Multiple sequence alignments

- Definition
- The need for MSA
- The MSA problem
- MSA methods

Editing and formatting alignments

- Software packages available
What is Multiple Sequence Alignment (MSA)?

- Multiple sequence alignment (MSA) can be seen as a generalization of Pairwise Sequence Alignment - instead of aligning two sequences, n sequences are aligned simultaneously, where n is > 2

- **Definition:**
  A multiple sequence alignment is an alignment of n > 2 sequences obtained by inserting gaps ("-") into sequences such that the resulting sequences have all length L and can be arranged in a matrix of N rows and L columns where each column represents a homologous position.

- **Note:**
  MSA applies both to nucleotide and amino acid sequences.
  To construct a multiple alignment, one may have to introduce gaps in sequences at positions where there were no gaps in the corresponding pairwise alignment.
  → multiple alignments typically contain more gaps than any given pair of aligned sequences.
Why do we need MSA?

- Multiple sequence alignment can help to develop a sequence “finger print” which allows the identification of members of distantly related protein family (motifs).

- Formulate & test hypotheses about protein 3-D structure.

- MSA can help us to reveal biological facts about proteins, e.g.:
  (e.g. how protein function has changed or evolutionary pressure acting on a gene)

- Crucial for genome sequencing:
  - Random fragments of a large molecule are sequenced and those that overlap are found by a multiple sequence alignment program.
  - There should be one correct alignment that corresponds to the genomic sequence rather than a range of possibilities.
  - Sequence may be from one strand of DNA or the other, so complements of each sequence must also be compared.
  - Sequence fragments will usually overlap, but by an unknown amount and in some cases, one sequence may be included within another.
  - All of the overlapping pairs of sequence fragments must be assembled into large composite genome sequence.

- To establish homology for phylogenetic analyses.

- Identify primers and probes to search for homologous sequences in other organisms.
The alignment problem

How do we generate a multiple alignment? Given a pairwise alignment, just add the third, then the fourth, and so on, until all have been aligned. Does it work?

Example: It is not self-evident how these sequences are to be aligned together. Here are some possibilities:

- Taxon A AGAC
- Taxon B --AC
- Taxon C AG--

- Taxon A AGAC
- Taxon B AC--
- Taxon C AG

- Taxon A AGAC
- Taxon C AG--
- Taxon B --AC

- Taxon B AC--
- Taxon C AG--
- Taxon A AGAC

- Taxon B --AC
- Taxon C --AG
- Taxon A AGAC

It depends not only on the various alignment parameters but also on the order in which sequences are added to the multiple alignment.
The alignment problem

What happens when a sequence alignment is wrong?

A: AGT  A: AGT  A: AGT  A: AGT-
B: AT   B: A-T  B: AT-  B: A-T-
C: ATC  C: ATC  C: ATC  C: A-TC
From pairwise to multiple alignments

In pairwise alignments, one has a two-dimensional matrix with the sequences on each axis. The number of operations required to locate the best “path” through the matrix is approximately proportional to the product of the lengths of the two sequences.

A possible general method would be to extend the pairwise alignment method into a simultaneous N-wise alignment, using a complete dynamical-programming algorithm in N dimensions. Algorithmically, this is not difficult to do.

But what about execution time?
The big-O notation

One of the most important properties of an algorithm is how its execution time increases as the problem is made larger (e.g. more sequences to align). This is the so-called \textit{algorithmic} (or computational) \textbf{complexity} of the algorithm.

There is a notation to describe the algorithmic complexity, called \textbf{the big-O notation}. If we have a problem size (number of input data points) $n$, then an algorithm takes $O(n)$ time if the time increases linearly with $n$. If the algorithm needs time proportional to the square of $n$, then it is $O(n^2)$.

It is important to realize that an algorithm that is quick on small problems may be totally useless on large problems if it has a bad $O()$ behavior. As a rule of thumb one can use the following characterizations, where $n$ is the size of the problem, and $c$ is a constant:

\begin{center}
\begin{tabular}{|l|l|}
\hline
$O(c)$ & utopian \\
$O(\log n)$ & excellent \\
$O(n)$ & very good \\
$O(n^2)$ & not so good \\
$O(n^3)$ & pretty bad \\
$O(c^n)$ & disaster \\
\hline
\end{tabular}
\end{center}
To compute a N-wise alignment, the algorithmic complexity is something like $O(c^{2n})$, where $c$ is a constant, and $n$ is the number of sequences.

