Outline for today

1. Homology and pairwise alignment
2. BLAST
3. Multiple sequence alignment
4. Phylogeny and evolution

Learning objectives: homology & alignment

1. You should know the definitions of homologs, orthologs, and paralogs
2. You should know how to determine whether two genes (or proteins) are homologous
3. You should know what a scoring matrix is
4. You should know how alignments are performed
5. You should know how to align two sequences using the BLAST tool at NCBI
Pairwise sequence alignment is the most fundamental operation of bioinformatics

- It is used to decide if two proteins (or genes) are related structurally or functionally
- It is used to identify domains or motifs that are shared between proteins
- It is the basis of BLAST searching (next topic)
- It is used in the analysis of genomes

Pairwise alignment: protein sequences can be more informative than DNA

- Protein is more informative (20 vs 4 characters); many amino acids share related biophysical properties
- Codons are degenerate: changes in the third position often do not alter the amino acid that is specified
- Protein sequences offer a longer "look-back" time
- DNA sequences can be translated into protein, and then used in pairwise alignments
Find BLAST from the home page of NCBI and select protein BLAST...

Choose align two or more sequences...

Enter the two sequences (as accession numbers or in the fasta format) and click BLAST. Optionally select “Algorithm parameters” and note the matrix option.
Pairwise alignment result of human beta globin and myoglobin:

The score is the sum of match, mismatch, gap creation, and gap extension scores.

<table>
<thead>
<tr>
<th>Score</th>
<th>Expect</th>
<th>Method</th>
<th>Composition-based state</th>
<th>Identities</th>
<th>Positives</th>
<th>Gaps</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.1</td>
<td>0.015</td>
<td>Nucleotide-Composition-based</td>
<td>93</td>
<td>12/14 (86)</td>
<td>14/28 (50)</td>
<td>2/16 (14)</td>
</tr>
</tbody>
</table>

Query 12  
Subject 11  

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

These scores come from a “scoring matrix”!
Gaps

Score = 18.1 bits (23). Expect = 0.015. Method: Composition-based stats.
Identities = 12/24 (50%), positives = 12/24 (50%), gaps = 2/24 (8%)

Query: I TRPGRRLQG--SYGEEBill 40
Query: I TRPGRRLQG--SYGEEBill 40
match  6 12 5 6 5 13 1 sum of matches: +30
mismatch -1 -1 -1 -2 -4 0 sum of mismatches: -12
gap open -1 -1 -1 -1 -1 -1 -1 total gap score: 0

First gap position scores -11
Second gap position scores -1
Gap creation tends to have a large negative score;
Gap extension involves a small penalty

Definitions

Pairwise alignment
The process of lining up two sequences
to achieve maximal levels of identity
(and conservation, in the case of amino acid sequences)
for the purpose of assessing the degree of similarity
and the possibility of homology.

Definitions

Homology
Similarity attributed to descent from a common ancestor.

Identity
The extent to which two (nucleotide or amino acid)
sequences are invariant.
Definitions: two types of homology

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

Paralogs
Homologous sequences within a single species that arose by gene duplication.

Orthologs: members of a gene (protein) family in various organisms. This tree shows globin orthologs.

Paralogs: members of a gene (protein) family within a species. This tree shows human globin paralogs.

You can view these sequences at www.bioinfbook.org (document 3.1)
Orthologs and paralogs are often viewed in a single tree

![Diagram showing relationships between homologs, orthologs, and paralogs.](Source: NCBI)

**Definitions**

**Similarity**
The extent to which nucleotide or protein sequences are related. It is based upon identity plus conservation.

**Identity**
The extent to which two sequences are invariant.

**Conservation**
Changes at a specific position of an amino acid or (less commonly, DNA) sequence that preserve the physico-chemical properties of the original residue.

**Calculation of an alignment score**

![Alignment example](Source: http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Alignment_Scores2.html)

\[ S = \sum \text{identites, mismatches} - \sum \text{gap penalties} \]

\[ \text{Score} = \text{Max}(S) \]
We will first consider the global alignment algorithm of Needleman and Wunsch (1970).

We will then discuss the local alignment algorithm of Smith and Waterman (1981).

Finally, we will consider BLAST, a heuristic version of Smith-Waterman. BLAST is faster but less rigorous.

Two kinds of sequence alignment: global and local

We will first consider the global alignment algorithm of Needleman and Wunsch (1970).

We will then discuss the local alignment algorithm of Smith and Waterman (1981).

