University of Bergen

BIO332 Phylogeny

Introducing PAUP* (last update May 2007 E.Willassen)

PAUP* is a state of the art program for phylogenetics. The Windows and Unix versions are operated on command lines or by batch scripts. You will be using pre-made scripts like the one coming with MODELTEST, and also learn how to make your own scripts. It is a good idea to read the the beginners guide first. The commands and options are well documented in the manual. An additional source of help is the FAQ pages at the PAUP*web site. It is highly reccomended that you read the paper by Swofford and Sullivan (2003) before we start the computer lab.

Here is what you want to learn for a start to operate PAUP*:

- handling data files
- the structure of NEXUS files and the various block formats
- logging runs
- computing pairwise distances
- basic commands for tree search with parsimony, distance, and maximum likelihood methods
- handling trees - displaying, saving, exporting, computation of consensus trees
- performing some standard tests

Importing / exporting data to / from PAUP

Import (non-nexus) data

Paup* may import the following file formats: FrePars, GCG MSF, Hennig86, MEGA, NBRF-PIR, Phylip 3.X, Simple text, Tab-delimited text. To import *myfile.phy* in phylip 3 format, the following commands should work.

```
Tonexus format=phylip fromfile=myfile.phy tofile=myfile.nex;
```

You may experience trouble because file formats are being revised. If so,try making nexus files with other programs. (For example from fasta to nexus with ClustalX)

Export data

`Export file=newnex.nex format=nexus interleave=no linebreaks=doswindows charsperline=1250;` [converts an interleaved matrix to a non-interleaved file with 1250 characters per line called newnex.nex]

Comment: if you want Winclada to read the nexus file it must be non-interleaved

Nexus file formats

You might find it convenient to use programmes such as Mesquite or MacClade to organise your data. If so, you should take the opportunity to study the resulting nexus files in an ordinary text editor. It will give you a good understanding of the structure of nexus files and how the information there is organised. For instance, take a look at Exercise1 to learn more about this.

Files always start with `#nexus`. Blocks start with `Begin` and stop with `End`. Notice the use of semicolon **;**. Data may be input in two slightly different nexus formats. One format has a taxa block and a characters block. The other has a data block instead:

A) a nexus file with a TAXA BLOCK and a CHARACTERS BLOCK

```
#NEXUS

Begin taxa;

        Dimensions ntax=4;
```

```
Taxlabels 'taxon1' 'taxon2' 'taxon3' 'taxon4'; [defines taxa / sequence names]

End;

[a taxa block is recommended, but not necessary if the taxa block and character
block are alternatively replaced by a data block (see below)]

Begin characters;

      Dimensions Newtaxa ntax=4 nchar=4;

Format datatype=standard symbols="0123" missing=? Interleave=no ; options
mstaxa=polymorph ; [defines datatype and symbols for character states. The last
expression allows for polymorphic states]



CHARLABELS

[1] 'head'

[2] 'shoulder'

[3] 'knee'

[4] 'toe' [defines character labels for four characters]

;

STATELABELS

[1] 'long_scull' 'short_scull' [labels two states for character #1]

;

MATRIX

taxon1      0000

taxon2      0(01)10 [notice polymorphism (01) in character 2]

taxon3      1321

taxon4      1412;

End;
```

Note: Make sure to use unique names for each taxon. Avoid using hyphen, space or similar symbols in taxon names.

B) A nexus file without taxa and characters blocks, but with a DATA BLOCK instead

```
#NEXUS

Begin data;

Dimensions ntax=4 nchar=10;
```

```
Format datatype=dna missing=? gap=- interleave; [notice interleave as opposed to
noninterleaved]

Taxon1      AGCTA

Taxon2      AGGTA

Taxon3      AGCCA

Taxon4      AACCA

Taxon1      TGCTT [additional block of characters, i.e. data are interleaved]

Taxon2      AGGTT

Taxon3      AGCCA

Taxon4      AACCA;

End;
```

Other blocks

may be added:

A SETS BLOCK defining character sets and character partitions.An ASSUMPTIONS BLOCK defining for instance character transformations types.

A TREES BLOCK defining trees in NEWICK, nexus, or altnexus formats

A PAUP BLOCK with commands. (PAUP BLOCKS can be stored in separate files for batch processing of data files.)

