

# Analyze CRISPRi growth competition data for growth at high CO2 gas feeds

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

september 30, 2024

## Contents

1 Aim of the analysis	1
2 Analysis	1
2.1 Diagnostic plot to check if control sgRNAs look ok	1
2.2 Diagnostic plot of fitness value distributions	2
2.3 Diagnostic plot to check if replicates correlate	4
2.4 Add annotation to results tables	7
2.5 Visualization	8
2.6 Growth curves of most interesting targets	10
2.7 GSEA	12
Session info	28

## 1 Aim of the analysis

Basic visualization of CRISPRi data for cultivation with different levels of CO2 (4% or 30% CO2). 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_controls/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)

count_matrix <- read_tsv("../results_controls/prepare/all_counts.tsv")
count_matrix$Gene <- NULL

df_samplesheet <- readr::read_csv("../input/samplesheet_CRISPRi_CO2_Elena.csv", col_types = cols()) %>%
  select(all_of(c("sample", "condition", "replicate", "time", "group", "reference_group")))
  dplyr::mutate(group = factor(`group`))
df_samplesheet$name <- paste("gen_", df_samplesheet$time, "_r_", df_samplesheet$replicate, sep="")
```

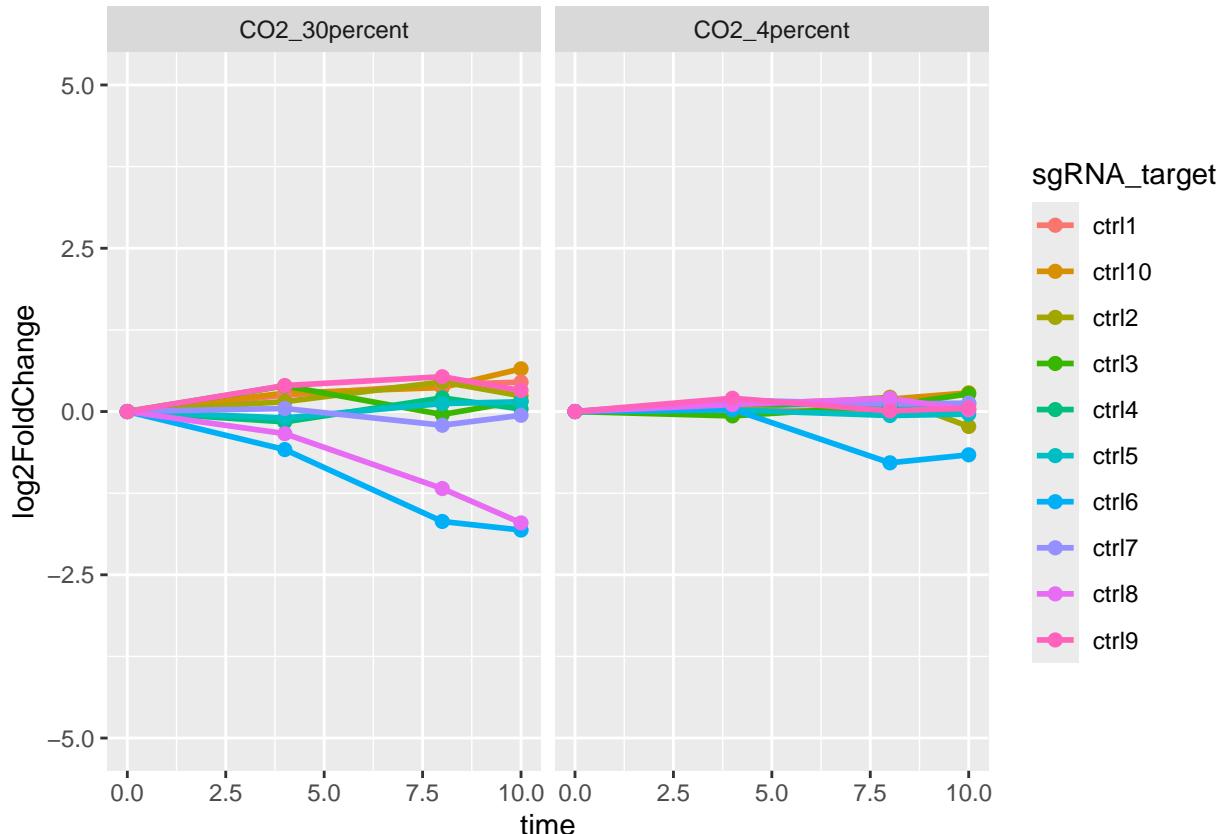
### 2.1 Diagnostic plot to check if control sgRNAs look ok

Several control sgRNAs are included in the CRISPRi library. These control sgRNAs do not target any specific gene and serve as a control.

```

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)

```



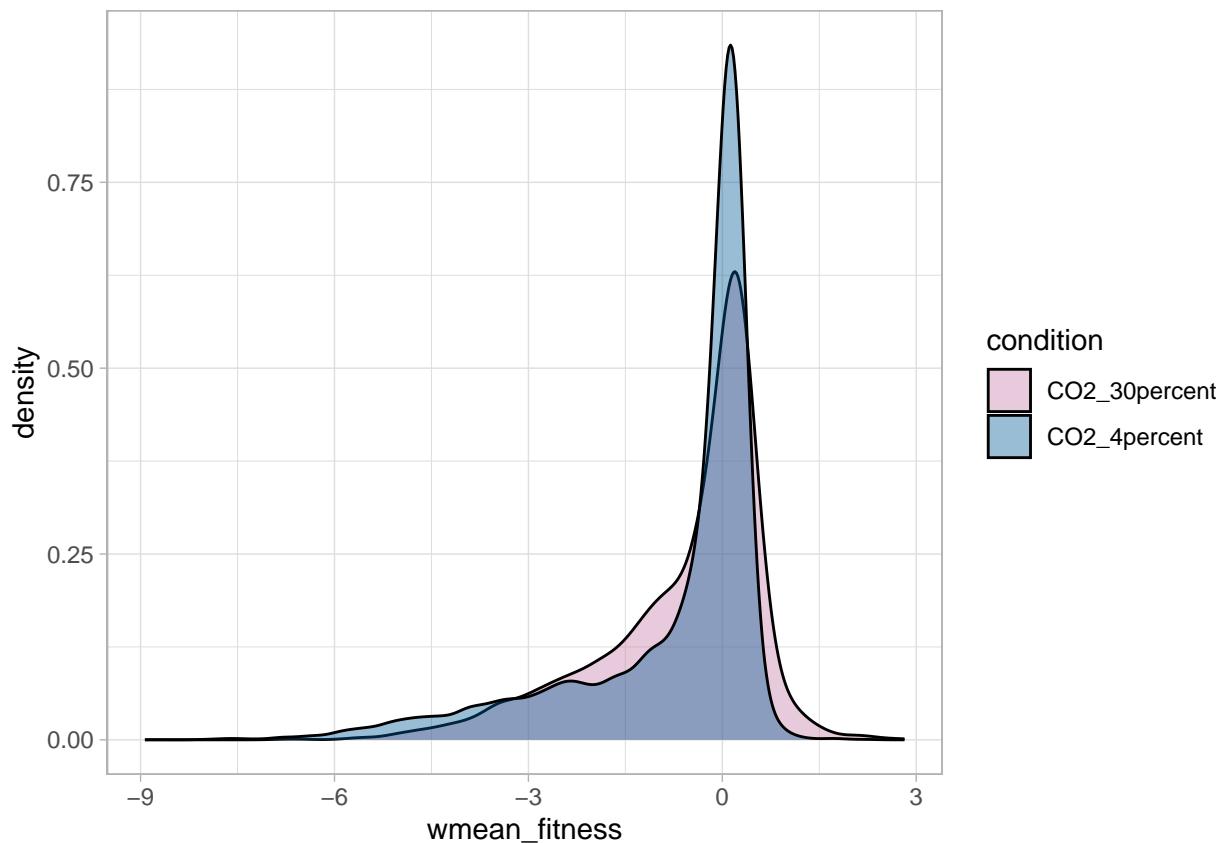
```
ggsave("../R_results_controls_sgRNAs/plot_control_sgRNAs.pdf", plot=plot_controls_sgRNAs, width=12, height=8)
```

## 2.2 Diagnostic plot of fitness value distributions

```

subs <- unique(DESeq_result_table[,c("sgRNA", "condition", "wmean_fitness")])
p <- ggplot(subs, aes(x=wmean_fitness, group=condition, fill=condition)) + geom_density(adjust=1.5, alpha=0.5)
p

```

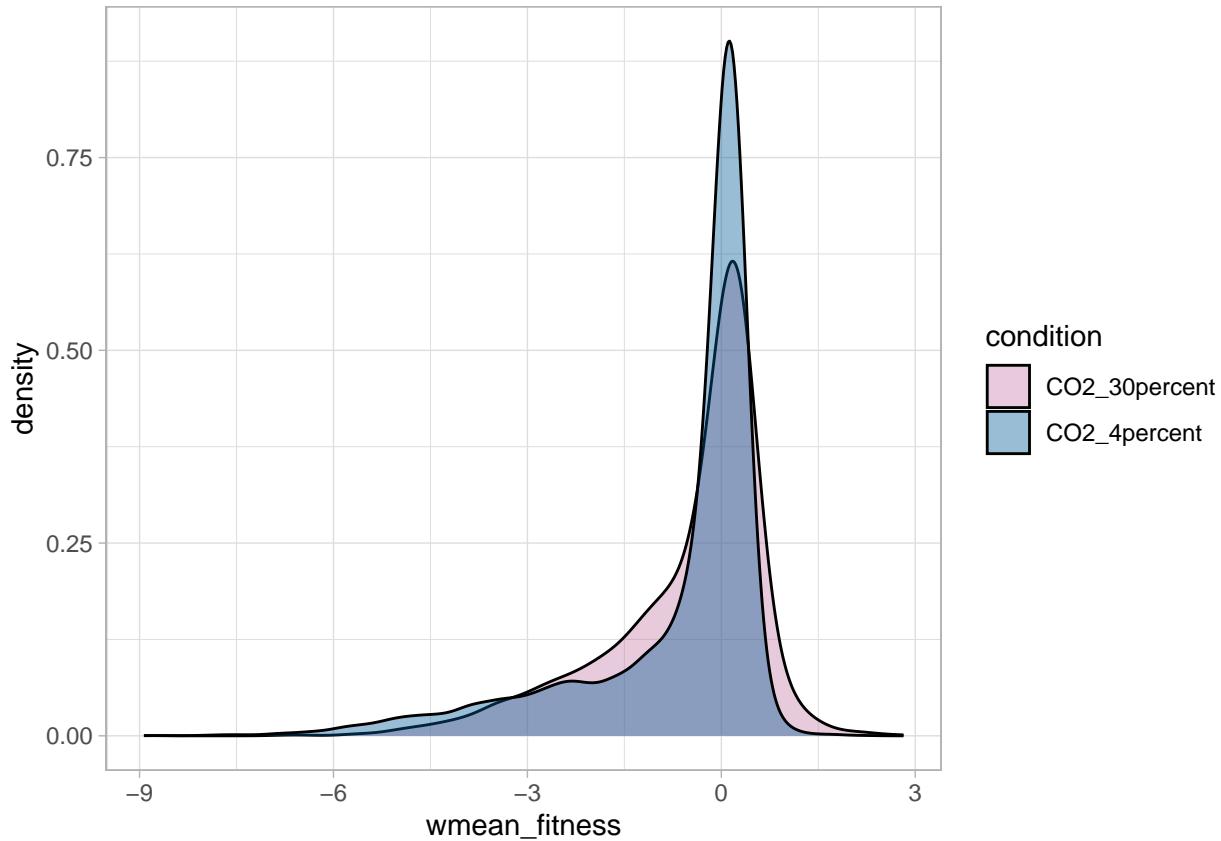


```

ggsave("../R_results_controlsgRNAs/wmean_fitness_sgRNA-level_densityplot.pdf", plot=p, width=11.5, height=8.5)
ggsave("../R_results_controlsgRNAs/wmean_fitness_sgRNA-level_densityplot.png", plot=p, width=11.5, height=8.5)

subs <- unique(DESeq_result_table[,c("sgRNA_target", "condition", "wmean_fitness")])
p <- ggplot(subs, aes(x=wmean_fitness, group=condition, fill=condition)) + geom_density(adjust=1.5, alpha=0.8)
p

```



