Protein phylogenetics

Robert Hirt

PAUP4.0* can be used for an impressive range of analytical methods involving DNA alignments. This, unfortunately is not the case for estimating protein phylogenies. Only protein parsimony analyses can be performed in PAUP (a very simple distance method based on straight similarity can be performed as well but it is not recommended, particularly for divergent sequences).

For sophisticated distance and maximum likelihood analyses we need to use alternative programs. The following practical aims at making you familiar with three sets of programs that can be used for protein distance and protein maximum likelihood analyses, they include several programs from PHYLIP, and also TREE-PUZZLE and MRBAYES.

PHYLIP and TREE-PUZZLE (or PUZZLE) can both read the PHYLIP data format. MRBAYES use the Nexus data format, as does PAUP. PHYLIP and PUZZLE have similar and straightforward menu-driven options whereas MRBAYES makes use of a command line interface or command blocks. PHYLIP, developed by Joe Felsenstein, is one of the most comprehensive packages for phylogenetic analyses. It contains a broad range of methods for DNA (parsimony, distance and maximum likelihood) and protein analyses including protein distances and a protein parsimony method.

During these exercises you will be using several programs available in the PHYLIP package in an effort to infer protein distance trees and perform bootstrapping. For protein maximum likelihood analyses we shall use the program TREE-PUZZLE5. For Bayesian analyses you will use MRBAYES 3.

In addition, PUZZLE can also be used in conjunction with PHYLIP to perform more complex distance estimates. Of particular interest in PUZZLE is the broad range of protein evolutionary models with several matrices (e.g. PAM, JTT, BLOSUM62 and WAG) and the possibility to incorporate a correction for rate heterogeneity between sites using a discreet gamma shape parameter with the possibility of assuming a fraction of constant sites.

The datasets you will be using are:

1. An alignment of 6 taxa with 760 aligned amino acids positions. It is a subset of the alignment of the largest subunit of the RNA polymerase II that we have analyzed in detail to investigate the phylogeny of Microsporidia (as discussed during the lectures - see also Hirt *et al*. 1999 `http://www.pnas.org/cgi/content/full/96/2/580` for more details). It is in the PHYLIP format which can be used by PHYLIP and PUZZLE you shall be using during the first part of the practicals.

2. There are two additional small datasets that were produced by simulation and that will be used for the Bayesian analyses. These are in the NEXUS format with a MrBayes command block. The command block will instruct the program to perform analyses with a specified evolutionary model and values that determine the way the chains are performed. It is also suggested to compare the Bayesian results with PHYLIP and PUZZLE results obtained for the same datasets. Hence the two datasets are also found in the PHYLIP format. One dataset was evolved on a resolved tree whereas the other dataset was produced using a star tree. The support values that you will obtain for these two datasets should reflect this difference.

PHYLIP programs

PHYLIP programs can be run on several platforms including UNIX and Macintosh machines. Programs can be used to perform parsimony, distance and maximum likelihood analyses of both DNA and protein datasets. There are also two programs for bootstrapping and calculating majority-rule consensus trees. For DNA analyses PAUP has essentially superseded PHYLIP in terms of the diversity and complexity of DNA evolution models. PHYLIP however is still very useful for protein analyses. During these practicals you will be using the following programs from PHYLIP:

PROTDIST calculate pairwise distances from protein alignments, it allows the calculation of distances with several mutational data matrices including the PAM matrix.

NEIGHBOR calculate a neighbor-joining (NJ) tree from a distance matrix. It does not assume a molecular clock. NJ trees are heuristic estimates of minimum evolution trees but no alternative trees are compared (NJ is an algorithm and does not have an objective function), a unique tree is always produced without any idea of the quality of the tree. Its main advantage is its speed of execution, which might be considered to be a significant advantage when numerous taxa have to analyzed.

FITCH calculate a least-square tree from a distance matrix. It does not assume a molecular clock. If the

distances are additive or close to be additive the program should find the best tree. The program conducts a search of tree-space and the best tree which optimizes the difference between calculated distances and the inferred distances on the tree is selected.

Bootstrapping is recommended in an effort to obtain some information on support for nodes within the tree. If you analyze few taxa, say less than 40, it is advisable to use the FITCH program described below. If the distances are additive (or nearly additive), the NJ method can identify the same tree as FITCH (see below).

SEQBOOT produces bootstrap replicates from DNA or protein alignments. It is simply a method of producing many resampled datasets from the original one. The bootstrapped data is then used by one of the distance (or parsimony or likelihood) calculation programs.

