Phylogenetics course

practice instructions

Distance methods

In this assignment we will construct phylogenies with distance based methods. Do all the exercises preferably in the provided order. You will practice the most basic tasks related to a general phylogenetic analysis including importing data, calculating phylogenies with different, alternative options, and generating tree plots.

The software what will be used in this practice is the *R statistical environment* with the following supporting libraries: *seqinr*, *ape*, and *phangorn*. This is a free and open-source software setting which is available for you also later on, therefore you are able to apply the learned skills and approaches in your own research or study projects.

Files you need

IL6_protein.aln – This file contains protein sequences of interleukin 6 from six mammal and a bird species. The sequences were aligned using Clustal Omega.

IL6_mRNA.aln – This file contains coding cDNA sequences of interleukin 6 gene from six mammal and a bird genomes. The sequences were aligned using Clustal Omega.

Exercises

1. Set up the environment
    1. Start R using the appropriate method depending on the operating system you use with your PC. R provides you command line interface where you can type or paste commands for an analysis. Some versions of R provide you also GUI-based menu and clicking based tools. For the sake of simplicity, we will ignore them here.
    2. R is a general software focused on the statistical analysis of large scale data. Supporting libraries, such as *sequinr* here, are used to provide task and topic specific functionality. You have to load libraries should be loaded before you can use the functions inside. We will load the *seqinr* and *ape* libraries here:

        ```
        library(seqinr)
        ```

        ```
        library(ape)
        ```

        Remember, if you restart R, you have to load the libraries again to repeat or continue your tasks.
    3. We have to tell R the directory, where we want to work. Usually, this directory contains

the input files and we want to save the results files there. When R starts it has configuration specific working directory what we can easily see:

```
getwd()
```

After this we have to set the working directory using the format we have seen in the output of getwd().

```
setwd('/home/path/to/your/files') #on Linux/UNIX

setwd('/Users/User Name/Documents/FOLDER') #on Mac

setwd('c:/path/to/my/directory/') #on Windows
```

2. Load data files
    1. R accesses data from local files using read() type functions. Many data types and file types has specialized functions. We will use read.alignment() function from *seqinr* package to access Clustal Omega aligned sequences.

    ```
    ali.prot<-read.alignment("IL6_protein.aln",format="fasta")

    ali.rna<-read.alignment("IL6_mRNA.aln",format="fasta")
    ```

    2. These create complex representations of the alignments in the ali.prot and ali.rna variables. We can have a good overview of their content by simply typing in variable names or using the str() unction.

    ```
    ali.prot

    str(ali.prot)
    ```

    3. Some functions in the *ape* library requires that the alignment is in a specific, DNAbin format. We have to convert the alignment object here to convert accordingly:

    ```
    ali.rna.b<-as.DNAbin(ali.rna)
    ```

3. Calculate distances
    1. As you have learned from the lecture, there are countless ways to calculate distances between aligned sequences. For protein sequences dist.alignment() function can be called, which can calculate the distances either taking into account of mutational similarity of codons or not.

    ```
    d.prot<-dist.alignment(ali.prot,matrix="similarity")
    ```

    2. For the cDNA sequences the dist.dna() function will be used from the *ape* library which offers many evolutionary and mathematical methods via its *model* parameter. For example, we can use simply the number of different sites as a (not very good) distance measure:

```
d.rna.n<-dist.dna(ali.rna.b,model="N")
```

Alternatively, we can use the number of transitions, transversion, the same way, or we can weight the number of differences with the length of the sequences.

```
d.rna.ts<-dist.dna(ali.rna.b,model="TS")
d.rna.tv<-dist.dna(ali.rna.b,model="TV")
d.rna.raw<-dist.dna(ali.rna.b,model="raw")
```

3. Many of the nucleotide substitution models which were mentioned in the lecture (nad even more which wasn't) are also usable here:

```
d.rna.JC69<-dist.dna(ali.rna.b,model="JC69")
d.rna.K80<-dist.dna(ali.rna.b,model="K80")
d.rna.F84<-dist.dna(ali.rna.b,model="F84")
d.rna.TN93<-dist.dna(ali.rna.b,model="TN93")
```

4. Creating trees

1. Now we have so many matrices, but how to create phylogenetic trees from them? In this practice we will use Neighbour-Joining alhorithm which is implemented in the nj() finction of the ape package. Note that Minimum Evolution as fastme.bal() or the BIONJ as bionj() algorithms are available and usable in a very similar manner.

```
t.prot<-nj(d.prot)
```