```
ggsave("../R_results_controlsgRNAs/wmean_fitness_sgRNAtarget-level_densityplot.pdf", plot=p, width=11.5)
ggsave("../R_results_controlsgRNAs/wmean_fitness_sgRNAtarget-level_densityplot.png", plot=p, width=11.5)
```

### 2.3 Diagnostic plot to check if replicates correlate

This is a diagnostic plot to check if replicates correlate. Samples are clustered according to correlation and the resulting tree is divided into 6 sections according to which samples cluster best. Replicates at different conditions and generations cluster and the 30% samples taken at generations 8 and 10 are most dissimilar to the other samples.

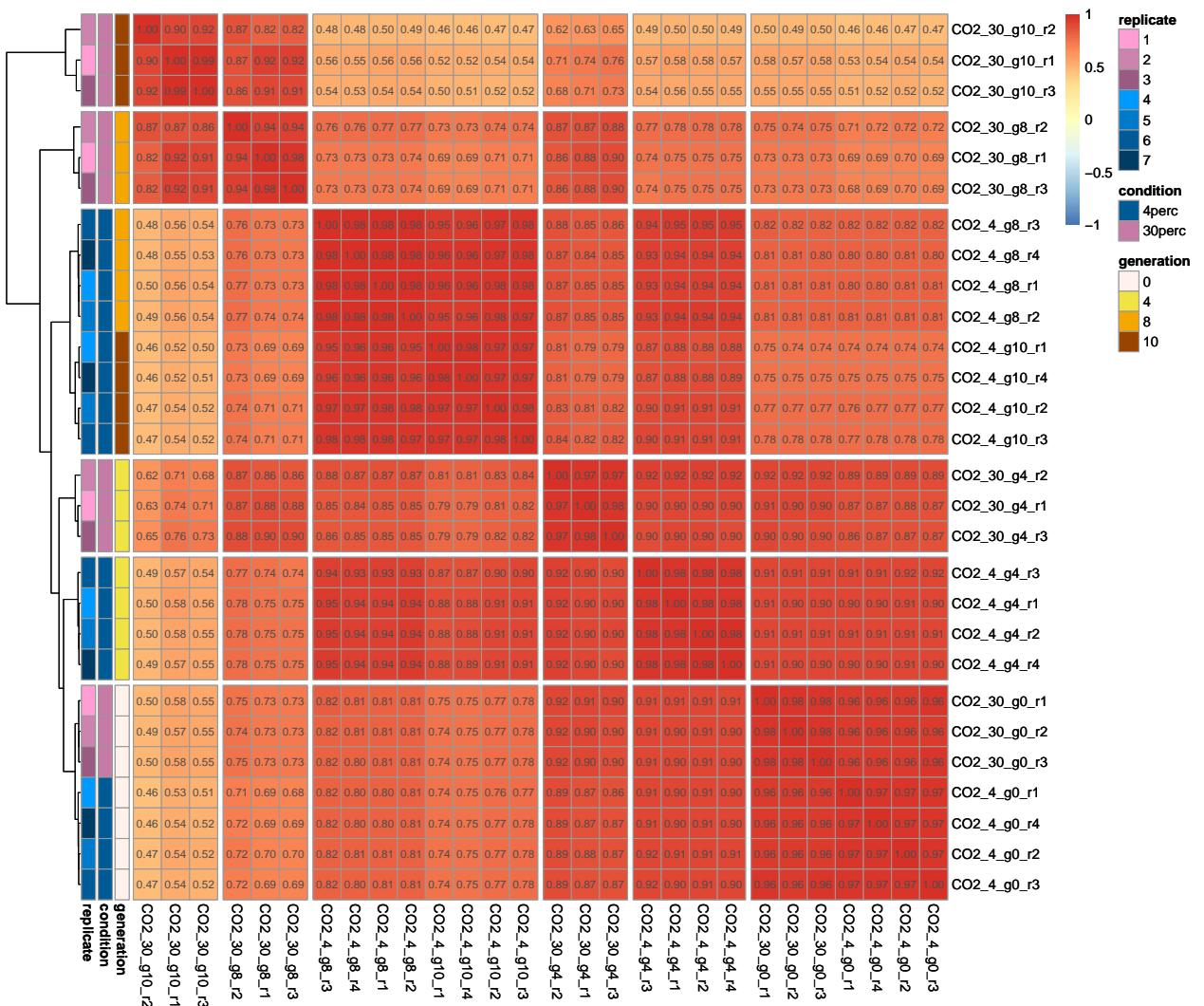
```
df_counts <- tidyverse::pivot_longer(count_matrix,
  cols = 2:ncol(count_matrix),
  names_to = "sample", values_to = "n_reads"
)

# sort
df_counts <- arrange(df_counts, sample)
#df_counts <- left_join(df_samplesheet, df_counts)

df_correlation <- df_counts %>%
  tidyverse::pivot_wider(names_from = "sample", values_from = "n_reads") %>%
  dplyr::select(-c(1)) %>%
  cor()

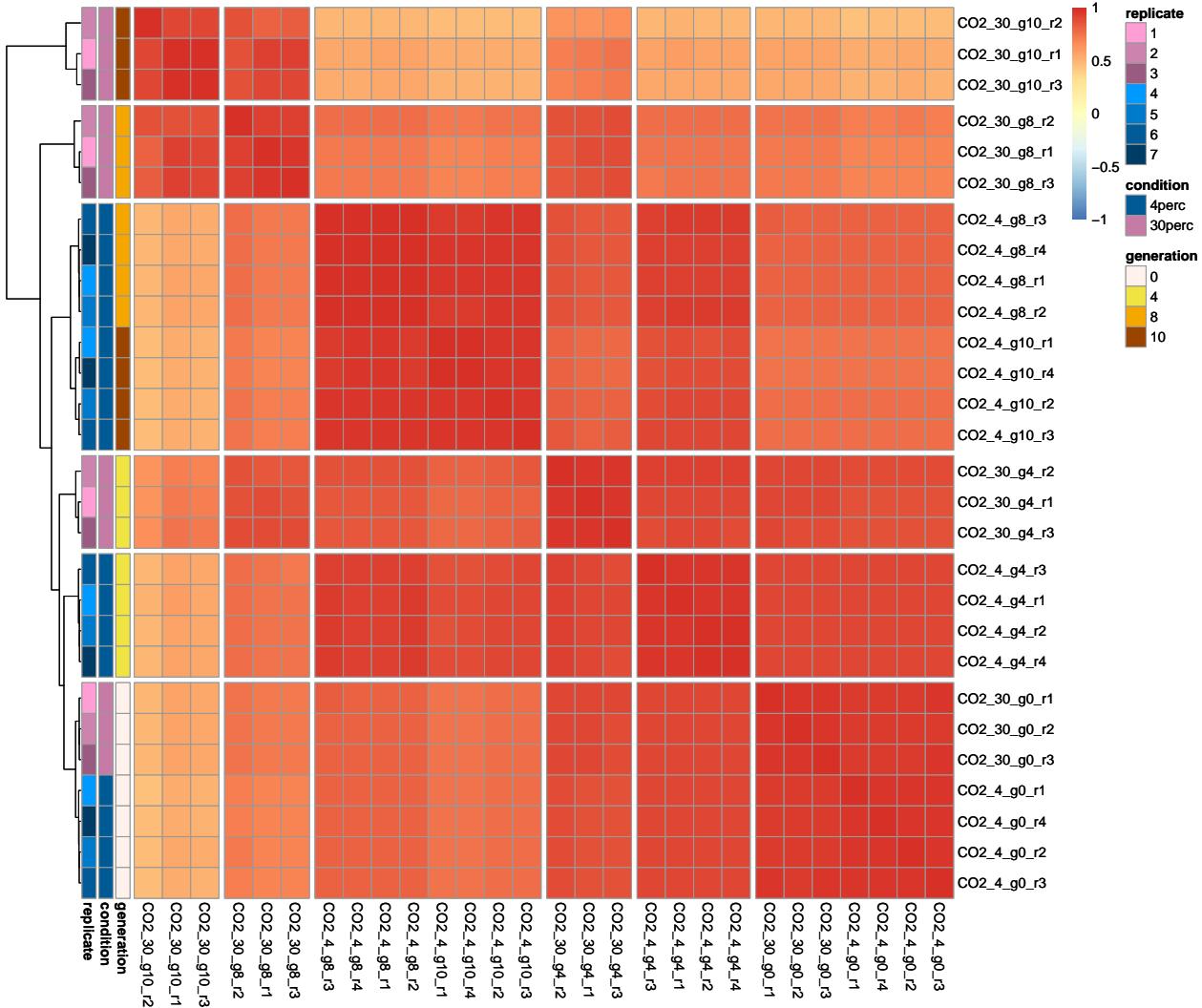
# https://bioinformatics.stackexchange.com/questions/22502/manually-set-range-of-colour-scale-in-pheatmap
color.divisions <- 100
```





```
ggsave("../R_results_controlsgRNAs/correlation/correlation_samples_clustering.png", plot=p, width=11.5  
ggsave("../R_results_controlsgRNAs/correlation/correlation_samples_clustering.pdf", plot=p, width=11.5
```

```
p <- pheatmap(df_correlation, display_numbers=FALSE, treeheight_col=0, cutree_rows = 6, cutree_cols = 6  
p
```



```
ggsave("../R_results_controlsgRNAs/correlation/correlation_samples_clustering_woNumbers.png", plot=p, )
ggsave("../R_results_controlsgRNAs/correlation/correlation_samples_clustering_woNumbers.pdf", plot=p, )
```

## 2.4 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/mjahn/R-notebook-crispri-lib/blob/master/sgRNA\\_library\\_V2/data/input/mapping\\_trivial\\_names.tsv](https://github.com/mjahn/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+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_controlsgRNAs/annotated_DESeq_result_table.tsv")
df_reduced_info <- unique(subset(DESeq_result_table, DESeq_result_table$time==8 | DESeq_result_table$time==30))
write_tsv(df_reduced_info, file="../R_results_controlsgRNAs/Reduced.annotated_DESeq_result_table.tsv")

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)
write_tsv(df_red_wide, file="../R_results_controlsgRNAs/Wide_DESeq_result_table.tsv")

```

## 2.5 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 at different gas conditions. This was performed separately for ncRNAs and protein-coding genes.