CONSENSE calculate a majority-rule consensus tree, is used in conjunction with SEQBOOT, PROTDIST and any tree inference program such as FITCH or NEIGHBOR.

SEQBOOT and CONSENSE can also be used in conjunction with PUZZLE, (see PUZZLEBOOT information), to allow more complex models to be used to estimate protein pairwise distances.

Here is a list of the files to be used during the exercises

inf6.760 A PHYLIP file containing the protein alignment (6 taxa, 760 aligned amino acids).

d.res.nex A NEXUS file containing a protein alignment (8 taxa 500 aligned amino acids). This dataset was simulated on a resolved (res) tree with a gamma shape parameter to include site rate variation.

d.res.phy As above but in the PHYLIP format

d.sta.nex A NEXUS file containing a protein alignment (8 taxa 500 aligned amino acids). This dataset was simulated on a star (sta) tree with a gamma shape parameter and a fraction of invariant sites to include site rate variation.

d.sta.phy As above in the PHYLIP format

Exercise 1. Calculate a distance matrix with PROTDIST

Execute the program PROTDIST by typing:

```
protdist
```

You will be asked the name of a file, type:

```
inf6.760
```

This allows protdist to read the sequence alignment (note that if a file is called infile, all PHYLIP programs automatically read this file whether or not it contains the appropriate data).

You will see a selection of commands, type

```
p
```

to change the distance matrix to be used. Repeat typing `p` until you select the Kimura formula (it approximates the PAM matrix and has the advantage of being much faster then the PAM estimates) see the PHYLIP WWW page for more information on this.

The distances are written to a file called `outfile` (note that *all* PHYLIP programs will output *all* results to a file called `outfile` which will overwrite any existing `outfile`).

You can have a look at the distances by typing:

```
more outfile
```

To allow later inspections of these distance save the outfile under a different name by using either the UNIX command `mv` or the command `cp`:

mv This command means 'move'

cp This command means 'copy'

Here are 2 examples of their usage:

```
mv outfile newfile  # This command deletes 'outfile'
cp outfile newfile  # This command does not delete 'outfile'
```

To estimate a tree from these distances you should first copy the outfile as infile. Then execute the FITCH program by typing:

```
fitch
```

As above you will have several options. Typically the user selects an outgroup by typing

```
o
```

and select taxon number 6 (note that all trees inferred by FITCH are formally unrooted).

It is common to use the *jumble* option to perform tree search with random addition of taxa, by typing

```
j
```

and type an odd number, say 67, and then select the number of jumble to be performed, say 10

Type

```
y
```

to start the analysis.

The results of a PHYLIP tree search is found in two files. The `outfile` contains in a text file the diagram of the tree topology, the number of trees searched and the branch lengths. The `outtree` contains the tree topology and branch lengths (if a distance methods is used) in a format (Newick or New Hampshire format) allowing the viewing and manipulation of trees in a series several programs such as TREEVIEW and NJPLOT.

For a quick look at the result type:

```
more outfile
```

What is the phylogenetic position of the Micropsporidion *Vairimorpha necatrix* among the other eukaryotes?

Abbreviations: `Tri.vag`: *Trichomonas vaginalis*, `Ara.tha`: *Arabidopsis thaliana*, `Hom.sap`: *Homo sapiens*, `Sac.cer`: *Saccharomyces cerevisiae*, `Pla.fal`. *Plasmodium falciparum*.

Exercise 2. Bootstrapping with PROTDIST and FITCH

In order to assess the amount of support from the alignment (how many characters are actually supporting these relationships), you will perform a bootstrap analyses. This is done by going through the following steps:

1. Use SEQBOOT to bootstrap the same alignment as above, type: `seqboot` and type `y` for yes - 100 bootstrap replicates will be performed. Copy the `outfile` as `infile` for the next step. You can look at the bootstrapped data by using the function `more` (`more outfile`). You can see that some positions are present more than once.

2. Use PROTDIST to calculate the distances for all 100 bootstrapped replicate. Type `p` to select the Kimura formula (speeds up the process!!! Important for today's practicals since many of you are using the same computer) and `m` to inform the program that you have 100 replicates to be analysed. Type `y` to start the distance calculations. Once finished copy the outfile (containing all the distances from each bootstrap replicates) as infile for the next step: `cp outfile infile`.

3. Use FITCH to estimate the 100 trees from the 100 bootstrapped replicates. Type `m` to inform the program that you have 100 distances to be analysed, type `j` for random addition of taxa with 1 replicate, type `o` (the letter O) to select the taxon 6 as the outgroup, and type `y` to start the analysis. Copy the `outtree` as `intree` for the next step.