Example:

A pairwise alignment of two sequences $[O(c^{2x2})]$, takes 1 second, then four sequences $[O(c^{2x4})]$, would take $10^4$ seconds (2.8 hours), five sequences $[O(c^{2x5})]$, $10^6$ seconds (11.6 days), six sequences $[O(c^{2x6})]$, $10^8$ seconds (3.2 years), seven sequences $[O(c^{2x7})]$, $10^{10}$ seconds (317 years), and so on.

This is disastrous!
How to optimize alignment algorithms?

- Use structural information:
  - reading frame
  - protein structure

- Sequence elements are not truly independent but related by phylogeny

**Raw**          **Alignment**

<table>
<thead>
<tr>
<th>Species</th>
<th>Raw</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>N Y L S</td>
<td>N – Y L S</td>
</tr>
<tr>
<td>Chimpanzee</td>
<td>N K Y L S</td>
<td>N K Y L S</td>
</tr>
<tr>
<td>Gorilla</td>
<td>N F S</td>
<td>N – F – S</td>
</tr>
<tr>
<td>Orangutan</td>
<td>N F L S</td>
<td>N – F L S</td>
</tr>
</tbody>
</table>

- Sequences often contain highly conserved regions
How to optimize alignment algorithms?

- Sequences often contain highly conserved regions

These regions can be used for an initial alignment

By analyzing a number of small, independent fragments, the algorithmic complexity can be drastically reduced!
**MSA methods**

- Progressive global alignment of the sequences starting with an alignment of the most alike sequences and then building an alignment by adding more sequences.

- Iterative methods that make an initial alignment of groups of sequences and then revise the alignment to achieve a better result.

- Alignments based on locally conserved patterns found in the same order in the sequences.
For a given group of sequences, there is no single "correct" alignment, only an alignment that is "optimal" according to some set of calculations. This is partly due to:

- the complexity of the problem,
- limitations of the scoring systems used,
- our limited understanding of life and evolution

Determining what alignment is best for a given set of sequences is really up to the judgment of the investigator.

Success of the alignment will depend on the similarity of the sequences. If sequence variation is great it will be very difficult to find an optimal alignment.
**MSA and gaps**

- **Gaps can occur:**

  Before the first character of a string
  
  \[
  \text{CTGCGGG}---\text{GGTAAT} \\
  \ | | | \ | | | \\
  \text{--GCCG--AGAGG--AA--}
  \]

  Inside a string
  
  \[
  \text{CTGCGGG}---\text{GGTAAT} \\
  \ | | | \ | | | \\
  \text{--GCCG--AGAGG--AA--}
  \]

  After the last character of a string
  
  \[
  \text{CTGCGGG}---\text{GGTAAT} \\
  \ | | | \ | | | \\
  \text{--GCCG--AGAGG--AA--}
  \]

- **Note:** In protein-coding nucleotide sequences most gaps have a length of 3N
**MSA and gaps**

**Gap Penalties**

- In the MSA scoring scheme, a penalty is subtracted for each gap introduced into an alignment because the gap increases uncertainty into an alignment.

- The gap penalty is used to help decide whether or not to accept a gap or insertion in an alignment.

- Biologically, it should in general be easier for a sequence to accept a different residue in a position, rather than having parts of the sequence chopped away or inserted. Gaps/insertions should therefore be more rare than point mutations (substitutions).

- In general, the lower the gapping penalties, the more gaps and more identities are detected but this should be considered in relation to biological significance.

- Most MSA programs allow for an adjustment of gap penalties.
**MSA with ClustalW**

- Works by progressive alignment: it aligns a pair of sequences then aligns the next one onto the first pair.

- Most closely related sequences are aligned first, and then additional sequences and groups of sequences are added, guided by the initial alignments. Uses alignment scores to produce a phylogenetic tree.

- Aligns the sequences sequentially, guided by the phylogenetic relationships indicated by the tree.

- Gap penalties can be adjusted based on specific amino acid residues, regions of hydrophobicity, proximity to other gaps, or secondary structure.

- Is available with a great web interface: [http://www.ebi.ac.uk/clustalw/](http://www.ebi.ac.uk/clustalw/).

- Also available as ClustalX (stand-alone MS-Windows software).
ClustalW Submission Form

ClustalW is a general purpose multiple sequence alignment program for DNA or proteins. It produces biologically meaningful multiple sequence alignments of divergent sequences. It calculates the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen. Evolutionary relationships can be seen via viewing Cladograms or Phylograms.