Finally, we will consider BLAST, a heuristic version of Smith-Waterman. BLAST is faster but less rigorous.

Global alignment with the algorithm of Needleman and Wunsch (1970)

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

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

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

• N-W is guaranteed to find optimal alignment(s)

Three steps to global alignment with the Needleman-Wunsch algorithm

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

Start Needleman-Wunsch with an identity matrix

Start Needleman-Wunsch with an identity matrix
(or, as here, use values from a scoring matrix)
(a) Fill in the matrix using “dynamic programming”

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>M</th>
<th>D</th>
<th>T</th>
<th>P</th>
<th>L</th>
<th>N</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>-16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>-18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) 

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

Score (this example) = +1 (match) 
-2 (mismatch) 
-2 (gap penalty)

(c) Fill in the matrix using “dynamic programming”

\[
F(i, j) = \begin{cases} 
F(i, j-1) - \text{gap penalty} \\
F(i-1, j) - \text{gap penalty} \\
F(i-1, j-1) + s(x_i, y_j) 
\end{cases}
\]

Page 78
Fill in the matrix using “dynamic programming”

(d) Sequence 2
   F   M
   0   -2  -4
   -2  -1  -3
   -4

Sequence 1
   F   K
   -2  -4

Page 78

Fill in the matrix using “dynamic programming”

(e) Sequence 2
   F   M
   0  -2  -4  -6
   -2  -1  -3  -5
   -4  -3  -5  -7
   -6

Sequence 1
   F   K
   -2  -4

Page 78

Fill in the matrix using “dynamic programming”

(f) Sequence 2
   F   M   D   T   P   L   N   E
   0  -2  -4  -6  -8  -10  -12  -14  -16
   F  -2  -1  -3  -5  -7  -9  -11  -13  -15
   K  -4  -3  -5  -7  -9  -11  -13  -15  -17
   H  -6  -5  -7  -9  -11  -13  -15  -17  -19
   M  -8  -7  -9  -11  -13  -15  -17  -19  -21
   E  -10  -9  -11  -13  -15  -17  -19  -21  -23
   D  -12  -11  -13  -15  -17  -19  -21  -23  -25
   N  -18  -17  -19  -21  -23  -25  -27  -29  -31

Sequence 1
   F   M   D   T   P   L   N   E
   0  -2  -4  -6  -8  -10  -12

Page 78
Traceback to find the optimal (best) pairwise alignment

(a) Sequence 2

(b) Sequence 2

Smith-Waterman allows local alignment (internal terminations)

Rapid, heuristic versions of Smith-Waterman: BLAST

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

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

BLAST provides a rapid alternative to S-W, although it’s not as accurate.
Outline for today

1. Homology and pairwise alignment
2. BLAST
3. Multiple sequence alignment
4. Phylogeny and evolution

Learning objectives: BLAST

1. You should know what the five basic BLAST programs are
2. You should be able to perform a BLAST search
3. You should be able to interpret the results of a BLAST search

BLAST

BLAST (Basic Local Alignment Search Tool) allows rapid sequence comparison of a query sequence against a database.

The BLAST algorithm is fast, accurate, and web-accessible.
Why use BLAST?

BLAST searching is fundamental to understanding the relatedness of any favorite query sequence to other known proteins or DNA sequences.

Applications include
- identifying orthologs and paralogs
- discovering new genes or proteins
- discovering variants of genes or proteins
- investigating expressed sequence tags (ESTs)
- exploring protein structure and function

Four components to a BLAST search

1. Choose the sequence (query)
2. Select the BLAST program
3. Choose the database to search
4. Choose optional parameters

Then click “BLAST”

Step 1: Choose your sequence

Sequence can be input in FASTA format or as accession number
Step 2: Choose the BLAST program

<table>
<thead>
<tr>
<th>Program</th>
<th>Input</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastn</td>
<td>DNA</td>
<td>DNA</td>
</tr>
<tr>
<td>blastp</td>
<td>protein</td>
<td>protein</td>
</tr>
<tr>
<td>blastx</td>
<td>DNA</td>
<td>DNA</td>
</tr>
<tr>
<td>tblastn</td>
<td>protein</td>
<td>DNA</td>
</tr>
<tr>
<td>tblastx</td>
<td>DNA</td>
<td>DNA</td>
</tr>
</tbody>
</table>

Fig. 4.3 page 91
Step 3: choose the database

nr = non-redundant (most general database)
dbest = database of expressed sequence tags
dbsts = database of sequence tag sites
gss = genomic survey sequences
htgs = high throughput genomic sequence

Step 4a: Select optional search parameters

organism
Entrez!
algorithm

Filter, mask
Scoring matrix
Word size
Expect
How a BLAST search works

"The central idea of the BLAST algorithm is to confine attention to segment pairs that contain a word pair of length \( w \) with a score of at least \( T \)."

