metabolism

Léna AILLIOT

2023-01-30

Installation des packages

```
#install.packages("ggplot2")
#install.packages("xfun")
#install.packages(c("gplots", "pheatmap"))
#install.packages("openxlsx")
```

```
library (gplots)
```

```
##
## Attachement du package : 'gplots'

## L'objet suivant est masqué depuis 'package:stats':
##
##     lowess
```

```
library (pheatmap)
library(ggplot2)
library(openxlsx)
```

Compararaison de 68 MAGs présents sur GTDB, d'1 genome et 3 MAGs de *Planktomarina* :

Ouverture des résultats fastANI dans R :

```
data <- read.table ("result.out", header = FALSE)
colnames(data) <- c("query", "ref", "ani", "align_f", "nb_f")
#print(data)
```

result.out : ANI des 68 génomes de *Planktomarina* (NCBI) plus ceux des 3 MAGs. Colonnes nommées.

```
data$query <- gsub("../genomes/planktomarina_", "", data$query)
data$ref <- gsub("../genomes/planktomarina_", "", data$ref)
data$query <- gsub(".fna.gz", "", data$query)
data$ref <- gsub(".fna.gz", "", data$ref)

data$query <- gsub("../MAG/", "", data$query)
data$ref <- gsub("../MAG/", "", data$ref)
data$query <- gsub(".fa", "", data$query)
data$ref <- gsub(".fa", "", data$ref)
```

Tronquage des noms des fichiers .fna.gz et .fa afin d'obtenir uniquement les n° d'acccession sur les figures.

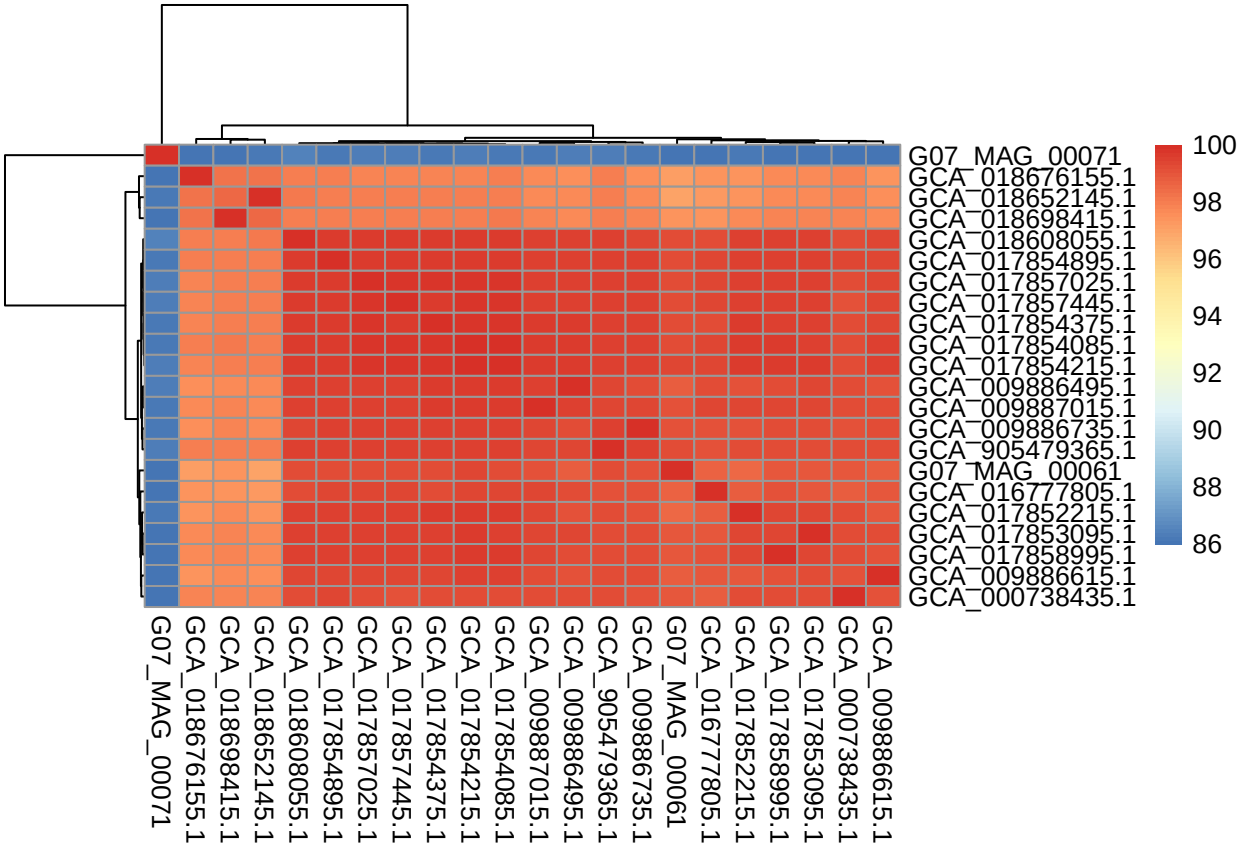
Matrice carrée des ANIs de chaque paire de génome :

```
df_ANI <- data.frame()
for (query in data[ ,1]){
  for (ref in data[ ,2]){
    df_ANI[query,ref] <- max(data[data$query == query & data$ref == ref,3], data[data$query == ref & da
  }
}
```

Dans chaque paire, les 2 génomes sont une fois Query et une fois Ref, il y a donc deux valeurs d'ANIs par paire. De manière à obtenir une matrice symétrique, l'ANI max de la paire est choisis.