4. Use CONSENSE to calculate the majority rule consensus tree. Type `R` to inform the program that your trees were previously rooted and type `y` to start the calculation. Look at the results by typing `more outfile`. Give a new name to the `outfile` so that you can compare that results with the next analyses.

What is the support for the phylogenetic position of the Microsporidia and how does it compare with the other bootstrap values in the tree? Is the tree topology well supported for the method used?

PUZZLE

PUZZLE can be run on several platforms including UNIX and Macintosh. The program can be downloaded from the web. It was originally written by Korbinian Strimmer and Arndt von Haeseler.

Because of the intensive calculations needed for maximum likelihood analyses there is often only a limited number of taxa that can be analyzed at a time. To reduce this limitation these authors have proposed a quartet approach to allow faster maximum likelihood analyses of large datasets (numerous taxa).

Instead of searching trees with the full range of taxa they proposed a method where a tree is estimate through a two step process. The first step involves the calculation of the best tree with maximum likelihood for all possible combination of four taxa (quartets), a simple task since there is only three possible topologies for a quartet.

All **quartets** are then combined into a single tree for all *n-taxa* using a consensus method, if all quartets are compatible a unique tree will always be found. This is very unlikely with real data and different trees can be obtained. This is typically dependent on the order in which quartets are combined.

To avoid quartet sampling order effects the last step is repeated numerous times with a different quartet order each time, this is the so called **puzzling step**. After numerous, typically 1,000-10,000, puzzling steps a majority consensus tree is calculated from all reconstructed *n-taxa* trees.

The final tree summarizes the result by suggesting an *n-taxa* tree. Support for the tree topology is indicated by the resolved branching pattern and PUZZLE support values (maximum 100%, i.e. all n-taxa trees recovered the specific **clade** or **polytomies** for really poorly supported branching patterns). These values are not bootstrap values but can apparently correspond well to them in some situations.

Exercise 3. PUZZLE analyses of the inf6.760 dataset

Copy the data file `inf6.760` to `infile` using the command

```
cp inf6.760 infile
```

and start PUZZLE by typing:

```
puzzle
```

You can select the different settings as in PHYLIP by typing the relevant letter. So, for instance, you could select the outgroup No. 6 and start the analysis by typing `y`. This would perform a puzzle analysis with the auto-selected model with no correction for rate heterogeneity. Puzzle produces three **output** files:

outfile This file contains the majority of the information such as the features of the dataset, the settings used, the result of a chi-squared test for sequence heterogeneity, the maximum likelihood value for each pairwise distances (not to be mistaken with the distances between taxa on the final consensus tree estimated with ML), the consensus tree with attached puzzle support values (how many times the n-taxa trees recovered the shown relationships), and additional data. Use `more outfile` to scroll through the results. Compare in particular:

- the ML pairwise distance values with the distances inferred using PROTDIST.
- the tree topology and the puzzle support values with the consensus tree obtained with bootstrapping in PHYLIP. Are trees and support values significantly different?

outdist contains the ML pairwise distances with the used model.

outtree contains the tree topology with branch length and the puzzle support values. It can be opened in programs such as TREEVIEW and NJPLOT, for future modification, to ease observations, to get printer friendly formats or to produce figures for publication. Do not forget the change the name of these files for later reference, `outfile`, `outtree` and `outdist` files will be overwritten each time you do a new analysis!

Repeat the analysis by changing the settings in the following way. Select outgroup No.6, type

```
o
```

Select the BLOSUM62 matrix by typing the letter

```
m
```

until that model is chosen. Then, start the analysis by typing

```
y
```

How does the ML value of the tree change?

MRBAYES

MRBAYES was one of the first programs to implement protein Bayesian analyses. MRBAYES is also used for DNA Bayesian analyses as previously discussed. One of the main advantages of such analyses is that all taxa are included in the tree estimation (not quartets as with PUZZLE) and that a thorough parameter space search is performed, including the tree space during the analysis. Below we specifically deal with protein alignment analyses by presenting two simple examples. MRBAYES, like PAUP, uses a NEXUS format for input. You can use the command line once you open the program or use a MRBAYES block that would contains all the instruction to run a specific analysis. The MRBAYES block can be found in the same file as the data or in separate files. Here is an example of such a file, containing both the information for the protein alignment and the MRBAYES block.

```
#NEXUS
begin data;
 dimensions ntax=4 nChar=26;
 format datatype=protein gap=- missing=?;
 matrix
  Taxon_A vavygymaldvngsergnfpsyleli
  Taxon_B vaiagylalevdaadqgifhsylavi
  Taxon_C pphckylglevdgseqgsfpsylsii
  Taxon_D vaiyrfvkldvdaspqgglnsylkml
 ;
end;

begin mrbayes;
 log start filename=d.res.nex.log replace;
 prset aamodelpr=fixed(wag);
 lset rates=invgamma Ngammacat=4;
 set autoclose=yes;
 mcmc ngen=5000 printfreq=500 samplefreq=10
  nchains=4 savebrlens=yes startingtree=random
  filename=d.res.nex.out;
quit;
end;
[
begin mrbayes;
 log start filename=d.res.nex.con.log replace;
 sumt filename=d.res.nex.out burnin=100
 contype=allcompat;
end;
]
```

