
So sequence of these parts does not change during evolution (= **conserved**)
Remainder of protein evolves normally

**Motifs**: conserved regions crucial for protein’s function
Proteins with the same function have the same structure

**Human cyclophilins:**
all isomerases of peptide bonds
that facilitate protein folding
Only functional parts of a protein sequence are conserved.

Part that binds the DNA
How much similarity is required to speak of homologs?

Depends on evolution rate of gene

- Very conserved genes: e.g. genes encoding histone proteins
  Histones have adopted ideal sequence for function early in evolution
  Function is not much influenced by environment so no need for adaptation
  => Histone sequences of different organisms are 99-100% identical

- Rapidly evolving genes: e.g. immunity-related genes like MHC genes
  Have to adapt all the time because pathogens also continuously change
  MHC genes of different organisms can have very different sequences
  but they still are homologs

Depends on evolutionary distance between organisms
Nevertheless, a simplistic rule of thumb is often used.

Most protein pairs > 25-30% identical residues over at least 100 amino acids are **structurally and thus functionally similar**.

Also proteins with < 10% identity can have similar structure.

Chances are that the structure is not the same.

http://peds.oxfordjournals.org/content/12/2/85.long
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3820096/
The BLAST/reBLAST strategy to find orthologs

Organism A

Protein A1  => probably orthologs

Protein A2  => probably no orthologs

Organism B

Best hit : protein B1

Ortholog or not ?

Duplications mess up this strategy:

You only find one ortholog while there can be several

OK when you need a fast approach to find pairs of orthologs in many species
Best strategy to identify orthologs

Gather similar sequences from many organisms and align them all

(see section on multiple sequence alignment)

Search for those that have high similarity in the same (functional) regions

Regions with functional importance in a protein sequence alignment should show no substitutions only substitutions of amino acids with similar properties because sequence changes in these regions will change function of protein
Databases of groups of orthologs

Sequence similarity based:


OrthoMCL: [http://www.orthomcl.org/](http://www.orthomcl.org/)

InParanoid: [http://inparanoid51.sbc.su.se/](http://inparanoid51.sbc.su.se/)

eggNOG: [http://eggnogdb.embl.de/](http://eggnogdb.embl.de/) (many organisms)

e.g. Homologs = sequences > 70% identical over > 70% of length

Phylogenetic tree based:

Ensembl (very good for higher eukaryotes)


Take into account evolutionary distances between organisms

Check multiple databases and see how they (dis)agree!

### Table 1: Comparison of selected phylogenomic databases

<table>
<thead>
<tr>
<th>Database</th>
<th>Nb. species and taxonomic range</th>
<th>Nb databases inquired for input data</th>
<th>Homology detection and clustering</th>
<th>Multiple sequence alignment and tree-building</th>
<th>Grouping strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compara (Ensembl 58)</td>
<td>47 chordates and outgroups</td>
<td>1</td>
<td>BlastP, Hclusteringg</td>
<td>M-Coffee, TreeBeSt (NJtree, NJ and ML, species tree)</td>
<td>(i) Phylogenetic trees, (ii) Ortholog groups from species pairs</td>
</tr>
<tr>
<td>eggNOG (release 2)</td>
<td>630 species</td>
<td>4</td>
<td>Blast RBH triangular linkage clustering</td>
<td>Muscle, MAFFT and filters PhyML</td>
<td>Hierarchical groups based on up to six taxonomic levels</td>
</tr>
<tr>
<td>HOGENOM (release 5)</td>
<td>964 species</td>
<td>12</td>
<td>BlastP2 (low complexity filters) single-linkage clustering ≥ 50% similarity, ≥ 80% overlap</td>
<td>Muscle, Gblocks BioNJ, PhyML, FASTTREE and TREEFINDER</td>
<td>Phylogenetic trees</td>
</tr>
<tr>
<td>InParanoid (release 7)</td>
<td>99 eukaryotes and E. coli</td>
<td>22</td>
<td>BLAST (compositional adjustment, SEG) ≥ 50% overlap</td>
<td>Kalign NJ (100 replicates)</td>
<td>Ortholog groups from species pairs</td>
</tr>
<tr>
<td>OMA (May-2010)</td>
<td>1000 species</td>
<td>12</td>
<td>Smith–Waterman with minimum length requirement</td>
<td>–</td>
<td>(i) Pure ortholog groups, (ii) Ortholog groups from species pairs and (iii) Hierarchical groups based on taxonomic nodes</td>
</tr>
<tr>
<td>OrthoDB (release 3)</td>
<td>40 vertebrates 23 arthropods 32 fungi</td>
<td>8</td>
<td>Smith–Waterman, RBH, triangular linkage clustering</td>
<td>–</td>
<td>Hierarchical groups based on a species phylogeny (i) Phylogenetic trees and (ii) Ortholog groups from species pairs</td>
</tr>
<tr>
<td>Panther (release 7)</td>
<td>48 species</td>
<td>13</td>
<td>BlastP, HSP, single-linkage clusters (SLC)</td>
<td>MAFFT GIGA</td>
<td></td>
</tr>
</tbody>
</table>

