Maximum Likelihood methods

1 Maximum Likelihood with PhyML

1.1 Overview of PhyML

You can download the binaries for PhyML from here. Alternatively, you can download the software Seaview, which includes a version of PhyML in it.

[Please read first] Let's start with a quick overview of how PhyML works. The program uses a heuristic search to move away from a starting tree. Instead of estimating branch lengths and parameters of the model after each swapping, it does the following

- start from a Neighbor-Joining tree
- make a NNI or SPR (or both) swap, keeping all model parameters and branch lengths the same
- keep the topology and model parameters the same, but update local branch lengths only
- keep the topology and branch lengths the same, but update model parameters
- do this until you do not improve the likelihood of the tree anymore

It starts from a Neighbor-Joining tree, which gives a good approximation of the true phylogeny as starting point of the heuristic search. PhyML uses a very quick way of calculating the likelihood, but can represent a dangerous shortcut when the data is not informative enough.

1.2 Running PhyML

During this practical, we will use data sets of clownfishes based on five DNA regions of length of 15 Kb from the nuclear genome (chromosomes 2, 8, 10, 18 and 22) and the full mitochondrial genome. You can find the data on the course web site (see below).

PhyML reads files that are in the extended phylip format (see document on file format on the course web page). You can transform a fasta file into this format using R (libraries: ape – read.dna and write.dna; seqinr – read.alignment). Here's an example:

```
library(ape) #use install.packages("ape") if you do not have it
dna<-read.dna(file="align_chr02_15kb.fasta", format="fasta")
nbSites<-dim(dna)[2]
write.dna(dna, file="align_chr02_15kb.phy", format="sequential", nbcol=nbSites,
    colsep="")
```

Before starting PhyML, **make sure to copy a version of the program where you have your data set**. It will then be easier to use the program.

▶ Explore first what PhyML can do with the file clownfish_mtdna.fasta (you need to change it into phylip format) containing part of the mitochondrial genome.

1. Start PhyML. A prompt 'Enter the sequence file name '>' should appear. Enter the file name containing the data set. If you use Seaview, the steps can be done through its specific GUI for PhyML.
2. If your data is in sequential format, change first that option by pressing 'I' once.
3. Give an ID to the run by pressing 'R' and adding any text relevant for you.
4. Press '+' and select the simplest model of evolution, i.e. JC69 by pressing 'M' several times.
5. Press 'R' once and then run the analysis by pressing 'y'.
6. Alternatively, you can run PhyML in command line like this:

```
phyml -i clownfish_mtdna.phy -m JC69 -b 0
```

7. What do the three resulting output files contain? Have a look in particular at the files finishing by 'stat.txt' and 'tree.txt'.
8. Open the tree file using FigTree or seaview or R (ape library).

We cannot know in advance what is the best evolutionary model for our DNA sequences. Fortunately, we can test the different models using Likelihood Ratio Tests (LRT). This can be done in R as follows (change the execname value to match what you have on your computer)

```
library(ape)
# run phyml sequentially on each DNA model

#change the execname based on what you have on your computer
phyml.test<-phymltest(seqfile="clownfish_mtdna.phy", execname="phyml_3.0_win32.exe -b 0 -o lr",
    append = FALSE, "sequential")

#takes a while...

#plot the results
plot(phyml.test)
```

▶ Find out which are the best models for each of the DNA regions available for the clownfishes:

- clownfish_chr02.fasta
- Clownfish_chr08.fasta
- Clownfish_chr10.fasta
- Clownfish_chr18.fasta
- Clownfish_chr22.fasta

and reconstruct a phylogenetic tree for each DNA region.

▶ For the mitochondrial region, did the topology change between the JC69 model and the best model selected? Try to explain why. What else is different between the two trees built with different models?

▶ For one of the DNA region (your choice), build a new phylogenetic tree by using the SPR branch swapping option instead of the NNI, which is selected by default. What are the consequences of setting this option? Compare the trees obtained and explain the differences if you see any.

▶ Assess the support for the different nodes of the phylogenetic trees built with each DNA region by running 100 bootstrap replicates. Use the best approach for the other options available in PhyML.

▶ Concatenate the six DNA regions using the following R code:

```
chr02<-read.dna("clownfish_chr02.fasta", format="fasta")
chr08<-read.dna("clownfish_chr08.fasta", format="fasta")
chr10<-read.dna("clownfish_chr10.fasta", format="fasta")
chr18<-read.dna("clownfish_chr18.fasta", format="fasta")
chr22<-read.dna("clownfish_chr22.fasta", format="fasta")
mtdna<-read.dna("clownfish_mtdna.fasta", format="fasta")

concat<-cbind(chr02,chr08,chr10,chr18,chr22,mtdna)
nbSites<-dim(concat)[2]
write.dna(concat, file="clownfish_concat.phy", format="sequential", nbcol=nbSites,
    colsep="")
```

Choose then the best options to estimate a single phylogenetic tree for the concatenated data set. Compare what you find with each gene trees obtained before.

2 Topology tests

The Maximum Likelihood analyses returned different topologies for the clownfish data set. The question is now to know if the incongruences are due to random errors in the tree reconstruction because of limited amount of data, or due to real biological differences between the DNA regions used. To do this, we can test if the topologies are significantly different given a data set. We will use the Shimodaira-Hasegawa test that is available in the *phangorn* library in R. You first need to install this library.

1. load the mitochondrial alignment into R using the function *read.dna*
2. load the six PhyML trees based on each DNA region using the functions *read.tree*
3. load the *phangorn* library and, for each tree, create a *pml* object (specific to the phangorn library) as follow (assuming that you named your tree *tree*, your DNA matrix *dna* and that the model of evolution is GTR+G+I):

   ```
   phylo1<-pml(unroot(tree), phyDat(dna), model="GTR", k=4, inv=0);
   phylo1.opt<-optim.pml(phylo1, optQ=TRUE, optBf=TRUE, optEdge=TRUE, optInv=TRUE, optGamma=TRUE);
   ```

4. repeat this process for the six DNA regions that we have (naming them, for example, $\texttt{phylo}_i\texttt{.opt}$)
5. use the function *SH.test* to perform the Shimodaira-Hasegawa test

   ```
   SH.test(phylo1.opt, phylo2.opt, phylo3.opt, phylo4.opt, phylo5.opt, phylo6.opt);
   ```

▶ Repeat the analyses for each DNA region (i.e. use each region as the DNA matrix and test if the trees explain their evolution equally well). Are the trees significantly different or not? Try to explain the results obtained?