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 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 is 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 Achromoba
```

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"


unique(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"
##  [7] "aspartate/glutamate racemase family protein"
##  [8] "Chain D, Arylmalonate decarboxylase"
##  [9] "aspartate/glutamate racemase family protein, partial"
## [10] "MULTISPECIES: arylmalonate decarboxylase"
## [11] "MAG: maleate cis-trans isomerase"
## [12] "maleate cis-trans isomerase family protein"
## [13] "maleate isomerase"
## [14] "MULTISPECIES: maleate cis-trans isomerase family protein"
## [15] "maleate cis-trans isomerase"
```

```
gene <- tolower(gene)
ind <- grep("arylmalonate decarboxylase", gene)
gene[ind] <- "arylmalonate decarboxylase"
ind <- grep("maleate cis-trans isomerase", gene)
gene[ind] <- "maleate cis-trans isomerase"
ind <- grep("aspartate/glutamate racemase family protein", gene)
gene[ind] <- "aspartate/glutamate racemase family protein"
unique(gene)
```

```
## [1] "arylmalonate decarboxylase"
## [2] "aspartate/glutamate racemase family protein"
## [3] "maleate cis-trans isomerase"
## [4] "maleate isomerase"
```

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 softerware:

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

```
##      accession        gene
## [1,] "Q05115"         "arylmalonate decarboxylase"
## [2,] "3DG9_A"         "arylmalonate decarboxylase"
## [3,] "2VLB_A"         "arylmalonate decarboxylase"
## [4,] "3DTV_B"         "arylmalonate decarboxylase"
## [5,] "Q05115.1"       "arylmalonate decarboxylase"
## [6,] "WP_280016214.1" "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")
```

Alignment (in R)

If you were able to install the msa package we can align the sequences in R.

```
library(msa)
mySeqs <- "data_raw/all_sequences.fas"
align <- msa(mySeqs, method="Muscle", type="protein")
align <- as.phyDat(align)
```

We export the sequences and also a much smaller

```
short_align <- phangorn:::remove_similar(align, k=30)
short_align
```

```
## 40 sequences with 264 character and 260 different site patterns.
## The states are A R N D C Q E G H I L K M F P S T W Y V
```

```
write.phyDat(short_align, "data/short_align.fas", format="FASTA")
write.phyDat(align, "data/align_muscle_3_8.fas", format="FASTA")
```

Handling Alignments

```
library(phangorn)
align <- read.phyDat("data/short_align.fas", format="FASTA", type="AA")
align
image(align)
image(align, scheme="Clustal")
```