Heatmap

```
pheatmap(as.matrix(df_ANI))
```



G07_MAG_00071 semble avoir une composition en nt similaire à ~86% avec les 20 génomes et l'autre MAG tandis que les autres sont globalement similaires à plus de 96%. Cet écart semble indiquer que 00071 est éloignée de l'espèce *P.temperata*.

Heatmap sur la proportion de fragement alignés par nombre de fragment pour chaque paire de génome :

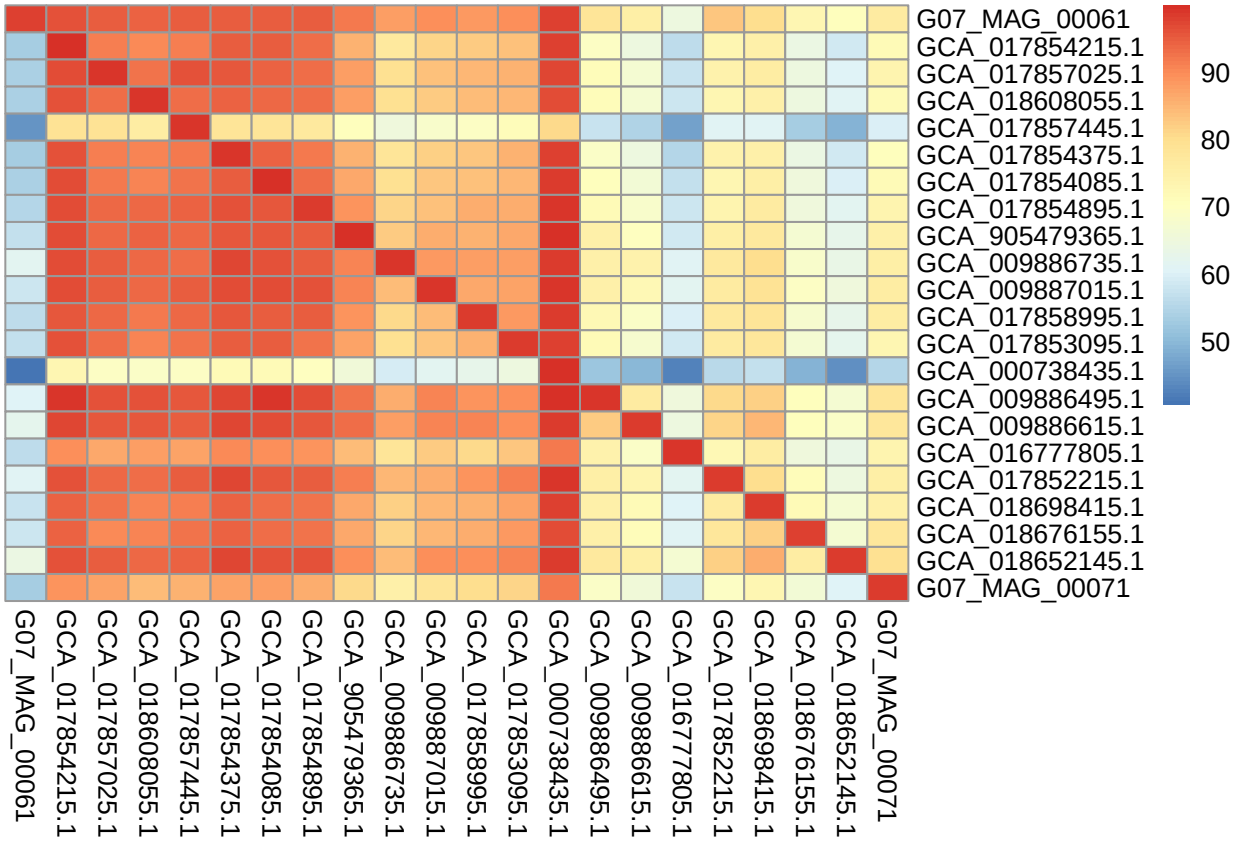
```
data$aligned_fraction <- 100*data$align_f/data$nb_f
```

Création nouvelle colonne contenant le % de fragment alignés par paire.

```
df2 <- data.frame()
for (query in data[ ,1]){
  for (ref in data[ ,2]){
    df2[query,ref] <- data[data$query == query & data$ref == ref, 6]
  }
}
```

Matrice carrée du % de fragment alignés pour chaque paire de génome.

```
df2 <- df2[colnames(df2),colnames(df2)]
pheatmap(as.matrix(df2), cluster_rows = FALSE, cluster_cols = FALSE)
```



Heatmap sans clustering réalisée à partir de df2 dans lequel les génomes des lignes et des colonnes sont dans le même ordre. G07_MAG_00061 possède peu de fragment alignés avec les autres génome lorsqu'il est utilisé comme référence (<50%)(beaucoup de fragment de la query ne s'y alignent pas), tandis qu'il semble aligné à plus de 85% avec plus de la moitié des génomes lorsqu'il est utilisé comme query. Il semble donc qu'il ne soit pas aussi long que les autres génomes. A l'inverse, GCA_000738435.1 présente de fortes similarité avec tous les génomes lorsqu'il est utilisé comme référence mais présente une faible similarité avec lesa utres quand utlilisé comme query –> car plus complet. Le génome de la souche isolée semble fortement aligné à tous les génomes.

