Clustering & PCA

SYSTEM: R version 4.3.2 (2023-10-31)

DATE: 2024-01-29 15:01:49.137418

Contents

1 Sample overview

1.1 Data import

- Importing result table(s):

```r
# load counts matrix
df_counts <- read.delim("../../results/prepare/all_counts.tsv")
df_counts <- tidyr::pivot_longer(df_counts,
    cols = 3:ncol(df_counts),
    names_to = "sample", values_to = "n_reads"
)
# sort
df_counts <- arrange(df_counts, sample)
print("Import of counts table complete.")
```

```
## [1] "Import of counts table complete."
```

1.2 Sample table

- overview of samples

```r
# list of samples + generic options
list_samples <- unique(df_counts$sample)
figwidth <- 9
figheight <- round(1 + (length(list_samples) / 4))
figheight2 <- 3 * figheight

# output sample table
test <- df_counts %>%
    dplyr::group_by(sample) %>%
    dplyr::summarize(
        barcodes = length(unique(sgRNA)),
        total_reads = sum(n_reads, na.rm = TRUE),
        min_reads = min(n_reads, na.rm = TRUE),
        mean_reads = mean(n_reads, na.rm = TRUE),
        max_reads = max(n_reads, na.rm = TRUE),
    )
```

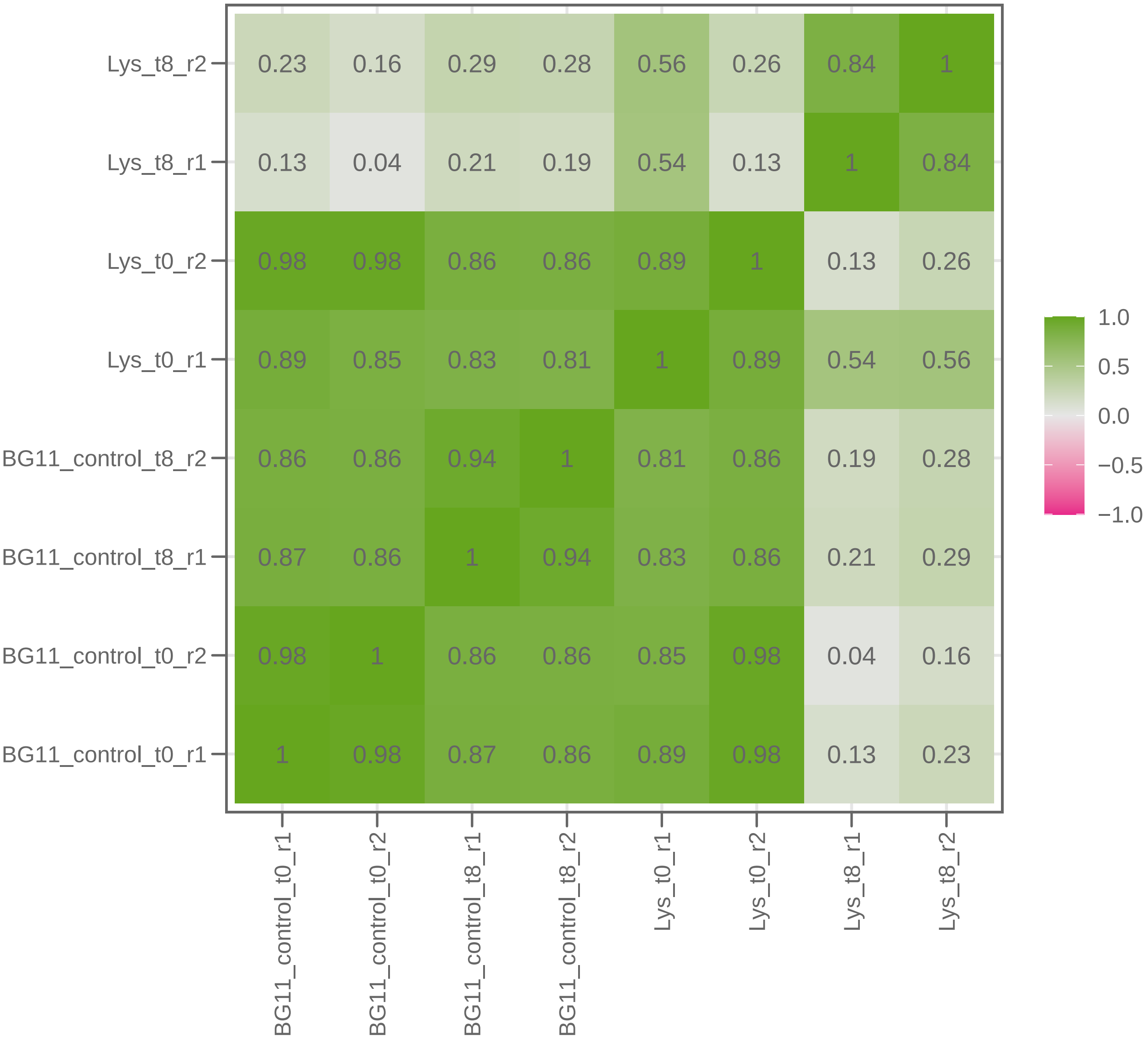
2 Quality control

```r
# define a custom ggplot2 theme (just for prettiness)
# custom ggplot2 theme that is reused for all later plots
custom_colors <- c("#E7298A", "#66A61E", "#E6AB02", "#7570B3", "#B3B3B3", "#1B9E77", "#D95F02", "#A6761
custom_range <- function(n = 5) {
    colorRampPalette(custom_colors[c(1, 5, 2)])(n)
}

custom_theme <- function(base_size = 12, base_line_size = 1.0, base_rect_size = 1.0, ...) {
    theme_light(base_size = base_size, base_line_size = base_line_size, base_rect_size = base_rect_size
        title = element_text(colour = grey(0.4), size = 10),
        plot.margin = unit(c(12, 12, 12, 12), "points"),
