

Useful code

Getting started

First we need to load all the necessary packages:

```
library(ape)
library(phangorn)
library(msa)
library(ggmsa)
```

If we get an error loading a package, this is usually an indication the package is not installed. In that case we need to install the package. This has to be done on a computer only once. This code now depends in which repository the package is stored. Most packages are on CRAN:

```
install.packages("ape")
install.packages("phangorn")
```

If the packages are from the bioconductor repository

```
install.packages("BiocManager")
library(BiocManager)
BiocManager::install("msa")
BiocManager::install("ggmsa")
```

Now we should be able to load all the packages:

```
library(ape)
library(phangorn)
library(msa)
library(ggmsa)
```

Tip

Often you will need to read or write the path to a file. Typing `tmp <- file.choose()` will store the path in the variable `tmp` and you can use it to read in the file.

Reading in sequences

Let's start with reading in the sequences from the BLAST searches.

```
reference <- read.FASTA("data_raw/Q05115.fasta", type="AA")
deltablast <- read.FASTA("data_raw/3dtv_deltablast.txt", type="AA")
blastp <- read.FASTA("data_raw/3dtv_pBLAST.txt", type="AA")
psyblast <- read.FASTA("data_raw/psiblast.txt", type="AA")
```

After this we combine the sequences and have a short look at them

```
aa <- c(reference, blastp, psyblast, deltablast)
aa
```

```
## 301 amino acid sequences in a list
##
## Mean sequence length: 235.6
##   Shortest sequence: 212
##   Longest sequence: 251
```

```
tmp <- names(aa)
head(tmp)
```

```
## [1] "sp|Q05115|AMDA_BORBO Arylmalonate decarboxylase OS=Bordetella bronchiseptica OX=518 PE=1 SV=1"
## [2] "3DG9_A Chain A, Arylmalonate decarboxylase [Bordetella bronchiseptica]"
## [3] "2VLB_A Chain A, ARYLMALONATE DECARBOXYLASE [Bordetella bronchiseptica]"
## [4] "3DTV_B Chain B, Arylmalonate decarboxylase [Bordetella bronchiseptica]"
## [5] "Q05115.1 RecName: Full=Arylmalonate decarboxylase; Short=AMDase [Bordetella bronchiseptica]"
## [6] "WP_280016214.1 MULTISPECIES: aspartate/glutamate racemase family protein [unclassified Achromobacter]"
```

Now the names are very long. Let's try to clean them up:

```
accession <- sapply(strsplit(tmp, " ") , \(\x)x[[1]])
accession[1] <- "Q05115"
accession <- gsub("\\:.*" , "", accession)

species <- sapply(strsplit(tmp[-1], "\\[" ) , \(\x) x[[2]])
species <- gsub("\\]", "", species)
species <- c("Bordetella bronchiseptica", species)

gene <- sapply(strsplit(tmp, " ") , \(\x) x[-1])
gene <- sapply(gene, paste0, collapse=" ")
gene <- gsub("\\[.*" , "", gene) |> trimws()
gene[1] <- "Arylmalonate decarboxylase"
```

Finally we write out our sequences with accession number as ID and create a table with the accession number, the gene name and the species name. And we save this data so that we can use later on or with other software:

```
X <- cbind(accession, gene, species)
head(X)

##      accession
## [1,] "Q05115"
## [2,] "3DG9_A"
## [3,] "2VLB_A"
## [4,] "3DTV_B"
## [5,] "Q05115.1"
## [6,] "WP_280016214.1"
##      gene
## [1,] "Arylmalonate decarboxylase"
## [2,] "Chain A, Arylmalonate decarboxylase"
## [3,] "Chain A, ARYLMALONATE DECARBOXYLASE"
## [4,] "Chain B, Arylmalonate decarboxylase"
## [5,] "RecName: Full=Arylmalonate decarboxylase; Short=AMDase"
## [6,] "MULTISPECIES: aspartate/glutamate racemase family protein"
##      species
## [1,] "Bordetella bronchiseptica"
## [2,] "Bordetella bronchiseptica"
## [3,] "Bordetella bronchiseptica"
## [4,] "Bordetella bronchiseptica"
## [5,] "Bordetella bronchiseptica"
## [6,] "unclassified Achromobacter"

write.table(X, file = "data/info.csv", row.names = FALSE)

names(aa) <- accession
```

```
write.FASTA(aa, "data_raw/all_sequences.fas")
```

Handling Alignments

```
library(phangorn)
align <- read.phyDat("data/MAFFT_aligned_sequences",
                    type="AA", format = "fasta")

align
image(align)
image(align, scheme="Clustal")
names(align)[1:5]

lab <- names(align)
accession <- sapply(strsplit(lab, " ") , \(x)x[[1]])
names(align) <- accession

write.phyDat(align, "data/Mafft_short_names.fas", format = "fasta")
```