

# Yeast CAGE analysis

Gabe Zentner

```
library(TSRExploreR)
library(filesstrings)
library(tidyverse)
library(BSgenome.Scerevisiae.UCSC.sacCer3)
library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)
```

## Data preparation

```
# Generate sample sheet
samples <- data.frame(sample_name = before_last_dot(list.files("ctss")),
                      file_1 = list.files("ctss", full.names = TRUE),
                      file_2 = NA,
                      condition = before_last_dot(before_last_dot(list.files("ctss"))))

# Create tsrexplorer object
exp <- tsr_explorer(sample_sheet = samples,
                      genome_assembly = BSgenome.Scerevisiae.UCSC.sacCer3,
                      genome_annotation = TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)

# Import CTSS
exp <- tss_import(exp, sample_sheet = samples)

# Format counts
exp <- format_counts(exp, data_type = "tss")
```

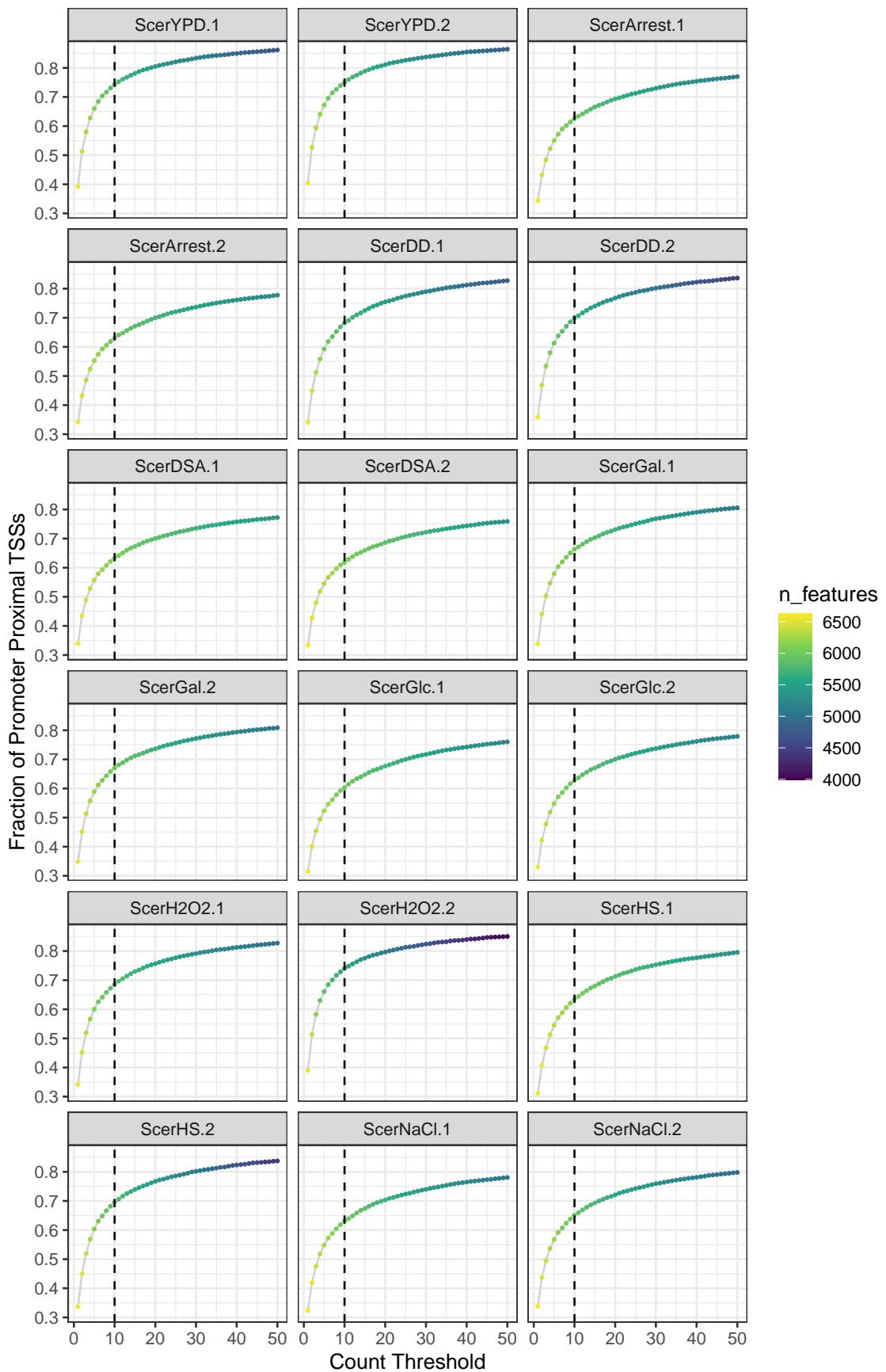
## Threshold exploration

```
# Annotate TSSs
exp <- annotate_features(exp, data_type = "tss", feature_type = "transcript",
                           upstream = 250, downstream = 100)

# Set order for all samples to be plotted
samples_ordered <- c("ScerYPD.1", "ScerYPD.2", "ScerArrest.1", "ScerArrest.2",
                      "ScerDD.1", "ScerDD.2", "ScerDSA.1", "ScerDSA.2",