Which method is the best?

**Phylogeny-based** for the difficult cases
Predict orthology for gene family with many duplications, gene losses
Obtain clear distinction of many-to-many, 1-to-many and 1-to-1 relations
Obtain orthology and paralogy predictions among many species
Know about gene losses

**Blast-based methods** much faster and provide good results.
Choice depends on question:
Find orthologs among more than 2 species? orthoMCL

Exercises:
Sequence similarity searches

Why make pairwise alignments?

Database searches are heuristic: fast but t

Pairwise alignment yields more precise alignments

Some applications of pairwise alignments

Convince yourself that two sequences are homologous

Identify repeats or long insertions or deletions

Compare a gene and its mRNA

Basis of multiple sequence alignment
Methods for making pairwise alignments?

**Computational:** dynamic programming

- Global alignment
- Local alignment

Local alignment will align the best aligning portion of your sequence!

Multiple similar regions =>

**Graphical:** dot plots

All methods can align protein and nucleic acid sequences

But all alignment algorithms are more reliable on proteins / DNA

**EMBOSS tutorial** [http://emboss.sourceforge.net/docs/emboss_tutorial/node3.html](http://emboss.sourceforge.net/docs/emboss_tutorial/node3.html)
Computational tools for making pairwise alignments

<table>
<thead>
<tr>
<th>Tools:</th>
<th>Global</th>
<th>Local</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EMBOSS Needle</td>
<td>EMBOSS Water</td>
</tr>
<tr>
<td>Very long sequences</td>
<td>EMBOSS Stretcher</td>
<td>EMBOSS Matcher</td>
</tr>
</tbody>
</table>

Access via EBI website: [http://www.ebi.ac.uk/Tools/psa](http://www.ebi.ac.uk/Tools/psa)

<table>
<thead>
<tr>
<th>Scoring:</th>
<th>Proteins</th>
<th>Nucleic acids</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores</td>
<td>BLOSUM62</td>
<td>DNAFull: match +5</td>
</tr>
<tr>
<td>Gap opening penalty</td>
<td>-10</td>
<td>mismatch: -4</td>
</tr>
<tr>
<td>Gap extension penalty</td>
<td>-0.5</td>
<td>-10</td>
</tr>
</tbody>
</table>

These are the default settings but you can change them

Graphical method: dot plots rapidly identify similar regions

Parameters:
Scoring matrix: identity matrix
Word size
Threshold: % of characters in a word that have to be identical to other word

Create a visual representation of the sequence similarity of two sequences
Very fast
Convenient for large molecules e.g. chromosomes
How to make a dot plot for visualizing sequence similarity?

One sequence is placed on the vertical axis, other on the horizontal axis.

If residue on Y-axis and X-axis or if 2 residues in 3 letter words are identical: dot.

- **words size = 1**
- **words size = 3 ; threshold = 2**
Dot plots generate patterns that can be interpreted

- Repeat
- Insertion in Sequence B
- Insertion in Sequence A
- Complex repeat
- Palindrome

When to use which method for pairwise alignment?

Dot plots:  Get general idea of similarity (What do you look for?)
Discover repeats and inversions
Discover long insertions and deletions

Local alignment  Comparing sequences with partial homology
Making high quality alignments

Global alignment  Comparing sequences over their entire length
Discover long insertions and deletions

Exercises:
Pairwise sequence alignment
How are multiple sequence alignments generated?

