Marine Biological Laboratory, Woods Hole, MA
Workshop on Molecular Evolution: multiple sequence analysis session
July 28, 2009, 7 to 10 PM

Multiple Sequence Alignment & Analysis with SeaView and MAFFT
Steven M. Thompson

More data yields stronger analyses — if done carefully! The patterns of conservation become ever clearer by comparing the conserved portions of sequences amongst a larger and larger dataset. Mosaic ideas and evolutionary ‘importance.’

But first a prelude: My definitions
Biocomputing and computational biology are synonymous and describe the use of computers and computational techniques to analyze any biological system, from molecules, through cells, tissues, organisms, and populations, to complete ecologies. Bioinformatics describes using computational techniques to access, analyze, and interpret the biological information in any of the available online biological databases. Sequence analysis is the study of molecular sequence data for the purpose of inferring the function, mechanism, interactions, evolution, and perhaps structure of biological molecules. Genomics analyzes the context of genes or complete genomes (the total DNA content of an organism) within and across genomes. Proteomics is a subdivision of genomics concerned with analyzing the complete protein complement, i.e. the proteome, of organisms, both within and between different organisms.
And a ‘way’ to think about it:
The reverse biochemistry analogy
from a ‘virtual’ DNA sequence to actual molecular
physical characterization, not the other way ‘round.
Using bioinformatics tools, you can infer all sorts
of functional, evolutionary, and, structural
insights into a gene product, without the need
to isolate and purify massive amounts of
protein! Eventually you can go on to clone and
express the gene based on that analysis using
PCR techniques.
The computer and molecular databases are an
essential part of this process.

The exponential growth of molecular
sequence databases & cpu power

<table>
<thead>
<tr>
<th>Year</th>
<th>Bases</th>
<th>Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1982</td>
<td>68336</td>
<td>914</td>
</tr>
<tr>
<td>1983</td>
<td>227525</td>
<td>2671</td>
</tr>
<tr>
<td>1984</td>
<td>339870</td>
<td>4710</td>
</tr>
<tr>
<td>1985</td>
<td>520430</td>
<td>5730</td>
</tr>
<tr>
<td>1986</td>
<td>801271</td>
<td>9890</td>
</tr>
<tr>
<td>1987</td>
<td>1526476</td>
<td>14684</td>
</tr>
<tr>
<td>1988</td>
<td>2200100</td>
<td>20579</td>
</tr>
<tr>
<td>1989</td>
<td>3470285</td>
<td>26791</td>
</tr>
<tr>
<td>1990</td>
<td>4577000</td>
<td>26923</td>
</tr>
<tr>
<td>1991</td>
<td>7147000</td>
<td>50247</td>
</tr>
<tr>
<td>1992</td>
<td>1001376</td>
<td>126454</td>
</tr>
<tr>
<td>1993</td>
<td>1574242</td>
<td>153892</td>
</tr>
<tr>
<td>1994</td>
<td>2771245</td>
<td>252721</td>
</tr>
<tr>
<td>1995</td>
<td>38079406</td>
<td>359693</td>
</tr>
<tr>
<td>1996</td>
<td>45570006</td>
<td>312271</td>
</tr>
<tr>
<td>1997</td>
<td>118638987</td>
<td>1786237</td>
</tr>
<tr>
<td>1998</td>
<td>200761784</td>
<td>2323017</td>
</tr>
<tr>
<td>1999</td>
<td>360119211</td>
<td>4864570</td>
</tr>
<tr>
<td>2000</td>
<td>1141588038</td>
<td>10582023</td>
</tr>
<tr>
<td>2001</td>
<td>1594992138</td>
<td>14476210</td>
</tr>
<tr>
<td>2002</td>
<td>28207870198</td>
<td>22288983</td>
</tr>
<tr>
<td>2003</td>
<td>28207870198</td>
<td>30986418</td>
</tr>
<tr>
<td>2004</td>
<td>4430741776</td>
<td>44683179</td>
</tr>
<tr>
<td>2005</td>
<td>6951065705</td>
<td>26817472</td>
</tr>
<tr>
<td>2006</td>
<td>9301375718</td>
<td>80366862</td>
</tr>
</tbody>
</table>

Doubling time about a year and half!