                      "ScerGal.1", "ScerGal.2", "ScerGlc.1", "ScerGlc.2",
                      "ScerH202.1", "ScerH202.2", "ScerHS.1", "ScerHS.2",
                      "ScerNaCl.1", "ScerNaCl.2")

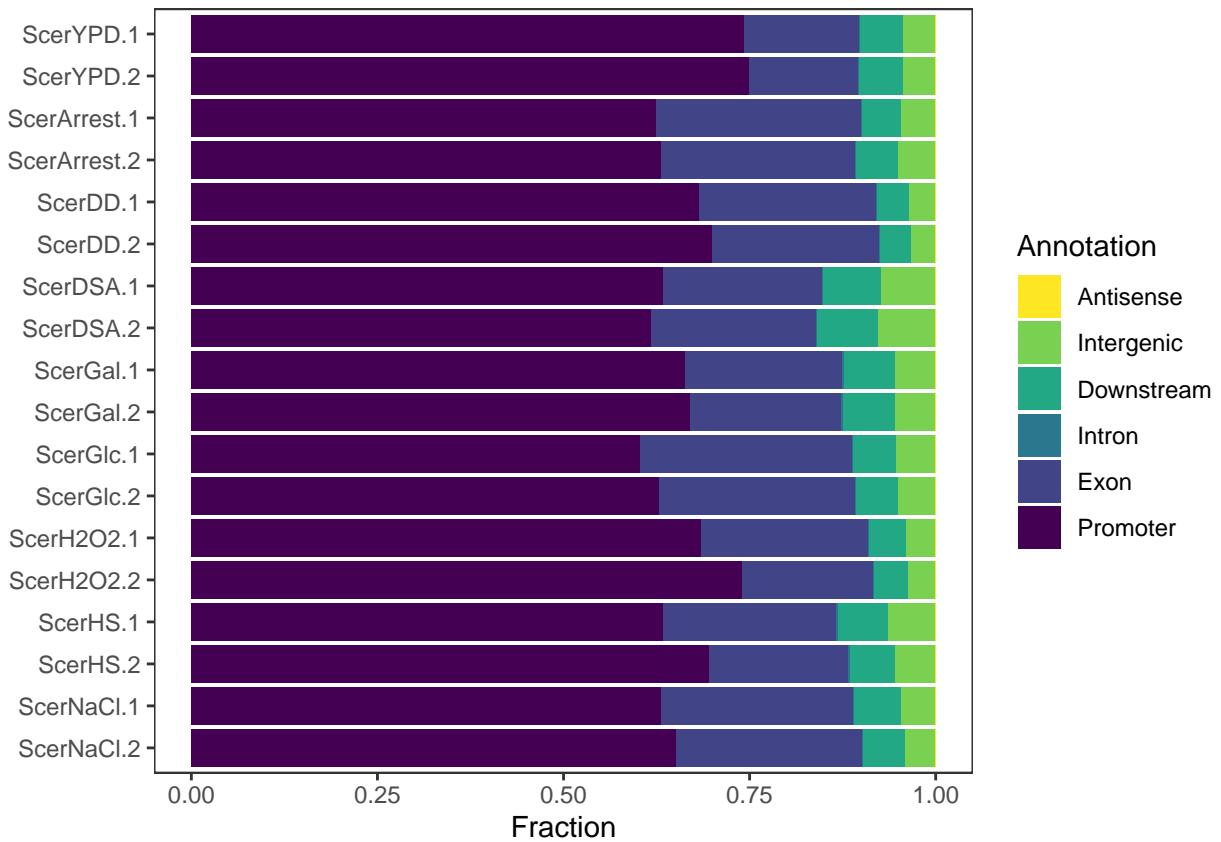
# Explore thresholds (Fig. 1B, S1)
plot_threshold_exploration(exp, samples = samples_ordered, max_threshold = 50,
                            steps = 1, ncol = 3, point_size = 0.5) +
  scale_color_viridis_c() +
  theme(text = element_text(color = "black")) +
```

```
geom_vline(xintercept = 10, linetype = 2)
```

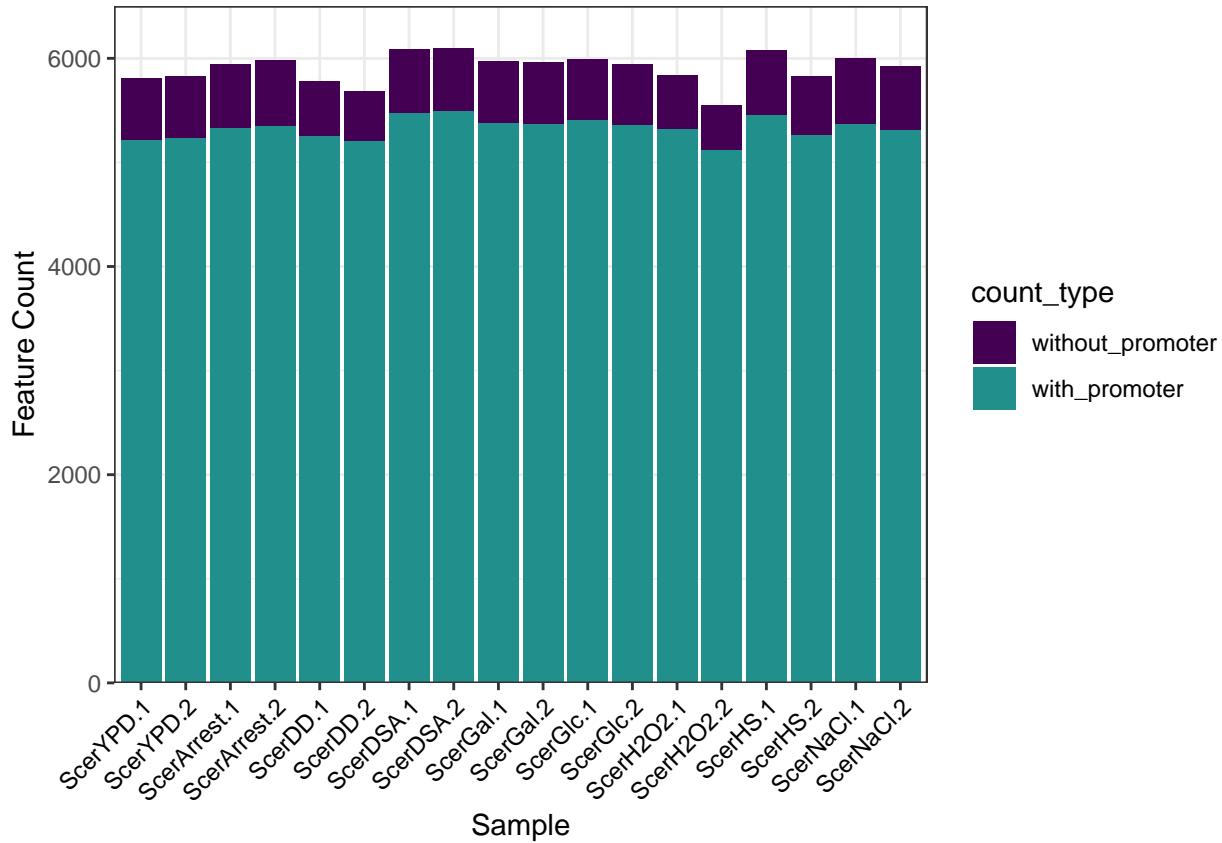


## TSS distribution relative to known features

```