Altschul et al. (1990)

How the original BLAST algorithm works: three phases

Phase 1: compile a list of word pairs (\( w=3 \)) above threshold \( T \)

Example: for a human RBP query ...

…FSGTWYA… (query word is in yellow)

A list of words (\( w=3 \)) is:

\[
\text{FSG SOT GTW TWY WYA} \\
\text{YSG TGT ATW SNY WPA} \\
\text{FTG SVT GSW TWF WYS}
\]

Phase 1: compile a list of words (\( w=3 \))

<table>
<thead>
<tr>
<th>neighborhood</th>
<th>GTW 6,5,11 22</th>
</tr>
</thead>
<tbody>
<tr>
<td>word hits</td>
<td>GSW 6,1,11 18</td>
</tr>
<tr>
<td>&gt; threshold</td>
<td>ATW 0,5,11 16</td>
</tr>
<tr>
<td>( (T=11) )</td>
<td>NTW 0,5,11 16</td>
</tr>
<tr>
<td>neighborhood</td>
<td>GTY 6,5,2 13</td>
</tr>
<tr>
<td>word hits</td>
<td>GNW 10</td>
</tr>
<tr>
<td>&lt; below threshold</td>
<td>GAW 9</td>
</tr>
</tbody>
</table>

Fig. 4.13

How a BLAST search works: 3 phases

Phase 2:
Scan the database for entries that match the compiled list.
This is fast and relatively easy.

Fig. 4.13

How a BLAST search works: 3 phases

Phase 3: when you manage to find a hit
(i.e. a match between a "word" and a database entry), extend the hit in either direction.

Keep track of the score (use a scoring matrix)

Stop when the score drops below some cutoff.

KENFDKARFSETWAMYAKKDPEG 50 RBP (query)
MKGLDIQKVAETWYSLAMAASD 44 lactoglobulin (hit)

extend Hit! extend

BLAST-related tools for genomic DNA

Recently developed tools include:

• MegaBLAST at NCBI.

• BLAT (BLAST-like alignment tool). BLAT parses an entire genomic DNA database into words (11mers), then searches them against a query. Thus it is a mirror image of the BLAST strategy. See http://genome.ucsc.edu

• SSAHA at Ensembl uses a similar strategy as BLAT. See http://www.ensembl.org
To access BLAT, visit http://genome.ucsc.edu

BLAT on DNA is designed to quickly find sequences of 95% and greater similarity of length 40 bases or more. It may miss more divergent or shorter sequence alignments. It will find perfect sequence matches of 33 bases, and sometimes find them down to 20 bases. BLAT on proteins finds sequences of 80% and greater similarity of length 20 amino acids or more. In practice DNA BLAT works well on primates, and protein blat on land vertebrates. --BLAT website

BLAT output includes browser and other formats. Try a beta globin protein search to view homologs.

How to interpret a BLAST search: expect value

The expect value $E$ is the number of alignments with scores greater than or equal to score $S$ that are expected to occur by chance in a database search.

An $E$ value is related to a probability value $p$.

The key equation describing an $E$ value is:

$$E = Kmn e^{-S}$$
How to interpret BLAST: $E$ values and $p$ values

Very small $E$ values are very similar to $p$ values. $E$ values of about 1 to 10 are far easier to interpret than corresponding $p$ values.

<table>
<thead>
<tr>
<th>$E$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.99995460</td>
</tr>
<tr>
<td>5</td>
<td>0.99326205</td>
</tr>
<tr>
<td>2</td>
<td>0.86466472</td>
</tr>
<tr>
<td>1</td>
<td>0.63212056</td>
</tr>
<tr>
<td>0.1</td>
<td>0.09516258 (about 0.1)</td>
</tr>
<tr>
<td>0.05</td>
<td>0.04877058 (about 0.05)</td>
</tr>
<tr>
<td>0.001</td>
<td>0.00099950 (about 0.001)</td>
</tr>
<tr>
<td>0.0001</td>
<td>0.0001000</td>
</tr>
</tbody>
</table>

