Algo Gene

##1- Importation

```
#- Importation du dataset
library(readr)
db<- read_csv("db.csv")
```

```
## Rows: 2370 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (8): id_global, nom_pays, pays, trimestre, sequence_adn, smoke, alcohol,...
## dbl (6): annee, charge_virale, CD4, anticorps, CA_15_3, CA_15_3_apres
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

##2- Ecriture de l'algo

Pour chaque séquence : - Transformer la séquence ADN en chr - Isoler chaque caractère de la séquence dans un vecteur - Parcourir le vecteur avec une boucle qui compare une sous-séquence taille 6 à gggccc - Implémenter la boucle en tenant compte de deux erreur par occurence (le souci viendrait du fait qu'on ne connaît pas le début de chaque séquence et sa fin) - Compte le nombre d'occurences de "gggccc" - Créer une nouvelle table avec colonne 1 = id_global de la ligne où a été prise la séquence ADN colonne 2 = pays correspondant et colonne 3 = nb_occurrences de gggccc

Les amorces à détecter :

```
primers <- c(
  "gggccc",
  "acctcca",
  "tttttta",
  "gggacggg",
  "atatatat",
  "gtacacgt"
)
```

La fonction générale de détection des amorces

```
count_primer <- function(adn_seq, primer, tolerance = 2) {

  if (is.na(adn_seq)) {
    return(0)
  }

  adn_seq <- tolower(as.character(adn_seq))

  if (nchar(adn_seq) < nchar(primer)) {
    return(0)
  }

  adn_vec <- strsplit(adn_seq, "")[[1]]
  primer_vec <- strsplit(primer, "")[[1]]

  primer_length <- length(primer_vec)
  nb_occurrences <- 0

  for (i in 1:(length(adn_vec) - primer_length + 1)) {

    sub_seq <- adn_vec[i:(i + primer_length - 1)]

    errors <- sum(sub_seq != primer_vec)

    if (errors <= tolerance) {
      nb_occurrences <- nb_occurrences + 1
    }
  }

  return(nb_occurrences)
}
```

##3- Application de toutes les amorces et à toutes les séquences

```
# Table de base
algo_results <- data.frame(
  id_global = db$id_global,
  country = db$pays
)

# Pour chaque amorce → créer une colonne
for (p in primers) {

  col_name <- paste0("nb_", p)

  algo_results[[col_name]] <- sapply(
    db$sequence_adn,
    count_primer,
    primer = p,
    tolerance = 2
  )
}
```

##4- Dataviz pour présenter les résultats