Comparaison des 3 MAGs et des 65 génomes proposés sur NCBI du genre *Planktomarina* :

Importation des ANIs des 65 génomes :

```
data2 <- read.table ("results_ANI_68.out", header = FALSE)
colnames(data2) <- c("query2", "ref2", "ani2", "align_f2", "nb_f2")
print(data2)
```

```
data2$query2 <- gsub("input/", "", data2$query2)
data2$ref2 <- gsub("input/", "", data2$ref2)
data2$query2 <- gsub(".fna.gz", "", data2$query2)
data2$ref2 <- gsub(".fna.gz", "", data2$ref2)
```

```
data2$query2 <- gsub(".fa", "", data2$query2)
data2$ref2 <- gsub(".fa", "", data2$ref2)
```

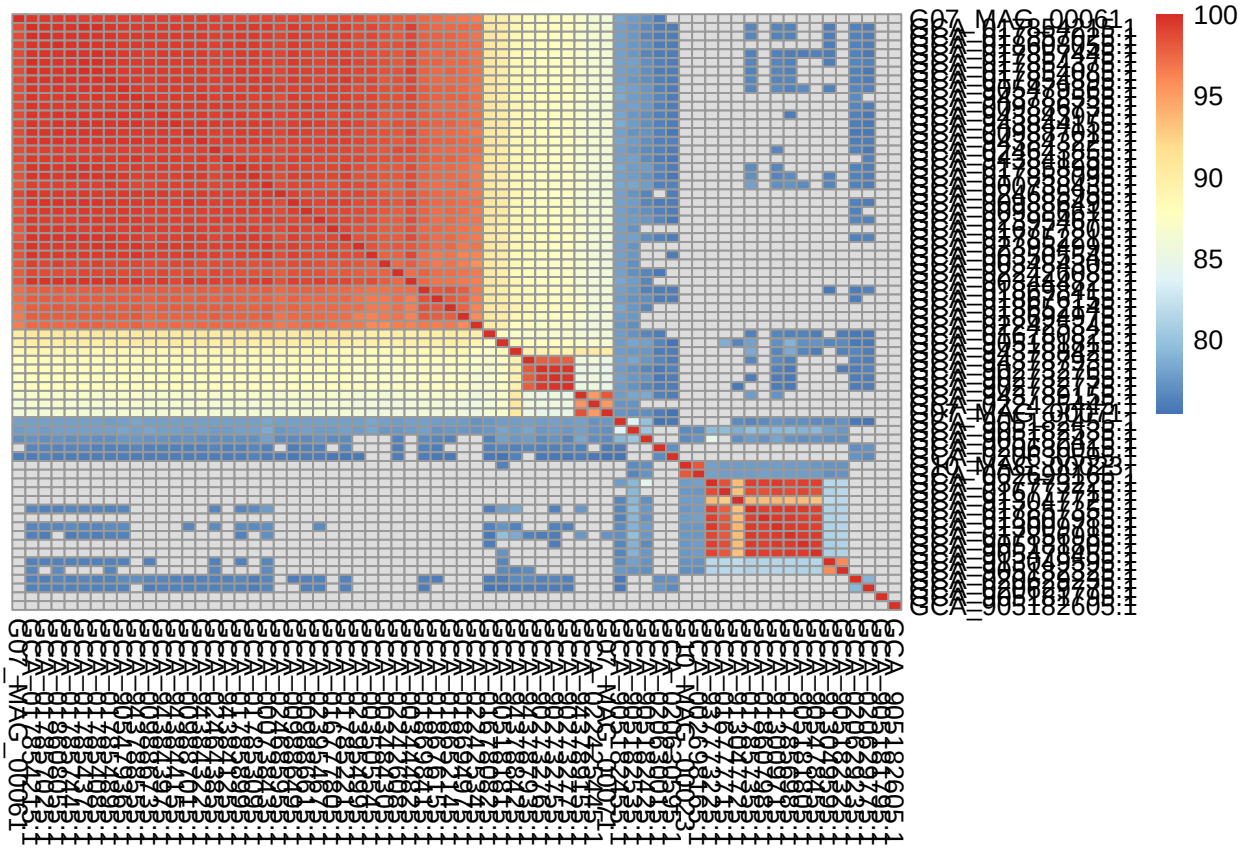
Matrice des ANIs par paire de génome :

```
df44 <- data.frame()
for (query2 in data2[ ,1]){
  for (ref2 in data2[ ,2]){
    if (nrow(data[data2$query2 == query2 & data2$ref2 == ref2,]) == 1) {df44[query2,ref2] <- data2[data
    else {df44[query2,ref2] <- NA}
  }
}
```

Certains génomes étant éloignés à plus de 25%, la valeur d'ANI n'est pas calculée, lorsqu'une paire ne possède pas de valeur, la valeur sera renseignée NA dans la matrice.

