PRACTICAL SESSION 2. ADVANCED PHYLOGENETIC RECONSTRUCTIONS.

We will start working with the same multiple alignment file derived in the previous session (bacteria_orthologs.fas or bacteria_orthologs.meg) and use it to obtain the corresponding phylogenetic trees using two character-based methods, maximum likelihood and Bayesian inference. We assume that the best model of evolution for this multiple alignment has already been determined (using jModeltest or its implementation in MEGA). Apparently, the model turned out to be TN92+I+G and, although the exact parameters were obtained, in the next applications we will just use the model and not the parameter values.

Maximum likelihood.

There are several possibilities, including using the ML implementation in MEGA (Analysis → Phylogeny → Construct/Test Maximum Likelihood Tree). However, we will use one of the two most popular and thoroughly tested implementations of ML: PHYML. The other software, (RAxML, is slightly more cumbersome to run and can be obtained in many different flavors from http://exelixis-lab.org/web/software/).

PHYML can be downloaded from http://www.atgc-montpellier.fr/phyml/binaries.php The current version is 3.0 and users are advised to read the manual (available from Recursos in Aula Virtual) and the manuscripts by Guindon & Gascuel (2003) and Guindon et al (2010).

PHYML can be run in an interactive form or from the command line. We will use the former method, as it allows us to explore the different options, but the manual describes how to obtain the same run by using the command line.

The input is a multiple alignment file in Phylip (interleaved or sequential) or Nexus format. These formats can be derived from the initial Fasta or MEGA format using the Export Data tool in MEGA. For this, select the complete display (all positions) and the Phylip 3.0 format. In order to retain the complete names of the species, it is better to use the sequential format, because Phyml, unlike Phylip, does accept species names longer than 10 characters. This has to be done by manually editing the export file before (or after) saving it with an appropriate name (e.g., bacteria_orthologs.phy). This file can be located in any directory of the computer as long as the complete path is provided to Phyml when requested. In any case, it is advisable to place a copy of it in the same location as the executable file.

The program is launched from the command line or by double clicking its icon (windows). The first thing it asks for is the input file name: [$PATH\]bacteria_orthologs.phy

The first screen shows the **Input Data** menu, with options to move to other sub-menus, specify the type of characters (DNA/Amino Acids), the input file format (change to SEQUENTIAL by typing I), etc. Each letter acts as a toggle that allows changing between alternatives. Once the selection for each sub-menu has been done we can move to the next sub-menu (+) or launch the analysis (Y). We will move now to the next menu (type +).

We enter now the **Substitution Model** menu.

[M] allows us to toggle among different models for nucleotide substitutions. The first model corresponds to HKY85, and it is possible to select any among: JC69, K80, F81, HKY85, F84, TN93, GTR, and a custom model to be specified by the user. The model selected in jModeltest was T92+I+G, which corresponds to the Tamura 3-parameter model with gamma and invariants. This is not included among the predefined models of Phyml and we have two alternatives: (i) use the custom model option and give the parameter estimates obtained with MEGA, or (ii) use a very similar model, such as TN93 (Tamura-Nei 1993), which also considers differences between the two types of transitions and transversions but does not take G+C bias into account. We will use the second option.

Once the TN93 model has been selected, we notice that the options available in the menu have changed to accommodate the different possibilities of the selected model. In this first run we will optimize the equilibrium frequencies, estimate the transition/transversion ratio as well as the proportion of invariable sites and the parameter of the gamma distribution (using 8 substitution rate categories). The remaining options are left as the default, but the manual should be read to be aware of our choices. Once finished we will hit '+' again to enter the next sub-menu.

We now enter the **Tree Searching** menu.

In the first run we will usually want to optimize the tree topology, which is the default option. The starting tree can be obtained by neighbor-joining (BioNJ to be precise), maximum parsimony, or provided by the user. This is only advisable when there are many sequences and we have already worked out the best topology previously.

Finally, we will be using a heuristic search (see Lesson 3, in the Theory folder), and we are offered several options. The default is NNI (Nearest Neighbor Interchange, a quick and dirty alternative useful in simple cases). Other options available are: SPR (Subtree Pruning and Regrafting, described as slow and accurate) with or without random starting trees, or a combination of NNI and SPR. For practical reasons, we will use the default NNI. We move to the next sub-menu hitting '+' again.

We are now entering the **Branch Support** menu.

We have essentially three possibilities here: (i) perform a non-parametric bootstrap analysis (hit B once to select this and then the number of replicates and whether we want to keep the trees and statistics in a file), (b) perform an approximate likelihood ratio test (this is the default option, see the manual for details of the three alternative possibilities), or (c) do not perform any test (type A to remove the default option so that both B and A show "No" as their alternative). We will use the default (SH-like support), so we can explore the results file.

We are now ready to launch the program (or we can navigate again through the menus by hitting '+' and '-'). To launch the program we hit 'Y'.

The programs shows its progress on the screen and, depending on the computer and analysis it may run from a few minutes to several weeks… This particular case should be finished in less than 5 minutes. (In my computer it took 0h2m7s)

The output is stored in two files whose names correspond to that of the input file with suffixes *_phyml_stats.txt and *_phyml_tree.txt

The first file gives all the details about the run we have just performed, including details on the input, model and choices we have made. It also prints out the parameter estimates according to our selection.

The tree file is a direct Newick formatted file with branch-length and node support estimates. It can be visualized with the TreeExplorer option of MEGA (Analysis → UserTree → Display Newick Trees).

Bayesian inference

We will use MrBayes version 3.2.2 to derive a phylogenetic reconstruction of these sequences using Bayesian inference. Unfortunately, there is not a single reference that can be recommended for learning the fundamentals of this method. A good reference list can be found in the manual (available in Aula Virtual). I have also uploaded a "quick reference" for the command line instructions of MrBayes.