The top part of the file contains the information about the data. It is a subset of the accepted nexus format that can be used with PAUP. Such a data could be analysed by PAUP if one would like to do so. The only protein specific feature (outside the alignment itself) here is `datatype=protein`. The second part, after the first `end;` is the first MRBAYES block. It contains instructions for the evolutionary model selected and how to run the Markov chain. `log`: is for feeding the output shown on the screen to a file. This will record all the details of the settings used for the analysis. This is always good to have to check what was done. `prset`: contains the information about the protein matrix used, the model of amino acid evolutionary changes used to calculate the lnL of each sites on a given tree. This a fixed model, which values were established empirically from another dataset, see for example Whelan and Goldman (2001) `http://mbe.oupjournals.org/cgi/content/full/18/5/691` (this is not like the GTR model used for DNA where all values of the model – rate matrix – can be estimated). The only element that can be changed here is to correct the values of the model with the amino acid frequency of the dataset to be analysed. Unfortunately this version of MRBAYES cannot do that automatically — note that TREE-PUZZLE does that and shows these values in the output file. One could use these values in MRBAYES.

rates=invgamma sets the additional component of the model used to calculate the lnL and here we have included a gamma shape parameter with a fraction of invariant sites.

Ngammacat sets the number of discrete values for the gamma shape parameter. Here we have 4 categories.

set autoclose yes will automatically close the program once the mcmcmc has finished its run.

mcmc gives the details about how the mcmcmc will be run.

gen gives the number of generations (usually a high number, 100000 or more).

printfreq defines the number of times the result of a chain will be printed out to the screen (and the log file if used). To avoid a big log files a large number is usually chosen here (e.g. 500 or 1000).

samplefreq will determine how many trees are kept and the results of the analysis recorded in the output files (see below). These are the data that will be used to calculate the final consensus tree. One needs to choose a value that will produce random points during the chain so that each results are not auto-correlated with the previous/following one.

nchains determines how many chains are run in parallel. The number of 4 is typically used and allows to use different heat, that is a chain with different capacity to jump at different part of the parameters space (including tree topology space), the hotter the chain the more "jumpy" it is in the parameter space. This allows to reduce the chance of the chain to be stuck in a suboptimal region of the tree (and other parameters) space and not find the best tree(s).

savebrlens means that the branch length of each saved tree will be saved in the tree file. This will allow to calculate the branch length of the consensus tree.

startingtree=random means that the analysis will start with a random tree that will be improved during the mcmcmc.

file name=xxxx defines the name of the files that will contain the results of the mcmcmc. Each analysis produces two files with the following suffix: - `xxxx.p` contained the lnL value for each saved tree and the values of any estimated parameter of the model we choose — here the alpha (gamma shape parameter) and pinv (fraction of invariant sites) `xxxx.t` contains the tree topology and branch length of all saved trees during the mcmcmc In addition to these files a `xxxx.log` file will be produced if you asked for it, as described above.

Once the mcmcmc has been run, one needs to investigate the shape of the variation of lnL and other parameter(s) estimated during the chain. This will allow you to choose the *burn-in* that is the number of mcmcmc that have to be discarded prior the values have converged (in this case: the tree lnL, alpha and pinv) and which will be ignored for calculating the consensus tree. Once the burn in is determined, you will need to introduce that value in the second MRBAYES block that contains the instruction for calculating the consensus tree (with branch length). Note that this block is found between brackets [ ] that is this block will be ignored and only the first block with mcmcmc instructions will be followed if this is used. To allow the second block to be run one needs to edit the file and place these [ ] just before and after, respectively, the first block and remove them from the second, the second block will now be instructing MRBAYES what to do. The second block has the following instructions:

log this line instruct the output that is shown on the screen to be feed into a file. This file contains different information, including the tree topology with posterior values attached to branches.

sumt gives the name of the file that will contain the consensus tree topology and branch length and posterior values and the burn in value that you will have previously determined. This file can be opened with TREEVIEW for example to manipulate the tree for publication.

contype=allcompat will allow to save the fully resolved consensus tree, even for low supported branches.