Heatmap :

```
df44 <- df44[colnames(df44),colnames(df44)]
pheatmap(as.matrix(df44), cluster_rows = FALSE, cluster_cols = FALSE)
```



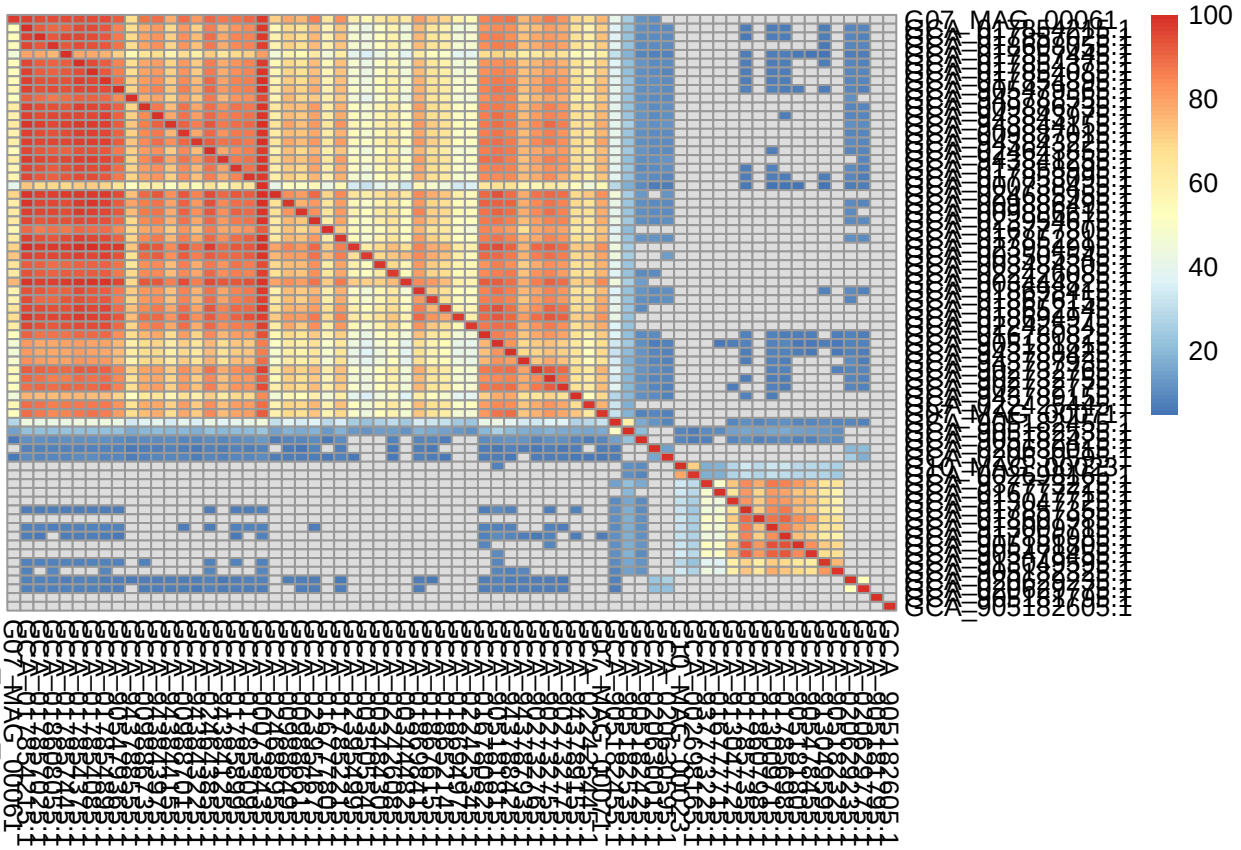
G07_MAG_00071 semble être proche d'un autre génomes uniquement, du genre *Planktomarina* mais pas de l'espèce *P.temperata*. GCA_02245445.1 n'est pas présent sur GTDB. 30 génomes semblent particulièrement proche, 3 proches à 85% des autres et 9 génomes ne possède pas de valeur d'ANIs avec la majorité des autres génomes.

