Multiple sequence alignment

How to do it

Follow the ensuing steps to perform the multiple sequence alignment:

1. Install package **seqinr**

```
>install.packages ("seqinr")
>library (seqinr)
```

2. Obtain a fasta file *fastaMSA.fasta* from Blackboard

   Question 1: List the species of the sequences in the file.
   Answer:

3. Perform multiple sequence alignment using Clustal Omega

   – Go to **ClustalOmega**.

   – Paste sequencess in *fastaMSA.fasta* into the text box

   – Choose PHYLIP as your output format

   – Submit your job

   – Click on ”Download Alignment File” and save your result into your R work place

4. Read in the multiple sequence alignment

```
# ClustalOmege.phylip is the alignment file generated by Clastal Omega
> myAlignment <- read.alignment(file = "ClustalOmege.phylip", format = "phylip")
```

```
> class (myAlignment)
[1] "alignment"

> names (myAlignment)
[1] "nb"  "nam" "seq" "com"
```

The *myAlignment* variable is a list variable that stores the alignment. An R list variable can have named elements, and you can access the named elements of a list variable by typing the variable name, followed by $ sign , followed by the name of the named element. The list variable *myAlignment* has named elements nb, nam, seq, and com. In fact, the named element seq contains the alignment, which you can view by typing:

```
> myAlignment$seq
[[1]]
[1] "mvewtdaertailglwgklnideigpqalsrclivypwtqryfatfgnlsspaaimgnpkvaahgrtvmggleraiknmdnvkntyaalsvmhsekl

[[2]]
[1] "mvhwtaeekqlitglwgkvnvaecgaealarllivypwtqrffasfgnlssptailgnpmvrahgkkvltsfgdavknldnikntfsqlselhcdkl

[[3]]
```

```
[1] "mvhltpeeksavtalwgkvnvdevggealgrllvvypwtqrffesfgdlstpdavmgnpkvkahgkkvlgafsdglahldnlkgtfatlselhcdkl

[[4]]
[1] "mvhltpeeksavtalwgkvnvdevggealgrllvvypwtqrffesfgdlstpdavmgnpkvkahgkkvlgafsdglahldnlkgtfatlselhcdkl

[[5]]
[1] "-vqlsgeekaavlalwdkvneeevggealgrllvvypwtqrffdsfgdlsnpgavmgnpkvkahgkkvlhsfgegvhhldnlkgtfaalselhcdkl
```

Viewing a multiple alignment

if you want to view a long multiple alignment, it is convenient to view the multiple alignment in blocks.

1. Install **Biostrings** if you have not installed the package

```
>install.packages ("Biostrings")
>library (Biostrings)
```

2. Load the R code file, `printMultipleAlignment.R` from Blackboard, then using the source function as follows:

```
> source ("printMultipleAlignment.R")
```

As its inputs, the function printMultipleAlignment() takes the input alignment, and the number of columns to print out in each block.

3. Print out the output To print out the multiple alignment of sequences in blocks of 60 columns, we type:

```
> printMultipleAlignment(myAlignment, 60)
[1] "MVEWTDAERTAILGLWGKLNIDEIGPQALSRCLIVYPWTQRYFATFGNLSSPAAIMGNPK 60"
[1] "MVHWTAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPM 60"
[1] "MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK 60"
[1] "MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK 60"
[1] "-VQLSGEEKAAVLALWDKVNEEEVGGEALGRLLVVYPWTQRFFDSFGDLSNPGAVMGNPK 59"
[1] " "
[1] "VAAHGRTVMGGLERAIKNMDNVKNTYAALSVMHSEKLHVDPDNFRLLADCITVCAAMKFG 120"
[1] "VRAHGKKVLTSFGDAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFS 120"
[1] "VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG 120"
[1] "VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG 120"
[1] "VKAHGKKVLHSFGEGVHHLDNLKGTFAALSELHCDKLHVDPENFRLLGNVLVVVLARHFG 119"
[1] " "
[1] "QAGFNADVQEAWQKFLAVVVSALCRQYH 180"
[1] "K-DFTPECQAAWQKLVRVVAHALARKYH 179"
[1] "K-EFTPPVQAAYQKVVAGVANALAHKYH 179"
[1] "K-EFTPPVQAAYQKVVAGVANALAHKYH 179"
[1] "K-DFTPELQASYQKVVAGVANALAHKYH 178"
[1] " "
```

Calculating genetic distances between protein sequences

A common first step in performing a phylogenetic analysis is to calculate the pairwise genetic distances between sequences.

The genetic distance is an estimate of the divergence between two sequences, and is usually measured in quantity of evolutionary change (an estimate of the number of mutations that have occurred since the two sequences shared a common ancestor).

We can calculate the genetic distances between protein sequences using the *dist.alignment()* function in the **seqinr** package. The *dist.alignment()* function takes a multiple alignment as input. Based on the multiple alignment that you give it, *dist.alignment()* calculates the genetic distance between each pair of proteins in the multiple alignment.

For example, to calculate genetic distances between the virus phosphoproteins based on the multiple sequence alignment stored in virusaln, we type:

```
> seqdist <- dist.alignment(myAlignment)  # Calculate the genetic distances
> seqdist
          sp|Q90486| sp|P02112| sp|P68871| sp|P68873|
sp|P02112|  0.4285714
sp|P68871|  0.3955535  0.3299144
sp|P68873|  0.3955535  0.3299144  0.0000000
sp|P02062|  0.4138029  0.3412306  0.2983975  0.2983975
```

The genetic distance matrix above shows **the genetic distance** between each pair of proteins. These distances actually depict how far the species are, or rather, the sequences on the evolutionary scale.

Phylogenetic analysis and tree plotting

To perform the phylogenetic analysis on sequences of your choice, follow the ensuing steps: Install and load the **ape** package by typing the following command:

```
> install.packages("ape")
> library(ape)
```

Use distance matrix for the sequences with the following *dist.alignment* function:

```
myphylo <- triangMtd(seqdist)
```

Then you can create the different kinds of phylogenetic trees for your analysis as follows:

1. The **phylogram** shows the evolutionary history where the length of each branch stem indicates the amount of evolution (such as the number of nucleotide substitutions that occurred between the connected branch points)

   ```
   > plot(myphylo, type="phylogram", edge.color="red", cex=1, edge.width=1,main="(A) Phylogram")
   ```

2. The second type of tree that we plotted is a **cladogram**. It does not represent the actual relation among the species but the branching during the evolution and branches join at hypothetical ancestors

```
> plot(myphylo, type="cladogram", edge.color="red", cex=1, edge.width=1, main="(B) Cladogram")
```

3. The third type of tree is a **fan-shaped** tree that shows radical branches for species.

```
> plot(myphylo, type="fan", edge.color="red", cex=1, edge.width=1, main="(C) Fan")
```

4. The **unrooted trees** illustrate the relatedness of the leaf nodes, completely ignoring the ancestry.

```
> plot(myphylo, type="unrooted", edge.color="red", cex=1, edge.width=1, main="(D) Unrooted")
```

Question 2: Save phylogenetic trees and submit.

Question 3: Change the heading format of fastaMSA.fasta so that the species names are shown in your phylogeneic tree (Bonus)