Heuristic: progressive alignment

1. Make all pairwise global alignments by dynamic programming
2. Construct a guide tree based on pairwise similarity scores
3. Align the two most similar sequences
4. Merge pairwise alignments following the structure of the guide tree
Step 1: Make all pairwise alignments using EMBOSS Needle

3 sequences:

MTDNLIEPPRDAWTPTMFISCR
MTNDLIQPHRDWAWTPTGTEFNSCR
MTDELDEPEMDANTESMDIMLEP

# Identity: 16/24 (66.7%)
# Similarity: 19/24 (79.2%)
# Gaps: 2/24 (8.3%)
# Score: 69.0

1 MTDNLIEPPRD-AWTP-TMFISCR 22
   ||::||:|.| | |||| |.|.|.||
2 MTNDLIQPHRDWAWTPTGTEFNSCR 24

# Identity: 11/23 (47.8%)
# Similarity: 12/23 (52.2%)
# Gaps: 1/23 (4.3%)
# Score: 42.0

1 MTDNLIEPPRDAWTPTMFISCR- 22
   |||.|.|.|.|.|.|.|.|.|.|.|...  
3 MTDELDEPEMDANTESMDIMLEP 23

# Identity: 7/27 (25.9%)
# Similarity: 10/27 (37.0%)
# Gaps: 7/27 (25.9%)
# Score: 19.5

2 MTNDLIQPHRDWAWTPTGTEFNSCR--- 24
   ||::|.:|..| ..||......
3 MTDELDEPEMD----ANTESMDIMLEP 23

Which sequences do you start with?
Step 2: Create similarity matrix and guide tree

Scores depend on scoring matrix and gap penalties

Although some tools use different approaches to calculate the scores

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>69</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>19.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>19.5</td>
<td></td>
</tr>
</tbody>
</table>

Seq1 0.01779
Seq2 0.25494
Seq3 0.52767
Step 3: Start with two most similar sequences

1  MTDNLIEPPRD–AWTP–TMFISCR  22
   ||::||:||.||  ||||  |.|.|||
2  MTNDLIQPHRDWAWTPEFNSCR  24
Step 4a: Merge pairwise alignments following guide tree

1  MTDNLIEPPRD-AWTP-TMFISCR  22
   ||:::||:|.| | | || | | | .| .| .|| |
2  MTNDLIQPHRDWAWTPGTEFNSCR  24

1  MTDNLIEPPRD-AWTP-TMFISCR-  22
   |||.|.||||..| |.|  :|.|... |
3  MTDEDELDEPEMD-ANTE-SMDIMLEP  23

Merging is mainly done by adding gaps

No guarantee that tree is optimal!

If error is made in early stage, it is propagated throughout the full alignment

Quality MSA is highly dependent on quality of guide tree
Iterative variant to improve MSA

All pairwise alignments
Similarity scores
Similarity matrix

Replace two sequences by model
Model of this alignment
Alignment of the two most similar sequences

Guide tree

! This is not a phylogenetic tree!

Iterative: repeatedly recalculate pairwise alignments + tree + resulting MSA
Calculate model of alignment of sequences

1  MTDNLIEPPRD-AWTP-TMFISCR
   ||::||:|.|| |||| |.|.|||

2  MTNDLIQPHRDWAWTGTEFNSCR

Model: MTBBLIZPXRDWAWTGTXFXSCR  = consensus

B = N or D
Z = E or Q
X = any amino acid

simplistic: Some algorithms use better models for alignments (see later)
Extra step to improve alignment even more

Select random sequence
Make model of alignment of all others
Realign chosen sequence to alignment model
Repeat until nothing changes anymore
Different algorithms for making MSA

ClustalW

Current standard
Can use iterative alignment

ClustalOmega

New alternative for ClustalW
EBI claims it’s **faster** and more accurate than ClustalW
Can use iterative alignment AND:
Letters in the sequences are replaced by numbers to speed up guide tree calculation
Instead of a consensus sequence a more accurate HMM is used (see later)

E[ducational paper on MSA](http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.0030123)
Preferred algorithms for making MSA

**Muscle**
Uses iterative alignment AND
Similarity scores based on counting the number of seeds between pairs of sequences
Much faster but trees are less accurate

**MAFFT**
Uses iterative alignment AND
Pairwise alignments generated by searching for seeds and extending from there

Multiple sequence alignment formats

Columns (~positions in the alignment)

<table>
<thead>
<tr>
<th>Sequences (~species)</th>
<th>GARFIELDTHELASTFAT-CAT</th>
<th>GARFIELDTHEFASTCAT----</th>
<th>GARFIELDTHEVERYFASTCAT</th>
<th>THEFAT------CAT</th>
<th>GARFIELDTHEVASTCAT----</th>
</tr>
</thead>
</table>

Delineates conserved positions
Can have many appearances depending on MSA tool

* positions where amino acid is the same in all sequences
  : positions with amino acids with strongly similar properties in all sequences
    score > 0.5 in PAM250 matrix
  . positions with amino acids with weakly similar properties in all sequences
    0 < score =< 0.5 in PAM250 matrix
Concluding remarks

Algorithms all behave differently although almost all based on same techniques

If you have hundreds of sequences to align, you want to do it fast:

MAFFT: best quality - bit slower

Clustal Omega: very fast - good quality

Most popular is CLUSTALW based on progressive alignment but it is slow

Try different algorithms and compare the results!