Heatmap sur la proportion de fragement alignés par nombre de fragment pour chaque paire de génome :

```
data2$aligned_fraction2 <- 100*data2$align_f2/data2$nb_f2
```

```
df44_2 <- data.frame()
for (query2 in data2[ ,1]){
  for (ref2 in data2[ ,2]){
    if (nrow(data[data2$query2 == query2 & data2$ref2 == ref2,]) == 1) {df44_2[query2,ref2] <- data2[da
    else {df44_2[query2,ref2] <- NA}
  }
}
```

```
df44_2 <- df44_2[colnames(df44_2),colnames(df44_2)]
pheatmap(as.matrix(df44_2), cluster_rows = FALSE, cluster_cols = FALSE)
```



Même proposition pour G07_MAG_00061 que pour la heatmap précédente (% fragments alignés)

```
write.xlsx(df44, '/Users/33640/OneDrive/Bureau/FAC/M1/Stage', colNames =TRUE, rowNames = TRUE)
```

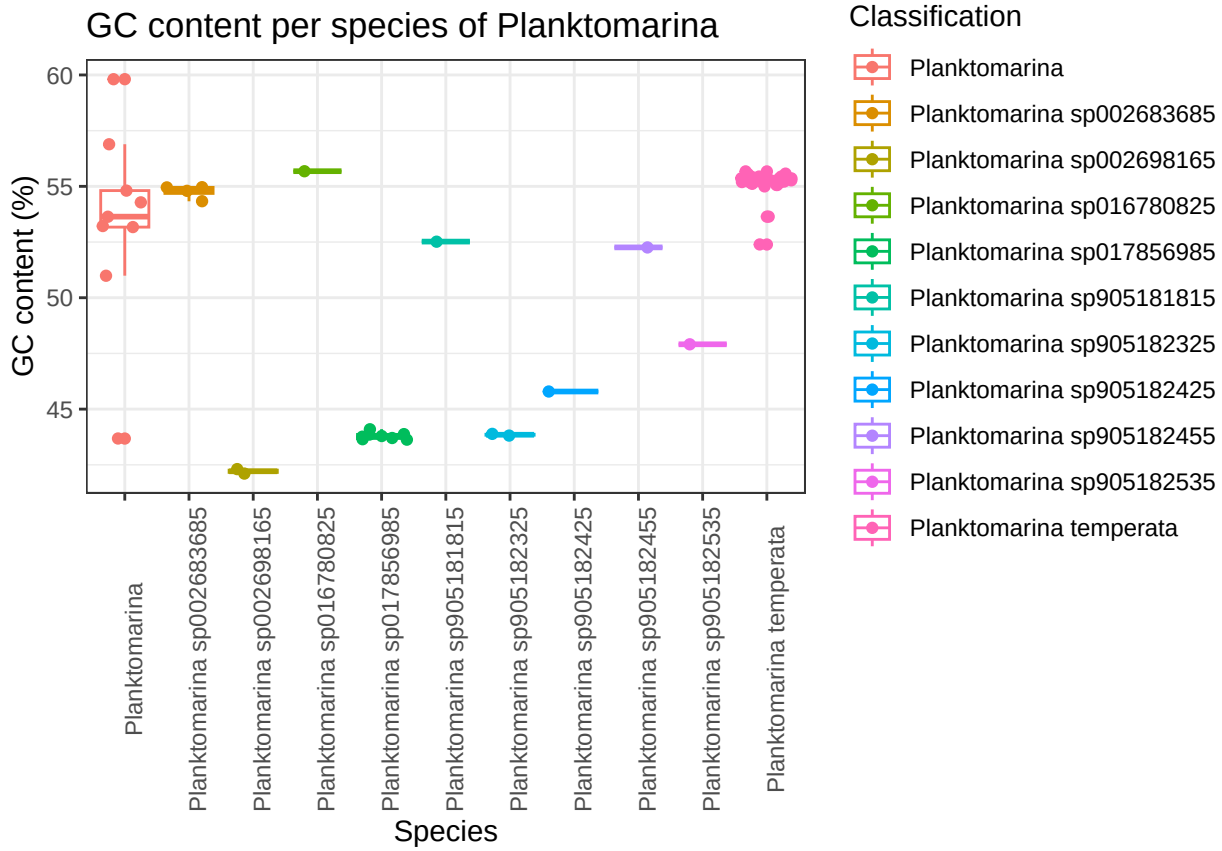
Visualisation des caractéristiques des 68 génomes d'après les résultats de Checkm et GTDBTK :

```
df_plot <- read.table("Résultats gtdb checkm binning - propre 2 (1).tsv", sep = "\t",header = TRUE)
df_plot2 <- df_plot[-c(1,2),]
```

Supression des deux premières lignes car MAGs classé dans autres genres que *Planktomarina*. Reste donc 66 génomes.