### 2.5.1 Protein-coding genes

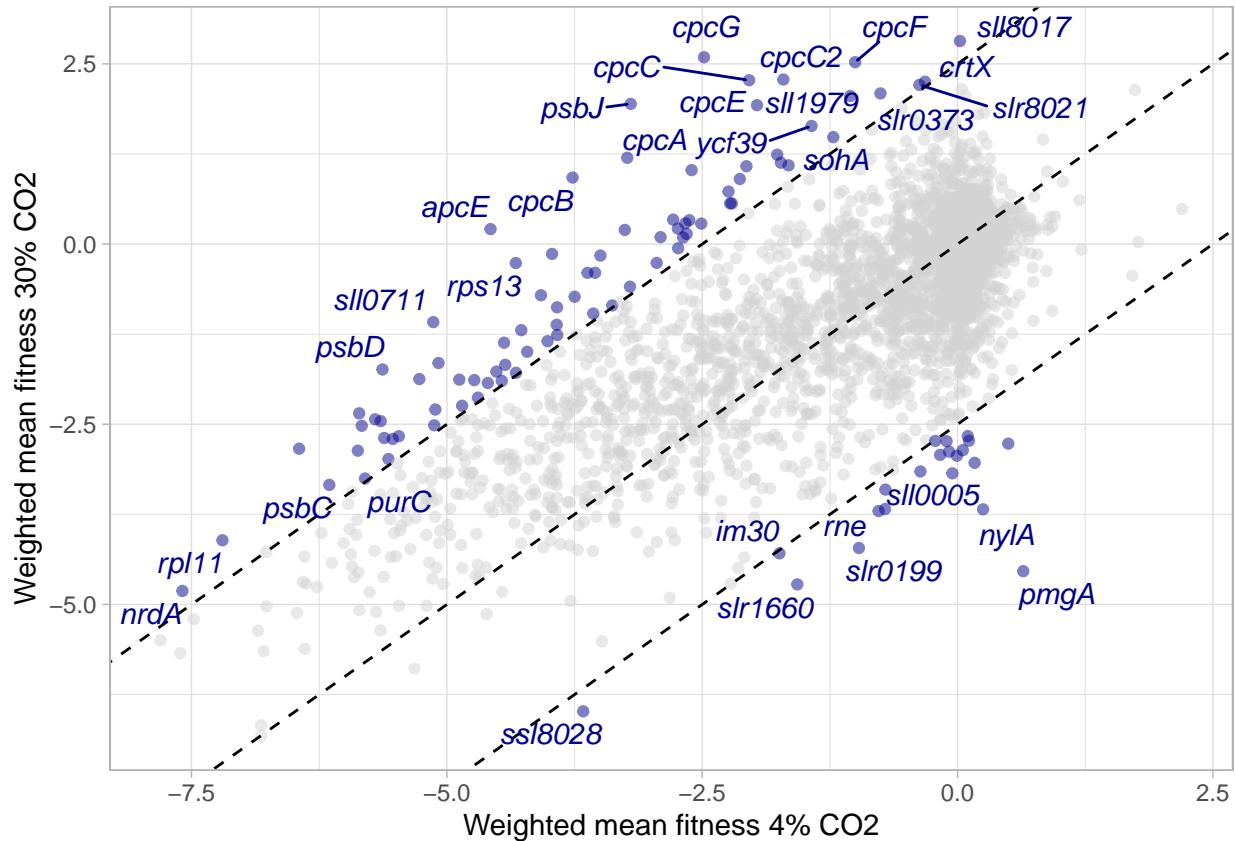
```

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="C02_4percent", x_axis_label="Weighted mean fitness 4% CO2", delabel="NO", filename_save="C02_4percent_fitness_fitness.pdf") {
  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]] < -2.5)] <- "YES"
  # 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", "#d3d3b2")
  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, color=delabel))
  p + theme_light() + labs(y=y_axis_label, x=x_axis_label) + theme(legend.position = "none") + geom_abline()
  ggsave(filename = filename_save, plot=p, width=12, height=12, units="cm")
}

plot_fitness_fitness(df_red_wide, "C02_30percent", y_axis_label="Weighted mean fitness 30% CO2", filename_save="C02_30percent_fitness_fitness.pdf")

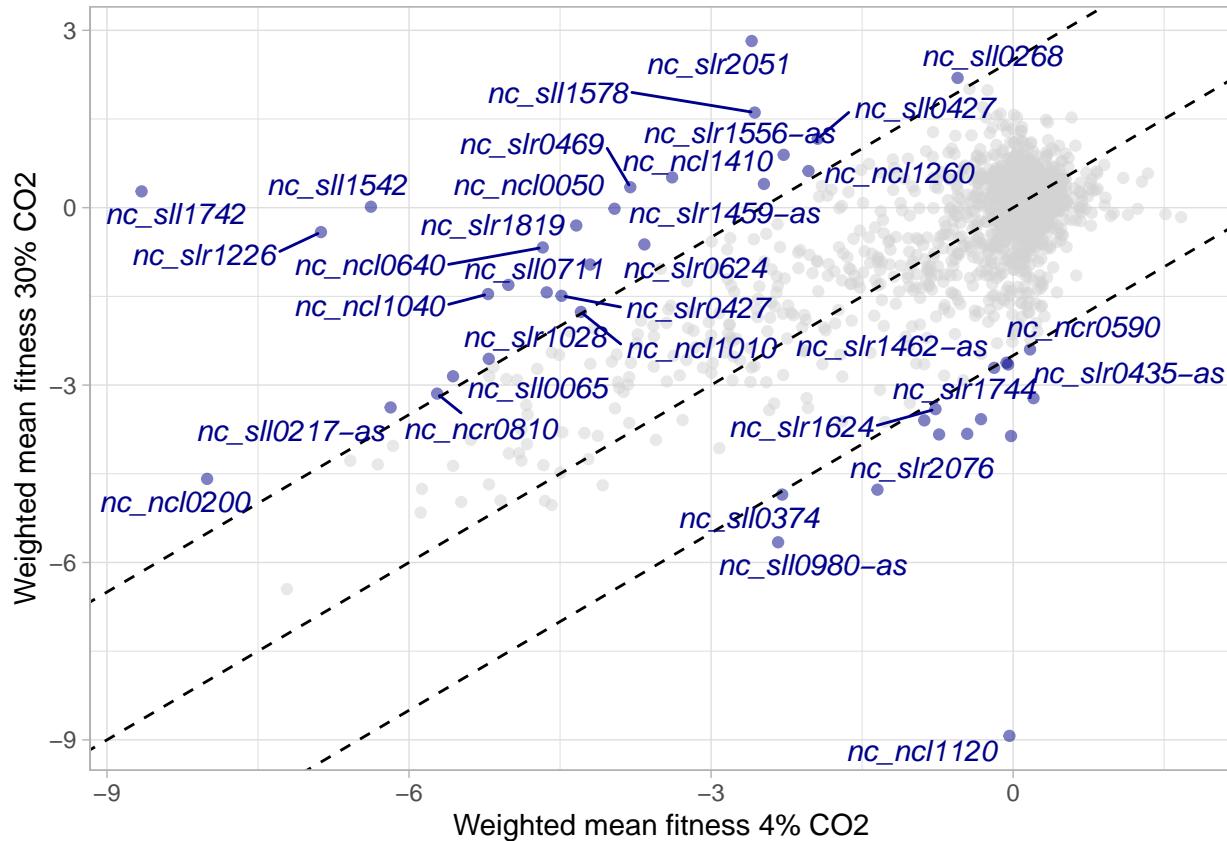
```



### 2.5.2 ncRNAs

These include antisense RNAs, but also other ncRNAs.

```
df_red_wide_ncRNA <- pivot_wider(df_red_ncRNAs, names_from="condition", values_from=c("wmean_fitness"))
plot_fitness_fitness(df_red_wide_ncRNA, "CO2_30percent", y_axis_label="Weighted mean fitness 30% CO2",
```



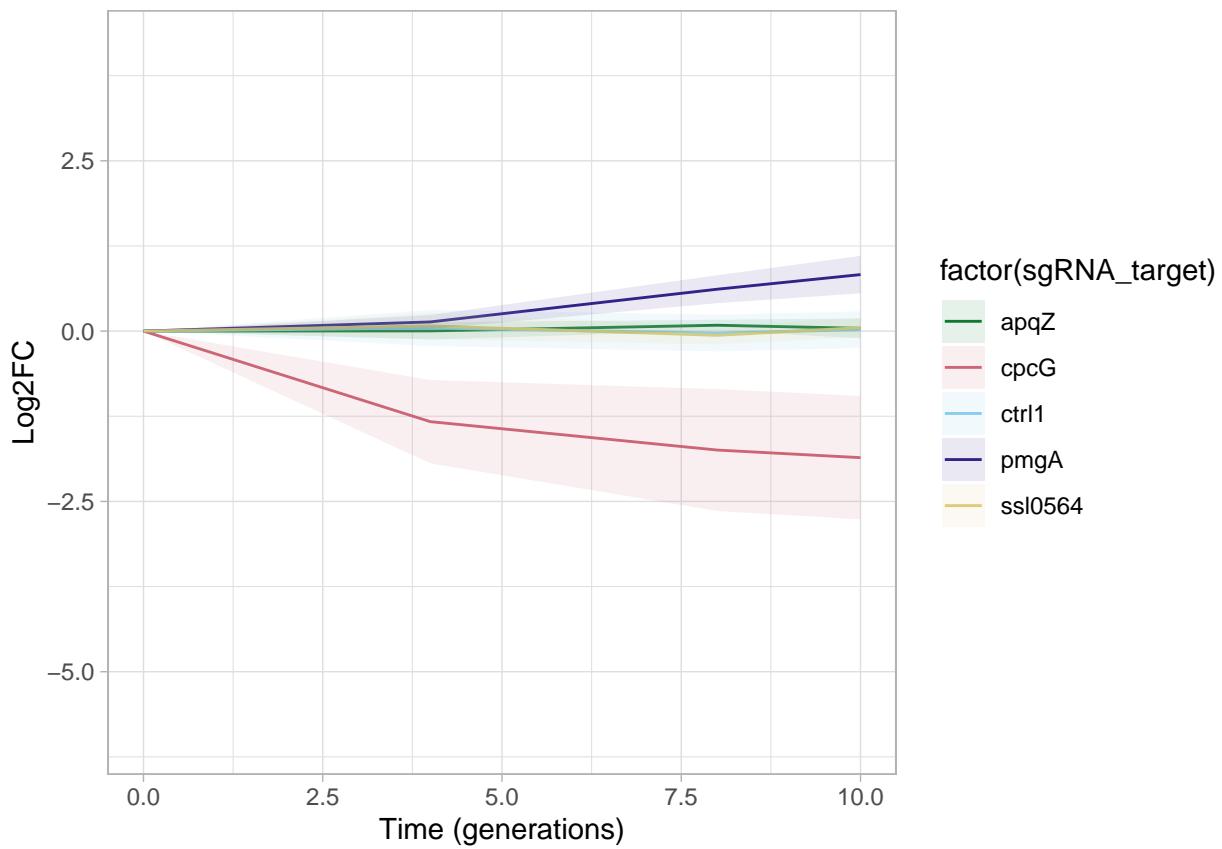
## 2.6 Growth curves of most interesting targets

```

smaller_subset <- unique(subset(DESeq_result_table, DESeq_result_table$sgRNA_target %in% c("ssl0564", "ctrl11", "cpcG", "apqZ")))

colors_vector <- c("apqZ"="#117733ff", "cpcG"="#cc6677ff", "ssl0564"="#ddcc77ff", "ctrl11"="#88cceeff", "ctrl2"="#4477aaee")
p <- lineplot_CVinterval_severalColours_meanlog2(smaller_subset) + ylim(-6,+4.2) + scale_color_manual(values=colors_vector)
p

```

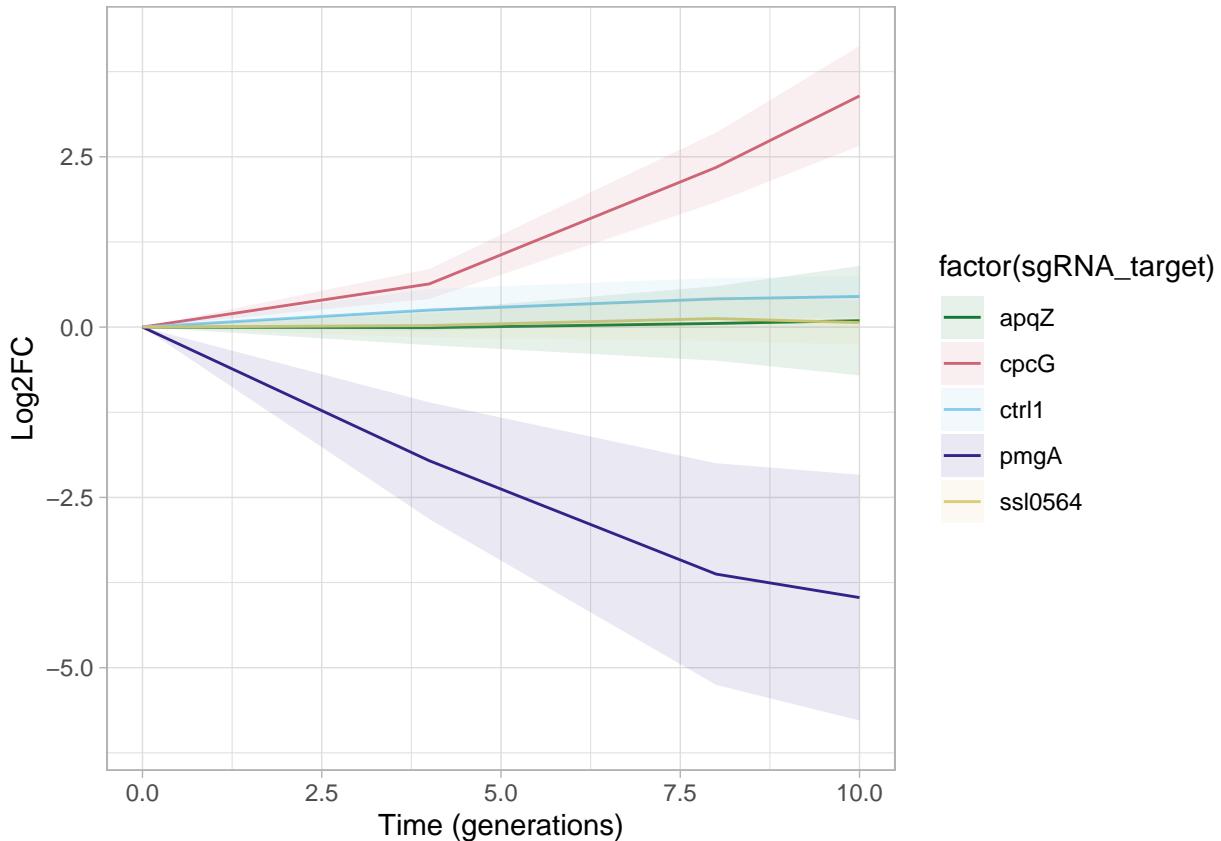


```

ggsave("../R_results_controlsgRNAs/growthCurves/variants_timeLinePlot_4perc.pdf", plot=p)
ggsave("../R_results_controlsgRNAs/growthCurves/variants_timeLinePlot_4perc.png", plot=p)

smaller_subset <- unique(subset(DESeq_result_table, DESeq_result_table$sgRNA_target %in% c("ssl0564", "ctrl1", "pmgA", "apqZ", "cpcG")))
p <- lineplot_CVinterval_severalColours_meanlog2(smaller_subset) + ylim(-6,+4.2) + scale_color_manual(values = c("ctrl1" = "#00FFFF", "pmgA" = "#4B0082", "apqZ" = "#228B22", "cpcG" = "#DC143C", "ssl0564" = "#FFDAB9"))
p

```



```
ggsave("../R_results_controlssRNAs/growthCurves/variants_timeLinePlot_30perc.pdf", plot=p)
ggsave("../R_results_controlssRNAs/growthCurves/variants_timeLinePlot_30perc.png", plot=p)
```

## 2.7 GSEA

Functional enrichment analyses and gene set enrichment analyses help to check if a certain pathway or specific group of genes is especially affected by a treatment. Here, gene set enrichment analyses were performed for either Gene Ontology terms or KEGG pathways. To perform a gene set enrichment analysis, genes are sorted according to some measure, e.g. the log2FC after a certain time or the calculated fitness. Here, we used the weighted fitness of several sgRNAs as measure. The mapping of locus tags to Gene Ontology terms was downloaded from UniProt on the 18th Jan. 2024. There is the possibility to somehow weigh the adjusted p value in these calculation, e.g. by multiplying the weighted mean with the adjusted p value. Here, only the first few rows of each table is given. Full tables with all found terms/pathways are available.