Now then, why even bother
— Applicability?
Molecular evolutionary analysis; plus
Probe/primer, and motif/profile design;
Graphical illustrations; and
Comparative ‘homology’ inference.
OK — here’s some examples.
Molecular evolution and phylogenetics

We all know multiple sequence alignments are necessary for phylogenetic inference, but does everybody here truly realize that the absolute positional homology of every column in a data matrix passed on to these programs is the most critical assumption that all the algorithms make (but see Bayesian coestimation)!

And what about this other stuff?

Multiple sequence alignments can be indispensable for primer design when you don’t have data on a particular taxa, yet data is available in related taxa. The conservation and variability within an alignment can help guide the design of universal or taxa specific primers.

Here’s an HPV L1 example

The ellipses show areas where PCR primers could differentiate the Type 16 clade from it’s closest relatives — areas of high L1 conservation in the Type 16 clade (red line) that correspond to areas of much weaker conservation in the others (blue line).
Motif and profile definition

An alignment of human SRY/SOX proteins illustrates the conservation of the HMG box. Conserved regions can be visualized with a sliding window approach and appear as peaks. Motifs and (better yet) HMM profiles can be created of the region to be used as a search tool to find other HMG box proteins.

One picture’s worth . . .

The HMG-box domain is strikingly conserved amongst the otherwise nearly unalignable human DNA regulatory paralogous protein family.

Structure/function homology inference

A Swiss-Model homology based model of *Giardia* EF1α superimposed over its eight most similar sequences with solved structure. Amazingly accurate inferences of both function and structure are possible using comparative methods.
On to aligning multiple sequences — dynamic programming’s complexity increases exponentially with the number of sequences being compared:

N-dimensional matrix . . .

complexity $O \left( \text{sequence length} \times \text{number of sequences} \right)$

A couple ‘global’ solutions using heuristic tricks
See —

MSA (‘global’ within ‘bounding box’) and
PIMA (‘local’ portions only) on the multiple alignment page at the
Both available at the Baylor College of Medicine’s Search Launcher —
http://searchlauncher.bcm.tmc.edu/ — but, severely limiting restrictions!

Therefore — pairwise, progressive dynamic programming . . .

. . . restricts the solution to the neighborhood of only two sequences at a time.
All sequences are compared, pairwise, and then each is aligned to its most similar partner or group of partners represented as a consensus. Each group of partners is then aligned to finish the complete multiple sequence alignment.
Enhancements on the theme

First enhancements came from ClustalW — variable sequence weighting, dynamically varying gap penalties and substitution matrices, and a neighbor-joining guide-tree.

Since the year 2000 a slew of new programs have tried other heuristic variations, all in attempts to build faster, more accurate multiple sequence alignments. The devil's in the details: Muscle, ProbCons, T-Coffee, MAFFT and many, many more.

Muscle

An iterative method that uses weighted log-expectation profile scoring along with a slew of optimizations. It proceeds in three stages — draft progressive using k-mer counting, improved progressive using a revised tree from the previous iteration, and refinement by sequential deletion of each tree edge with subsequent profile realignment.

ProbCon

Uses Hidden Markov Model (HMM) techniques and posterior probability matrices that compare random pairwise alignments to expected pairwise alignments. Probability consistency transformation is used to reestimate the scores, and a guide-tree is then constructed, which is used to compute the alignment, which is then iteratively refined. Incredibly accurate.

T-Coffee

Uses a preprocessed, weighted library of all the pairwise global alignments between your sequences, plus the ten best local alignments associated with each pair. This helps build the NJ guide-tree and the progressive alignment. The library is used to assure consistency and help prevent errors, by allowing ‘forward-thinking’ to see whether the overall alignment will be better one way or another after particular segments are aligned one way or another. The institutional schedule analogy . . . .

T-Coffee can even tie together multiple methods as external modules, making consistency libraries from the results of each, as long as all the specified methods are installed on your system. T-Coffee is one of the most accurate multiple sequence alignment methods available because of this consistency based rationale, but it is not the fastest. Regardless, I encourage you to check it out!
MAFFT — today’s example

— has many modes, among them: a couple of progressive, approximate modes, using a fast Fourier transformation (FFT); a couple of iteratively refined methods that add in weighted-sum-of-pairs (WSP) scoring; and several iterative methods that use WSP scoring combined with a T-Coffee-like consistency based scoring scheme. Speed and accuracy are inversely proportional for these from fast and rough, to slow and accurate, respectively.

MAFFT provides command aliases for all of these, from fast to slow — FFTNS with or without retree, FFTNSI with or without maxiterate, and the three combined approaches EINSI, LINSI, and GINSI.