```
library(ggplot2)
library(tidyr)
library(dplyr)
```

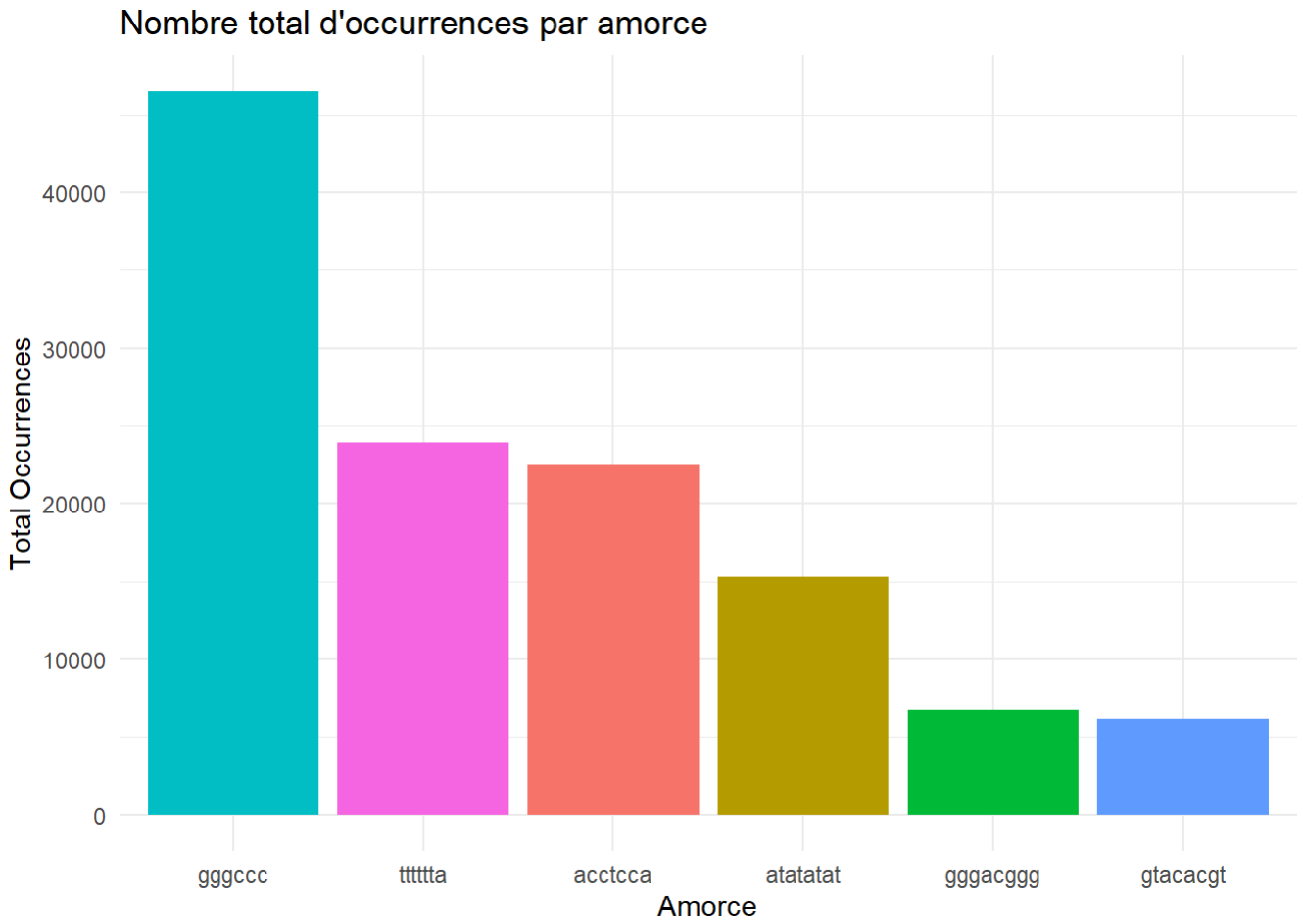
```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Transformer les données au format long pour ggplot
results_long <- algo_results %>%
  pivot_longer(
    cols = starts_with("nb_"),
    names_to = "primer",
    values_to = "count"
  ) %>%
  mutate(primer = gsub("nb_", "", primer))
```