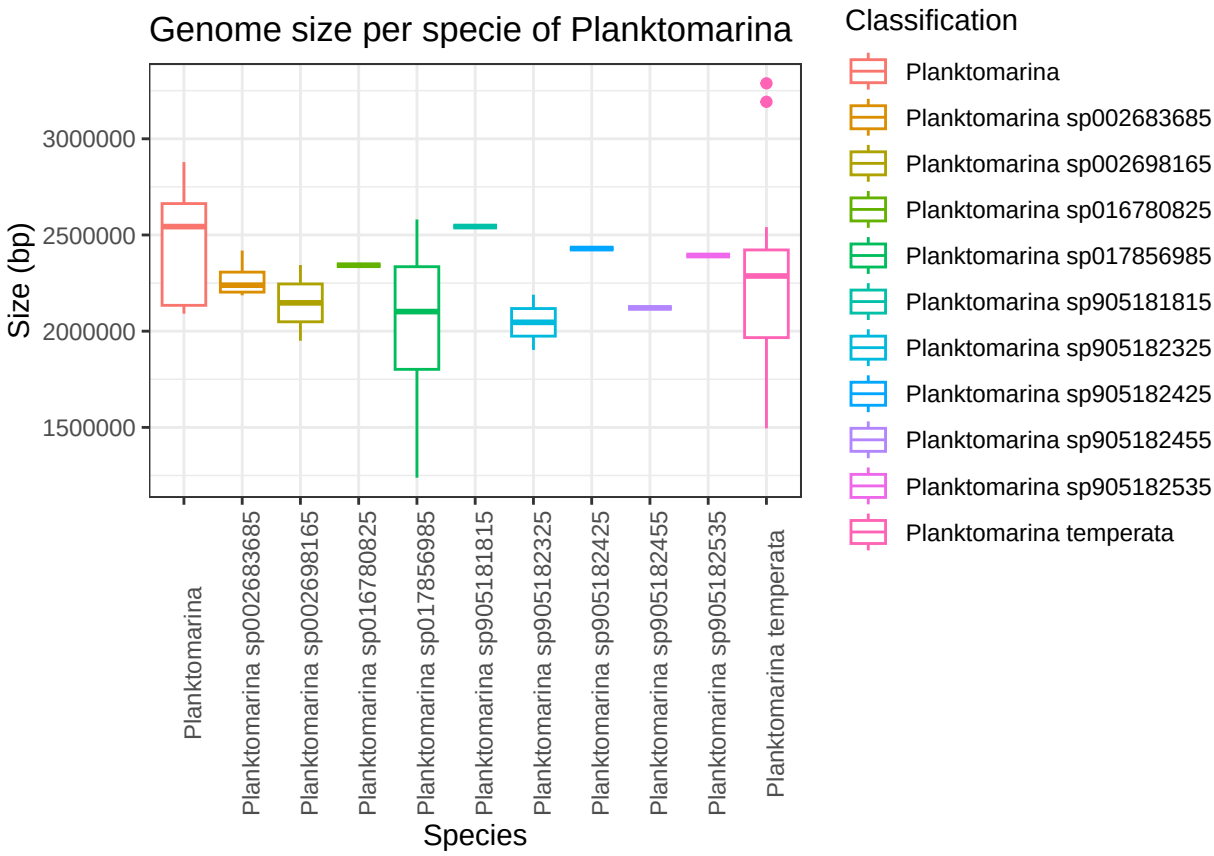
Plot du GC% par espèce de *Planktomarina*

```
GC_plot <- ggplot(data = df_plot2) + geom_boxplot(mapping = aes(x=Classification, y=GC.content...., col
GC_plot + theme_bw() + theme(axis.text.x = element_text(angle = 90)) + labs(y="GC content (%)", x="Spec
```