Defining sets

```
Begin sets;

      charset 12S = 1-605; [mitochondrial small subunit rDNA]

      charset cytB = 606-1147; [mitochondrial coding cytochrome B gene]

           charset 1st = 606-1147\3; [defining 1st codon position]

           charset 2nd = 607-1147\3; [defining 2nd codon position]

           charset 3nd = 608-1147\3; [defining 3rd codon position]

         charpartition source = 1:12S, 2:cytB; [defining two partitions, here collectively called source]

      taxset birds = 'taxon1' 'taxon2'; [define sets of taxa]

      taxset reptiles = 'taxon3' 'taxon4';

      end;
```

Running PAUP

Log the run

```
begin paup;

log file=log.txt start; [start logging to the file log.txt]

……

log stop; [stop logging]

log start append; [starts logging again and appends results to the file log.txt]

log file=log.txt replace; [overwrites the file log.txt]
```

Analyse data

Compute observed and adjusted distances (pairwise)

```
Begin Paup;

set criterion=distance;

dset distance =abs subst=all; [calculate pairwise numbers of substitutions]

savedist format=onecolumn file=subs.txt; [save results in one column to file subst.txt]

dset subst=tv; [calculate pairwise number of  transversions only]

savedist format=onecolumn file=no_tv.txt; [save results in one column to file no_tv.txt]

[dset subst=ti;] [calculate pairwise number of transitions only]

[savedist format=onecolumn file=no_ti.txt;] [save results in one column to file no_ti.txt]

dset distance =F84 subst=all; savedist format=onecolumn file=F84.txt; [save F84 distances]
```

*Comment: This procedure can be used to produce saturation plots. Tab delimited text files can be imported to MS-Excel.

Compute trees

Parsimony (MP) is the default criterion, but must be reset if you just did a run with a different criterion

```
begin paup;

set criterion=parsimony;
set maxtrees = 500 increase=no;

hsearch nreps=100 addseq=random swap=tbr; [do 10 heuristic search replicates with random addition of taxa and TBR branch swapping]

savetrees file=mp.tre brlens=yes; [save trees with branch lengths to file named mp.tre]

end;
```

Distance methods

```
begin paup;

set criterion=distance;

dset distance=logdet objective=me negbrlen=setzero;  [use logdet distance and minimum evolution. see manual for other distances]

hsearch nreps=10 swap=tbr; [do 10 replicate searches, and use TBR swapping]

savetrees file=me.tre brlens=yes;
```

Maximum likelihood ML

```
begin paup;

set criterion=likelihood;

lset nst=2 basefreq=empirical rates=gamma ncat=4; [sets the model for base substitutions*]

lset tratio=estimate shape=estimate; [estimate the tr/ti ratio and the alpha parameter of the gamma distribution]

lscore; [show the estimated log likelihood score for the tree]

lset tratio=previous shape=previous; [use the previous estimates for the tr/ti ratio and the alpha parameter]

hsearch nreps=1 swap=tbr start=1; [do a heuristic search with TBR branch swapping starting from tree 1 currently in memory]

savetrees file=ml.tre brlens=yes; [save trees with branch length to a file named ml.tre]

end;
```

*See Models for different model settings

Models

Evolutionary models and how to define them for PAUP* is explained in the chapter called Model testing. Also, see the file *modelblock* coming with Crandall and Posada's *Modeltest* or Johan Nylander's 'Models in PAUP* and

MrBayes' (link).

Stepmatrix

Stepmatrices can be defined to specify models of evolution in parsimony. See more about parsimony models in the chapter called Parsimony. Among the assumptions in the stepmatric below is that it takes two evolutionary steps (cell with red number) for a character to evolve from state 1 to state 3. This is how we make a stepmatrix that we choose to name `mystepmatr`:

```
Begin assumptions;

        usertype mystepmatr = 4 [may be applied to morphological characters with 4 states]

    [states] 0       1       2       3

    [0]      .       1       2       3

    [1]      1       .       1       2

    [2]      2       1       .       1

    [3]      3       2       1       .

  ;

End;
```

In order to make the stepmatrix active, we need to provide character type commands (ctype) like the following to PAUP*:

```
begin paup;

ctype mystepmatr:10-15 18; [apply 'mystepmatrix' to character nos 10-15 and
18:]
```

Trees

Tree file formats

Notice that trees can be defined in two ways:

A) with a taxa label specification>

```
Begin trees;

Translate

      Taxon1,

      Taxon2,

      Taxon3,
            Taxon4;

        Tree*mytree =  (((1,2),3),4);
```

B) without a taxa label specification

```
        Begin trees;

        Tree*mytree =  (((Taxon1,Taxon2),Taxon3),Taxon4);
```

If you have a huge tree file, alternative A) takes less space. You may convert to alternative B) with the **altnex** option (see **Saving trees**).