In a first step, GSEAs were calculated for the two different CRISPRi libraries separately. The depletion of essential pathways related to “Ribosomes” or “photosynthesis” is a first good quality measure for a CRISPRi screen.

### 2.7.1 4% CO<sub>2</sub>

```
DESeq_result_table_4percent <- unique(subset(DESeq_result_table, DESeq_result_table$condition=="CO2_4pe")
geneList <- DESeq_result_table_4percent$wmean_fitness
names(geneList) <- DESeq_result_table_4percent$locus
geneList = sort(geneList, decreasing = TRUE)
set.seed(513)
go_gsea_object <- GSEA(geneList, TERM2GENE = term_to_gene, TERM2NAME=term_to_name, seed=TRUE)
print(head(go_gsea_object [,columns_to_show]))
```

```

##                                     Description setSize enrichmentScore      NES
## GO:0003735 structural constituent of ribosome      56      -0.8549876 -1.522873
## GO:0005737                               cytoplasm     304      -0.7005033 -1.277611
## GO:0006412                               translation     63      -0.8285052 -1.476260
## GO:0005524                               ATP binding    302      -0.6816709 -1.243252
## GO:0005829                               cytosol       222      -0.7004435 -1.272849
## GO:0019843                               rRNA binding     36      -0.8465441 -1.491552
##          p.adjust      qvalue
## GO:0003735 6.050000e-09 4.315789e-09
## GO:0005737 6.050000e-09 4.315789e-09
## GO:0006412 6.611612e-09 4.716417e-09
## GO:0005524 6.611612e-09 4.716417e-09
## GO:0005829 2.367647e-08 1.688970e-08
## GO:0019843 4.826859e-06 3.443257e-06

write.csv(go_gsea_object, "../R_results_controlsgRNAs/GSEA_output/GO_GSEA_CRISPRi_4percent.csv")

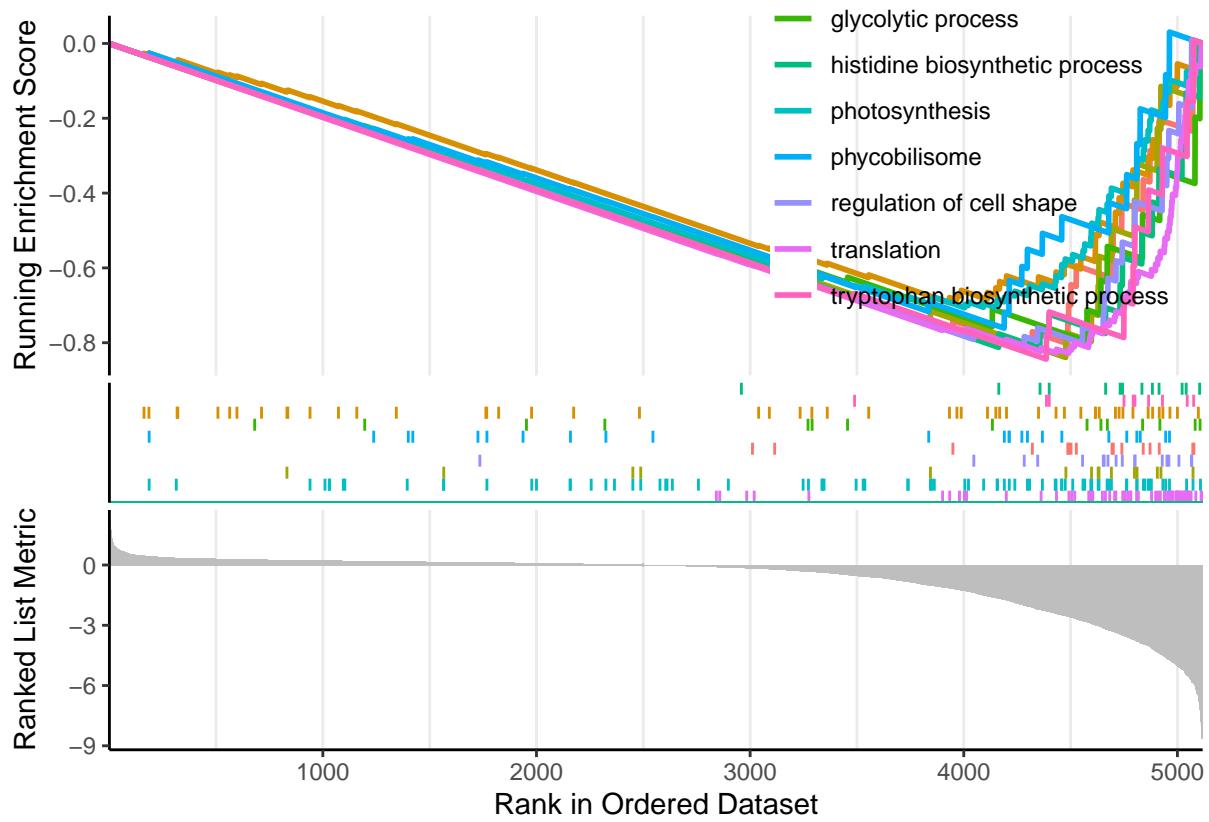
set.seed(914)
kegg_gsea_object <- gseKEGG(geneList, organism="syn", minGSSize=10, pvalueCutoff = 0.05, seed=TRUE)
print(head(kegg_gsea_object)[,columns_to_show_KEGG])

##          ID
## syn03010 syn03010
## syn01110 syn01110
## syn01230 syn01230
## syn01120 syn01120
## syn01240 syn01240
## syn01232 syn01232
##                                     Description
## syn03010           Ribosome - Synechocystis sp. PCC 6803
## syn01110   Biosynthesis of secondary metabolites - Synechocystis sp. PCC 6803
## syn01230   Biosynthesis of amino acids - Synechocystis sp. PCC 6803
## syn01120 Microbial metabolism in diverse environments - Synechocystis sp. PCC 6803
## syn01240           Biosynthesis of cofactors - Synechocystis sp. PCC 6803
## syn01232           Nucleotide metabolism - Synechocystis sp. PCC 6803
##          setSize enrichmentScore      p.adjust
## syn03010      54      -0.8637207 3.350000e-09
## syn01110      298      -0.7476045 3.350000e-09
## syn01230       91      -0.7804156 2.256075e-08
## syn01120      128      -0.7266786 1.115400e-06
## syn01240      139      -0.7000490 2.121112e-05
## syn01232       30      -0.8433933 7.998244e-05

write.csv(kegg_gsea_object, "../R_results_controlsgRNAs/GSEA_output/KEGG_GSEA_CRISPRi_4percent.csv")

p <- gseaplot2(go_gsea_object, geneSetID =c(3, 10, 15, 18, 21, 22, 23, 24,25,26))
p

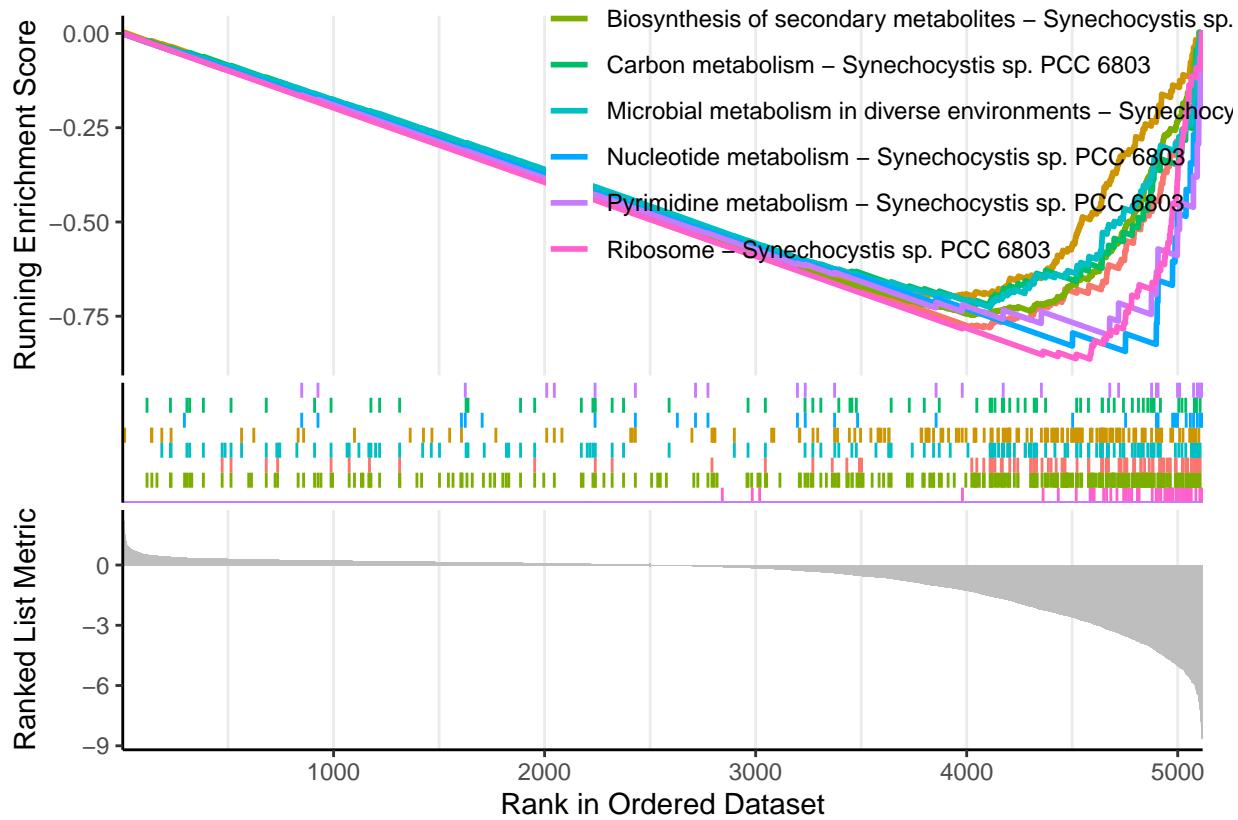
```



```

ggsave("../R_results_controls/gSEA_output/G0_GSEA_4perc_subset.pdf", plot=p, width=20, height=25,
p <- gseaplot2(kegg_gsea_object, geneSetID =1:8)
p

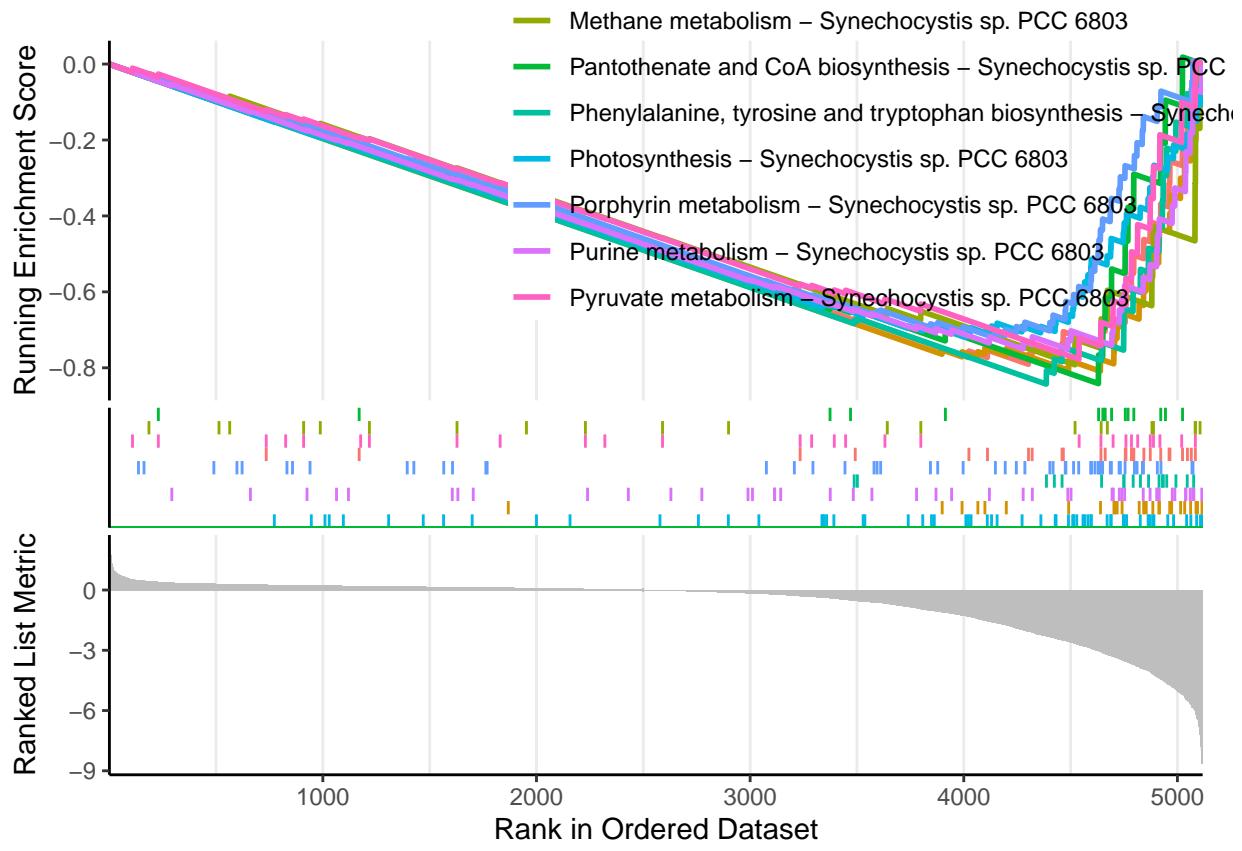
```



```

ggsave("../R_results_controls_gRNAs/GSEA_output/KEGG_GSEA_4perc_part1.pdf", plot=p, width=20, height=25)
p <- gseaplot2(kegg_gsea_object, geneSetID = 9:17)
p

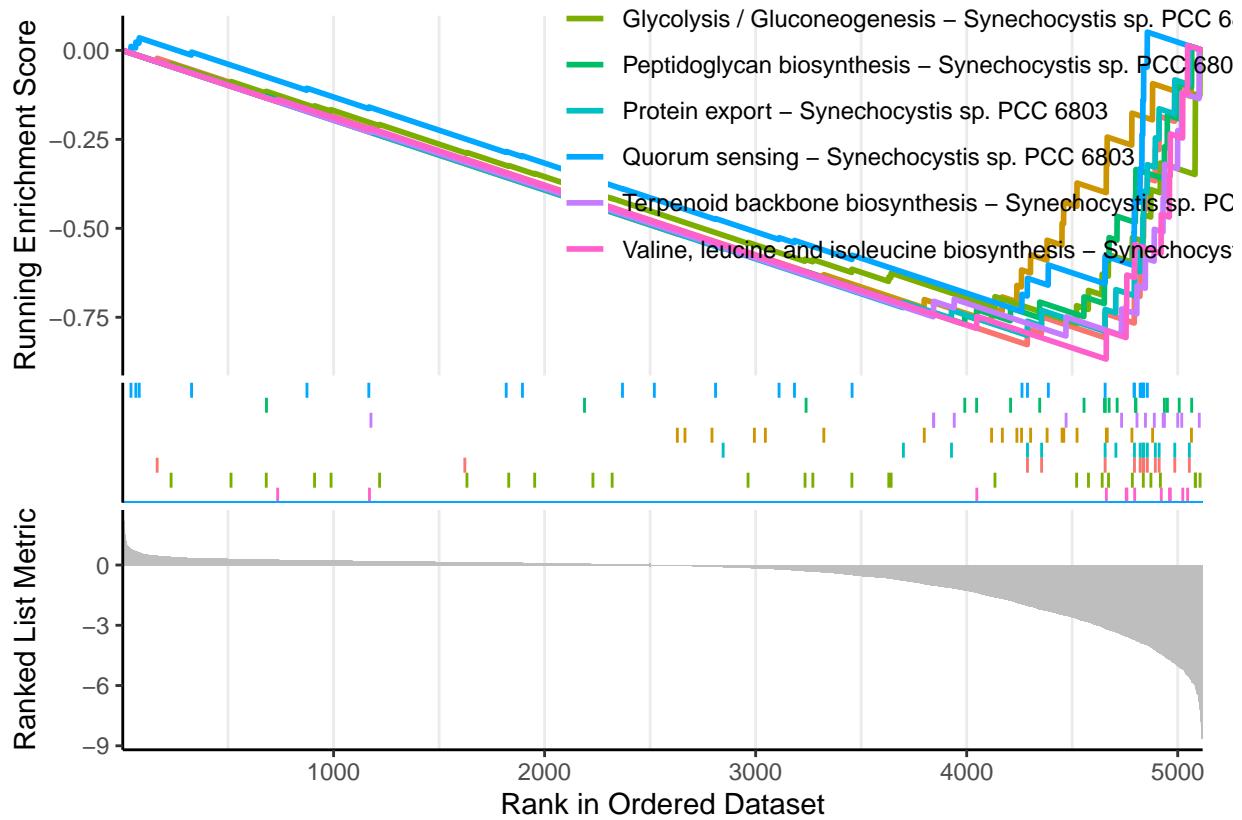
```



```

ggsave("../R_results_controls/gSEA_output/KEGG_GSEA_4perc_part2.pdf", plot=p, width=20, height=25)
p <- gseaplot2(kegg_gsea_object, geneSetID =18:25)
p