Often you need to change the alignments manually to improve them
Alignment editors to view and edit alignments

Each row = one protein

Colours can reflect conservation

Clean up the alignment before creating a phylogenetic tree by removing

- Sequences that are different over long regions in the alignment
- Uninformative positions: only one sequence has a residue, all others have a gap

https://www.reddit.com/r/bioinformatics/comments/305se3/what_are_some_of_the_best_multiple_alignment/
Why are MSA important?

Basis of **phylogenetic trees**

Detect **conserved regions** (that have not/little changed during evolution)

-> functionally important regions: protein **motifs** and **domains**

-> can be used to find more distant homologs

**Predict protein structure**
Domains are large conserved parts of a protein

Part of protein structure that can exist, fold, function and evolve independently

Most proteins contain multiple domains

Each domain of a protein folds into a stable structure

Each domain has a specific biological function

**Long**: 20-100s amino acids
Motifs are small conserved parts of a domain

Part of the sequence that recurs in all sequences of the alignment
Most conserved amino acids of the domain
Performs the biological function -> active site
Short, not necessarily contiguous
Pyruvate kinase: a protein with three domains

Domain that binds to sugars
Phylogenetic trees represent evolutionary relations.
Components of a phylogenetic tree

- Branches
- Nodes
- Leaves
- Root
- Evolutionary Time
Phylogram versus cladogram

Branch lengths are meaningless

Dendrograms

Cladograms

TAXON A
TAXON B
TAXON C

TREE1

TAXON A
TAXON B
TAXON C

TREE2

Branch lengths reflect amount of evolutionary divergence

Phylograms

TAXON A
TAXON B
TAXON C

TREE3

TAXON A
TAXON B
TAXON C

TREE4
Four methods to create phylogenetic trees

1. **Bayesian methods:** Mathematics best suited for phylogeny

2. **Maximum likelihood:** Model reflects evolution better than NJ
   Investigate all possible trees for the most likely

3. **Neighbour joining:** Model does not reflect evolution

4. **UPGMA:** Similar to NJ but even more simple
   Produces a single rooted tree
How phylogenetic trees could be calculated

1. Create all possible tree structures

2. Select the most likely one given the data (= MSA):
   For each tree calculate a score expressing how well the data fits the tree structure
   Retain the tree with the highest score

3. Calculate the lengths of the branches in the retained tree:
   For each branch find all trees that contain that branch
   Combine the scores of all these trees
   Branch length is inversely proportionate to combined score
Models to score how well the tree fits the data

HMM (see later) of sequence evolution

Describes rates at which one aa / nucleotide replaces another during evolution

Jukes & Cantor
\[ \text{freq}(A) = \text{freq}(T) = \text{freq}(C) = \text{freq}(G) \]

mutation rates are equal

Kimura
\[ \text{freq}(A) = \text{freq}(T) = \text{freq}(C) = \text{freq}(G) \]

distinguishes transitions:
- \( A \leftrightarrow G \)  purine to purine
- \( C \leftrightarrow T \)  pyrimidine to pyrimidine

transversions: purine \( \leftrightarrow \) pyrimidine

These two are used for nucleotide and protein sequences

But more advanced models exist (e.g. PAM)
However the number of possible trees is often too high

10 sequences -> 2 million possible trees

How to select the most likely one given the data then?

1. Take an initial possible tree and calculate its score (= score A)

2. Go to another not too different tree and calculate its score (= score B)
   - score B > score A => take tree B as initial tree
   - score B < score A => in most cases: take tree A as initial tree
     sometimes: take tree B as initial tree to avoid local optima

Take a next no too different tree and repeat many times...

