Analyze CRISPRi growth competition data after treatment with or without Lysine, use data with limma Normalization

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

januari 12, 2024

Contents

1 Aim of the analysis

Basic visualization of CRISPRi data for cultivation with lysine. Data analysis was performed using nf-core-crispriscreen pipeline (https://github.com/MPUSP/nf-core-crispriscreen).

2 Analysis

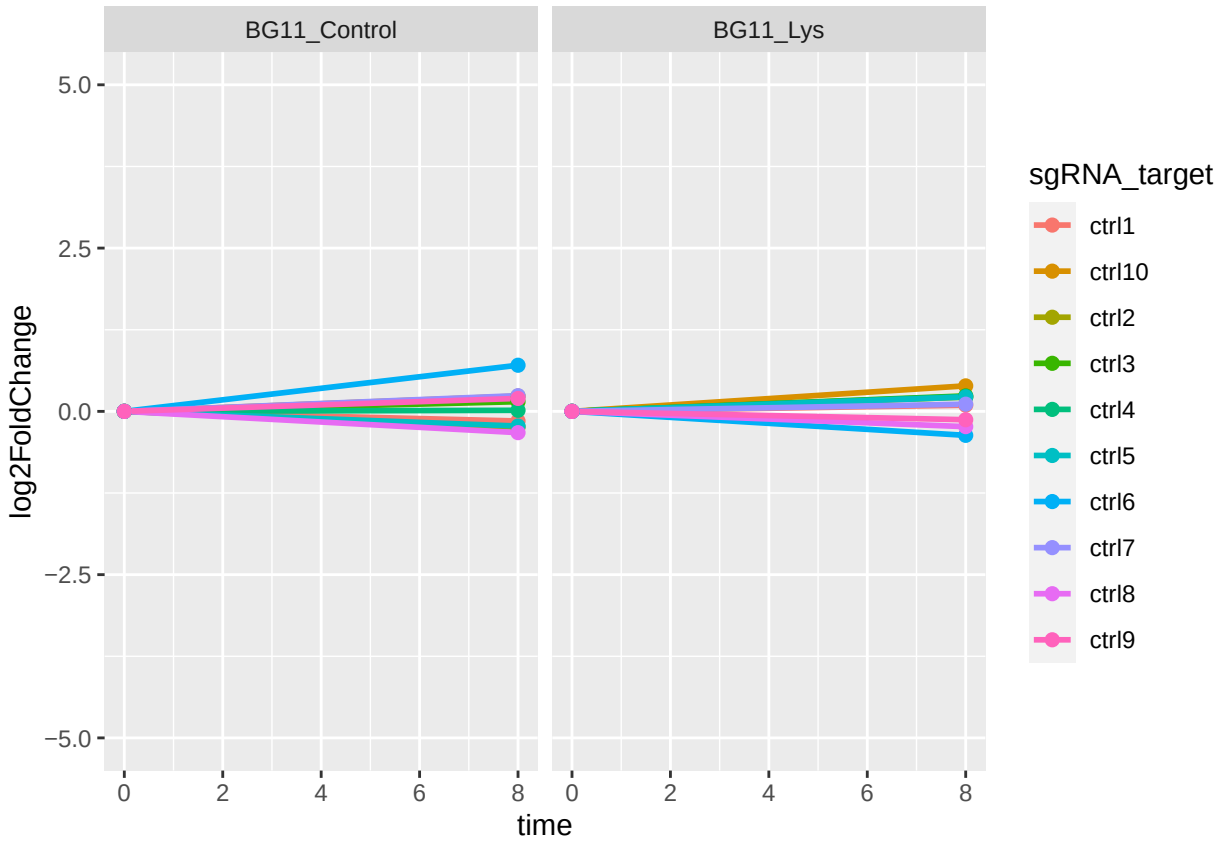
In a first step, the results given by the Nextflow pipeline are loaded.

```
load("../results_limmaNormalization/fitness/result.Rdata")
```

2.1 Diagnostic plot to check if control sgRNAs look ok

Several control sgRNAs are included in the CRISPRi library. These control sgRNAs do not target any specific gene and serve as a control. Here, most of them behave neutrally, except control sgRNA 6 under certain conditions.

```
plot_controls_sgRNAs <- DESeq_result_table %>% filter(grepl("ctrl", sgRNA_target)) %>%
  ggplot(aes(x = time, y = log2FoldChange, color = sgRNA_target)) +
  geom_line(linewidth = 1) + geom_point(size = 2) + ylim(-5, 5) + facet_wrap(~ condition, ncol = 4)
print(plot_controls_sgRNAs)
```




```
ggsave("../R_results/limmaNorm_plot_control_sgRNAs.pdf", plot=plot_controls_sgRNAs, width=12, height=12
```

