Exploration des sources de variabilité (variables explicatives) qui influencent la qualité des Runs RNA-seq (variables réponses) Approche Spearman

Mickael Coquerelle

2025-06-25

```
library(tidyverse)   # manipulation & ggplot2
```

1. Préparation du *data-frame* et des données d'entrée

Le stage vise à comprendre comment certains **délais techniques** et variables de laboratoire peuvent dégrader la robustesse d'un protocole RNA-seq ciblé.

Dans la littérature, on distingue rarement une véritable "cause à effet" d'une simple **corrélation** ; ainsi, j'explore ici des *relations monotones éventuelles* sans présupposer de linéarité.

```
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)

# Les métriques CRAC *_reads représentent les deux brins ;
# Je divise par 2 pour les rendre comparables à STAR.
df_raw[grep("^CRAC_.*_reads$", names(df_raw))] <-
  df_raw[grep("^CRAC_.*_reads$", names(df_raw))] / 2
```

2. Choix des variables explicatives et réponses

Je sélectionne ci-dessous les variables **explicatives** (délais, concentration, etc.) et les variables **réponses** (métriques d'alignement STAR & CRAC) pertinentes pour l'analyse.La liste n'est pas exhaustive mais couvre les principaux points critiquesidentifiés au laboratoire.

```
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", "Delta_Recep_Ext", "Delta_Prel_Recep", "STAR_Type")

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")

# Garder uniquement les colonnes réellement présentes dans le df.
vars_exp  <- intersect(vars_exp,  colnames(df_raw))
vars_resp <- intersect(vars_resp, colnames(df_raw))

# Ne conserver ici que les variables *numériques* pour la corrélation Spearman.
vars_exp_num  <- vars_exp [sapply(df_raw[vars_exp ], is.numeric)]
vars_resp_num <- vars_resp[sapply(df_raw[vars_resp], is.numeric)]
```

3. Pourquoi choisir Spearman ?

- Les métriques d'alignement (nombre de reads non mappés, % multi-mappés, ...) ne suivent pas une distribution gaussienne stricte; les délais sont des **comptes** entiers (0,1,2jours...).
- Le test **Spearman**:
  - ne suppose pas la normalité,
  - détecte des relations **monotones** oumême si la forme n'est pas linéaire,
  - reste robuste aux outliers.

Je remplace donc l'ancien couplage *Pearson+lm()* par un **test de Spearman complet** (corrélation p et p-value sur les rangs).Pour fournir un indicateur d'explicabilité, je rapporte "R2 equivaut rho2", qui quantifie la proportion de variance *monotone* partagée.

4. Fonction personnalisée pour Spearman

```
corr_metrics <- function(x, y) {
  ok          <- !is.na(x) & !is.na(y)
  x_ok        <- x[ok]
  y_ok        <- y[ok]

  # exiger un effectif minimal (30 couples).
  if (length(x_ok) < 30)
    return(c(cor = NA, R2 = NA, pval = NA))

  # Test de corrélation de Spearman
  res         <- cor.test(x_ok, y_ok, method = "spearman", exact = FALSE)
  rho         <- as.numeric(res$estimate)     #
  pval        <- res$p.value                 # p-value des rangs
  R2_mono     <- rho^2                       # proportion de variance monotone

  c(cor = rho, R2 = R2_mono, pval = pval)
}
```

5. Calcul des matrices de corrélation (p), R² et p-values

```
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
```

```r
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"]
  }
}
```

