Exploration des sources de variabilite (variables explicatives) qui influencent la qualité des Runs RNASeq (variables réponses)

Mickael Coquerelle

2025-06-25

```
library(tidyverse)
```

1. Préparation du df et des données d'entrée

Ce qui est intéressant en biologie et particulièrement en génétique, c'est de constater à quel point dans la littérature on prends des précautions sur la véracité des relations de cause à effet entre une variable A et une variable B. D'ailleurs dans la majorité des articles ou sites que j'ai pu parcourir on parle rarement de cause à effet mais de potentiel corrélation (ce qui est très différent). Du reste, on peut vite se perdre dans les méandres méthodologiques statistiques tant il y'a d'approche et de conditions à respecter pour démontrer une significativité dans la corrélation de deux variables, tant il y'a de garde fou à considérés (hypothèse de normalité, homoscédasticité , correction des p-value ...). Au vu de la période de stage et du temps imparti, j'ai essayer de faire un focus sur certaines variables explicatives qui m'ont paru pertinente à récolter avec l'aide de l'équipe du laboratoire.

```
fichier <- "Stats_Log_merge_with_deltas.csv"
df_raw <- read_csv(fichier,
                   locale = locale(encoding = "ISO-8859-1"),
                   guess_max = 100,
                   show_col_types = TRUE)

df_raw[grep("^CRAC_.*_reads$", names(df_raw))] <- df_raw[grep("^CRAC_.*_reads$", names(df_raw))]/ 2
```

3. Choix des variables explicatives et réponses pour l'exploration

Ci-dessous, je stocke dans deux vecteurs les variables explicatives et les variables réponses que j'ai choisies pour mon analyse de corrélation. Il a fallu faire des choix : cette sélection n'est pas exhaustive. On peut bien sûr discuter de la pertinence de telle ou telle métrique, mais, dans le cadre de ces analyses univariées, je cherche à balayer rapidement les éléments qui me semblent les plus susceptibles d'influencer la robustesse expérimentale.
J'ai dû faire ces choix en tenant compte du temps imparti et du cadre du stage, car il y avait beaucoup de points à aborder, mais aussi en fonction de la disponibilité et de l'accessibilité des données. Comme je l'évoque plus haut dans mon rapport, lorsqu'on cherche à évaluer la corrélation entre deux variables, il faut poser des limites, et ici, le critère temporel a joué un rôle évident :

```
vars_exp <- c("Ville_Prescripteur", "Date_Prelevement", "Date_Recep", "Date_extraction",
              "Concentration_ARN", "Purete_proteique", "Date_Lib", "Date_Lancement",
              "Delta_Run_Prel", "Delta_Run_Recep", "Delta_Run_Ext", "Delta_Run_Lib", "Delta_Ext_Lib","D

vars_resp <- c("STAR_Total_reads", "STAR_Unique_reads", "STAR_Unique_pct", "STAR_Multi_reads",
```

```
                "STAR_Multi_pct", "STAR_No_map_reads", "STAR_No_map_pct_sum",
                "CRAC_Total_reads", "CRAC_Unique_reads", "CRAC_Unique_pct", "CRAC_Multi_reads",
                "CRAC_Multi_pct", "CRAC_No_map_reads", "CRAC_No_map_pct")
```

J'ai également choisi de filtrer les variables qualitatives (par manque de temps). Même si je les stocke dans les vecteurs initiaux, je n'ai pas eu le temps de les explorer pour l'instant, mais elles sont incluses afin de pouvoir étudier leur impact ultérieurement.

```
vars_exp <- intersect(vars_exp, colnames(df_raw))
vars_resp <- intersect(vars_resp, colnames(df_raw))

vars_exp_num <- vars_exp[sapply(df_raw[vars_exp], is.numeric)]
vars_resp_num <- vars_resp[sapply(df_raw[vars_resp], is.numeric)]
```

4. Fonction personnalisée pour les corrélations

J'encapsule le calcul de plusieurs métriques dans une fonction : le coefficient de corrélation de Pearson, le coefficient de détermination R2, ainsi que la p-valeur associée à la pente du modèle linéaire (qui se trouve à la ligne 2, colonne 4 du tableau des coefficients). Cette p-valeur permet d'évaluer la significativité statistique de la relation linéaire entre les variables testées x et y. Documentation : lm() R manual (https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html)

Avec le coefficient de Pearson : je cherche à mesurer la direction de la relation entre mes deux variables x et y.

