

Clann: Investigating Phylogenetic Information through Supertree Analyses

```
*****
*
*                               Clann  3.0.2
*
* web:  http://bioinf.may.ie/software/clann
* email: chris.creevey@gmail.com
*
* Copyright Chris Creevey 2003-2005
*
*****
```

James McInerney, National University of Ireland Maynooth.

This short practical will bring you step by step through the process of analysing a dataset using different optimality criteria and methods of investigating the data. For the purposes of this practical, a file called `real.ph` has been included with the distribution of the software. Please note that these are only suggestions that you may use when analysing data, there is no excuse for just following a recipe and not thinking about your data.

1) There are two ways of reading in a file of source trees into Clann. The first is to use the `exe` command at the `"clann>"` prompt. For example:

```
clann>exe real.ph
```

The second way is to place the name of the file as an argument when you start Clann. For example:

```
%> clann real.ph
```

This starts the Clann software and immediately executes the source trees. Then after the execution of the source trees file a summary of the number of trees etc is displayed to the screen. To find out what commands are available to you, you need to issue the **help** command. Do this now and read what appears on the screen.

2) Next it is necessary to decide which optimality criterion is to be used to find the best supertree. By default Clann will use the **dfit** criterion (most similar supertree method), however for the purposes of this tutorial we will begin with an MRP search. To choose the MRP criterion type:

```
set criterion=mrp
```

the following should display to screen:

```
"Scoring criterion set to matrix representation using parsimony  
(MRP)"
```

3) To perform a heuristic search of supertree space, you will use the following command:

```
hs;
```

Take a look at the options that are available to you using the question mark (?) placed after the command. Clann creates the matrix representation and writes it to a file called **coding.nex**. Clann then calls PAUP to perform the Parsimony step of the algorithm. The best tree is displayed to screen and also saved to a file called

`MRP.tree`. If PAUP* cannot be found by Clann on your system an error message will display asking you to execute the file `coding.nex` in PAUP* to get the final results.

4) Next, switch to the dfit optimality criterion using the command:

```
set criterion=dfit
```

5) Once again perform a heuristic search, but this time specify that only one repetition of the search is required (by default Clann will carry out ten repetitions) and also that you do not wish to carry out the default method of branch swapping (SPR), but rather, the less computationally-intensive NNI branch swapping method. When the search is over, you wish to save the resulting supertree to a sensibly-named file, such as `heuristicNNI.tre`

```
hs nreps=1 swap=nni savetrees=heuristicNNI.tre;
```

As stated earlier, after each search of tree-space, Clann will display the tree to screen in addition to writing it to file in two formats: The first is in nested parentheses format (which can be user-specified); The second format is written to a file called `supertree.ps` which is a graphical representation of the tree in postscript format.

6) To perform a YAPT test, you use the `yapt` command.

Take a look at the options available to you for this test. Find out what these options mean. Perform a short YAPT test. You should use no more than 10 reps (this is because of potential computational intensity, normally, you would perform much more

100-1000).

The result of this test is written to a file (which can be user specified) and contains the best trees found after randomizing the input data and the scores of these best trees. For the purposes of the test it is only necessary to record the scores. This gives a distribution of scores expected from random data, which can be compared to the real data.

7) To perform a bootstrap analysis, you will use the following command:

```
boot;
```

If further options are required (like the type of search to be used and the number of bootstrap repetitions) type something like the following:

```
boot nreps=10 swap=nni;
```

The results of this analysis are written to a file called [bootstrap.txt](#) (or the user-specified file). This file contains multiple trees, each a result from a bootstrap replicate. It is necessary to use a program like CONSENSE from the PHYLIP package (<http://evolution.genetics.washington.edu/phylip.html>) to summarise these results in a majority-rule consensus tree. You can start the consense program by typing its name.

Question: Do you think you should always bootstrap your input trees? Would it work well if there were 2 input trees where the taxon-overlap was poor?

8) Perform a search of tree space using your favourite method. Do you get a good tree? How do you know? Are random trees much worse?

9) Perform searches using the random data. Perform the YAPTP test, carry out a heuristic search of tree space. How does the random data differ from the real data in terms of phylogenetic signal? Can you tell the difference?