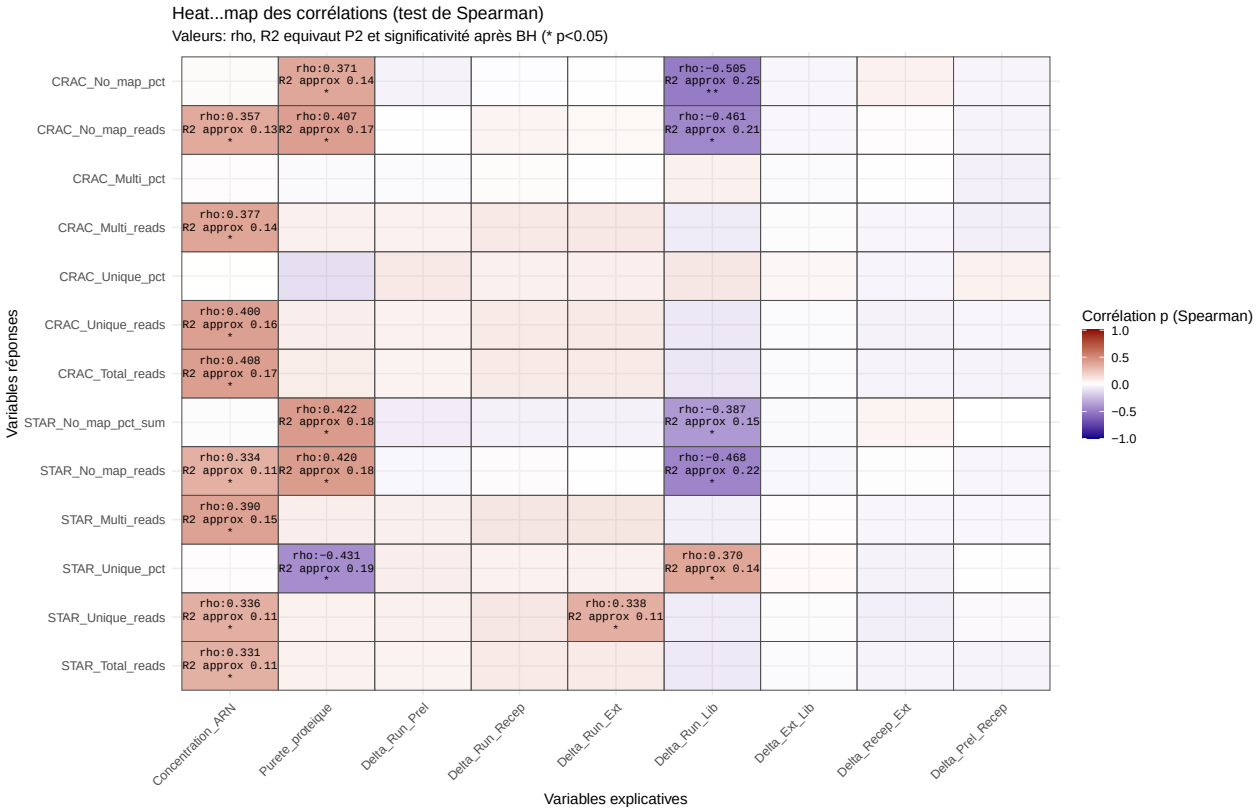
6. Formatage des résultats + correction de multiplicité (Benjamini–Hochberg)

```r
df_corr <- as.data.frame(as.table(cor_mat)) %>%
  rename(Response = Var1, Explicative = Var2, Correlation = Freq) %>%
  mutate(
    R2   = as.vector(R2_mat),
    pval = as.vector(pval_mat),
    pval_adj = p.adjust(pval, method = "BH"),
    R2_label = ifelse(!is.na(R2), sprintf("R2 approx %.2f", R2), ""),
    signif = case_when(
      is.na(pval_adj)      ~ "NS",
      pval_adj < 0.001     ~ "***",
      pval_adj < 0.01      ~ "**",
      pval_adj < 0.05      ~ "*",
      TRUE                 ~ "NS"),
    label = ifelse(!is.na(Correlation) & signif != "NS",
                   paste0("rho:", sprintf("%.3f", Correlation), "\n", R2_label, "\n", signif),
                   "")
  )
```

7. Heat-map des corrélations (Spearman)

```r
heatmap_plot <- ggplot(df_corr, aes(x = Explicative, y = Response, fill = Correlation)) +
  geom_tile(color = "grey30", aes(alpha = signif != "NS")) +
  scale_alpha_manual(values = c(`TRUE` = 1, `FALSE` = 0.35), guide = "none") +
  geom_text(aes(label = label), size = 3, lineheight = 1, family = "mono") +
  scale_fill_gradient2(low = "darkblue", mid = "white", high = "darkred",
                       midpoint = 0, limits = c(-1,1), na.value = "grey90",
                       name = "Corrélation p (Spearman)") +
  theme_minimal(base_size = 12) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  labs(title    = "Heat-map des corrélations (test de Spearman)",
       subtitle = "Valeurs: rho, R2 equivaut P2 et significativité après BH (* p<0.05)",
       x = "Variables explicatives", y = "Variables réponses")

print(heatmap_plot)
```



8. Sauvegarde des résultats

```
# Graphique
fname_plot <- "Heatmap_Correlation_Spearman.png"

ggsave(fname_plot, heatmap_plot, width = 14, height = 9, dpi = 300)

# CSV des résultats numériques
write_csv(df_corr %>%
            select(Response, Explicative, Correlation, R2, pval, pval_adj, signif),
          "Results_Spearman_correlations.csv")
```

#9. Complément d'enquete

Scatterplot simple avec ggplot2

```
ggplot(df_raw, aes(x = Delta_Run_Lib, y = CRAC_No_map_reads)) +
  geom_point(size = 2, alpha = 0.7, color = "#0072B2") +
  geom_smooth(method = "loess", se = TRUE, color = "darkred", linetype = "solid") +
  labs(
    title = "Relation entre le délai extraction → lancement (Delta_Run_Lib)",
    subtitle = "et le nombre de reads non mappés (STAR_No_map_reads)",
    x = "Delta_Run_Lib (jours)",
    y = "STAR_No_map_reads",
    caption = "Courbe de tendance: loess (locale, non linéaire)"
  ) +
```

```
  theme_minimal(base_size = 13)
```