Table 4.4 page 107

Sometimes a real match has an $E$ value $> 1$

<table>
<thead>
<tr>
<th>Sequence name</th>
<th>Strain</th>
<th>Score</th>
<th>E value</th>
<th>Bit score</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>M13203</td>
<td>E. coli</td>
<td>123</td>
<td>0.0001</td>
<td>100</td>
<td>90%</td>
</tr>
<tr>
<td>M13203</td>
<td>E. coli</td>
<td>123</td>
<td>0.0001</td>
<td>100</td>
<td>90%</td>
</tr>
<tr>
<td>M13203</td>
<td>E. coli</td>
<td>123</td>
<td>0.0001</td>
<td>100</td>
<td>90%</td>
</tr>
<tr>
<td>M13203</td>
<td>E. coli</td>
<td>123</td>
<td>0.0001</td>
<td>100</td>
<td>90%</td>
</tr>
</tbody>
</table>

...try a reciprocal BLAST to confirm

Fig. 4.18 page 110

Outline for today

1. Homology and pairwise alignment
2. BLAST
3. Multiple sequence alignment
4. Phylogeny and evolution
Multiple sequence alignment: definition

- a collection of three or more protein (or nucleic acid) sequences that are partially or completely aligned
- homologous residues are aligned in columns across the length of the sequences
- residues are homologous in an evolutionary sense
- residues are homologous in a structural sense

Example: someone is interested in caveolin

Step 1: at NCBI change the pulldown menu to HomoloGene and enter caveolin in the search box

Step 2: inspect the results. We’ll take the first set of caveolins. Change the Display to Multiple alignment.
Step 3: inspect the multiple alignment. Note that these eight proteins align nicely, although gaps must be included.

Here's another multiple alignment, Rac:

This insertion could be due to alternative splicing.

HomoloGene includes groups of eukaryotic proteins. The site includes links to the proteins, pairwise alignments, and more.
Example: globins

Let's look at a multiple sequence alignment (MSA) of five globins proteins. We'll use five prominent MSA programs: ClustalW, Praline, MUSCLE (used at HomoloGene), ProbCons, and TCoffee. Each program offers unique strengths.

We'll focus on a histidine (H) residue that has a critical role in binding oxygen in globins, and should be aligned. But often it's not aligned, and all five programs give different answers.

Our conclusion will be that there is no single best approach to MSA. Dozens of new programs have been introduced in recent years.
Conclusions: ClustalW (the most popular program) gives different answers than a set of recent, improved alternatives. No one method is ideal.
Multiple sequence alignment: properties

• not necessarily one “correct” alignment of a protein family

• protein sequences evolve...

• ...the corresponding three-dimensional structures of proteins also evolve

• may be impossible to identify amino acid residues that align properly (structurally) throughout a multiple sequence alignment

• for two proteins sharing 30% amino acid identity, about 50% of the individual amino acids are superposable in the two structures

Multiple sequence alignment: features

• some aligned residues, such as cysteines that form disulfide bridges, may be highly conserved

• there may be conserved motifs such as a transmembrane domain

• there may be conserved secondary structure features

• there may be regions with consistent patterns of insertions or deletions (indels)

Multiple sequence alignment: uses

• MSA is more sensitive than pairwise alignment to detect homologs

• BLAST output can take the form of a MSA, and can reveal conserved residues or motifs

• Population data can be analyzed in a MSA (PopSet)

• A single query can be searched against a database of MSAs (e.g. PFAM)

• Regulatory regions of genes may have consensus sequences identifiable by MSA
Feng-Doolittle MSA (implemented in ClustalW and other programs) occurs in 3 stages

[1] Do a set of global pairwise alignments (Needleman and Wunsch's dynamic programming algorithm)

[2] Create a guide tree

[3] Progressively align the sequences

Progressive MSA stage 1 of 3: generate global pairwise alignments
Number of pairwise alignments needed

For \( n \) sequences, \((n-1)n / 2\)
For 5 sequences, \((4)(5) / 2 = 10\)
For 200 sequences, \((199)(200) / 2 = 19,900\)

Feng-Doolittle stage 2: guide tree

- Convert similarity scores to distance scores
- A tree shows the distance between objects
- Use UPGMA (defined in the phylogeny lecture)
- ClustalW provides a syntax to describe the tree
- A guide tree is not a phylogenetic tree

Progressive MSA stage 2 of 3: generate a guide tree calculated from the distance matrix (5 distantly related globins)

```{mermaid}
graph LR
  A-->B
  B-->C
  C-->D

beta_globin -- 6.36622 --> myoglobin -- 5.39056 --> horseglobin -- 5.39024 --> cowheme -- 5.36700 --> zebrafish -- 5.36544 --> 10.91452 --> 10.96100
```
**Feng-Doolittle stage 3: progressive alignment**

- Make a MSA based on the order in the guide tree
- Start with the two most closely related sequences
- Then add the next closest sequence
- Continue until all sequences are added to the MSA
- Rule: "once a gap, always a gap."