# Genomic distribution (Fig. 1C)
plot_genomic_distribution(exp, data_type = "tss", threshold = 10, samples = samples_ordered) +
  scale_fill_viridis_d(direction = -1, name = "Annotation") +
  theme(text = element_text(color = "black"))
```



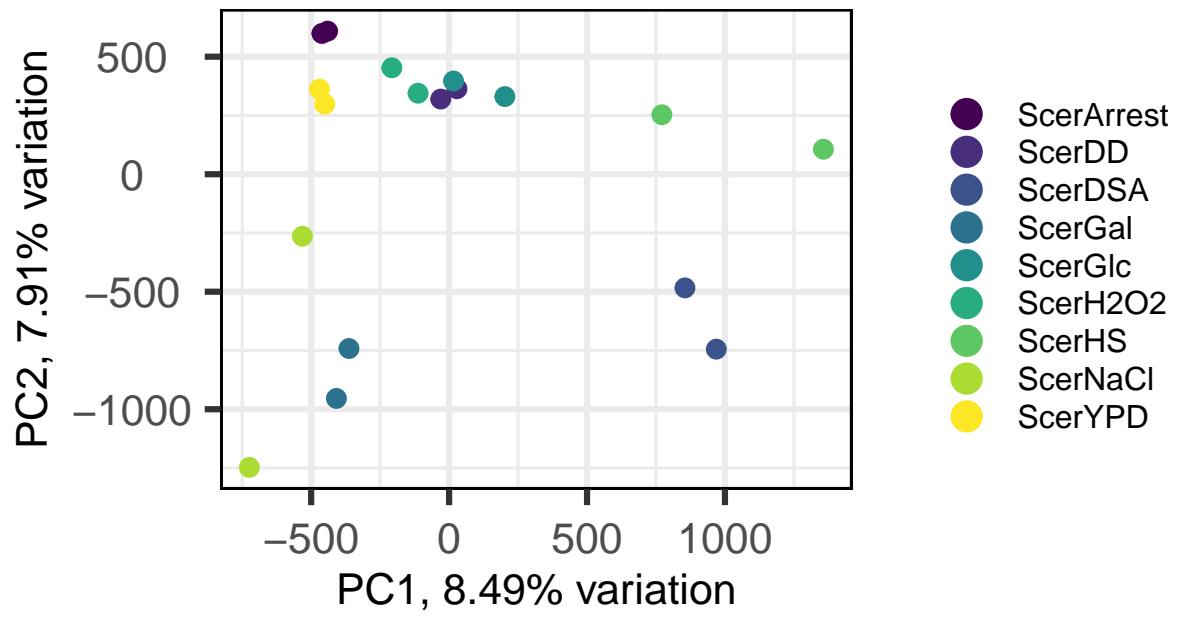
```
# Detected features (Fig. 1D)
plot_detected_features(exp, data_type = "tss", threshold = 10, samples = samples_ordered) +
  theme_bw() +
  scale_fill_viridis_d(end = 0.5) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 6500)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, color = "black"))
```



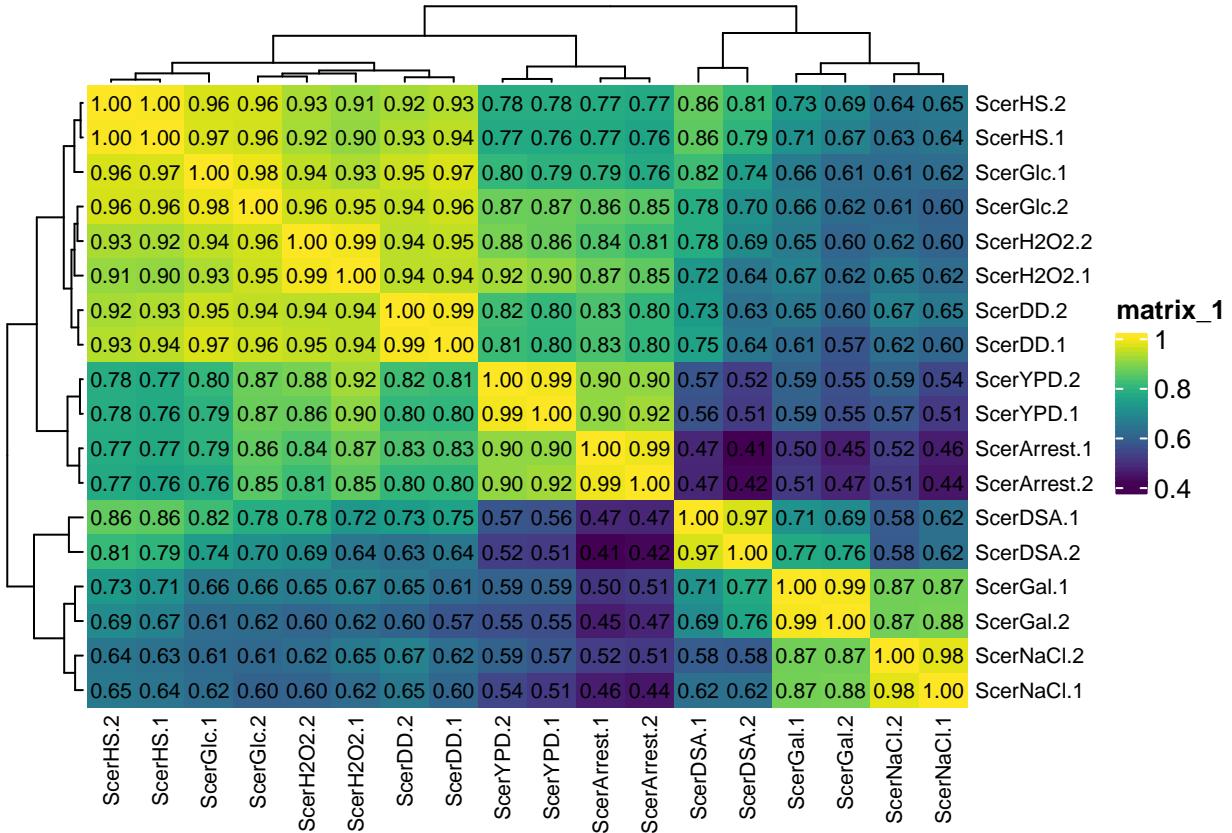