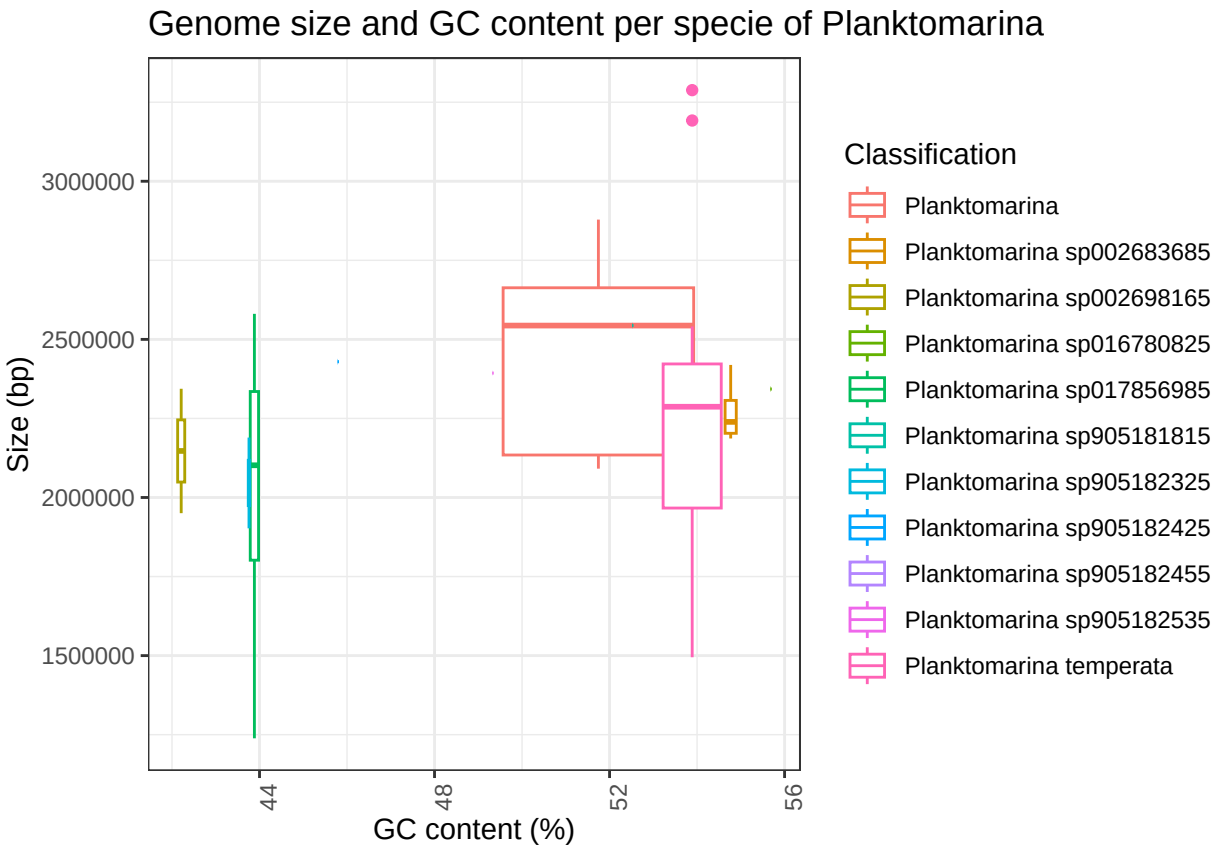


Plot de la tailles du génome par espèce

```
size_plot <- ggplot(data = df_plot2) + geom_boxplot(mapping = aes(x=Classification, y=Genome.size..bp.,
size_plot + theme_bw() + theme(axis.text.x = element_text(angle = 90)) + labs(y="Size (bp)", x="Species
```



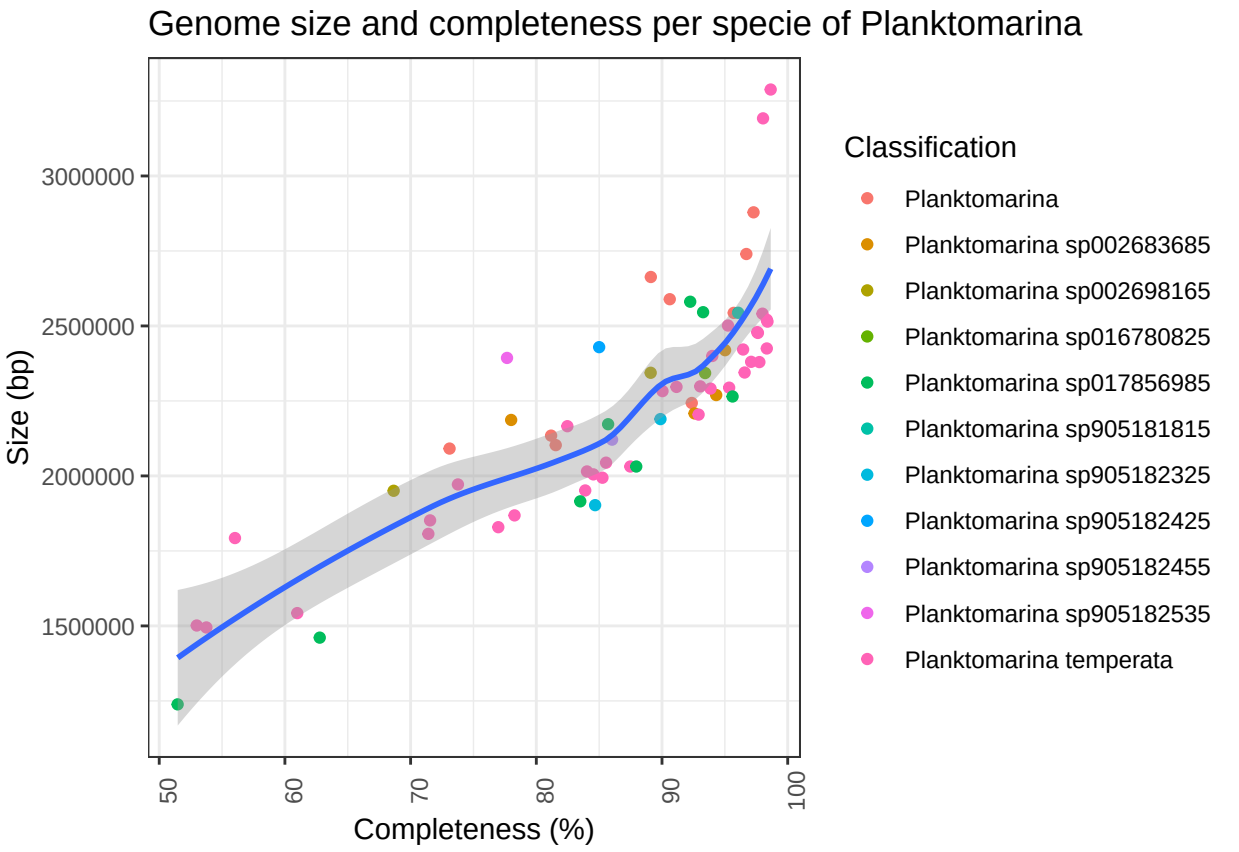

```
size_GC_plot <- ggplot(data = df_plot2) + geom_boxplot(mapping = aes(x=GC.content...., y=Genome.size..bp
print(size_GC_plot)
```