C) trees may also be saved with branch length and node support data:

```
        Tree*mytree =
        (((Taxon1:0.4500,Taxon2:0.4378):0.5607,Taxon3):0.7860,Taxon4:0.4399);
```

Consensus trees

`Contree 5-. / strict=no majrule=yes grpfreq=yes indices=yes treefile=`*`my_cons`*`.tre;` [computes a majority rule consensus tree and saves it. Displays a frequency table of groups occurring together, and consensus indices]

Saving trees

`Savetrees file=`*`myfile`*`.tre format=nexus brlens=yes;` [Saves all trees with branch lengths. The nexus tree file begins with taxon labels referring to *numbered* taxa in the parenteses]

`Savetrees from=6 to=8 file=`*`myfile`*`.tre format=altnexus brlens=yes;` [Saves trees 6-8 with branch length. Altnexus tree files have taxa names included in the parenthesis notation. You may want this format to communicate with other programs.]

Deleting trees

`Cleartrees;` [erase all trees from memory]

Show trees

`Showtrees` 1-.; [show all trees in memory]

Describe trees

`Describetrees / diagnose` [show tree / and character diagnostics]

Loading trees

`Gettrees` file=myfile.tre; [loads previously saved trees in *myfile.tre* into memory]

trees already in memory will be replaced if you don't use a *mode* option (see below).

```
Gettrees file=myfile.tre mode=7; [appends trees in the file to those already in memory]
```

note: a tree file may contain several tree blocks. To load all blocks, include the allblocks=yes option:

```
Gettrees file=myfile.tre allblocks=yes mode=7 storetreewts=yes;
```

Tests

Bootstrap test

```
Bootstrap nreps=250 treefile=myboot.tre [set up 250 replicates and save trees from each replicate to myboot.tre]
```

Prefered search conditions can be specified in the same command, for instance as follows:

```
search= heuristic/ addseq=random nreps=100; [do heuristic search, and do 100 addition sequence replicates in random order]
```

The bootstrap consensus tree is printed to the screen (and to the log file, if you are logging). One or several bootstrap tree files may be used to compute branch support with the GETTREES and CONTREE commands at any time. However, if you want to save the consensus tree with the bootstrap support values in the tree file, the following command must be run immediately after the bootstrap command:

```
savetree from=1 to=1 file=myboot_con.tre savebootp=nodelabels [saves bootstrap proportions as node labels in the tree named myboot_con.tre]
```

Homogeneity partition test (ILD-test)

Assuming you have defined a character partition called 'gene' (see charpartitions

```
Hompart partition=gene search=heuristic nreps=100 seed=123;
```

LRT (Likelihood ratio test>) (here applied as molecular clock test)

```
Lset nst=2; hsearch; [compute ml tree with your chosen model]
```

```
describetree / plot=ph brlens=yes; [show phylogram, list parameters, and branch lengths]
```

Take note of the likelihood score for the unconstrained tree: -lnL$_{\mathrm{noclock.}}$. Then select outgroup and root the tree.

```
outgroup 1; [select one or more outgroups by typing taxon numbers or names]
```

```
root; [root tree]
```

```
lscores / clock=yes; [compute likelihood score under molecular clock constraint]
```

```
describetree / plot=ph brlens=yes; [the 'phylogram' is now an ultrametic tree, list parameters, and linearised branch lengths]
```

Take note of the likelihood score for the constrained tree: -lnL$_{\mathrm{clock}}$

Compute times two the difference in lnL: d(-lnL) = 2 (-lnL$_{\mathrm{noclock}}$- (-lnL$_{\mathrm{clock}}$))

Use the programme MODELTEST or examine your result for significance in a chi square table. The test has (n-2) degrees of freedom, n = number of terminal taxa. The null hypothesis is that sequences evolve with similar rates, *i.e.* if the lscores are significantly different the clock hypothesis is rejected.