        axis.ticks.length = unit(0.2, "cm"),
        axis.ticks = element_line(colour = grey(0.4), linetype = "solid", lineend = "round"),
        axis.text.x = element_text(colour = grey(0.4), size = 10),
        axis.text.y = element_text(colour = grey(0.4), size = 10),
        panel.grid.major = element_line(size = 0.6, linetype = "solid", colour = grey(0.9)),
        panel.grid.minor = element_blank(),
        panel.border = element_rect(linetype = "solid", colour = grey(0.4), fill = NA, size = 1.0),
        panel.background = element_blank(),
        strip.background = element_blank(),
        strip.text = element_text(colour = grey(0.4), size = 10, margin = unit(rep(3, 4), "points")),
        legend.text = element_text(colour = grey(0.4), size = 10),
        legend.title = element_blank(),
        legend.background = element_blank(),
        ...
    )
}
```

2.1 Sample and replicate correlation coefficent (R)

```r
p <- df_counts %>%
    tidyr::pivot_wider(names_from = "sample", values_from = "n_reads") %>%
    dplyr::select(-c(1:2)) %>%
    cor() %>%
    dplyr::as_tibble() %>%
    dplyr::mutate(sample1 = colnames(.)) %>%
    tidyr::pivot_longer(
        cols = !sample1,
        names_to = "sample2", values_to = "cor_coef"
    ) %>%
    ggplot(aes(x = sample1, y = sample2, fill = cor_coef)) +
    geom_tile() +
    geom_text(color = grey(0.4), aes(label = round(cor_coef, 2))) +
    custom_theme() +
    labs(title = "", x = "", y = "") +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
    scale_fill_gradientn(
        colours = c(custom_colors[1], grey(0.9), custom_colors[2]),
        limits = c(-1, 1)
    )
p
```



```
ggsave("correlation_samples.pdf", plot=p, width=18, height=18, units="cm")
```