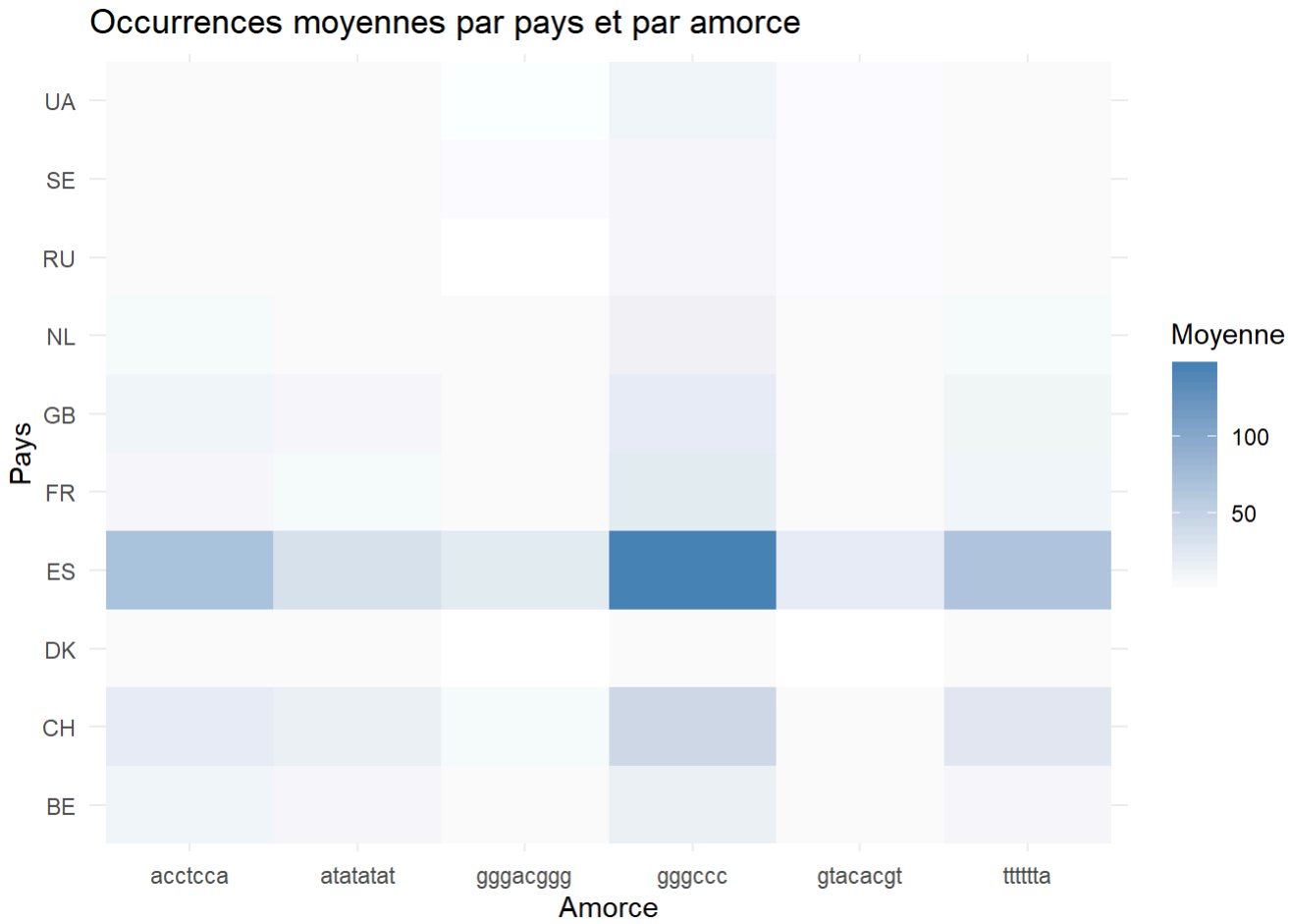
```r
# Graphique des occurrences totales
ggplot(results_long, aes(x = reorder(primer, -count, sum), y = count, fill = primer)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Nombre total d'occurrences par amorce",
       x = "Amorce", y = "Total Occurrences") +
  theme(legend.position = "none")
```



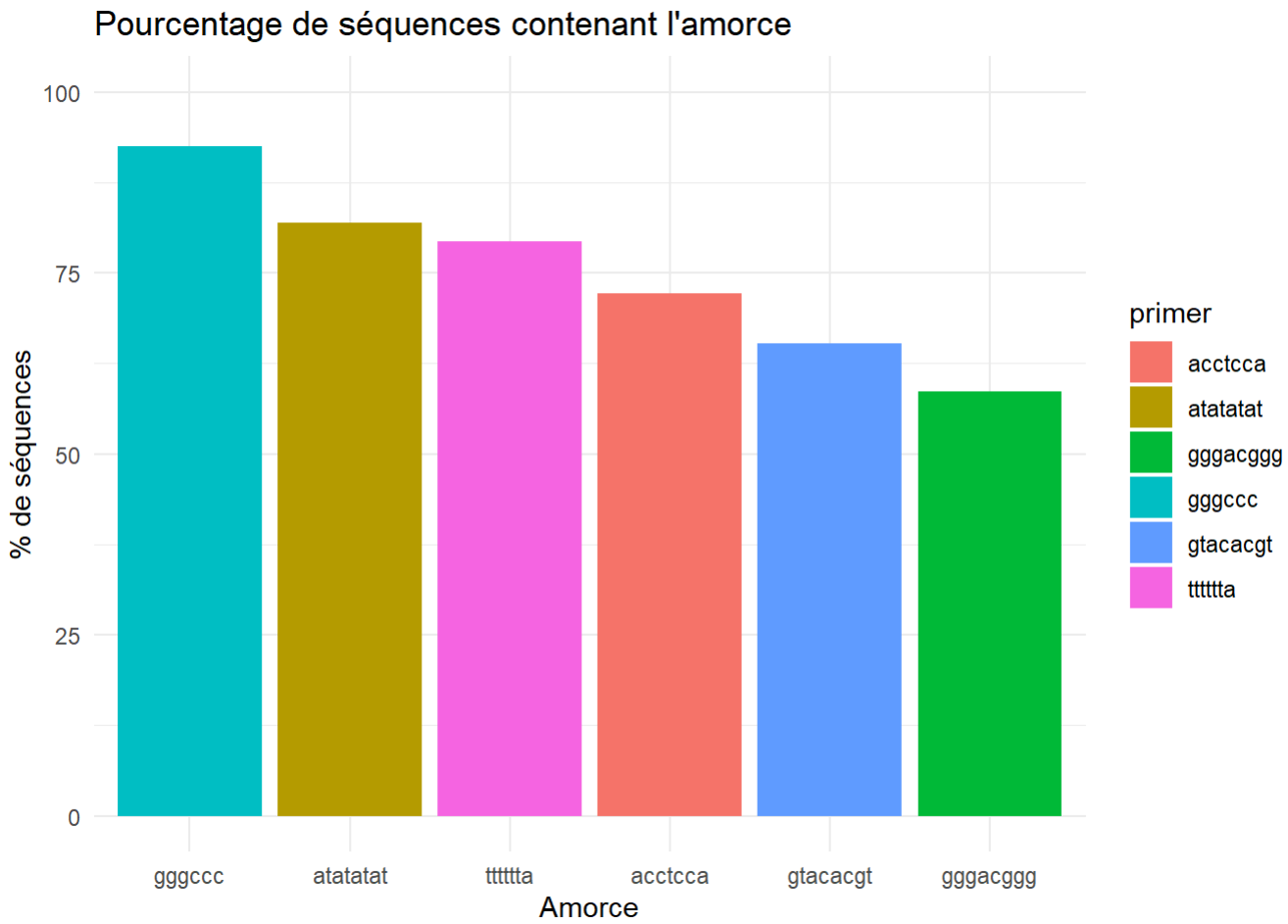