2.2 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 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-crispriscreen/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/limmaNorm_annotated_DESeq_result_table.tsv")
df_reduced_info <- unique(subset(DESeq_result_table, DESeq_result_table$time==8 | DESeq_result_table$ti
write_tsv(df_reduced_info, file="../R_results/Reduced_annotated_DESeq_result_table.tsv")
```

```
df_red_wide <- pivot_wider(df_reduced_info, names_from=condition, values_from=c(wmean_fitness, sd_fitnes
write_tsv(df_red_wide, file="../R_results/limmaNorm_Wide_DESeq_result_table.tsv")
```

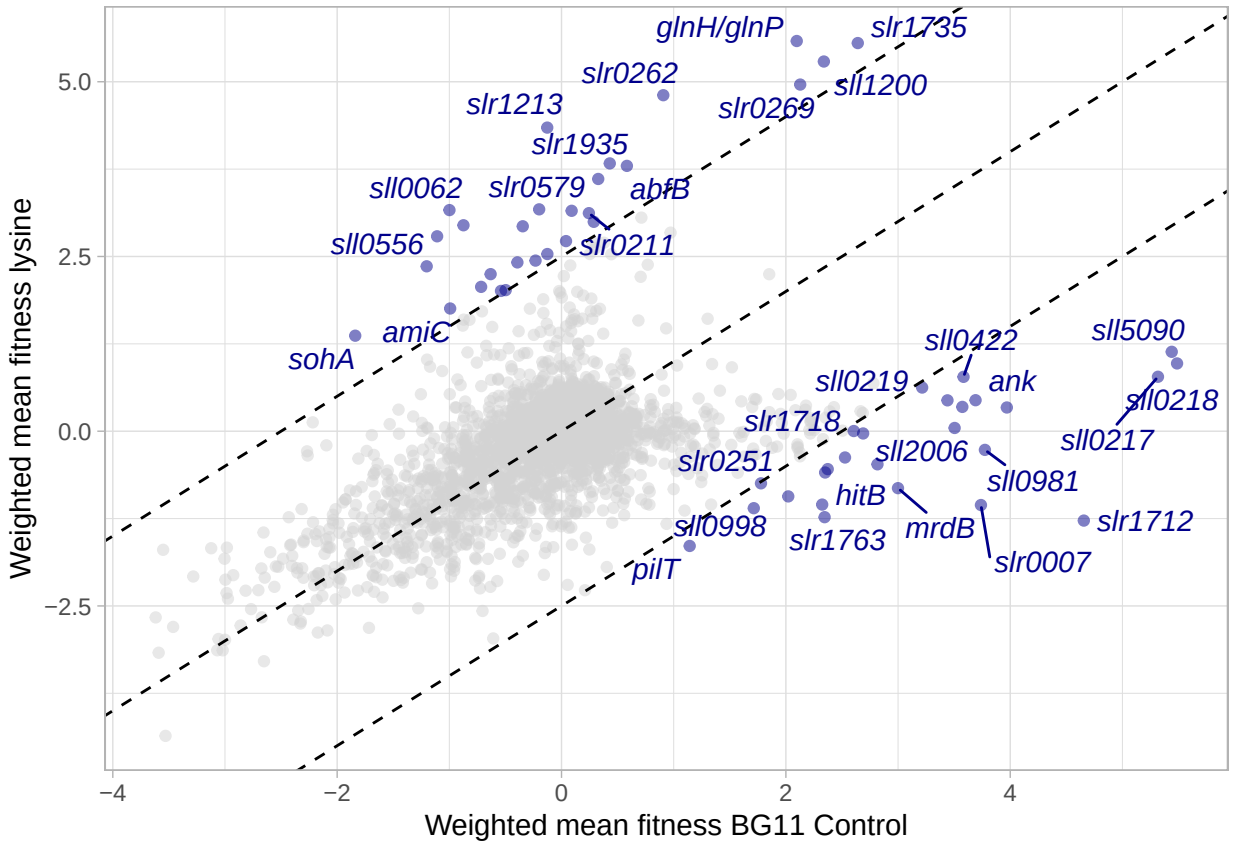
2.3 Visualization

The weighted mean fitness value combines the values of the different sgRNAs targeting the same gene. Fitness-fitness plots were created to identify genes which behave differently when amino acids were added compared to the control cultivation. This was performed separately for ncRNAs and protein-coding genes.

2.3.1 Protein-coding genes