The files produced from this analysis are: `xxxx.con`, the treefile with topology, posteriors (support values) and branch length, `xxxx.parts`, all the partitions of the consensus tree, `xxxx.trprobs`, give the probability for all the trees (that were recovered during the mcmcmc) used to calculate the consensus tree, `xxxx.con.log`, the log of the analysis.

Exercise 4. Perform Bayesian protein analyses and compare results with other methods

For this exercise you will use two datasets in two format, NEXUS and PHYLIP (four files) obtained by simulation using p4. The simulation was performed on:

- A resolved tree with a gamma shape parameter to include site rate heterogeneity. The files for this dataset are called:

    1. `d.res.nex` the file in a NEXUS format to be used with MRBAYES. It contains two MRBAYES blocks, that is a set of commands that will specify the way the mcmcmc will be run and with which protein evolutionary model and the way the consensus tree will be calculated
    2. `d.res.phy` same file as above in the PHYLIP format to be used with PHYLIP and PUZZLE

- A star tree with a gamma shape parameter and a fraction of invariant site to include site rate heterogeneity. The files for this data are called:

    3. `d.sta.nex` the file in the NEXUS format to be used with MRBAYES, it has two MRBAYES blocks as d.res.nex.
    4. `d.sta.phy` same file as above (3) in the PHYLIP format to be used with PHYLIP and PUZZLE

The following instructions will allow you to go through the process of running MRBAYES on these two datasets, analyse the result of the mcmcmc and choose a "burn in" to calculate the consensus tree with its posteriors. We suggest that you also try to analyse the same datasets with PHYLIP and PUZZLE, using the two PHYLIP format files described above and following the instructions of exercise 2 and 3. This will allow you to compare the different support values obtained with the different methods for these two datasets.

Running MRBAYES

Running mcmcmc chain will take some time so we suggest you to start up the chains for the two datasets `d.res.nex` and `d.sta.nex` at the beginning of the practicals. During the time these are running you can then perform PHYLIP distance analyses and the PUZZLE analyses. Since the

dataset file contain MRBAYES block you can start the analysis by instructing MRBAYES to read the `d.res.nex` and do the same with the `d.sta.nex` file. Typing the following command line

```
mb d.res.nex
```

will start up MRBAYES (mb) and read the content of the file `d.res.nex`, that is the alignment and the MRBAYES command block with all the instructions on how to run the chain. MRBAYES output will be shown on the screen and will feed into a file that can be looked at later on . Do the same for the `d.sta.nex` file on a different window. After a few generations of the mcmcmc chain, several files are produced. Type `ls`, list, (in other window if necessary) to list them to check that these files were produced indicating that all processes are running smoothly. The analysis will last a few minutes to several minutes depending on the available cpu and the speed of the machine.

Determine the "burn in" and calculate the consensus tree

After the last generation of the chain calculations are done you can plot the lnL versus the number of generations. You can do this in a spreadsheet or using the MRBAYES plot function. You should observe an increase of the lnL in early generations and then a stabilisation of the lnL values. This suggests that the chain successfully reached the region of optimal values of tree lnL and estimated parameters (here the alpha shape parameter and pinv, the fraction of invariant sites). Based on this graph you can choose the "burn in" that is the number of generation to ignore for the calculation of the consensus tree. To do so you have to edit the datafile. Open the file in a text editor and add "[" before and "]" after the first MRBAYES command block and remove these "[" and "]" found in the second command block found below the first one. Save the change and then type the command line

```
mb d.res.nex
```

This will read the datafile and the tree file containing all the trees produced during the chain but will ignore the trees of the "burn in". MRBAYES will produce a majority rule consensus tree file from the selected trees (all of the ones obtained by the chain after the "burn in") and partition matrix file. The consensus tree will have the posteriors, i.e. the support values for the different clades on that tree, the probability that these clades are correct. These support values, posteriors, correspond to the number of trees during the chain that recovered that clade. Repeat the same steps with the second dataset `d.sta.nex`.

Compare posteriors support values for clades with other support values

In addition you can perform PHYLIP distance bootstrap analyses and PUZZLE ML analyses (as described previously) and compare the support values obtained by bootstrapping and puzzling steps for the "resolved" and "star" trees datasets.

How do these support values differ?

Do they reflect the original simulation performed to obtain these data?