```
# Heatmap par Pays (Top 10 pays)
top_countries <- algo_results %>%
  count(country, sort = TRUE) %>%
  top_n(10) %>%
  pull(country)
```

```
## Selecting by n
```

```
results_long %>%
  filter(country %in% top_countries) %>%
  group_by(country, primer) %>%
  summarise(mean_count = mean(count), .groups = 'drop') %>%
  ggplot(aes(x = primer, y = country, fill = mean_count)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "steelblue") +
  theme_minimal() +
  labs(title = "Occurrences moyennes par pays et par amorce",
       x = "Amorce", y = "Pays", fill = "Moyenne")
```



```r
#Pourcentage de présence
results_long %>%
  group_by(primer) %>%
  summarise(presence_pct = mean(count > 0) * 100) %>%
  ggplot(aes(x = reorder(primer, -presence_pct), y = presence_pct, fill = primer)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Pourcentage de séquences contenant l'amorce",
       x = "Amorce", y = "% de séquences") +
  ylim(0, 100)
```



Note sur les données : L’algorithme a identifié que l’amorce gggccc est la plus fréquente avec un grand nombre d’occurrences par séquence, tandis que gggacggg et gtacacgt sont beaucoup plus rares. Ces différences peuvent être dues à la longueur intrinsèque des amorces ou à leur importance biologique.