

# Analyze CRISPRi growth competition data for growth at high CO2 gas feeds, calculate adjusted p values between two conditions

Ute Hoffmann (Science For Life Laboratory (KTH), Stockholm, Sweden)

oktober 22, 2024

## Contents

1 Aim of the analysis	1
2 Analysis	1
3 Add annotation to results tables	2
Session info	2

## 1 Aim of the analysis

Calculation of Wilcoxon rank sum test between sgRNA fitness values between 4% CO2 data and 30% CO2 data.

## 2 Analysis

In a first step, the results given by the Nextflow pipeline are loaded.

```
load("../results_controls/sgRNAs/fitness/result.Rdata")
num_sgRNAs <- read_tsv("../input/number_sgRNAs_per_target.tsv", col_names = c("sgRNA_target", "num_sgRNAs"))
DESeq_result_table <- left_join(DESeq_result_table, num_sgRNAs)
```

```
get_controls <- function(condition, sgRNA_target){
  sgRNA_spec = sgRNA_target
  cond_spec = condition
  control_table <- DESeq_result_table[DESeq_result_table$condition != cond_spec & DESeq_result_table$sgRNA_target == sgRNA_target]
  control_table$fitness
}
```

```
DESeq_result_table <- dplyr::left_join(
  DESeq_result_table,
  DESeq_result_table %>%
    dplyr::group_by(sgRNA_target, condition, time) %>%
    dplyr::summarize(
      .groups = "keep",
      # apply Wilcoxon rank sum test against other condition
      p_fitness_condition = stats::wilcox.test(
        x = fitness,
        y = get_controls(condition, sgRNA_target),
```

```

        alternative = "two.sided"
      )$p.value
    ),
    by = c("sgRNA_target", "condition", "time")
  )

DESeq_result_table <- DESeq_result_table %>%
  group_by(condition, time) %>%
  mutate(
    p_fitness_condition_adj = stats::p.adjust(p_fitness_condition, method = "BH")
  )

save(DESeq_result_table, file = "../R_results_controlssgRNAs/result.Rdata")
readr::write_tsv(DESeq_result_table, file = "../R_results_controlssgRNAs/result.tsv")

```

### 3 Add annotation to results tables

In the following, annotation is added to the results table provided by the Nextflow pipeline. Mapping of the sgRNA targets to slr-locus tags is given in this file, downloaded on 24/02/23: [https://github.com/m-jahn/R-notebook-crispri-lib/blob/master/sgRNA\\_library\\_V2/data/input/mapping\\_trivial\\_names.tsv](https://github.com/m-jahn/R-notebook-crispri-lib/blob/master/sgRNA_library_V2/data/input/mapping_trivial_names.tsv). The appended annotation is based on Uniprot and Cyanobase, partially edited manually. The table used for annotation was created beginning of 2021. Therefore, it does not include several genes which were only recently characterized. For a detailed description of all the columns given in the results tables, consult <https://mpusp.github.io/nf-core-crispriscreeen/output> or <https://www.biorxiv.org/content/10.1101/2023.02.13.528328v1.full.pdf+htmls>.

```

mapping_gene_locus <- read_tsv("../input/2023-02-24_mapping_trivial_names.tsv", show_col_types=FALSE)
names(mapping_gene_locus) <- c("sgRNA_target", "locus")
DESeq_result_table <- DESeq_result_table %>% left_join(mapping_gene_locus)

annotation <- read_tsv("../input/annotation_locusTags_stand13012021.csv", show_col_types = FALSE)
annotation_2 <- annotation[,c(1,2,3)]
names(annotation_2) <- c("locus", "Gene name", "Product")
DESeq_result_table <- DESeq_result_table %>% left_join(annotation_2)

write_tsv(DESeq_result_table, file="../R_results_controlssgRNAs/annotated_DESeq_result_table_comparison")
df_reduced_info <- unique(subset(DESeq_result_table, DESeq_result_table$time==8 | DESeq_result_table$time==12))
write_tsv(df_reduced_info, file="../R_results_controlssgRNAs/Reduced_annotated_DESeq_result_table_comparison")

df_red_wide <- pivot_wider(df_reduced_info, names_from=condition, values_from=c(wmean_fitness, sd_fitness))
df_red_wide$impact_score <- (df_red_wide$wmean_fitness_C02_30percent - df_red_wide$wmean_fitness_C02_4percent)
df_red_wide$combined_impact_score <- abs(df_red_wide$impact_score) * -log10(df_red_wide$p_fitness_condition_adj)
write_tsv(df_red_wide, file="../R_results_controlssgRNAs/Wide_DESeq_result_table_comparisonsConditions")

```

### Session info

```

## R version 4.4.1 (2024-06-14)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.4 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/libopenblas-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] magrittr_2.0.3  lubridate_1.9.3 forcats_1.0.0  stringr_1.5.1
## [5] dplyr_1.1.4     purrr_1.0.2     readr_2.1.5    tidyr_1.3.1
## [9] tibble_3.2.1    ggplot2_3.5.1    tidyverse_2.0.0 knitr_1.48
##
## loaded via a namespace (and not attached):
## [1] bit_4.5.0      gtable_0.3.5    crayon_1.5.3    compiler_4.4.1
## [5] tidyselect_1.2.1 parallel_4.4.1  scales_1.3.0    yaml_2.3.10
## [9] fastmap_1.2.0  R6_2.5.1        generics_0.1.3  munsell_0.5.1
## [13] pillar_1.9.0   tzdb_0.4.0      rlang_1.1.4     utf8_1.2.4
## [17] stringi_1.8.4  xfun_0.48       bit64_4.5.2     timechange_0.3.0
## [21] cli_3.6.3      withr_3.0.1     digest_0.6.37   grid_4.4.1
## [25] vroom_1.6.5    rstudioapi_0.17.0 hms_1.1.3       lifecycle_1.0.4
## [29] vctrs_0.6.5    evaluate_1.0.1  glue_1.8.0      fansi_1.0.6
## [33] colorspace_2.1-1 rmarkdown_2.28  tools_4.4.1     pkgconfig_2.0.3
## [37] htmltools_0.5.8.1

```