```
df_reduced <- unique(subset(DESeq_result_table, DESeq_result_table$time==8)[,c(2,4,20)])
df_red_ncRNAs <- subset(df_reduced, grepl("nc_", df_reduced$sgRNA_target))
df_red_no_ncRNAs <- subset(df_reduced, !grepl("nc_", df_reduced$sgRNA_target))
df_red_wide <- pivot_wider(df_red_no_ncRNAs,names_from="condition", values_from=c("wmean_fitness"))
```

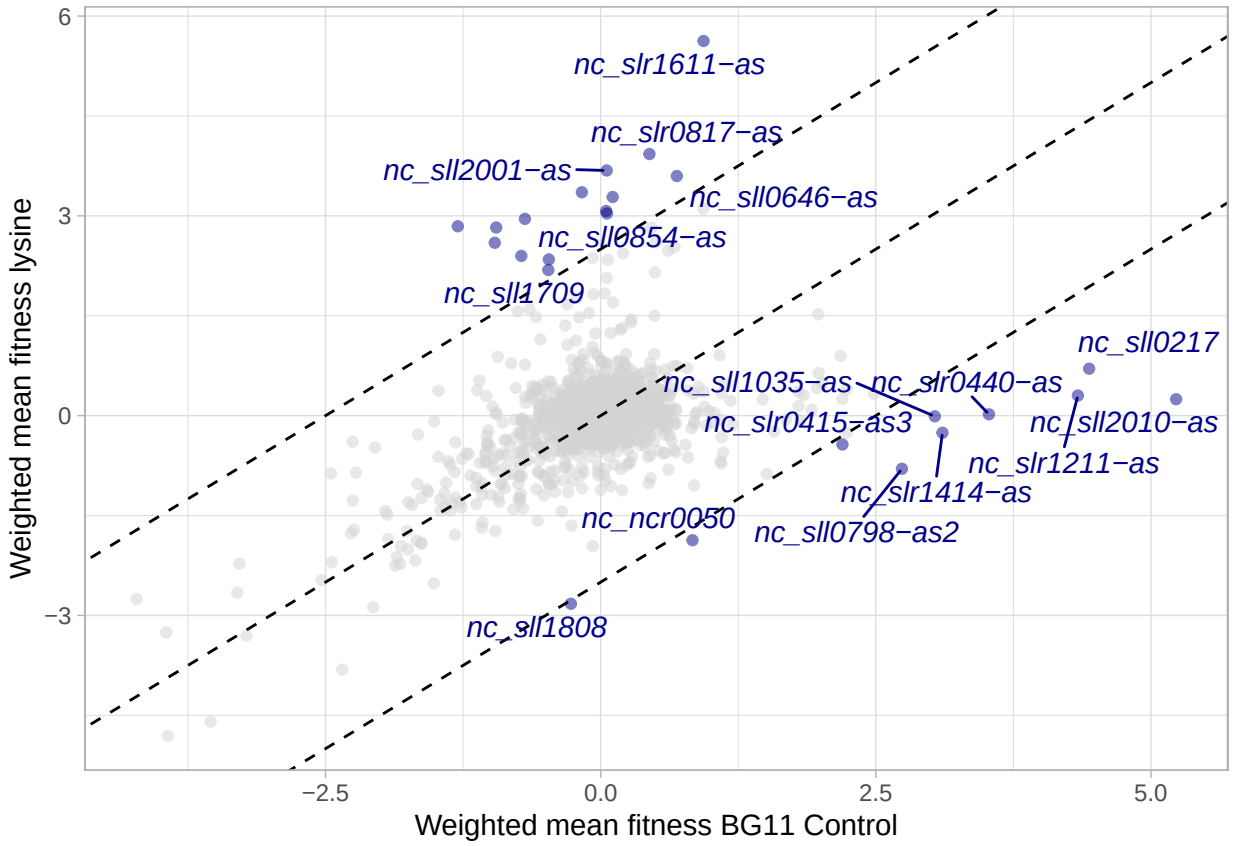
```
plot_fitness_fitness <- function(df_input, y_axis, y_axis_label, x_axis="BG11_Control", x_axis_label="W
  df_input$diff <- "NO"
  df_input$diff[(df_input[[x_axis]] - df_input[[y_axis]] > 2.5) | (df_input[[x_axis]] - df_input[[y_axis
  # prepare labels for plot
  df_input$delabel <- NA
  df_input$delabel[df_input$diff !="NO"] <- df_input$sgRNA_target[df_input$diff != "NO"]
  mycolors <- c("darkblue",  "#d3d3d3b2")
  names(mycolors) <- c("YES", "NO")
  p <- ggplot(data=df_input, aes(x=eval(parse(text=x_axis)), y=eval(parse(text=y_axis)), label=delabel,
    theme_light() + labs(y=y_axis_label, x=x_axis_label) + theme(legend.position = "none") + geom_ablin
  ggsave(filename = filename_save, plot=p, width=12, height=12, units="cm")
return(p)
}

plot_fitness_fitness(df_red_wide, "BG11_Lys", y_axis_label="Weighted mean fitness lysine", filename_sav
```



2.3.2 ncRNAs

These include antisense RNAs, but also other ncRNAs.