```



```
ggsave("../R_results_controlsGSEAs/GSEA_output/KEGG_GSEA_4perc_part3.pdf", plot=p, width=20, height=25)
```

## 2.7.2 30% CO<sub>2</sub>

```
DESeq_result_table_30percent <- unique(subset(DESeq_result_table, DESeq_result_table$condition=="CO2_30"))
geneList <- DESeq_result_table_30percent$wmean_fitness
names(geneList) <- DESeq_result_table_30percent$locus
geneList = sort(geneList, decreasing = TRUE)
set.seed(513)
go_gsea_object <- GSEA(geneList, TERM2GENE = term_to_gene, TERM2NAME=term_to_name, seed=TRUE)
print(head(go_gsea_object)[,columns_to_show])

##                                     Description setSize enrichmentScore      NES
## GO:0005737                      cytoplasm     304    -0.6260446 -1.460051
## GO:0005524                      ATP binding     302    -0.5926620 -1.382089
## GO:0006412                      translation      63    -0.7350610 -1.652183
## GO:0003735 structural constituent of ribosome     56    -0.7444203 -1.670604
## GO:0005829                      cytosol       222    -0.6020811 -1.397232
## GO:0008360 regulation of cell shape       18    -0.8773434 -1.818514
##          p.adjust      qvalue
## GO:0005737 1.210000e-08 8.421053e-09
## GO:0005524 3.184444e-08 2.216229e-08
## GO:0006412 2.633458e-07 1.832768e-07
## GO:0003735 4.617481e-07 3.213558e-07
## GO:0005829 4.617481e-07 3.213558e-07
## GO:0008360 6.683709e-06 4.651559e-06
```

```

write.csv(go_gsea_object, "../R_results_controlsgRNAs/GSEA_output/GO_GSEA_CRISPRi_30percent.csv")

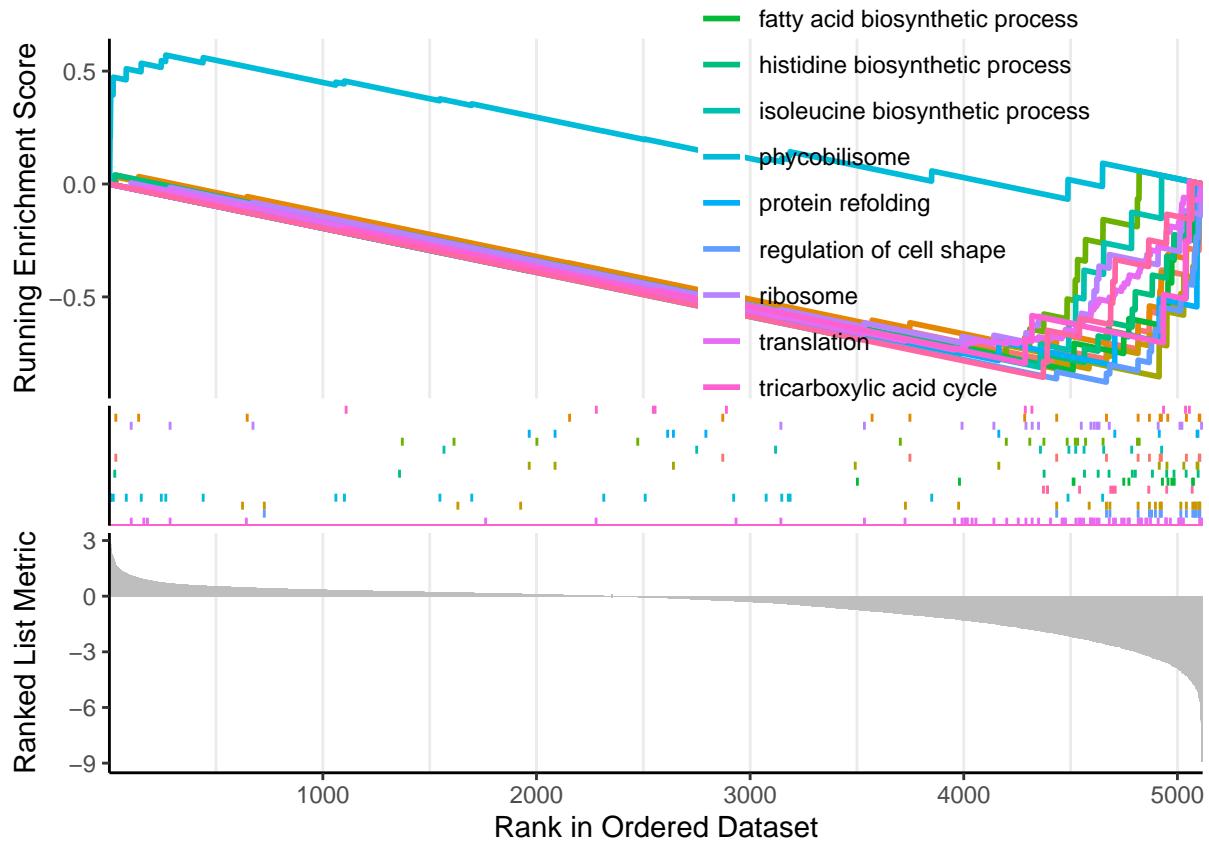
set.seed(914)
kegg_gsea_object <- gseKEGG(geneList, organism="syn", minGSSize=10, pvalueCutoff = 0.05, seed=TRUE)
print(head(kegg_gsea_object)[,columns_to_show_KEGG])

## ID
## syn01240 syn01240
## syn01110 syn01110
## syn01230 syn01230
## syn03010 syn03010
## syn01120 syn01120
## syn01210 syn01210
## Description
## syn01240 Biosynthesis of cofactors - Synechocystis sp. PCC 6803
## syn01110 Biosynthesis of secondary metabolites - Synechocystis sp. PCC 6803
## syn01230 Biosynthesis of amino acids - Synechocystis sp. PCC 6803
## syn03010 Ribosome - Synechocystis sp. PCC 6803
## syn01120 Microbial metabolism in diverse environments - Synechocystis sp. PCC 6803
## syn01210 2-Oxocarboxylic acid metabolism - Synechocystis sp. PCC 6803
## setSize enrichmentScore p.adjust
## syn01240 139 -0.6725422 3.300000e-09
## syn01110 298 -0.6527395 3.300000e-09
## syn01230 91 -0.7109200 4.635172e-09
## syn03010 54 -0.7491109 3.923199e-07
## syn01120 128 -0.6417441 1.933613e-06
## syn01210 25 -0.8243374 3.375637e-06

write.csv(kegg_gsea_object, "../R_results_controlsgRNAs/GSEA_output/KEGG_GSEA_CRISPRi_30percent.csv")

p <- gseaplot2(go_gsea_object, geneSetID =c(3, 6,9,15,16,17,18,19,23,27,28,29,30,32,33))
p

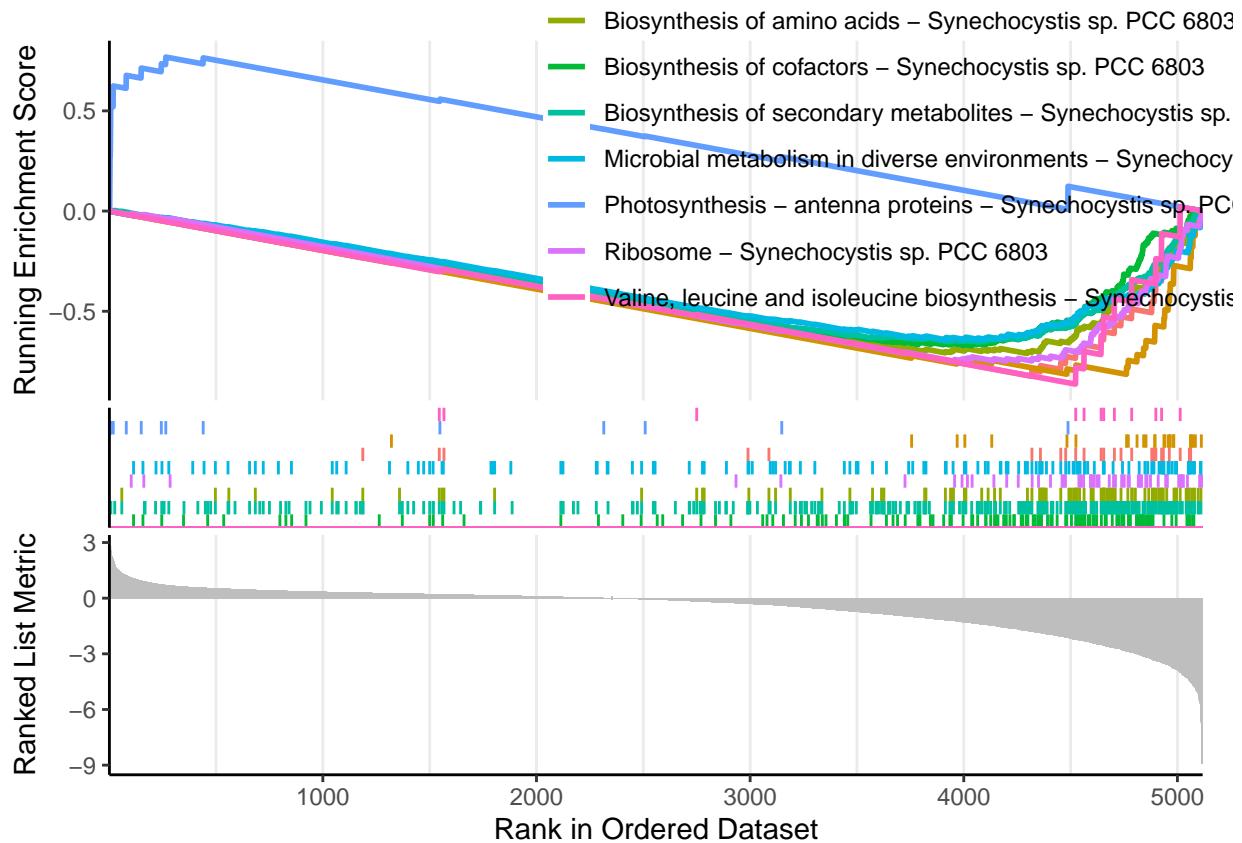
```