3. Identify the highest scoring tree
Four methods to create phylogenetic trees

1. Bayesian methods: Mathematics best suited for phylogeny

2. Maximum likelihood: Model reflects evolution better than NJ
   Investigate all possible trees for the most likely

3. Neighbour joining: Model does not reflect evolution

4. UPGMA: Similar to NJ but even more simple
   Produces a single rooted tree
UPGMA: fast but unreliable

Calculate distance matrix using pairwise alignments

Select shortest distance

Divide equally over both branches

Replace B and C by BC

Recalculate distance matrix

Select shortest distance

Add to tree

...
Four methods to create phylogenetic trees

1. Bayesian methods: Mathematics best suited for phylogeny
2. Maximum likelihood: Model reflects evolution better than NJ
   Investigate all possible trees for the most likely
3. Neighbour joining: Model does not reflect evolution
4. UPGMA: Similar to NJ but even more simple
   Produces a single rooted tree
Neighbour Joining

Similar to UPGMA

But **different way to calculate branch lengths**

Smallest elements? AB and CD

Calculate branch lengths:

\[
\frac{(56-38)}{(4-2)} = 9
\]

\[
A-AB: \frac{(13+9)}{2} = 11
\]

\[
B-AB: \frac{(13-9)}{2} = 2
\]

\[
\frac{(48-46)}{(4-2)} = 1
\]

\[
C-CD: \frac{(13-1)}{2} = 6
\]

\[
D-CD: \frac{(13+1)}{2} = 7
\]
Non Bayesian methods are followed by bootstrapping

Tree construction is repeated 100-1000 times

Each time making a small change to the data e.g. remove **column(s)** from MSA

You count the number of times two sequences are on the same branch

This number reflects the confidence of the branch

! You do not remove sequences (rows) you remove columns !
Bootstrapping results are displayed on top of the branches.

What does this 95 mean?

Values > 70 are considered trustworthy.

Trees of distant species are interpreted differently as trees of close species.
Tips for creating/checking phylogenetic trees

Different methods give different trees

However, if they completely disagree it means you don’t have enough data

Use NJ for exploratory research and Bayesian/ML for publishing

Use protein sequences!

Include 1 out-group

Checks:

Is the outgroup on the outside of the tree?

Do the placing of the organisms make sense?

Are the gene families grouped together?
Consensus of ML + Bayes for publication

Bootstrap
Bayes posteriors

http://www.nature.com/nature/journal/v438/n7069/full/nature04338.html
How to use multiple proteins for a phylogenetic tree

Two methods:

• Alignments are concatenated into a superalignment to generate the tree
• Trees are inferred separately from each protein and a consensus tree is created

A lot of discussion on which is the best

You can do concatenation in MEGA and construct the tree in MrBayes:

  Align each sequence separately with Muscle/MAFFT in MEGA
  Save the alignment of each protein
  Import the alignment file of each protein using the "Analyze" button
  Transform to "meg" format
  Concatenate the gene matrices

You can make the individual alignments in Ugene

Do concatenation in Sequence Matrix: http://www.ggvaidya.com/taxondna/
Results of BLAST can be displayed in multiple alignment

NCBI have written their own MSA tool: COBALT


Tree is based on pairwise alignments not on a MSA

If sequence matches query in different regions only the highest scoring one is shown

Not a real phylogenetic tree, just to recognize unusual sequences!
MSA in Uniprot are based on ClustalOmega

You can also make MSA in Uniprot: [http://www.uniprot.org/align/](http://www.uniprot.org/align/)

The tree that is shown here is a guide tree not a phylogenetic tree!

Comparative genomics in eukaryotes training: on demand -> bits@vib.be

Exercises:
Multiple sequence alignment: toy example
Simple and complex ways to represent motifs

Motif/domain can be represented by

1. Regular expression
2. Frequency matrix
3. Hidden Markov Model

Dr Motifs blog  http://drmotifs.genouest.org/category/motifs-for-dummies/
Regular expression is the simplest way to represent motifs

A “flexible” word used to "match" slightly different variations on the word

e.g. “%car%" matches all words containing “car” such as car, cartoon or bicarbonate

You need a syntax to create regular expressions

<table>
<thead>
<tr>
<th>position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATPKAE</td>
<td>[AKT]</td>
<td>[KLT]</td>
<td>P</td>
<td>[AK]</td>
<td>[APT]</td>
<td>[ADEKT-]</td>
</tr>
<tr>
<td>KKPKAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKPKAK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TKPKPA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKPKT-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKPAAK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLPKAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKPKAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consensus: **AKPKAA**