Avec le coefficient de determination R2 : je quantifie "l'explicabilité de la variance de y vers x . (on peut dire cela plus simplement en parlant de la qualité de la relation linéaire)

```
corr_metrics <- function(x, y) {
  # Nettoyer les couples xi/yi qui contiennent au moins un NA.
  ok <- !is.na(x) & !is.na(y)
  x_ok <- x[ok]
  y_ok <- y[ok]

  # Au moins 30 valeurs pour avoir un effectif minimum :
  if(length(x_ok) < 30) return(c(cor = NA, R2 = NA, pval = NA))

  # Calcul du coefficient de pearson:
  cor_val <- cor(x_ok, y_ok, method = "pearson")
  # Détermination du modèle linéaire pour chaque variable nettoyée
  lmfit <- lm(y_ok ~ x_ok)
  sum_lm <- summary(lmfit)
  # Extraction du coefficient de détermination:
  R2 <- sum_lm$r.squared

  # Qualité de la relation à travers la valeur de pvalue :
  pval <- coef(summary(lmfit))[2,4]
  c(cor = cor_val, R2 = R2, pval = pval)
}
```

5. Calcul des matrices de corrélation

```
cor_mat  <- matrix(NA_real_, nrow = length(vars_resp_num),
                   ncol = length(vars_exp_num),
                   dimnames = list(vars_resp_num, vars_exp_num))
R2_mat   <- cor_mat
pval_mat <- cor_mat

for (resp in vars_resp_num) {
  for (expv in vars_exp_num) {
    m <- corr_metrics(df_raw[[expv]], df_raw[[resp]])
    cor_mat[resp, expv]  <- m["cor"]
    R2_mat[resp, expv]   <- m["R2"]
    pval_mat[resp, expv] <- m["pval"]
  }
}
```

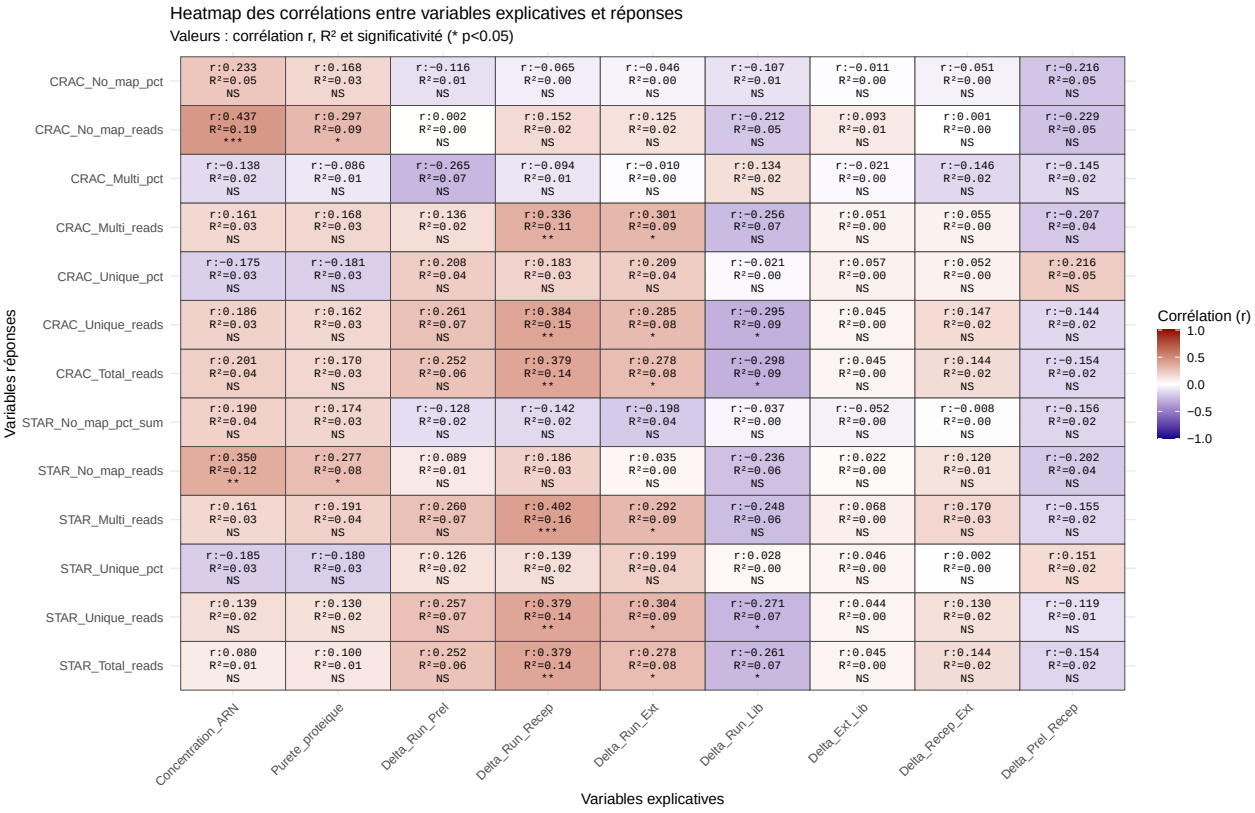
6. Formatage de la pval et des colonnes du csv de sortie.

```
df_corr <- as.data.frame(as.table(cor_mat))
colnames(df_corr) <- c("Response", "Explicative", "Correlation")

df_corr$R2 <- as.vector(R2_mat)
df_corr$pval <- as.vector(pval_mat)

df_corr <- df_corr %>%
  mutate(
    R2_label = ifelse(!is.na(R2), sprintf("R²=%.2f", R2), ""),
    signif = case_when(
      is.na(pval) ~ "NS",
      pval < 0.001 ~ "***",
      pval < 0.01  ~ "**",
      pval < 0.05  ~ "*",
      TRUE         ~ "NS"
    ),
    label = ifelse(!is.na(Correlation),
                   paste0("r:", sprintf("%.3f",Correlation), "\n", R2_label, "\n", signif),
                   "")
  )
```