---

**Clustal W alignment of 5 distantly related globins**

<table>
<thead>
<tr>
<th>MSA Name</th>
<th>Length</th>
<th>MSA Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta_globin</td>
<td>147</td>
<td>legumiposa</td>
<td>147</td>
</tr>
<tr>
<td>cytoplasm</td>
<td>147</td>
<td>legumiposa</td>
<td>147</td>
</tr>
<tr>
<td>mmaximiliana</td>
<td>147</td>
<td>legumiposa</td>
<td>147</td>
</tr>
<tr>
<td>legumiposa</td>
<td>147</td>
<td>legumiposa</td>
<td>147</td>
</tr>
<tr>
<td>beta_globin</td>
<td>147</td>
<td>legumiposa</td>
<td>147</td>
</tr>
</tbody>
</table>

*Fig. 6.3*
Clustal W alignment of 5 closely related globins

* asterisks indicate identity in a column

Fig. 6.5
Page 189

Why “once a gap, always a gap”?

• There are many possible ways to make a MSA
• Where gaps are added is a critical question
• Gaps are often added to the first two (closest) sequences
• To change the initial gap choices later on would be to give more weight to distantly related sequences
• To maintain the initial gap choices is to trust that those gaps are most believable

Outline for today

1. Homology and pairwise alignment
2. BLAST
3. Multiple sequence alignment
4. Phylogeny and evolution

Page 189
Learning objectives: phylogeny

1. You should know how to create a phylogenetic tree from a multiple sequence alignment
2. You should know the parts of a tree
3. You should know how to interpret the biological (historical) meaning of a tree

Molecular clock hypothesis

In the 1960s, sequence data were accumulated for small, abundant proteins such as globins, cytochromes c, and fibrinopeptides. Some proteins appeared to evolve slowly, while others evolved rapidly.

Linus Pauling, Emanuel Margoliash and others proposed the hypothesis of a molecular clock:

For every given protein, the rate of molecular evolution is approximately constant in all evolutionary lineages.
Molecular clock hypothesis: conclusions

Dickerson drew the following conclusions:

• For each protein, the data lie on a straight line. Thus, the rate of amino acid substitution has remained constant for each protein.

• The average rate of change differs for each protein. The time for a 1% change to occur between two lines of evolution is 20 MY (cytochrome c), 5.8 MY (hemoglobin), and 1.1 MY (fibrinopeptides).

• The observed variations in rate of change reflect functional constraints imposed by natural selection.

Five stages of phylogenetic analysis

[1] Selection of sequences for analysis
[2] Multiple sequence alignment
[3] Selection of a substitution model
[4] Tree building
[5] Tree evaluation

MEGA software for phylogeny:
http://www.megasoftware.net/
How to use MEGA to make a tree

1. Enter a multiple sequence alignment (.meg) file
2. Under the phylogeny menu, select one of these four methods...

- Neighbor-Joining (NJ)
- Minimum Evolution (ME)
- Maximum Parsimony (MP)
- UPGMA

Use of MEGA for a distance-based tree: UPGMA

Click compute to obtain tree

Click green boxes to obtain options
Use of MEGA for a distance-based tree: UPGMA

UPGMA is an unweighted pair group method using arithmetic mean.

Tree-building methods: UPGMA

Step 1: compute the pairwise distances of all the proteins. Get ready to put the numbers 1-5 at the bottom of your new tree.
Tree-building methods: UPGMA

Step 2: Find the two proteins with the smallest pairwise distance. Cluster them.

Fig. 7.26
Page 257

Tree-building methods: UPGMA

Step 3: Do it again. Find the next two proteins with the smallest pairwise distance. Cluster them.

Fig. 7.26
Page 257

Tree-building methods: UPGMA

Step 4: Keep going. Cluster.

Fig. 7.26
Page 257
Tree-building methods: UPGMA

Step 4: Last cluster! This is your tree.