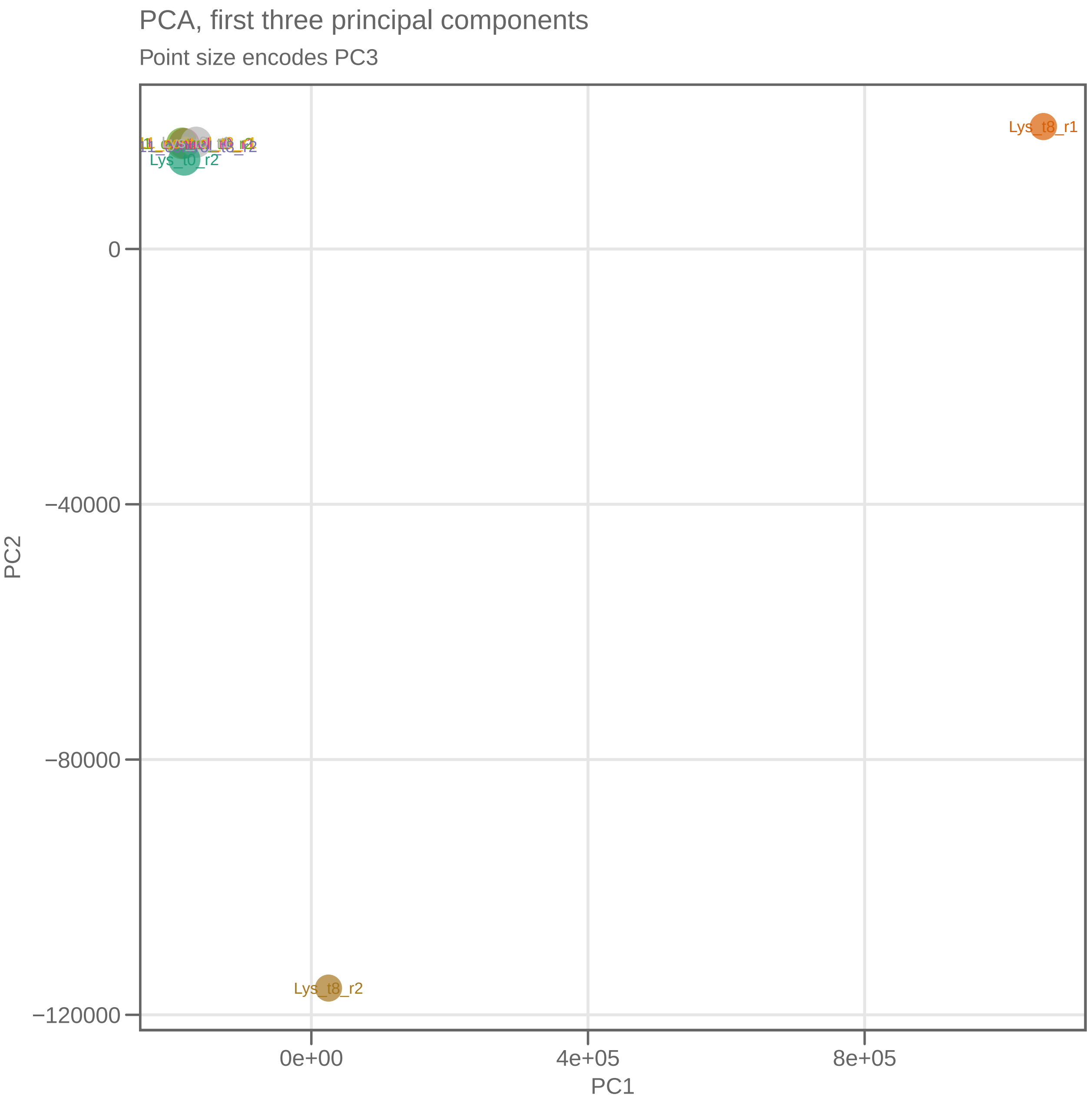
2.2 Sample and replicate similarity with PCA

```
pca_result <- df_counts %>%
    tidyr::pivot_wider(names_from = "sample", values_from = "n_reads") %>%
    dplyr::select(-c(1:2)) %>%
    as.matrix() %>%
    t() %>%
    replace(., is.na(.), 0) %>%
    prcomp()

df_PCA <- pca_result$x %>%
    as_tibble(rownames = "sample")

p <- df_PCA %>%
    ggplot(aes(x = PC1, y = -PC2, size = PC3, color = sample, label = sample)) +
    geom_point(alpha = 0.7) +
    geom_text(size = 2.5, show.legend = FALSE) +
    labs(
        title = "PCA, first three principal components",
        subtitle = "Point size encodes PC3", x = "PC1", y = "PC2"
    ) +
    custom_theme(legend.position = 0, aspect = 1) +
    scale_color_manual(values = colorRampPalette(custom_colors)(nrow(df_PCA))) +
    guides(size = "none")

p
```



```
ggsave("PCA.pdf", plot=p, width=18, height=18, units="cm")
```