MAFFT’s basic algorithm

MAFFT’s fast Fourier transform provides a huge speedup over previous methods. Homologous regions are quickly identified by converting amino acid residues to vectors of volume and polarity, thus changing a twenty-character alphabet to six, rather than by using an amino acid similarity matrix. Similarly, nucleotide bases are converted to vectors of imaginary and complex numbers. The FFT trick then reduces the complexity of the subsequent comparison to $O(N \log N)$. FFT identifies potential similarities though, without localizing them; a sliding window step using the BLOSUM62 matrix is used for this.

Then MAFFT constructs a distance matrix, and hence a progressive guide tree, on the number of shared six-tuples from this Fourier transform, rather than on a ranking based on full-length, pairwise sequence similarity. The user can specify how many times a new guide tree is subsequently recalculated from a previous alignment as many times as desired; the alignment is reconstructed using the Needleman Wunsch algorithm each time.

Some of MAFFT’s many modes

And each mode has a bunch of additional options!

1) Most basic, fastest modes — just progressive.
   a) FFTNS1 (fftns --retree 1)
   b) FFTNS2 (fftns) (same as mafft --retree 2)

   Suitable for 1,000’s of easily aligned sequences.
   A rough distance matrix is built from the sequences using FFT and the shared number of six-mers.
   A modified UPGMA guide tree is built from this matrix.
   The sequences are aligned according to the rough, initial guide tree (as in ‘traditional’ methods).
   FFTNS2 adds a recomputation of the guide tree (retree 2) from the original alignment, from which a new progressive alignment is built.
MAFFT’s iterative refinements

2) Intermediate modes — progressive + iterations to maximize the WSP objective function.
   a) FFTNSI (fftnsi) default two cycles, or e.g. fftnsi --maxiterate 1000
   b) NWNSI (nwnsi) same as FFTNSI, but no FFT, Needleman Wunsch only.

Progressive alignment and retree as before, with or without FFT, and then . . . .
Iterative refinement is cycled twice (default), or repeatedly until there is no further improvement, or until you reach your specified limit number.
Suitable for 100’s through 1000’s of sequences.

MAFFT’s most accurate modes

3) Advanced modes — progressive + iterations to maximize the objective WSP and T-Coffee-like consistency functions. Options differ according to the way the pairwise alignments are calculated.
   a) EINSI (einsi) most general of these.
      Uses a Smith Waterman style local algorithm with generalized affine gap costs for the pairwise step. Most appropriate for sequences with multi-shared, similarly ordered domains, in an otherwise nearly unalignable ‘mess’. e.g:

   oooooooooo-----XXXXXXXXXX--XXXXXXX----------
   ------XXXXXXXXXXXXXooo--------------------XXXXXXXXXXXXXXXXXX-XXXXXXXX----------
   --ooooXXXXXX---XXXXooooooooooo------------XXXXX----XXXXXXXXXXXXXXXXXXoooooooooo
   ------XXXXX----XXXXoooooooooooooooooooooooXXXXX-XXXXXXXXXXXX--XXXXXXX----------
   ------XXXXX----XXXX-----------------------XXXXX---XXXXXXXXXX--XXXXXXXooooo-----

MAFFT’s most accurate modes, cont.

3) Advanced modes — progressive + iterations to maximize the objective WSP and T-Coffee-like consistency functions. Options differ according to the way the pairwise alignments are calculated.
   b) LINSI (linsi) strictly local.
      Uses a Smith Waterman style local algorithm with affine gap costs for the pairwise step. Most appropriate for sequences with only one single, shared domain, in an otherwise nearly unalignable ‘mess’. e.g:

   ---------------------------XXXXXXXXXXXX-XXXXXXXXXXXXX
   ---------------------------XXXXXXXXXXXXXXXXXX-XXXXXXXX
   ---------------------------XXXXX----XXXXXXXXXXXXXXXXXX
   oooooooooooXXXXX-XXXXXXXXXXXX--XXXXXXX----------
   ---------------------------XXXXX---XXXXXXXXXX--XXXXXXXooooo-----
MAFFT’s most accurate modes, cont.

3) Advanced modes — progressive + iterations to maximize the objective WSP and T-Coffee-like consistency functions. Options differ according to the way the pairwise alignments are calculated.

c) GINSI (ginai) strictly global.

