FILE FORMAT

1 File format

Most important advice when writing your data files: stick to letters, figures and '_' in taxa names. Do not use other weird characters (i.e. avoid spaces within names, '-', '#', '/', '*', parentheses, etc.), and keep them as short as possible. It will then be far easier to translate from one file format to another and to use different software.

1.1 fasta

This is the most used format for unaligned sequences. There is no need to specify either the number of sequences present in the file, nor the length of each sequence. The sequence being usually unaligned, each has potentially a different length.

Each sequence is represented by two or more lines. The first line contains the name of the sequence. The first character of this first line has to be a '>', followed by any numbers of characters representing the name or description of the sequence. The sequence is then given on any number of lines.

New sequences are simply concatenated by repeating this scheme.

Example

```
>Homo sapiens, cytochrome B
AACAGTTAGACGTGACTTAGGATAG
AAGCCCCCCCCATTATTTATAT
>Gorilla gorilla, cytochrome B (partial)
AAAACCGTTAAGGATTTATTTATTATACCCAG
```

1.2 phylip

This format can be read by almost any software, and thus is very useful. The first line of the file starts with a space (or tab), followed by the number of sequences present in the file, another space (or tab) and the total number of characters in any given sequence (they must have the same length, i. e. be aligned). The next lines contains **exactly ten** characters for the taxa name followed by the sequence itself, which has to start at the 11th characters. Longer taxa names have to be cut down to ten characters. For shorter names, spaces have to be added so that **the sequence starts at the position 11**.

It is a very strict format in its original layout (for the phylip package). No special characters are allowed within the taxon name (e.g. space, '-', parentheses, etc.). This format has been adapted by a number of software (e.g. paml and phyml), and the authors have relaxed somewhat the number of characters for names by defining a (or two for paml) space(s) as the separator between taxon name and the sequence itself.

The sequences can either be sequential (i.e. taxon name and whole sequence is on a single line), or interleaved (i.e. taxon name and part of the sequence is repeated in blocks).

Example sequential

```
 3 20
HomosapienAACAGTTAGCCCCAGCTAGC
GorillagorAATAGTTTTCCCTCGCTTTC
Pansp      AACAGTTATCCCCACTCCTC
```

Example interleaved

```
 3 20
HomosapienAACAGTTAGC
GorillagorAATAGTTTTC
Pansp     AACAGTTATC

HomosapienCCCAGCTAGC
GorillagorCCTCGCTTTC
Pansp     CCCACTCCTC
```

1.3 nexus

This format has become very popular because it is associated with popular software such as paup, macclade, mesquite, mrbayes, BEAST or R. It's an evolving format, and only the simplest description (accepted by all software) is given here.

At the top of the file, a defining `#NEXUS` starts the format layout. Then, the file is separated in sections beginning with the word `begin` *`keyword`* and ending with `end;`. The most important section is the data itself, which is defined with `begin data;` (note and don't forget the `;` at the end). Then two lines are necessary to describe how many sequences and what type of sequences you have, one defining the dimensions, and one defining the format of your sequences. Finally the word `matrix` indicates the start of the real data. After your sequences, a `;` indicates the end of the data sets. To finish, a `end;` is added to terminate the data section.

The nexus format accepts either sequential or interleaved layout. The taxon name has to be separated by at least one space, but more than one can be used.

With this format, it is possible (and very useful) to specify commands that the program will have to execute on the data set. This is done by adding a `begin paup;` or `begin mrbayes;` section. Then each line will represent a command that the software can understand. We will see these command sections later when we are using them. Don't forget to add an `end;` to finish the command section.

For a more detailed description of this format, you can refer to the PAUP manual on the web site (http://www2.unil.ch/phylo/teaching/phylo/paup_manual.pdf).

Example

```
#NEXUS

begin data;
dimensions ntax=3 nchar=10;
format type=nucleotide missing=? gap=-;
matrix
Home_sapiens AACAGTTAGC
Gorilla_gorilla AATAGTTTTC
Pan_sp AACAGTTATC
;
end;
```

1.4 Tree formats

The hierarchical tree structure is represented by parentheses, and is often referred as 'newick' format. Some program, such as PAUP or MrBayes, add more details on top of this format.

Two terminal taxa grouped by a node in the tree are enclosed in one level of parentheses and are separated by a ',' and a ';' is added to specify the end of the tree description. So if taxa A and B form a clade, the newick format will be (A, B);. If a third taxon C forms a new clade with taxa A and B, a second level of parentheses will enclose the previous node and the new taxa, a ',' being added to separate the two sides: $((A, B), C)$;. New levels of parentheses are added everytime we go down the tree structure. For example, if two already formed nodes $((A, B), C)$ and (D, E) are grouped, the tree will be $(((A, B), C), (D, E))$;. Note that this last tree is identical to this one: $((D, E), (C, (A, B)))$;. The order of the taxa names does not matter at all, the structure being conserved in the two examples.

Polytomies are simply formed by removing the corresponding level of parentheses, keeping the ',' inbetween. For example, in the last tree with five taxa, if the node (A, B) is not resolved, the tree reduces to $((A, B, C)(D, E))$;.

Branch lengths are specified by preceding a number with ':', and is attached to the taxa or node subtended by the branch with that length. For example, if all the branches in our first five taxa tree above were of length 0.5, the tree will be written as $(((A : 0.5, B : 0.5) : 0.5, C : 0.5) : 0.5, (D : 0.5, E : 0.5) : 0.5)$;.

Finally, rooted and unrooted trees result in a slightly different newick format. The difference can be seen at the highest level of parentheses, where one grouping disappears in unrooted trees. The first five taxa tree above was in fact rooted (two branches are leaving the root node, $((A, B), C)$ and (D, E)). An unrooted tree has always three branches connecting the root node: $(((A, B), C), D, E)$;. Here one branch leads to $((A, B), C)$, one to D and one to E.

In phylip-related software (e.g. phylip, paml, phyml), a line containing the number of taxa and trees found in the file is simply added before the parentheses structure:

Example phylip tree file

```
 5 1
(((A,B),C),(D,E));
```

Furthermore, spaces can be introduced after the ',' or the parenthesis. The tree can even be split on several lines.

```
 5 1
(((A,
B), C),
(D , E));
```

In nexus-related software (e.g. paup, macclade, mrbayes), the tree is enclosed in a `begin trees;` section, and a name is given to the tree itself. Furthermore each line containing a tree starts with the structure `tree` *`treename`* `=`. Rooted and unrooted trees are defined by adding respectively `[&R]` or `[&U]` after the `=`. A `end;` always closes the tree section.

Example nexus tree file

```
#NEXUS

begin trees;
tree one = [&R] (((A,B),C),(D,E));
tree second_one = [&U] (((A,B),C),D,E);
end;
```

An additional feature of nexus tree format is the translation table for taxa names. The taxa names can be replaced by numbers in the trees and a corresponding table between the numbers and the names is placed below the `begin trees;` section. Software reading nexus tree files will then replace the numbers by the correct names when representing the trees. To use a translation table, the keyword `translate` is placed after `begin trees;` and one pair of numbers / names followed by a ',' is then written on each line. In the last pair, a ';' replaces the ','.

Example nexus tree with translation table

```
#NEXUS

begin trees;
translate
1 A,
2 B,
3 C,
4 D,
5 E;
tree one = [&R] (((1,2),3),(4,5));
end;
```