<table>
<thead>
<tr>
<th>Operational options</th>
<th>Output options</th>
<th>Input options, matrix choice, gap opening penalty</th>
<th>Gap information, output tree type</th>
<th>File input in GCG, FASTA, EMBL, GenBank, Phylip, or several other formats</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>YOUR EMAIL</strong></td>
<td><strong>ALIGNMENT TITLE</strong></td>
<td><strong>CPU MODE</strong></td>
<td><strong>ALIGNMENT</strong></td>
<td><strong>OUTPUT FORMAT</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OUTPUT ORDER</strong></td>
<td><strong>COLOR ALIGNMENT</strong></td>
<td><strong>K1TUP (WORD SIZE)</strong></td>
<td><strong>WINDOW LENGTH</strong></td>
</tr>
<tr>
<td>aligned</td>
<td>no</td>
<td>aligned</td>
<td>no</td>
<td>def</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOPDAG</strong></td>
<td><strong>PAIRGAP</strong></td>
<td><strong>PHYLOGENETIC TREE</strong></td>
<td><strong>MATRIX</strong></td>
<td><strong>GAP OPEN</strong></td>
</tr>
<tr>
<td>def</td>
<td>def</td>
<td>def</td>
<td>def</td>
<td>def</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>END GAPS</strong></td>
<td><strong>GAP EXTENSION</strong></td>
<td><strong>GAP DISTANCES</strong></td>
<td><strong>TREE TYPE</strong></td>
<td><strong>TREE GRAPH DISTANCES</strong></td>
</tr>
<tr>
<td>def</td>
<td>def</td>
<td>def</td>
<td>def</td>
<td>def</td>
</tr>
</tbody>
</table>

Enter or Paste a set of Sequences in any supported format
MSA with PILEUP

- PILEUP is the MSA program that is part of the Genetics Computer Group (GCG) sequence analysis package

- Sequences are aligned pairwise using dynamic programming algorithm

- The scores are used to produce a phylogenetic tree, which is then used to guide the alignment of the most closely related sequences and groups of sequences

- Resulting alignment is a global alignment produced by the Needleman-Wunsch algorithm
Iterative MSA methods

- Attempt to correct initial alignment problems by repeatedly aligning subgroups of the sequences and then by aligning these subgroups into a global alignment of all the sequences.

- MultAlin – recalculates pair-wise scores during the production of the progressive alignment and uses these scores to recalculate the tree.

- PRRP – initial alignment is made to predict a tree, the tree is used to produce weights where the sequences are analyzed for the presence of aligned regions that include gaps.

- SAGA – based on genetic algorithm that is a machine-learning algorithm that attempts to produce alignments by the simulations of evolutionary changes in sequences.
Editing and formatting alignments

◆ Sequence editors are used for:

- manual alignment/editing of sequences
- visualization of data
- data management
- import/export of data
- graphical enhancement of data for presentations

◆ Examples:

- **CINEMA** (Color Interactive Editor for Multiple Alignments) web applet  
  http://www.biochem.ucl.ac.uk/bsm/dbbrowser/CINEMA2.02/kit.html

- **GDE** (Genetic Data Environment) - UNIX based  
  http://bimas.dcrt.nih.gov/gde_sw.html

- **GeneDoc** - MS Windows http://www.psc.edu/biomed/genedoc/

- **MACAW** - local multiple sequence alignment program and sequence editing tool available by anonymous FTP from ncbi.nih.gov/pub/schuler/macaw

- **BioEdit** - sequence alignment editor for MS Windows with web access and accessory applications (BLAST, local BLAST, ClustalW, Phylip and more)
Summary MSA

- **Definition:**
  A multiple sequence alignment is an alignment of $n > 2$ sequences obtained by inserting gaps ('-') into sequences such that the resulting sequences have all length $L$ and can be arranged in a matrix of $N$ rows and $L$ columns where each column represents a homologous position.

- **Why do we need MSA?**
  - Formulate & test hypotheses about protein 3-D structure
  - MSA can help us to reveal biological facts about proteins
  - Crucial for genome sequencing
  - To establish homology for phylogenetic analyses
  - Identify primers and probes to search for homologous sequences in other organisms

- **The MSA problem**
  - Most pairwise alignment algorithms are too complex to be used for $n$-wise alignments
  - Alignment algorithms need to be optimized
    - use structural information
    - use phylogenetic information
    - use conserved regions

- **MSA methods**
  - Progressive global alignment (starts with the most alike sequences)
    - e.g., ClustalW, ClustalX, Pileup
  - Iterative methods (initial alignment of groups of sequences that are revised)
    - MultAlin, PRRP, SAGA
  - Alignments based on locally conserved patterns

- **Sequence editors**
  - CINEMA GDE, GeneDoc, MACAW, BioEdit