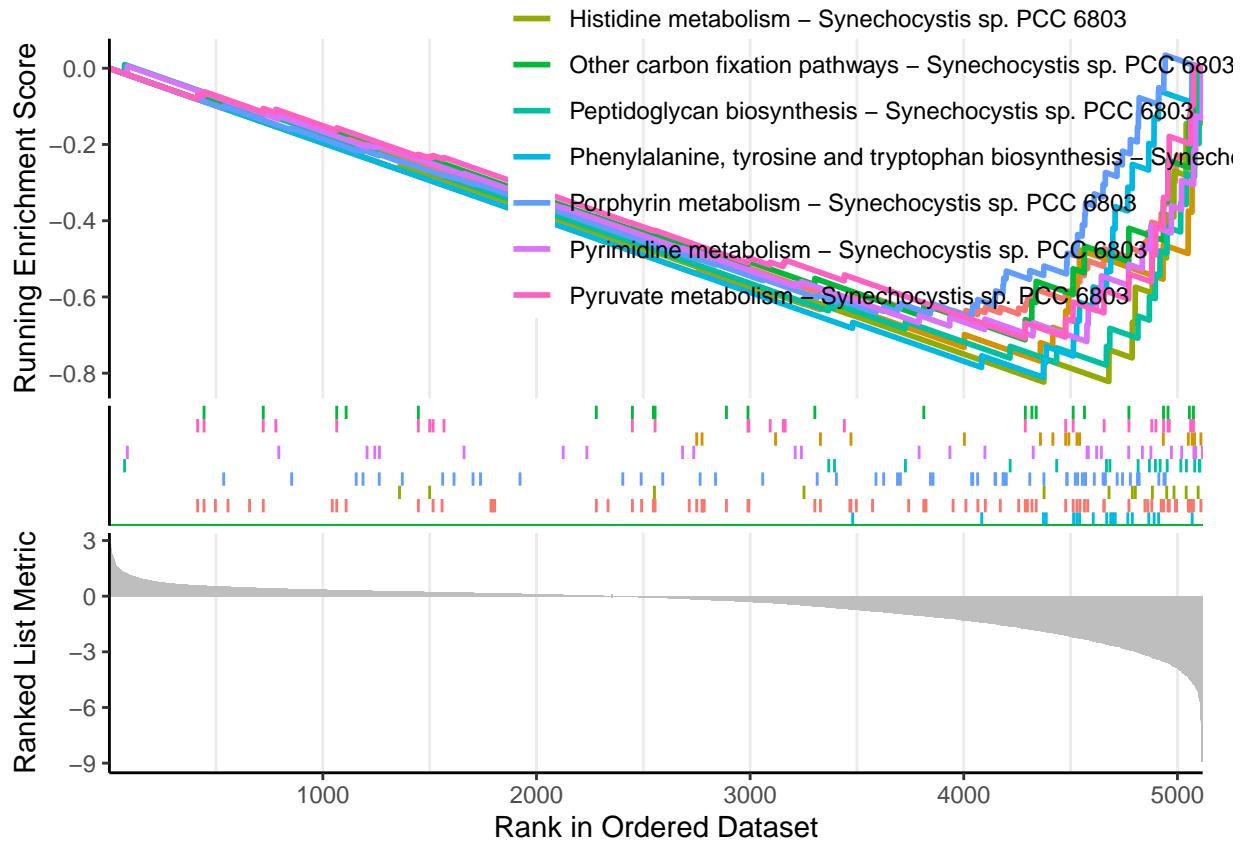
```

ggsave("../R_results_controls/sgRNAs/GSEA_output/GO_GSEA_30perc_subset.pdf", plot=p, width=20, height=25)
p <- gseaplot2(kegg_gsea_object, geneSetID =1:9)
p

```



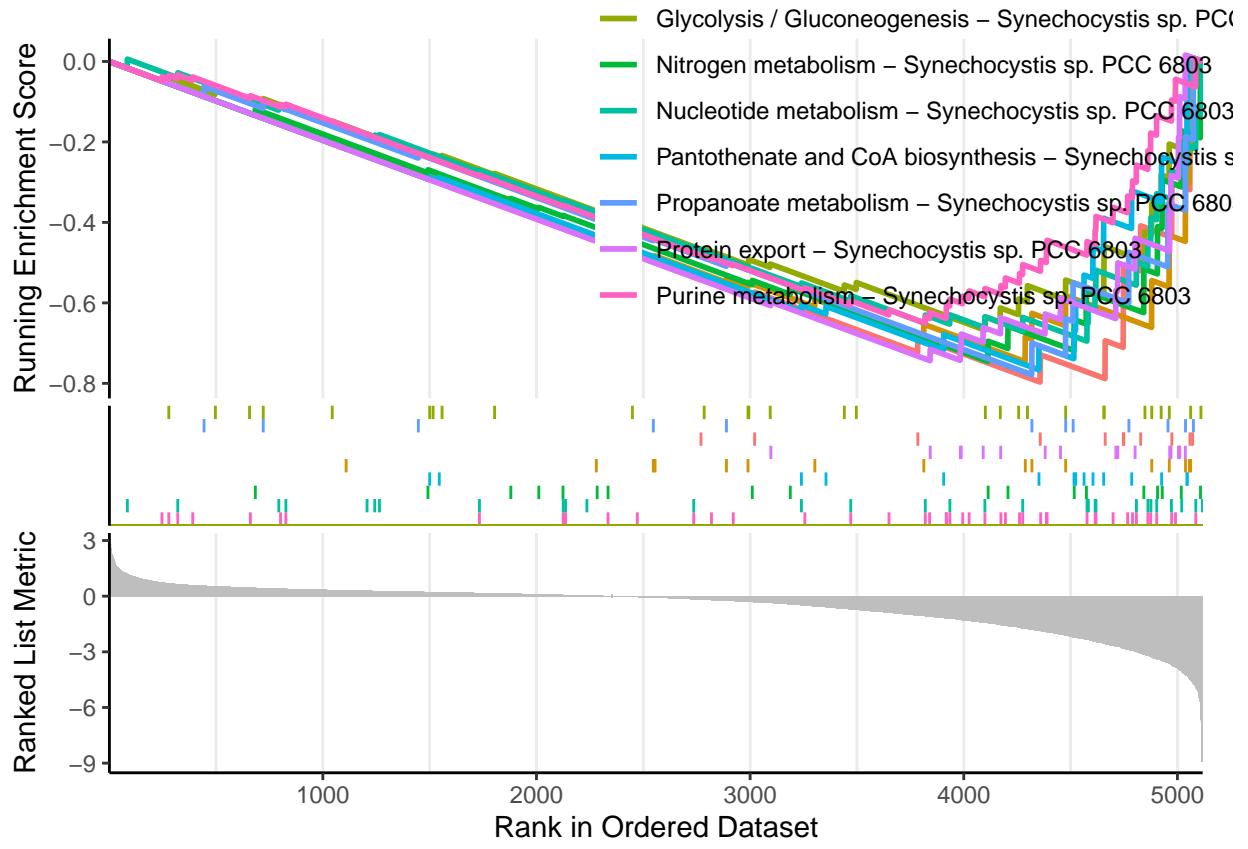
```
ggsave("../R_results_controlsRNAs/GSEA_output/KEGG_GSEA_30perc_part1.pdf", plot=p, width=20, height=20)
p <- gseaplot2(kegg_gsea_object, geneSetID =10:18)
p
```



```

ggsave("../R_results_controls_gRNAs/GSEA_output/KEGG_GSEA_30perc_part2.pdf", plot=p, width=20, height=20)
p <- gseaplot2(kegg_gsea_object, geneSetID = 19:27)
p

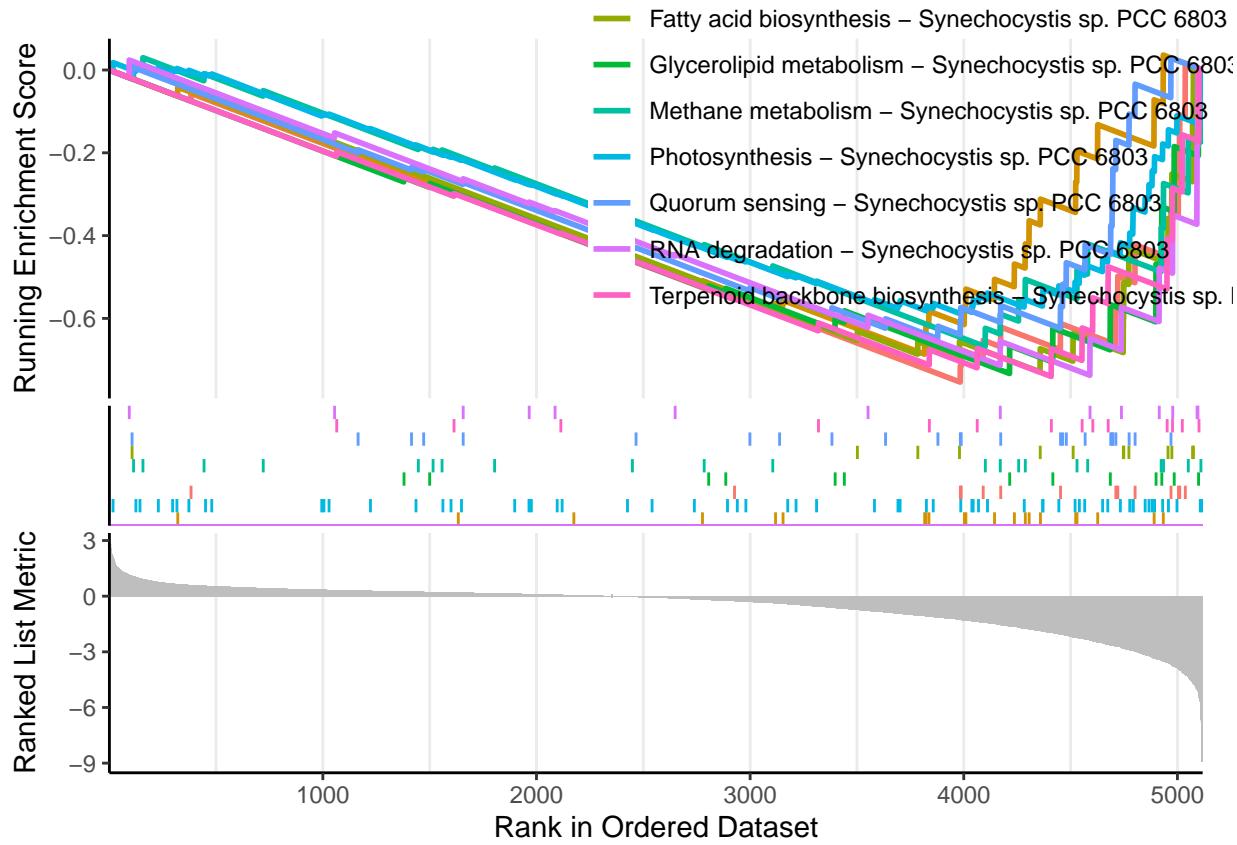
```



