# Multiple sequence alignment with POY and ClustalW

The sample dataset mammals.fas is a small dataset of 16S ribosomal rDNA from the mitochondrial genome of 16 mammals and an alligator as an outgroup. This type of DNA is not bound by the triplet code of protein coding genes, and has both highly variable and highly conserved regions.

| **Scientific name** | **Common name** | **Order** | **Family** |
|---|---|---|---|
| *Alligator mississippiensis* | American Alligator | Crocodilia | Alligatoridae |
| *Bos taurus* | Domestic cattle | Artiodactyla | Bovidae |
| *Balaenoptera musculus* | Blue whale | Cetacea | Balaenopteridae |
| *Didelphis virginiana* | Virginia Opossum | Didelphimorphia | Didelphidae |
| *Macropus robustus* | Eastern Wallaroo | Diprotodontia | Macropodidae |
| *Ornithorhynchus anatinus* | Duck-billed Platypus | Monotremata | Ornitorhynchidae |
| *Tachyglossus aculeatus* | Short-beaked Echidna | Monotremata | Tachyglossidae |
| *Equus caballus* | Horse | Perissodactyla | Equidae |
| *Equus asinus* | Donkey | Perissodactyla | Equidae |
| *Rhinoceros unicornis* | Indian Rhiniceros | Perissodactyla | Rhinocerotidae |
| *Papio hamadryas* | Hamadryas Baboon | Primates | Cercopithidae |
| *Homo sapiens* | Human | Primates | Hominidae |
| *Gorilla gorilla* | Western Gorilla | Primates | Hominidae |
| *Pan paniscus* | Bonobo | Primates | Hominidae |
| *Pan troglodytes troglodytes* | Central Chimpanzee | Primates | Hominidae |
| *Pongo pygmaeus* | Bornean Orangutan | Primates | Hominidae |
| *Hylobates agilis* | Agile Gibbon | Primates | Hylobatidae |

This is roughly similar to the data used by Kjer et al. (2007), but with the addition of the Bonobo and the alligator outgroup.

## About ClustalW

Clustal is the most popular multiple sequence alignment software, and has been actively developed since 1988. It is fast for small datasets and can produce output in several common formats. The Clustal algorithm works by creating a Neighbor-Joining tree from pairwise distances calculated from the Needleman-Wunsch algorithm, then using the NJ tree as a guide for the multiple alignment.

Point your favorite web browser at http://www.ebi.ac.uk/Tools/clustalw2/index.html

There are multiple options available, but for now just make sure to set the following:

OUTPUT FORMAT: set this to **aln w/numbers**
OUTPUT ORDER: set this to **input**

Either upload the file mammals.fas as a file, or open it in a text editor and paste it into the sequence input field.

Push **Run**. Wait.

When the page refreshes you will be taken to the results page. If you push the Start Jalview button, you can watch your alignment in a Java viewer. Try different color settings. Find out what different things they will tell you about the alignment.

Go back in the browser until you are back at the input page. Change the setting of OUTPUT FORMAT to **phylip**, and run the program again. When it finishes, select **View Alignment File.** Don't close the browser!

### Maximum likelihood analysis with RAxML

Open a new browser tab an go to RAxML Blackbox at

http://phylobench.vital-it.ch/raxml-bb/

Copy and paste your aligned sequences from Clustal. Enter Alligator_ as outgroup.  Push **run**. Wait. This can take a few minutes.

Refresh the link once in a while until you reach the results page. You can view your tree with branch lengths and bootstrap support values. Don't close the browser yet!

If you have time, go back to Clustal and try different settings for the multiple alignment, the most important being GAP OPEN and GAP EXTENSION.

### Optimization alignment with POY4

POY4 is a software package for phylogenetic analysis of all types of biological data. For this exercise we are going to focus on its optimization alignment features where treespace is explored without ever creating a static alignment.

There is a rudimentary GUI for Windows and OSX, but since it's focused on running prewritten scripts, we are going to use the Linux version in a terminal.

Start PuTTY and log on to your account on **mordor.stat.osu.edu** or any of the other stat department servers (**gondor/rohan/rivendell/shire.stat.osu.edu**)

Place the poy binary and mammals.fas in the same directory.

Start poy by entering `./poy` at the command prompt.

You will be presented with several 'windows'. The text in the output window can be scrolled up and down with the arrow keys. Push 'Tab' to get back to the interactive console.

Load your data into POY by entering

```
read("mammals.fas")
```

POY will tell you some stats about the data: number of taxa and number of gene fragments for each one.

Set the outgroup

```
set(root:1)
```

At this point the cost parameters can be set. They can be changed at any time, and the trees in memory re-evaluated under the new costs.

```
transform(tcm:(substitution,gap))
```

`transform(tcm:(1,1))` will be most similar to a static alignment analysis in PAUP* if gaps are treated like a 5th state instead of missing data. Default setting in POY is `transform(tcm:(1,2))`, which makes gaps twice as expensive as substitutions.

Next step is to build starting trees with the command `build`. Default number is 10 trees, but can be specified like this `build(number,method)`

`build(random)` will build 10 completely random trees
`build(nj)` will create a single Neighbor-Joining tree. This is most similar to Clustal.
`build()` will create 10 Wagner trees. These are better than random, and will speed up the search compared to starting from completely random trees for large datasets.

Compare the tree lengths of random trees compared to Wagner and NJ trees.

Search treespace by performing branch swapping on the stored trees

`swap()` will perform TBR branch swapping on all stored trees

Keep the best trees, throw out suboptimal trees

```
select()
```

Repeat the search under different weights and starting trees. Do the optimal trees have different tree lengths under different weights? Why?

Compare the POY trees to the tree you created using Clustal and RAxML. Are they different?

If you at any point want to see some information about the analysis, use the command report.

`report(treestats)` - stored trees, their lengths under the current weights
`report(terminals)` - the names and numbers of the taxa in the dataset
`report(asciitrees)` - all trees in memory in ascii graphics

If you want to delete all stored trees from memory, there is no specific command for that, but `select(best:0)` will do the trick.

If you want to redo a command you typed in earlier Ctrl-p will give you the last command in the history.