3 Report info

This analysis is based on code by Michael Jahn (Science For Life Laboratory (KTH), Stockholm, Sweden; Max-Planck-Unit for the Science of Pathogens, Berlin, Germany), which is part of the nf-core-crispriscreen pipeline (https://github.com/MPUSP/nf-core-crispriscreen)

Date: 2024-01-29

Author: Ute Hoffmann (SciLifeLab, Stockholm, Sweden)

4 Session Info

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.3 LTS
## 
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3 
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so;  LAPACK version 3.10.0
## 
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              
##  [3] LC_TIME=sv_SE.UTF-8        LC_COLLATE=en_US.UTF-8    
##  [5] LC_MONETARY=sv_SE.UTF-8    LC_MESSAGES=en_US.UTF-8   
##  [7] LC_PAPER=sv_SE.UTF-8       LC_NAME=C                 
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C            
## [11] LC_MEASUREMENT=sv_SE.UTF-8 LC_IDENTIFICATION=C       
## 
## time zone: Europe/Stockholm
## tzcode source: system (glibc)
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
## [1] Hmisc_5.1-1   tidyr_1.3.0   ggplot2_3.4.4 dplyr_1.0.10  knitr_1.45   
## 
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    stringi_1.7.12    digest_0.6.33    
##  [5] magrittr_2.0.3    evaluate_0.23     grid_4.3.2        fastmap_1.1.1    
##  [9] nnet_7.3-19       backports_1.4.1   DBI_1.1.3         Formula_1.2-5    
## [13] gridExtra_2.3     purrr_1.0.2       fansi_1.0.5       scales_1.2.1     
## [17] textshaping_0.3.7 cli_3.6.1         rlang_1.1.2       munsell_0.5.0    
## [21] base64enc_0.1-3   withr_2.5.0       yaml_2.3.7        tools_4.3.2      
## [25] checkmate_2.1.0   htmlTable_2.4.2   colorspace_2.1-0  assertthat_0.2.1 
## [29] vctrs_0.6.4       R6_2.5.1          rpart_4.1.23      lifecycle_1.0.4  
## [33] stringr_1.5.0     htmlwidgets_1.5.4 ragg_1.2.6        foreign_0.8-86   
## [37] cluster_2.1.6     pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.4     
## [41] glue_1.6.2        data.table_1.14.8 systemfonts_1.0.4 highr_0.10       
## [45] xfun_0.41         tibble_3.2.1      tidyselect_1.2.0  rstudioapi_0.14  
## [49] farver_2.1.1      htmltools_0.5.7   labeling_0.4.2    rmarkdown_2.25   
## [53] compiler_4.3.2
```