For every position the most frequently occurring amino acid
Use regular expression to see if new sequences match the pattern

|----------|---------|---------|-----|--------|---------|-------------|

? Does this sequence match: A K P K T E

? And this sequence: K K P E T E

? And what about this one: T L P A T E

Very straightforward but not very accurate:

In a position all observed amino acids have same importance regardless of their frequency
Frequency matrices include the chance of observing an amino acid

For each position of a motif: a list of all amino acids with their frequency

Not all observed amino acids in a position have equal importance

More sensitive

<table>
<thead>
<tr>
<th>position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATPKAE</td>
<td>A</td>
<td>5/8</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>6/8</td>
</tr>
<tr>
<td>KKPKAA</td>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AKPKAK</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TKPKPA</td>
<td>K</td>
<td>2/8</td>
<td>6/8</td>
<td>0</td>
<td>7/8</td>
<td>0</td>
</tr>
<tr>
<td>AKPKT</td>
<td>L</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AKPAAK</td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td>KLPKAD</td>
<td>T</td>
<td>1/8</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
</tr>
</tbody>
</table>

Consensus: **AKPKAA**

Frequencies are transformed into scores: Position Score Matrix

<table>
<thead>
<tr>
<th>position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATPKA_E</td>
<td>A</td>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>KKPKA_A</td>
<td>D</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>AKPKA_K</td>
<td>E</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>TKPKPA</td>
<td>K</td>
<td>1</td>
<td>3</td>
<td>-2</td>
<td>3</td>
<td>-2</td>
</tr>
<tr>
<td>AKPKT_</td>
<td>L</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>AKPAAK</td>
<td>P</td>
<td>-2</td>
<td>-2</td>
<td>3</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>KLKPAD</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>AKPKAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Consensus: **AKPKAA**

Scores can be used to measure how good a sequence matches a frequency matrix

? Query: AKPKTE Score = 2 + 3 + 3 + 3 + 0 + 0 = 11

? Query: KKPETE Score =

? Query: TLPATE Score =
Hidden Markov Model are adept at representing insertions /deletions

Advanced statistical models of MSAs

Graphs containing probabilities reflecting:

- which amino acids are likely to occur in a certain position
- how likely insertion or deletion occurs in a certain position

Nature’s primer on HMMs  http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html
Building a HMM based on a multiple sequence alignment

Simple nucleotide example not taking into account gaps

States

**Match**: probability for each nucleotide

Numbers on arrows: probability of going from one state to another

No gaps so probability of going from one Match state to the next = 1

Nature’s primer on HMMs  [http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html](http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html)
Building a HMM based on a multiple sequence alignment

Simple nucleotide example taking into account insertions

In 10% of sequences in the alignment an extra nucleotide is inserted in this position

In 90% of sequences in the alignment no insertion

1 column in the alignment

States

Insert: probability of inserting between those positions is shown for each nucleotide

Match: probability for each nucleotide

Nature’s primer on HMMs  http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html
Building a HMM based on a multiple sequence alignment

Simple nucleotide example taking into account gaps and insertions

In 20% of the sequences in the alignment there is a gap on this position

**States**

**Insert**: probability of inserting between those positions is shown for each nucleotide

**Match**: probability for each nucleotide

**Delete**: probability of a gap on this position

*Dr Motifs blog*  [http://drmotifs.genouest.org/2010/07/hiden-markov-models-hmm/]
HMM are just text files containing scores

For each position + for each amino acid:

