1- Drawing of *CifA* and *CifB* protein domains

First, an information matrix containing the total gene length and the start and end positions of each domain on each gene are prerequisites for the attached R code. An example of the first rows of the cifA matrix can be found attached to this GitHub repository as **'cifA.txt**,' where 'Apopt' is the abbreviation for the Apoptosis regulator-like domain and 'RNA' stands for the RNA-binding-like domain. Same for the cifB matrix with **'cifB.txt'**, where 'AAA' is the abbreviation for the AAA-ATPases-like domain, 'IPPDE' for the PD-(D/E)XK nuclease N-terminal, 'IIPDDE' for the PD-(D/E)XK nuclease C-terminal, 'DUB' for the Deubiquitinase DUB domain, 'TOXIN' for the Pore forming toxin TcdA/TcdB domain, and 'LATRO' for the Latrotoxin domain.

Then, these two matrices are imported into RStudio v4.3.1 (http://www.rstudio.com/):

```
CIFA <- read.table("cifA.txt",sep="\t",dec=",",header =T,row.names=NULL)
CIFB <- read.table("cifB.txt",sep="\t",dec=",",header =T,row.names=NULL)
```

The following libraries are loaded : ggplot2 v.3.4.4 (doi: 10.1007/978-3-319-24277-4), cowplot v1.1.1 (doi: 10.32614/CRAN.package.cowplot), and gridExtra v.2.3 (doi: 10.32614/CRAN.package.gridExtra).

```
library(ggplot2)
library(cowplot)
library(gridExtra)
```

Here is the command to create a plot specifically for the different *cifA* ORF. Briefly, we first draw a line corresponding to the length of the *cifA* ORF for each symbiont, then we add the specific domains for each *cifA* ORF, and finally, we set the titles and theme for better representation:

```
plot_CIFA <-ggplot(CIFA) +
   geom_segment(aes(x = start, xend = end, y = cifA, yend = cifA), color =
"black", size = 1) +
   geom_segment(aes(x = RNA_START, xend = RNA_END, y = cifA, yend = cifA),
color = "purple", size = 10) +
   geom_segment(aes(x = Apopt_START, xend = Apopt_END, y = cifA, yend =
cifA), color = "orange", size = 10) +
   labs(x = "Size in amino acids", y = NULL, title = "cifA-like") +
   theme(panel.background = element_rect(fill = "white"),
         axis.line.x = element_line(color = "black", size = 0.5))
```

We can do the same for *cifB*:

```
plot_ CIFB<-ggplot(CIFB) +
  geom_segment(aes(x = start, xend = end, y = cifB, yend = cifB), color =
"black", size = 1) +
  geom_segment(aes(x = AAA_START, xend = AAA_END, y = cifB, yend = cifB),
color = "green4", size = 10) +
  geom_segment(aes(x = IPDDE_START, xend = IPDDE_END, y = cifB, yend =
cifB), color = "#CC0000", size = 10) +
  geom_segment(aes(x = IIPDDE_START, xend = IIPDDE_END, y = cifB, yend =
cifB), color = "#CC0000", size = 10) +
  geom_segment(aes(x = DUB_START, xend = DUB_END, y = cifB, yend = cifB),
color = "#0099FF", size = 10) +
  geom_segment(aes(x = TOXIN_START, xend = TOXIN_END, y = cifB, yend =
cifB), color = "pink", size = 10) +
  geom_segment(aes(x = LATRO_START, xend = LATRO_END, y = cifB, yend =
cifB), color = "deeppink3", size = 10) +
```

```
  labs(x = "Size in amino acids", y = NULL, title = "cifB-like") +
  theme(panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(color = "black", size = 0.5))
```

Finally, we can combine the *cifA* and *cifB* of each symbiont on the same plot and add a scale to visualize the length of the two genes:

```
  max_length <- max(max(CIFA$end), max(CIFB$end))

plot_CIFA <- plot_CIFA +
  geom_vline(xintercept = seq(0, max_length, by = 500), color =
"lightgray", linetype = "solid", size = 0.1) +
  xlim(0, max_length)

plot_CIFB <- plot_CIFB +
  geom_vline(xintercept = seq(0, max_length, by = 500), color =
"lightgray", linetype = "solid", size = 0.1) +
  xlim(0, max_length) +
  scale_y_discrete(labels = NULL)

combined_plot <- plot_grid(plot_CIFA, plot_CIFB, ncol = 2, align = "v")
print(combined_plot)
```

2- *cifA-cifB* operon phylogeny

First, we used Clustal Omega (doi: 10.1002/pro.3290) implemented in Unipro UGENE v51.0 (doi: 10.1093/bioinformatics/bts091) to align three conserved domains of the *cif* operon: the RNA-binding-like domain of *cifA*, and the AAA-ATPase-like and PD-(D/E)XK N-and-C nucleases of *cifB*. Each domain alignment was trimmed to avoid any gaps, and the three domains were manually concatenated for each symbiont. The final concatenated alignment was saved in a .faa file named 'alignment.faa'.

The best ML substitution models were evaluated using modeltest v0.1.5 according to AICc criterion (https://github.com/ddarriba/modeltest, doi: 10.1093/molbev/msz189) :

```
modeltest-ng -i alignment.faa -p 12 -T raxml -d aa
```

Then, we constructed the phylogenetic tree with RAxML-NG v1.1.0 (https://github.com/amkozlov/raxml-ng, doi:10.1093/bioinformatics/btz305) with JTT+G as the best ML model for alignment.faa:

```
raxml-ng --all --msa alignment.faa --model JTT+G --prefix Name_of_your_job-
raxmlng --seed 5 --threads 4 --bs-trees 1000

raxml-ng --support --tree Name_of_your_job-raxmlng.raxml.bestTree --bs-
trees 1000 --prefix Name_of_your_job-boot --threads 2
```