7. Génération de la heatmap des corrélations

```
heatmap_plot <- ggplot(df_corr, aes(x = Explicative, y = Response, fill = Correlation)) +
  geom_tile(color = "grey30") +
  geom_text(aes(label = label), size = 3, lineheight = 1, family = "mono") +
  scale_fill_gradient2(low = "darkblue", mid = "white", high = "darkred", midpoint = 0,
                       na.value = "grey90", limits = c(-1, 1), name = "Corrélation (r)") +
  theme_minimal(base_size = 12) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  labs(
    title = "Heatmap des corrélations entre variables explicatives et réponses",
```

```
    subtitle = "Valeurs : corrélation r, R² et significativité (* p<0.05)",
    x = "Variables explicatives",
    y = "Variables réponses"
  )

print(heatmap_plot)
```



8. Sauvegarde du graphique et du csv résultats

```
ggsave("Heatmap_Correlation.png",
       heatmap_plot, width = 14, height = 9, dpi = 300)
ggsave("~/Latex_Project/Rapport_M1_irmb/Heatmap_Correlation_spearman.png",
       heatmap_plot, width = 14, height = 9, dpi = 300)

write.csv(df_corr %>% select(Response, Explicative, Correlation, R2, pval, signif),
          file = "Correlation_Results_numeric_only_spear.csv",
          row.names = FALSE, fileEncoding = "UTF-8")
```

#9 . Complement d'enquete Dans cette partie, pour mieux explorer visuellement les relations linéaires entre variables explicatives et variables réponses, je réalise des scatterplots avec une régression linéaire simple. Pour éviter de surcharger un seul graphique avec toutes les combinaisons, j'ai choisi de regrouper les combinaisons par groupes de 10 plots, chaque groupe correspondant à une figure distincte.

Cela permet de visualiser clairement chaque relation tout en respectant une lisibilité optimale des facettes. # Scatterplot simple avec ggplot2

```r
combinaisons <- expand.grid(x_var = vars_exp, y_var = vars_resp, stringsAsFactors = FALSE) %>%
  mutate(group = ceiling(row_number() / 10))  # regroupe par 10 paires
# Fonction pour créer un graphique pour un groupe de 10 combinaisons
plot_group <- function(group_num, df, comb_df) {
  comb_subset <- comb_df %>% filter(group == group_num)

  df_long <- purrr::map2_df(comb_subset$x_var, comb_subset$y_var, function(x, y) {
    df %>%
      select(all_of(c(x, y))) %>%
      rename(x = !!x, y = !!y) %>%
      mutate(x_var = x, y_var = y)
  })

  ggplot(df_long, aes(x = x, y = y)) +
    geom_point(size = 1.5, alpha = 0.6, color = "#0072B2") +
    geom_smooth(method = "lm", se = TRUE, color = "darkred", linewidth = 0.7) +
    facet_wrap(~ paste0("Y: ", y_var, "\nX: ", x_var), scales = "free", ncol = 2) +
    labs(
      title = paste("Régressions linéaires (Groupe", group_num, ")"),
      x = "Variable explicative",
      y = "Variable réponse",
      caption = "Modèle: lm (régression linéaire simple)"
    ) +
    theme_minimal(base_size = 11) +
    theme(strip.text = element_text(size = 10))
}
```