<table>
<thead>
<tr>
<th>HMM</th>
<th>A</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>P</td>
<td>Q</td>
<td>R</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>V</td>
<td>W</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m→m</td>
<td>2.12153</td>
<td>4.81395</td>
<td>3.16315</td>
<td>2.61552</td>
<td>3.77980</td>
<td>3.20233</td>
<td></td>
</tr>
<tr>
<td>m→i</td>
<td>3.77136</td>
<td>3.26437</td>
<td>1.94471</td>
<td>2.88123</td>
<td>3.93566</td>
<td>3.17937</td>
<td>2.96027</td>
</tr>
<tr>
<td>m→d</td>
<td>2.92525</td>
<td>2.58806</td>
<td>2.76214</td>
<td>2.84783</td>
<td>5.39514</td>
<td>3.90746</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.68604</td>
<td>4.42260</td>
<td>2.77554</td>
<td>2.73006</td>
<td>3.46388</td>
<td>2.40547</td>
<td></td>
</tr>
<tr>
<td>i→m</td>
<td>3.72529</td>
<td>3.29300</td>
<td>2.67775</td>
<td>2.69389</td>
<td>4.24264</td>
<td>2.90321</td>
<td>2.73774</td>
</tr>
<tr>
<td>i→i</td>
<td>2.89835</td>
<td>2.37886</td>
<td>2.77501</td>
<td>2.98518</td>
<td>4.58511</td>
<td>3.61538</td>
<td></td>
</tr>
<tr>
<td>d→m</td>
<td>0.59577</td>
<td>1.16419</td>
<td>1.99007</td>
<td>1.52955</td>
<td>0.24415</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>d→d</td>
<td>Match scores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>position</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>m→m</td>
<td>2.30925</td>
</tr>
<tr>
<td>m→i</td>
<td>3.84927</td>
</tr>
<tr>
<td>m→d</td>
<td>3.16346</td>
</tr>
<tr>
<td>i→m</td>
<td>2.68618</td>
</tr>
<tr>
<td>i→i</td>
<td>3.72494</td>
</tr>
<tr>
<td>d→m</td>
<td>2.89801</td>
</tr>
<tr>
<td>d→d</td>
<td>State transition scores</td>
</tr>
<tr>
<td>0.02181</td>
<td>4.23210</td>
</tr>
<tr>
<td>0.72197</td>
<td>4.18920</td>
</tr>
</tbody>
</table>
Why HMMER is better than BLAST for defining homologs

HMMER: set of tools for similarity searches using HMMs

http://www.ebi.ac.uk/Tools/hmmer/search/hmmsearch

BLAST takes as an input a single sequence

- it will focus the search on the complete sequence

HMMER takes as an input a complete MSA consisting of multiple sequences

- it will focus the search on the conserved parts of the sequences
Some protein BLAST variants search based on motifs/domains

**PSI-BLAST** finds sequences similar to a PSM of a MSA

Unfortunately you have little control on how the MSA is created

**PHI-BLAST** protein BLAST with matching of a motif

User provides a regular expression which all BLAST results should satisfy


Find all sequences that contain the motif and are similar in the vicinity of the motif

**DELTAS-BLAST** to identify distantly related proteins

Find a PSM from NCBIs domain database that matches the query sequence

Searches a sequence database with this PSM
Databases of models of protein domains

Interpro: http://www.ebi.ac.uk/interpro/

Pfam: http://xfam.org/

Prosite: http://expasy.org/prosite/

SMART: http://smart.embl-heidelberg.de/
PROSITE is a high quality database of protein motifs

PROSITE contains info on protein **domains** and **motifs**

**Best curated** and annotated database of protein **motifs**

**Patterns:** short **motifs** -> regular expressions

**Profiles:** longer **domains** -> PSMs

Kringle domain signature.

[FY]-C-[RH]-[NS]-x(7,8)-[WY]-C.

Kringle domain profile.

```
/GENERAL_SPEC: ALPHABET='ABCDEFGHIJKLMNOPQRSTUVWXYZ'; LENGTH=79;
/DISJOINT: DEFINITION=PROTECT; N1=6; N2=74;
/NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=.7529; R2=.00952475; TEXT='-LogE';
/CUT_OFF: LEVEL=0; SCORE=813; N_SCORE=8.5; MODE=1; TEXT='!';
/CUT_OFF: LEVEL=-1; SCORE=603; N_SCORE=6.5; MODE=1; TEXT='?';
/DEFAULT: D=-20; I=-20; B1=-50; E1=-50; MI=-105; MD=-105; IM=-105; DM=-105;
```