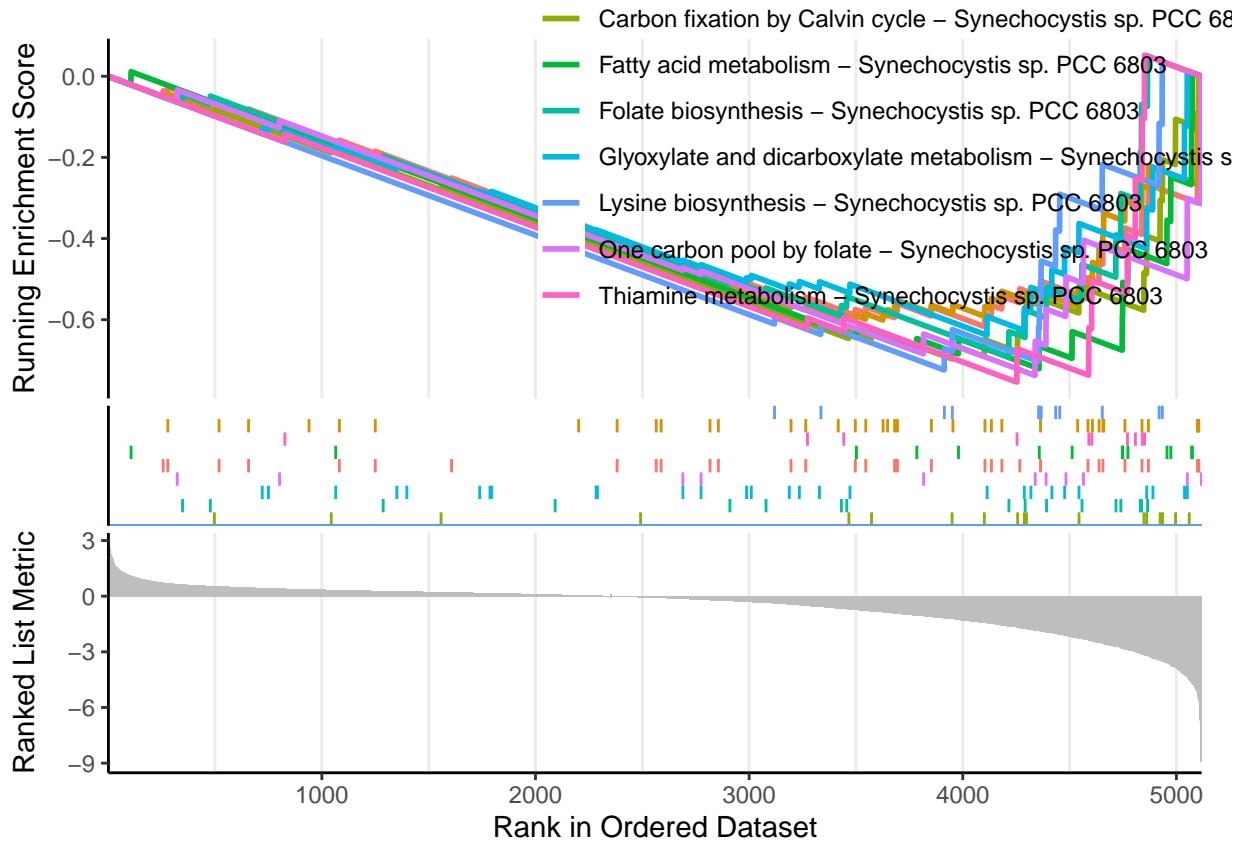
```

ggsave("../R_results_controls/gSEA_output/KEGG_GSEA_30perc_part3.pdf", plot=p, width=20, height=20)
p <- gseaplot2(kegg_gsea_object, geneSetID = 28:36)
p

```



```
ggsave("../R_results_controlsGSEAs/GSEA_output/KEGG_GSEA_30perc_part4.pdf", plot=p, width=20, height=20)
p <- gseaplot2(kegg_gsea_object, geneSetID =37:45)
p
```



```
ggsave("../R_results_controlsgRNAs/GSEA_output/KEGG_GSEA_30perc_part5.pdf", plot=p, width=20, height=20)
```

### 2.7.3 Difference between 4% and 30%

In a next step, we tried to check which GO terms or KEGG pathways show a divergent enrichment or depletion in the two libraries. For this, weighted fitness means belonging to the two conditions were subtracted from each other. These differences were used as input for the GSEA.

30% data was subtracted from 4% data. Hence, if a value is high or a pathway enriched, it should have a tendency of being enriched in 4%. The other way round, depletion means a pathway should be more enriched in the 30% data set. These enrichments might not be completely reliable since normalization between both data sets is not perfect.

```
df_difference <- unique(subset(DESeq_result_table, DESeq_result_table$time==8 & !is.na(DESeq_result_table$locus))
df_difference_wide <- pivot_wider(df_difference, names_from=condition, values_from=wmean_fitness)
df_difference_wide$difference <- df_difference_wide$C02_4percent - df_difference_wide$C02_30percent

df_difference_wide.annotated <- df_difference_wide %>% left_join(annotation_2)
write.csv(df_difference_wide.annotated, "../R_results_controlsgRNAs/fitness_values_differences_annotation.csv")

geneList <- df_difference_wide$difference
names(geneList) <- df_difference_wide$locus
geneList = sort(geneList, decreasing = TRUE)
set.seed(513)
go_gsea_object <- GSEA(geneList, TERM2GENE = term_to_gene, TERM2NAME=term_to_name, seed=TRUE)
print(head(go_gsea_object)[,columns_to_show])
```

	Description	setSize	enrichmentScore
##	translation	63	-0.7837152
## GO:0006412			

```

## GO:0003735 structural constituent of ribosome 56 -0.7914832
## GO:0031676 plasma membrane-derived thylakoid membrane 114 -0.6753590
## GO:0030089 phycobilisome 24 -0.8439869
## GO:0015979 photosynthesis 82 -0.6253269
## GO:0022625 cytosolic large ribosomal subunit 20 -0.8076978
## NES p.adjust qvalue
## GO:0006412 -2.375029 4.000000e-09 3.298246e-09
## GO:0003735 -2.337654 4.000000e-09 3.298246e-09
## GO:0031676 -2.197410 4.000000e-09 3.298246e-09
## GO:0030089 -2.157268 5.800770e-07 4.783091e-07
## GO:0015979 -1.960991 7.620861e-06 6.283868e-06
## GO:0022625 -2.017981 7.519467e-05 6.200262e-05

write.csv(go_gsea_object, "../R_results_controlsgRNAs/GSEA_output/GO_GSEA_difference.csv")

set.seed(914)
kegg_gsea_object <- gseKEGG(geneList, organism="syn", minGSSize=10, pvalueCutoff = 0.05, seed=TRUE)
print(head(kegg_gsea_object [,columns_to_show_KEGG]))

## ID
## syn03010 syn03010
## syn01110 syn01110
## syn01230 syn01230
## syn00196 syn00196
## syn00195 syn00195
## syn00260 syn00260

## Description
## syn03010 Ribosome - Synechocystis sp. PCC 6803
## syn01110 Biosynthesis of secondary metabolites - Synechocystis sp. PCC 6803
## syn01230 Biosynthesis of amino acids - Synechocystis sp. PCC 6803
## syn00196 Photosynthesis - antenna proteins - Synechocystis sp. PCC 6803
## syn00195 Photosynthesis - Synechocystis sp. PCC 6803
## syn00260 Glycine, serine and threonine metabolism - Synechocystis sp. PCC 6803

## setSize enrichmentScore p.adjust
## syn03010 54 -0.8035296 3.350000e-09
## syn01110 298 -0.5440240 3.350000e-09
## syn01230 91 -0.6555711 1.464460e-08
## syn00196 15 -0.9231287 5.891602e-08
## syn00195 63 -0.6399369 2.577030e-05
## syn00260 17 -0.7786594 2.457856e-03

write.csv(kegg_gsea_object, "../R_results_controlsgRNAs/GSEA_output/KEGG_GSEA_difference.csv")

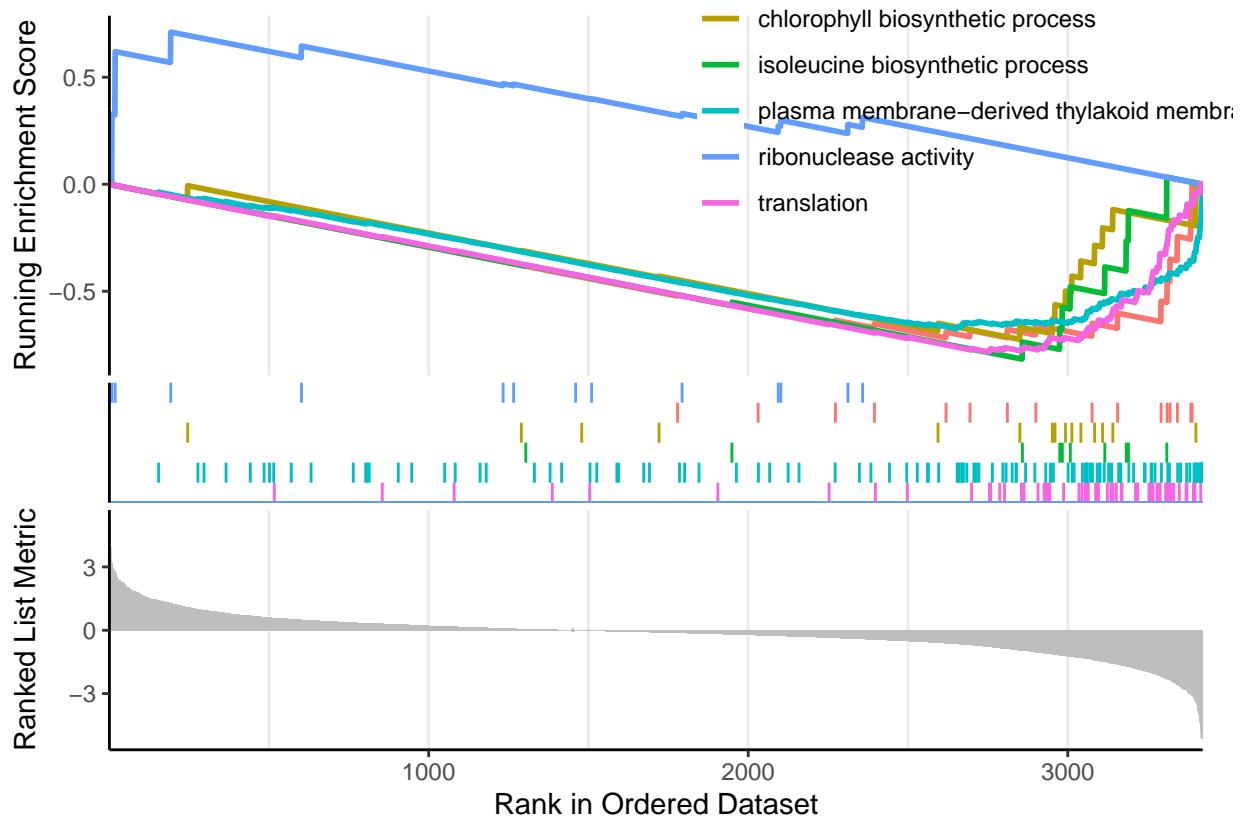
```

Results for the comparison of the two libraries were visualized. These plots are separated in three panels. The lowest panel (“Ranked List Metric”) shows the metric according to which the genes were sorted. In this case, this was the weighted fitness mean associated with the different genes. The upper panel shows the running enrichment score of terms/pathways which were enriched/depleted in a statistically significant manner. The middle panel shows where the genes associated with these terms are located within the ranked list of genes in the same color code as used in the upper panel.