2. We have now a tree in the t.prot variable. Let's get some information about it. How many species are represented on this tree? What are those? Is this a rooted tree?

```
t.prot$Nnode
t.prot$tip.label
is.rooted(t.prot)
```

3. As you see, this tree is not rooted. That is a problem, because almost all further methods need a rooted tree. We can use the outgroup method to root this tree. From the list of included species we have seel that species number 7 is *Gallus gallus* (chicken) which is the only bird seuqnce, all the others are from mammals. Birds are clearly an outgroup compared to the mammal sequences, so we can use it to root the tree:

```
t.prot<-root(t.prot,outgroup=7,resolve.root=T)
t.prot<-root(t.prot,outgroup=7,resolve.root=T)
```

4. Let's finally see how the tree's graphic representation looks like:

```
plot(t.prot)

nodelabels()
```

5. If we do not like the exact arrangement here, we can swap the tree at internal nodes.

```
t.prot<-rotate(t.prot,9)

plot(t.prot,main="Protein sequences")
```

5. Compare trees from different distance matrices

1. Now we have all the tools to create comparative plots to see the real differences between different distance calculations. To help further the visual inspection, we will plot the trees on the same pages, and we will add bars indicating the distance measure on the trees. Let's investigate the mathematical distances firs:

```
par(mfrow=c(2,2))

plot(root(nj(d.rna.n),outgroup=7,resolve.root=T),main="mRNA sequences",sub="N")

add.scale.bar(length=10)

plot(root(nj(d.rna.ts),outgroup=7,resolve.root=T),main="mRNA sequences",sub="TS")

add.scale.bar(length=10)

plot(root(nj(d.rna.tv),outgroup=7,resolve.root=T),main="mRNA sequences",sub="TV")

add.scale.bar(length=10)

plot(root(nj(d.rna.raw),outgroup=7,resolve.root=T),main="mRNA sequences",sub="Raw")

add.scale.bar(length=0.05)

par(mfrow=c(1,1))
```

2. And the same for the evolutionary distances:

```
par(mfrow=c(2,2))

plot(root(nj(d.rna.JC69),outgroup=7,resolve.root=T),main="mRNA sequences",sub="JC69")

add.scale.bar(length=0.05)

plot(root(nj(d.rna.K80),outgroup=7,resolve.root=T),main="mR
```

```
NA sequences",sub="K80")

add.scale.bar(length=0.05)

plot(root(nj(d.rna.F84),outgroup=7,resolve.root=T),main="mR
NA sequences",sub="F84")

add.scale.bar(length=0.05)

plot(root(nj(d.rna.TN93),outgroup=7,resolve.root=T),main="m
RNA sequences",sub="TN93")

add.scale.bar(length=0.05)

par(mfrow=c(1,1))
```

3. We have a possibility to compare two trees directly. For example, are the trees coming from the proten and mRNA sequences the same or not?

   A<-matrix(t.rna$tip.label,nrow=7,ncol=2)

   cophyloplot(t.rna,t.prot,A,space=25,lty=2)

6. Saving plots into files

   1. If we want to save these nice plots into files, we have to use the graphics facility R offers. R is capable of using several devices for plotting. A device can be our monitor in front of us, or a PDF, JPEG, or TIFF file the same way. For example, to save the last nice co-plot with two trees into a PDF file, we have to call the pdf() function, and all plotting command will draw into a file. We have to close the file using the dev.off() function to have R release the file. After that it can be opened with other software.

      ```
      pdf("IL6_cophyloplot.pdf",paper="a4")

      cophyloplot(t.rna,t.prot,A,space=25,lty=2)

      dev.off()
      ```

   2. Or having amore complex example:

      ```
      pdf("IL6_trees.pdf",paper="a4")

      par(mfrow=c(2,2))

      plot(root(nj(d.rna.n),outgroup=7,resolve.root=T),main="mRNA
      sequences",sub="N")

      add.scale.bar(length=10)

      plot(root(nj(d.rna.raw),outgroup=7,resolve.root=T),main="mR
      NA sequences",sub="Raw")
      ```

```
add.scale.bar(length=0.05)

plot(root(nj(d.rna.JC69),outgroup=7,resolve.root=T),main="m
RNA sequences",sub="JC69")

add.scale.bar(length=0.05)

plot(t.prot,main="Protein sequences",sub="Similarity")

add.scale.bar(length=0.05)

dev.off()
```