PLot du rapport entre la taille du génome et la complétion

```
size_comp_plot <- ggplot(data = df_plot2) + geom_jitter(mapping = aes(x=Completeness...., y=Genome.size
print(size_comp_plot)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
#size_GC_plot2 <- ggplot(data = df_plot2) + geom_jitter(mapping = aes(x=GC.content...., y=Genome.size..
#print(size_GC_plot2)
```

Visualisation des modules KEGGs au sein des génomes

```
df <- read.table("kegg-metabolism_modules.txt", sep = "\t", header = TRUE)
```

Trie des n° d'accession par ordre alphabétique d'espèce

```
taxo <- read.table("gtdbtk_68.tsv", sep = "\t", header = TRUE)
taxo$classification[taxo$classification == ""] <- "Planktomarina"
taxo_ordered <- taxo[order(taxo$classification),]
```

Création tableau de la complétion des modules par génomes et par module

```
modules <- data.frame()
for (genome in unique(df[ ,3])){
  for (m in unique(df[ ,4])){
```

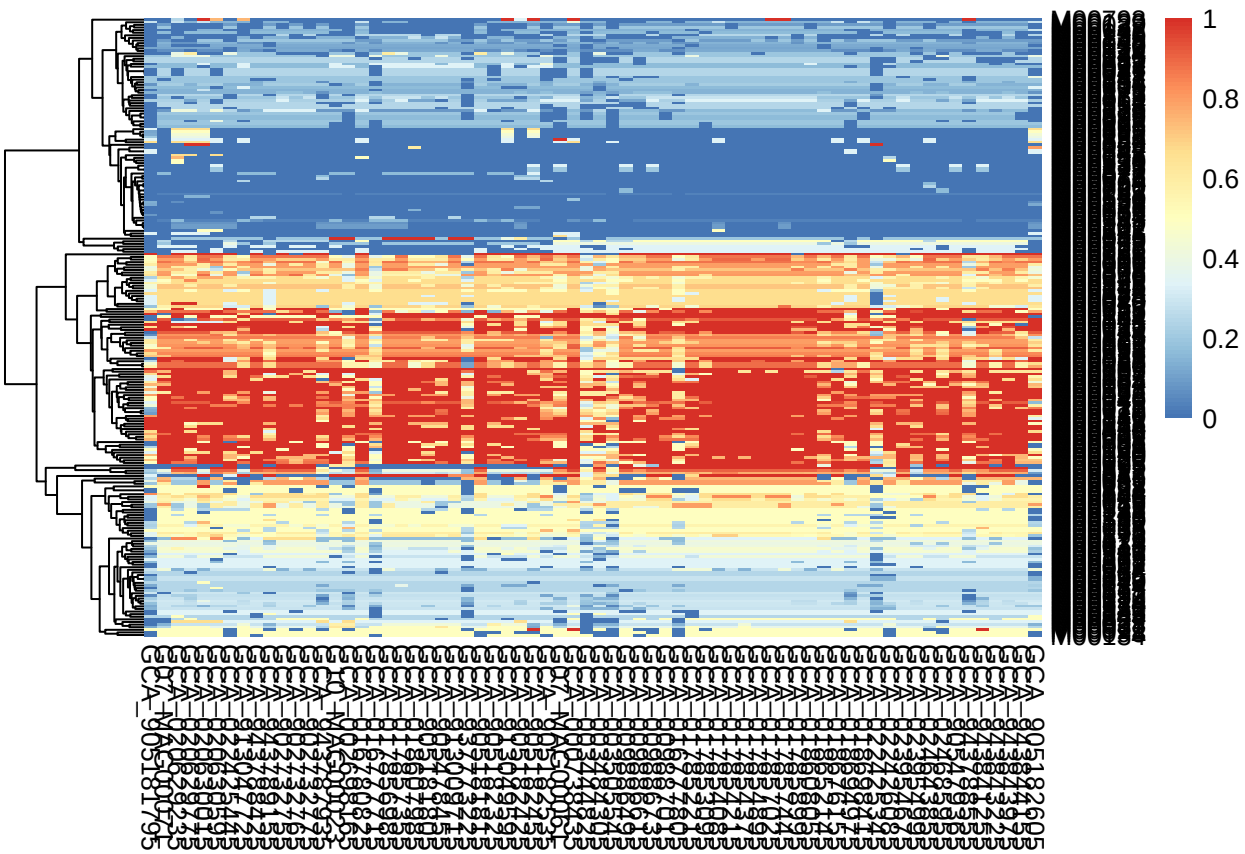
```
    if (nrow(df[df$kegg_module == m & df$db_name == genome,]) == 1) {modules[m,genome] <- df[df$kegg_mo
    else {modules[m,genome] <- 0}
  }
}
```

Tableau "modules" réoganisé par ordre alphabétique d'espèce

```
modules_ordered <- modules[taxo_ordered$name]
```

Heatmap de la completion des modules par génome

```
pheatmap(as.matrix(modules_ordered), cluster_rows = TRUE, cluster_cols = FALSE)
```



Fonctions du CORE génome

```
data2 <-read.table("Planktomarina_Pan_gene_clusters_summary.txt", sep = "\t", header = TRUE, quote="\""
```