The program is launched from the command line or by clicking its icon in windows. In both cases, the program presents a prompt for command line instructions along with some very brief instructions on how to get help. The easiest way to run the program is by launching a predefined file which will include the data, the model, and the run specifications.

To build this file, we will start with the sequential Nexus (Paup 3.0) file exported from MEGA previously. We will make some changes to fulfill the specific requirements of the Nexus format as implemented in MrBayes. The headline should look like:

```
#NEXUS
begin data;
   dimensions ntax=37 nchar=486;
   format datatype=DNA missing=? gap=- matchchar=.;
   matrix
```

The multiple alignment will come next

```
Treponema_pallidum
ATGGCGGCGTT------------------------GAGTAATGAACAGAT  (etc.)
GAAGTCAA------GTAA
;
End;
```

The last semi-colon and the "end;" are essential because it indicates the end of the data matrix. Without it, the program would keep trying to interpret the following symbols as data…

We type now the instructions for mrbayes. We use two blocks, each will be finished by "end;"

```
begin assumptions;
options deftype=unord;
end;

begin mrbayes;
lset nst=6  rates=invgamma  ;
mcmc ngen=1000000 printfreq=1000 samplefreq=100 nchains=4;
[sumt outputname=TriatomaITS2.nex burnin=500
contype=halfcompat;]
[sump outputname=TriatomaITS2.nex burnin=500 hpd=yes;]
end;
```

We save this file and name it “ortologos.nex”. A copy of it will be placed at the same directory than the executable file.

The execution is launched by typing: exec ortologos.nex

The progress of the run is shown on the screen. We have asked for 1,000,000 generations, which will take a few minutes, so we will launch the program first and then will consider what we have asked for (you should do it the other way round).

The interesting choices are specified in the **mrbayes** block of the Nexus file. We start by specifying the nucleotide evolution model, which in this case allows for a GTR model with invariant sites and gamma distribution. The parameters of this model (7 parameters) will be evaluated during the run using the default priors (Dirichlet with parameter 1.0 for the rates, exponential with initial 2.0 for the gamma shape parameter, and uniform in [0.0 – 1.0] for the p-invar value, see below]. In MrBayes it is not possible to specify a T92+I+G model, so we opt for a more general one that encompasses it.

The next line sets the parameters for the Markov chain Monte Carlo execution and launches it. Specifically, we indicate that the run will proceed for 1000000 generations and parameter values will be sampled every 100 generations and printed on the screen every 1000 generations. In addition, each of the two runs that will be launched will consist of 4 chains (one cold, three hot), to allow for a better exploration of the space parameter. These values are just illustrative and each case should be studied and run for as long as necessary to attain convergence.

The next two lines are commented (enclosed between brackets). They invoke a pair of built-in instructions to summarize and analyze the parameter values and phylogenetic trees obtained in these runs. We are asking for summarizing values after discarding the first 500 values as recorded in the run (out of 10,000 = 1,000,000 / 100). This is a “blind” choice, and we prefer to evaluate these parameters with the help of other programs. We will use Tracer for evaluating the parameters and TreeAnnotator for the trees. The resulting tree will be visualized with FigTree.

Combining trees produced by MrBayes into a single file. There is no tool to do this easily, but there are two alternatives. One is just to use a text editor. Remove the burnin from each file separately, combine them and save them as a single file to be processed with TreeAnnotator. Alternatively, for those more computer-geeks, you can use the following shell script in Linux or Mac:

```
#!/bin/bash
#change working directory
cd /file/path/to
#gets the taxon block from one of the files (in this case the first
one) and makes a new tree file called "combined.t"
grep -v -e "tree rep" -e "end;" run1.t > combined.t
#searches for the trees using the term "tree rep" in both tree files
and dumps them into alternate lines in the combined file
paste -d"\n" <(grep "tree rep" run1.t) <(grep "tree rep" run2.t) >>
combined.t
#adds the closing nexus code
echo "end;" >> combined.t
```

You can use TreeAnnotator for removing the burnin and summarizing the remaining trees.

We first inspect the convergence of the parameters in the two runs, both separately and combined. We will launch Tracer and open the *.p files of those runs. It is possible to combine more than two runs in one single analysis. We have to pay attention to the ESS (Estimated Sample Size) for each parameter, checking that all values are >200. If this is the case they are shown in black color, whereas orange and red are used for values below that threshold. In this case, both runs attain very good convergence and, in principle, they could be analyzed separately. If we take the first run, we can determine that the burnin can be as low as the usual 10% and use this value for the analysis of the corresponding trees.

In light of these results, we will use only the trees produced from run 1 to analyze the phylogeny of this gene. We will launch TreeAnnotator, open the tun1.t treefile and apply a 10% burnin (this corresponds to 1000 trees, because the treefile contains 1000000/100 = 10,000 trees). We select a name for the output file and run the program. The Log of the run provides information on the total number of trees read and kept for analysis and a summary of the results.

Next we open the resulting summary file in FigTree. This is a very complete program for visualizing and annotating phylogenetic trees, but we will pay attention only to its capabilities for displaying the posterior probabilities of the nodes in the consensus tree obtained from the 9000 trees kept for analysis in TreeAnnotator. These values can be visualized by clicking on “Node Labels” and then selecting “Display: posterior”. The values shown next to each node can now be compared with the SH-like support values (or bootstrap, should we have used this method) obtained in Phyml or the bootstrap values for the Neighbor-Joining tree obtained in MEGA.

The tree obtained by the three different methods can be summarized into one single tree (I usually use the ML tree) with the support for the nodes introduced by editing the corresponding figure in PowerPoint or other graphics editor.