/GENERAL_SPEC: ALPHABET='ABCDEFGHIJKLMNOPQRSTUVWXYZ'; LENGTH=79;
/DISJOINT: DEFINITION=PROTECT; N1=6; N2=74;
/NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=.7529; R2=.00952475; TEXT='-LogE';
/CUT_OFF: LEVEL=0; SCORE=813; N_SCORE=8.5; MODE=1; TEXT='!';
/CUT_OFF: LEVEL=-1; SCORE=603; N_SCORE=6.5; MODE=1; TEXT='?';
/DEFAULT: D=-20; I=-20; B1=-50; E1=-50; MI=-105; MD=-105; IM=-105; DM=-105;

```
I:  B1=0; BI=-105; BD=-105;
M:  SY='D'; M=-15, 29,-30, 44, 37,-36,-15, 1,-34, 5,-25,-24, 10, -6, 13, -4, 0,-10,-30,-34,-19, 25;
M:  SY='C'; M=-10,-20,120,-30,-30,-20,-30,-30,-30,-20,-20,-20,-20,-40,-30,-30,-10,-10,-10,-50,-30,-30;
M:  SY='Y'; M=-11,-21,-25,-25,-20,16,-27, -1, 10,-12, 9, 15,20,-25,-12,-12,-18, -9, 3, 1, 31,-18;
M:  SY='H'; M=-13,-8,-26,-9, 0,-9,-23,16,-13,-2,-9, -1,-5,-15, 2, 2,-8,-6,-13,-19, 4,-1;
M:  SY='G'; M=-4,-5,-11, -4,-14,-29,45,-17,-38,-18,-28,-21, 0,-21,-17,-19, -1,-17,-27,-26,-28,-16;
M:  SY='N'; M=-9,19,-22, 11, 2,-22,-10, 1,-19, 4,-22,-14,26,-17, 5, 5, 5, 0,-21,-32,-14, 3;
```
PROSITE comes with tools for searching the database


**Sequence search:**
You can check whether your query sequence contains known protein motifs

**Motif search:**
Search the Prosite database for sequences that contain a motif (regular expression)
Example: `[DE](2)-H-S-{P}-x(2)-P-x(2,4)-C>`

Regular expression syntax is specific, but not difficult:
Pfam is a high quality database of protein domains

Pfam contains info on protein **domains**

**HMMs** to identify them

**Best curated** and annotated database of protein **domains**

Use Pfam to find homologs for your favourite protein

Use Prosite to select those that contain the active site (the functional ones)
SMART is **visually** the most attractive domain database
Interpro is a composite protein domain database

UniProt uses Interpro for annotation

Online training on using Interpro  http://www.ebi.ac.uk/interpro/training.html
About InterproScan  http://www.ebi.ac.uk/interpro/interproscan.html
Interpro user manual  http://www.ebi.ac.uk/interpro/user_manual.html
List of databases where Interpro gets its data from  http://www.ebi.ac.uk/interpro/about#about_08
Summary of scoring how similar sequences are

You can calculate similarity scores to express how similar two sequences are.

For proteins, scoring matrices provide a means to determine similarity in an objective way. These scoring matrices are derived from analysis of multiple alignments of related sequences:

- PAM matrices
- BLOSUM matrices

The scoring matrices capture the similarity in properties between amino acids.

For nucleotide sequences, you simply use match/mismatch scores.

A matrix does not capture insertions and deletions: gap penalties are used to deal with this.

There is a link between similarity and homology, but they are not the same.

Homology has biological implications.
Summary of sequence alignment methods

Similarity searches are based on sequence alignments

You can make global and local sequence alignments

You can use different approaches for making sequence alignments
  
dynamic programming
  
heuristics
  
combinations

Dynamic programming calculates solutions to recurring subproblems once and stores them

Dynamic programming is very accurate but slow

Heuristic rely on experience (gut feeling) to make assumptions to speed up calculations

In most cases a combination of heuristics and dynamic programming is used
Summary of pairwise and multiple sequence alignments

**Pairwise sequence alignments** are based on dynamic programming

You can use global and local alignments

EMBOSS provides algorithms for both global and local pairwise alignments

Dot plots provide a graphical method to rapidly identify regions with similar sequence

Dot plots generate typical patterns which can be interpreted

**Multiple sequence alignment** is based on heuristic: progressive alignment

Different tools generate different alignments

Try and compare results from multiple tools ...

... and still you will need to edit your alignments manually

Many tools for doing so exist
Summary on functional classification

Functional classification is the holy grail of biology

Basis of functional classification is a protein multiple sequence alignment

Highly conserved regions in the alignment are conserved because of functionality

Highly conserved functional regions in protein sequences are called motifs

Motifs are often associated with domains (3D functional parts)

Three ways to represent motifs

- regular expressions
- position score matrices
- hidden Markov models

You can use these representations to search for protein motifs / domains

Domain databases
Applications of MSA

Multiple sequence alignment is the basis for
- Phylogenetic trees
- Defining homologs
  Databases of orthologs
- Defining protein motifs and domains
  There is a link between sequence – structure – function
  Functionally important parts of proteins are conserved: motifs + domains
  Different ways to represent motifs / domains
  Databases of motifs / domains

Exercises:
Protein motifs and domains: Exercises with *
Group exercises