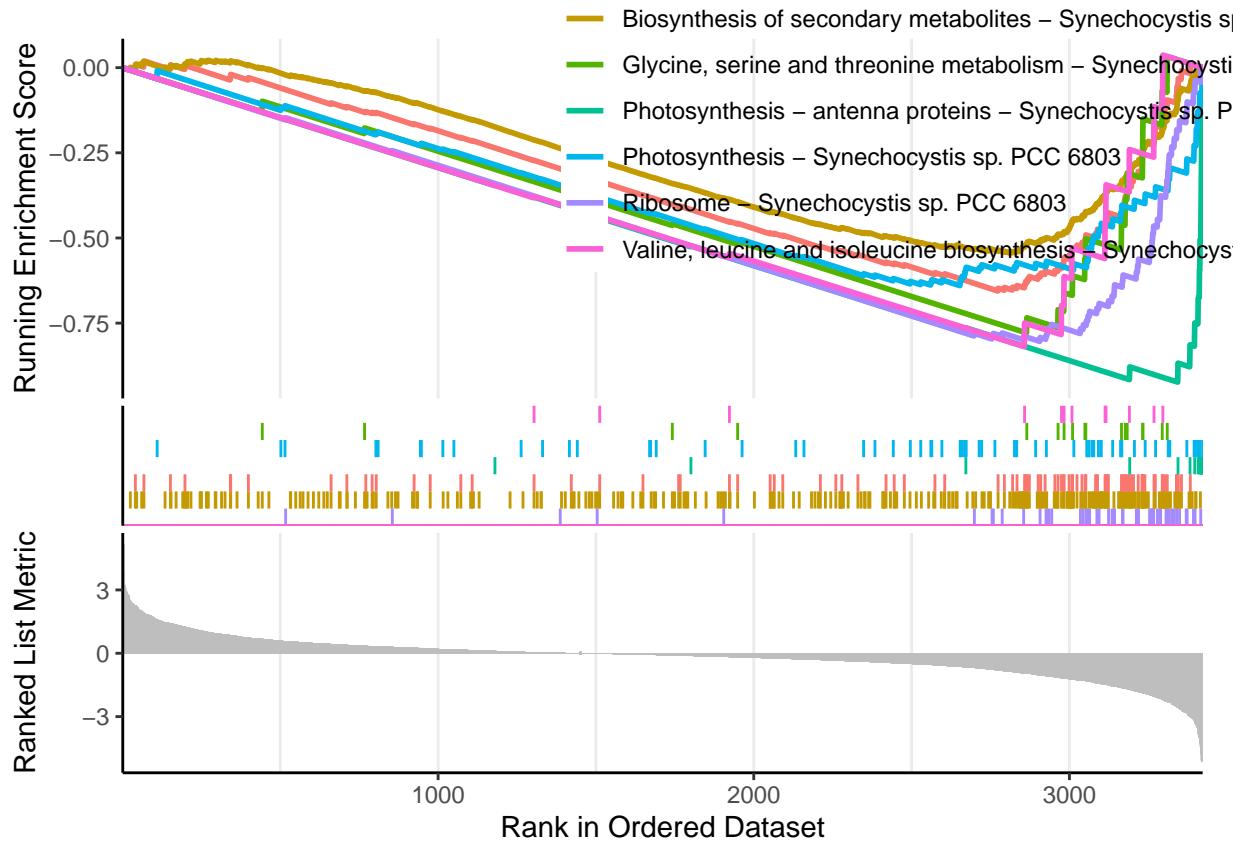
```

p <- gseaplot2(go_gsea_object, geneSetID = c(1,3,12,17,18,19))
p

```



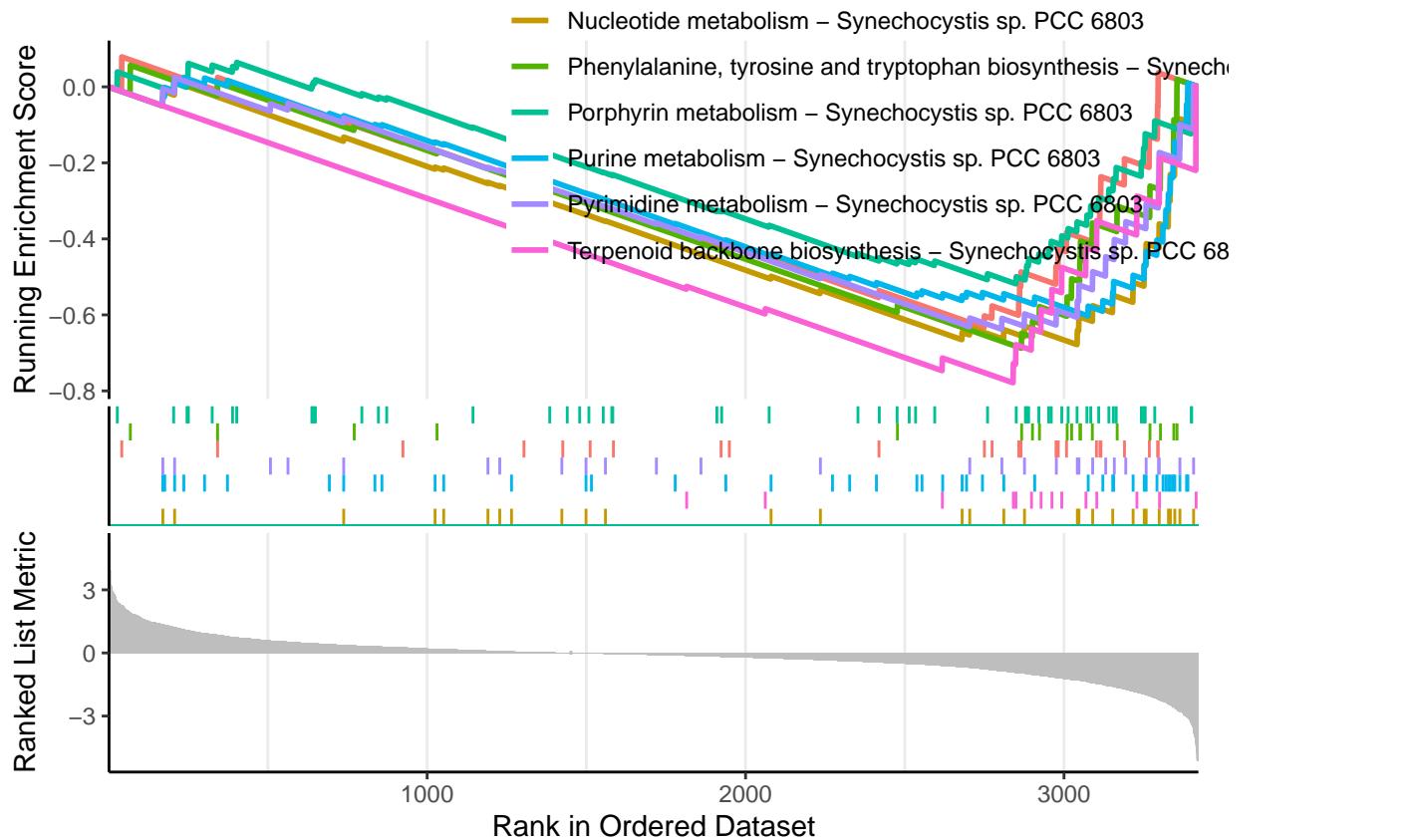
```
ggsave("../R_results_controls/sgRNAs/GSEA_output/GO_GSEA_differences_subset.pdf", plot=p, width=20, height=10)
p <- gseaplot2(kegg_gsea_object, geneSetID =1:7)
p
```



```

ggsave("../R_results_controls_gRNAs/GSEA_output/KEGG_GSEA_differences_part1.pdf", plot=p, width=20, height=10)
p <- gseaplot2(kegg_gsea_object, geneSetID =8:14)
p

```



```
ggsave("../R_results_controlsRNAs/GSEA_output/KEGG_GSEA_differences_part2.pdf", plot=p, width=20, height=10)
```

## 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-p0.3.20.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] pheatmap_1.0.12      enrichplot_1.24.0    clusterProfiler_4.12.0
## [4] magrittr_2.0.3       lubridate_1.9.3     forcats_1.0.0
## [7] stringr_1.5.1       dplyr_1.1.4        purrr_1.0.2
## [10] readr_2.1.5         tidyverse_2.0.0    tibble_3.2.1
## [13] tidyverse_2.0.0    ggpibr_0.6.0      ggrepel_0.9.5
## [16] ggplot2_3.5.1      knitr_1.47        clusterProfiler_4.12.0
##
## loaded via a namespace (and not attached):
## [1] RColorBrewer_1.1-3   rstudioapi_0.16.0   jsonlite_1.8.8
## [4] farver_2.1.2        rmarkdown_2.27     ragg_1.3.2
## [7] fs_1.6.4            zlibbioc_1.50.0   vctrs_0.6.5
## [10] memoise_2.0.1      ggtree_3.12.0     tinytex_0.51
## [13] rstatix_0.7.2      htmltools_0.5.8.1 broom_1.0.6
## [16] gridGraphics_0.5-1 plyr_1.8.9       cachem_1.1.0
## [19] igraph_2.0.3       lifecycle_1.0.4   pkgconfig_2.0.3
## [22] Matrix_1.6-5      R6_2.5.1        fastmap_1.2.0
## [25] gson_0.1.0         GenomeInfoDbData_1.2.12 digest_0.6.35
## [28] aplot_0.2.2        colorspace_2.1-1   patchwork_1.2.0
## [31] AnnotationDbi_1.66.0 S4Vectors_0.42.0   textshaping_0.4.0
## [34] RSQLite_2.3.7       labeling_0.4.3    fansi_1.0.6
## [37] timechange_0.3.0   httr_1.4.7       polyclip_1.10-6
## [40] abind_1.4-5        compiler_4.4.1   bit64_4.0.5
## [43] withr_3.0.0        backports_1.5.0   BiocParallel_1.38.0
## [46] carData_3.0-5     viridis_0.6.5    DBI_1.2.2
## [49] highr_0.11         ggforce_0.4.2    ggsignif_0.6.4
## [52] MASS_7.3-61        HDO.db_0.99.1   tools_4.4.1
## [55] ape_5.8            scatterpie_0.2.2 glue_1.7.0
## [58] nlme_3.1-165       GOSemSim_2.30.0   grid_4.4.1
## [61] shadowtext_0.1.3   reshape2_1.4.4   fgsea_1.30.0
## [64] generics_0.1.3     gtable_0.3.5    tzdb_0.4.0
## [67] data.table_1.15.4 hms_1.1.3       tidygraph_1.3.1
## [70] car_3.1-2          utf8_1.2.4     XVector_0.44.0
## [73] BiocGenerics_0.50.0 pillar_1.9.0    yulab.utils_0.1.4
## [76] vroom_1.6.5         splines_4.4.1   tweenr_2.0.3
## [79] treeio_1.28.0      lattice_0.22-5  bit_4.0.5
## [82] tidyselect_1.2.1   GO.db_3.19.1   Biostrings_2.72.0
## [85] gridExtra_2.3       IRanges_2.38.0  stats4_4.4.1
## [88] xfun_0.44           graphlayouts_1.1.1 Biobase_2.64.0
## [91] stringi_1.8.4      UCSC.utils_1.0.0 lazyeval_0.2.2
## [94] gggfun_0.1.5        yaml_2.3.8     evaluate_0.23
## [97] codetools_0.2-19   ggraph_2.2.1   qvalue_2.36.0
## [100] ggplotify_0.1.2   cli_3.6.2     systemfonts_1.1.0
## [103] munsell_0.5.1     Rcpp_1.0.12    GenomeInfoDb_1.40.1
## [106] png_0.1-8          parallel_4.4.1 blob_1.2.4
## [109] DOSE_3.30.1        viridisLite_0.4.2 tidytree_0.4.6
## [112] scales_1.3.0       crayon_1.5.2   rlang_1.1.3
## [115] cowplot_1.1.3.9000 fastmatch_1.1-4 KEGGREST_1.44.0

```