```
df_red_wide <- pivot_wider(df_red_ncRNAs,names_from="condition", values_from=c("wmean_fitness"))
plot_fitness_fitness(df_red_wide, "BG11_Lys", y_axis_label="Weighted mean fitness lysine", filename_sav
```



Session info

```
## 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] enrichplot_1.20.3     clusterProfiler_4.8.3 magrittr_2.0.3
```

```
##  [4] forcats_0.5.2         stringr_1.5.0         dplyr_1.0.10
##  [7] purrr_1.0.2           readr_2.1.4           tidyr_1.3.0
## [10] tibble_3.2.1          tidyverse_1.3.1       ggpubr_0.6.0
## [13] ggrepel_0.9.4         ggplot2_3.4.4         knitr_1.45
##
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3      rstudioapi_0.14         jsonlite_1.8.7
##   [4] farver_2.1.1            rmarkdown_2.25          ragg_1.2.6
##   [7] fs_1.6.3                zlibbioc_1.46.0         vctrs_0.6.4
##  [10] memoise_2.0.1           RCurl_1.98-1.13         ggtree_3.8.2
##  [13] rstatix_0.7.2           htmltools_0.5.7         haven_2.5.1
##  [16] broom_1.0.1             cellranger_1.1.0        gridGraphics_0.5-1
##  [19] plyr_1.8.9              lubridate_1.8.0         cachem_1.0.8
##  [22] igraph_1.5.1            lifecycle_1.0.4         pkgconfig_2.0.3
##  [25] gson_0.1.0              Matrix_1.6-3            R6_2.5.1
##  [28] fastmap_1.1.1           GenomeInfoDbData_1.2.10 digest_0.6.33
##  [31] aplot_0.2.2             colorspace_2.1-0        patchwork_1.1.3
##  [34] AnnotationDbi_1.62.2    S4Vectors_0.38.2        textshaping_0.3.7
##  [37] RSQLite_2.3.3           labeling_0.4.2          fansi_1.0.5
##  [40] httr_1.4.4              polyclip_1.10-6         abind_1.4-5
##  [43] compiler_4.3.2          bit64_4.0.5             withr_2.5.0
##  [46] downloader_0.4          backports_1.4.1         BiocParallel_1.34.2
##  [49] carData_3.0-5           viridis_0.6.4           DBI_1.1.3
##  [52] highr_0.10              ggforce_0.4.1           ggsignif_0.6.4
##  [55] MASS_7.3-60             HDO.db_0.99.1           tools_4.3.2
##  [58] scatterpie_0.2.1        ape_5.7-1               glue_1.6.2
##  [61] nlme_3.1-163            GOSemSim_2.26.1         shadowtext_0.1.2
##  [64] grid_4.3.2              reshape2_1.4.4          fgsea_1.26.0
##  [67] generics_0.1.3          gtable_0.3.4            tzdb_0.4.0
##  [70] data.table_1.14.8       hms_1.1.3               tidygraph_1.2.3
##  [73] xml2_1.3.5              car_3.1-2               utf8_1.2.4
##  [76] XVector_0.40.0          BiocGenerics_0.46.0     pillar_1.9.0
##  [79] vroom_1.6.4             yulab.utils_0.1.0       splines_4.3.2
##  [82] tweenr_2.0.2            treeio_1.24.3           lattice_0.22-5
##  [85] bit_4.0.5               tidyselect_1.2.0        GO.db_3.17.0
##  [88] Biostrings_2.68.1       gridExtra_2.3           IRanges_2.34.1
##  [91] stats4_4.3.2            xfun_0.41               graphlayouts_1.0.2
##  [94] Biobase_2.60.0          stringi_1.7.12          lazyeval_0.2.2
##  [97] ggfun_0.1.3             yaml_2.3.7              evaluate_0.23
## [100] codetools_0.2-19        ggraph_2.1.0            qvalue_2.32.0
## [103] ggplotify_0.1.2         cli_3.6.1               systemfonts_1.0.4
## [106] munsell_0.5.0           modelr_0.1.9            Rcpp_1.0.11
## [109] GenomeInfoDb_1.36.4     readxl_1.4.3            dbplyr_2.2.1
## [112] png_0.1-8               parallel_4.3.2          assertthat_0.2.1
## [115] blob_1.2.3              DOSE_3.26.2             reprex_2.0.2
## [118] bitops_1.0-7            viridisLite_0.4.2       tidytree_0.4.5
## [121] scales_1.2.1            crayon_1.5.2            rlang_1.1.2
## [124] cowplot_1.1.2           fastmatch_1.1-4         KEGGREST_1.40.1
## [127] rvest_1.0.3
```