Uses a Needleman Wunsch style global algorithm with affine gap costs for the pairwise step. Most appropriate for sequences where only one single, shared domain extends the full length of all of the sequences, e.g:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX--XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX--XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX--XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

How to know when to use what

for MAFFT — see “tips,” 2, 3, and 4 pages, for all of them — Take home message:

For simple cases it doesn’t really matter what program to use. For complicated situations it may, and what you use will depend on the size of your dataset, personal preferences, time allotted, and how much hand editing you want to do.


The rest of my references can be found in my tutorial manuscript.

You can do a lot of this stuff on the Web, if you need to — some resources for multiple sequence alignment:

http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html
http://pbil.univ-lyon1.fr/alignment.html
http://www.ebi.ac.uk/clustalw/
http://searchlauncher.bcm.tmc.edu/

However, problems with very large datasets and huge multiple alignments make doing multiple sequence alignment on the Web impractical after your dataset has reached a certain size. You’ll know it when you’re there!
If large datasets become intractable for analysis on the Web, what other resources are available?

Desktop software solutions — all of these programs are available in public domain open source, but . . . they can be complicated to install, configure, and maintain. User must be pretty computer savvy.

So, commercial software packages are available, e.g. MacVector, DS Gene, DNAsis, DNAStar, etc., but . . . license hassles, big expense per machine, lack of most recent programs, underperformance, and Internet and/or CD database access all complicate matters!

Therefore, I argue for UNIX server-based solutions . . .

UNIX servers — pros and cons
Free/public domain solutions still available, but now a very cooperative systems manager needs to maintain everything for users. If you have such a person, then:

You end up with a more powerful, and usually faster computer, with larger storage capabilities. Plus, connections can be made from any networked terminal or workstation anywhere!

Operating system: UNIX command line operation hassles; communications software — telnet, ssh, and terminal emulation; X graphics; file transfer — ftp, and scp/sftp; and editors — vi, emacs, pico/nano (or desktop word processing followed by file transfer [save as “text only!”]). See my supplement pdf file.

Reliability and the Comparative Approach — explicit homologous correspondence;
manual adjustments should be encouraged — based on knowledge, especially structural, regulatory, and functional sites.

Therefore, editors like SeaView and databases like the Ribosomal Database Project:
http://rdp.cme.msu.edu/index.jsp
Coding DNA issues

Work with proteins! If at all possible.
Twenty match symbols versus four, plus similarity versus identity!
Way better signal to noise.
Also guarantees no indels are placed within codons. So translate, then align.
Nucleotide sequences will only reliably align if they are very similar to each other. And they will likely require extensive and carefully considered hand editing with an editor like SeaView.

Beware of aligning apples and oranges [and grapefruit]!

- receptors and/or activators with their namesake proteins;
- paralogous versus orthologous;
- genomic versus cDNA;
- mature versus precursor.

Mask out uncertain areas —
Complications —
Order dependence.
   Not that big of a deal.
Substitution matrices and gap penalties.
   Can be a very big deal!
Regional ‘realignment’ becomes incredibly important, especially with sequences that have areas of high and low similarity. SeaView let’s you do this!

Complications cont. —
Format hassles!
   Specialized format conversion tools such as GCG’s SeqConv+ program and PAUPSearch, and
   Don Gilbert’s public domain ReadSeq program.
   Plus, some programs like SeaView can read and write several formats.

Still more complications —
Indels and missing data symbols (i.e. gaps) designation discrepancy headaches —
   ., -, ~, ?, N, or X
   . . . . Help!
Conclusions —

Gunnar von Heijne in his very old but quite readable treatise, *Sequence Analysis in Molecular Biology; Treasure Trove or Trivial Pursuit* (1987), provides a very appropriate conclusion:

> “Think about what you’re doing; use your knowledge of the molecular system involved to guide both your interpretation of results and your direction of inquiry; use as much information as possible; and do not blindly accept everything the computer offers you.”

He continues:

> “…if any lesson is to be drawn … it surely is that to be able to make a useful contribution one must first and foremost be a biologist, and only second a theoretician … . We have to develop better algorithms, we have to find ways to cope with the massive amounts of data, and above all we have to become better biologists. But that’s all it takes.”

---

On to a demonstration of some of SeaView’s multiple sequence dataset capabilities —

The HPV L1 gene and complete genome . . . the tutorial:

How to use SeaView with MAFFT.