## Correlation

```
# Normalize counts
exp <- normalize_counts(exp, data_type = "tss", method = "deseq2")

# PCA
plot_reduction(exp, data_type = "tss", legendPosition = "right",
               colby = "condition", labSize = 0, drawConnectors = FALSE,
               colkey = c(ScerArrest = "#440154FF",
                         ScerDD = "#472D7BFF",
                         ScerDSA = "#3B528BFF",
                         ScerGal = "#2C728EFF",
                         ScerGlc = "#21908cff",
                         ScerH2O2 = "#27AD81FF",
                         ScerHS = "#5DC863FF",
                         ScerNaCl = "#AADC32FF",
                         ScerYPD = "#FDE725FF"))
```



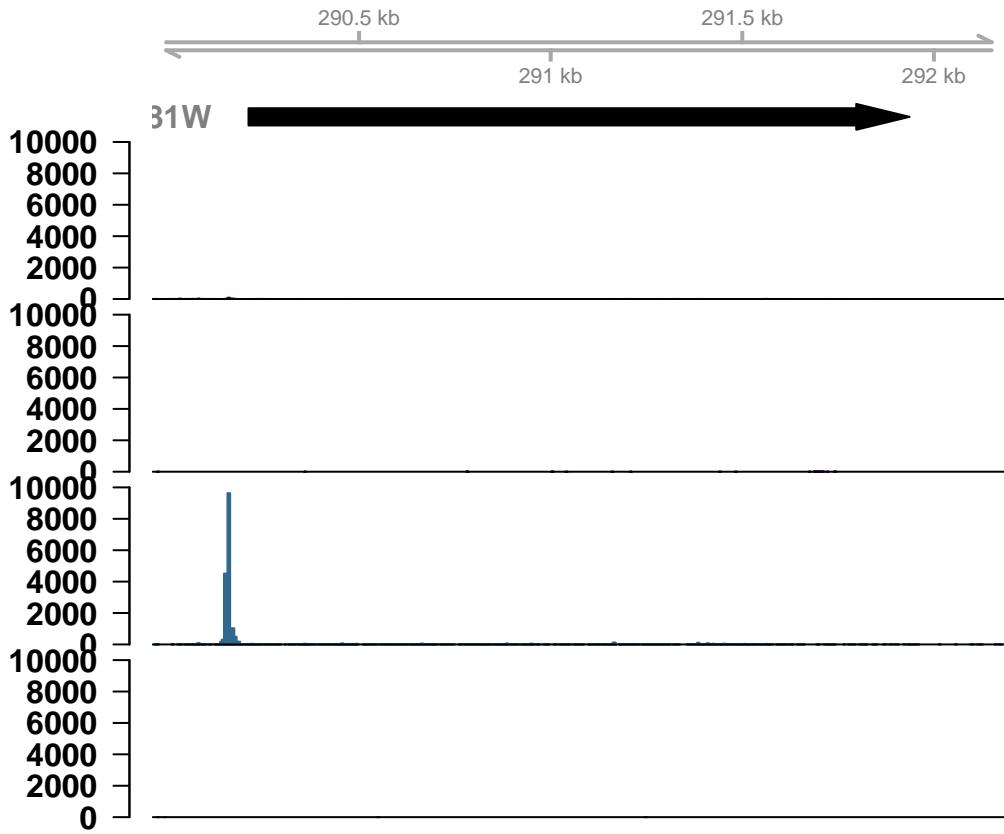
```
# Correlation (Fig. 1E)
plot_correlation(
  exp, data_type = "tss",
  font_size = 8,
  use_normalized = TRUE,
  cluster_samples = TRUE,
  correlation_metric = "pearson",
  heatmap_colors = viridis::viridis(100)
)
```



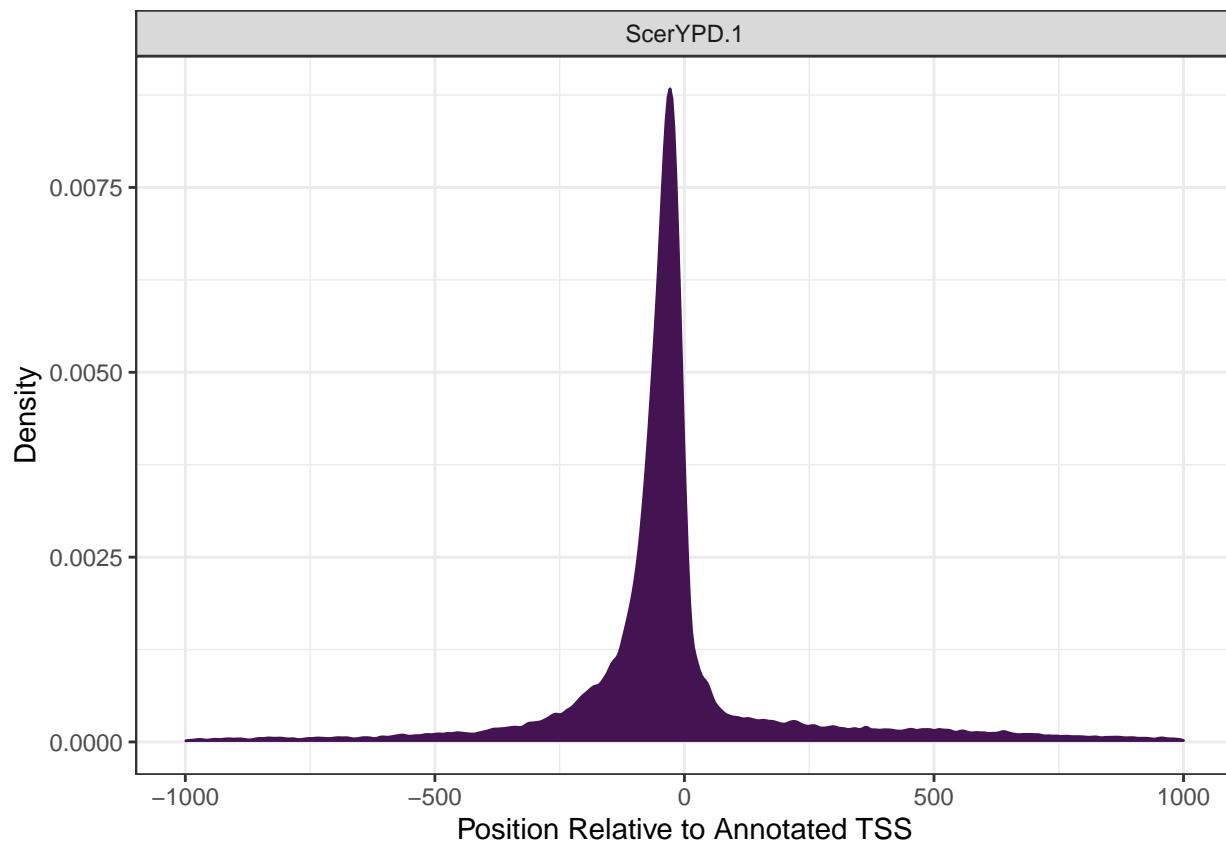
## Visualization

```
# Tracks (Fig. 1F)
gene_tracks(exp, feature_name = "YLR081W", promoter_only = FALSE,
            samples = c(TSS = "ScerYPD.1", TSS = "ScerGal.1"),
            ymax = 10000, tss_colors = viridis::viridis(4),
            use_normalized = TRUE, axis_scale = 1, anno_pos = "top")
```

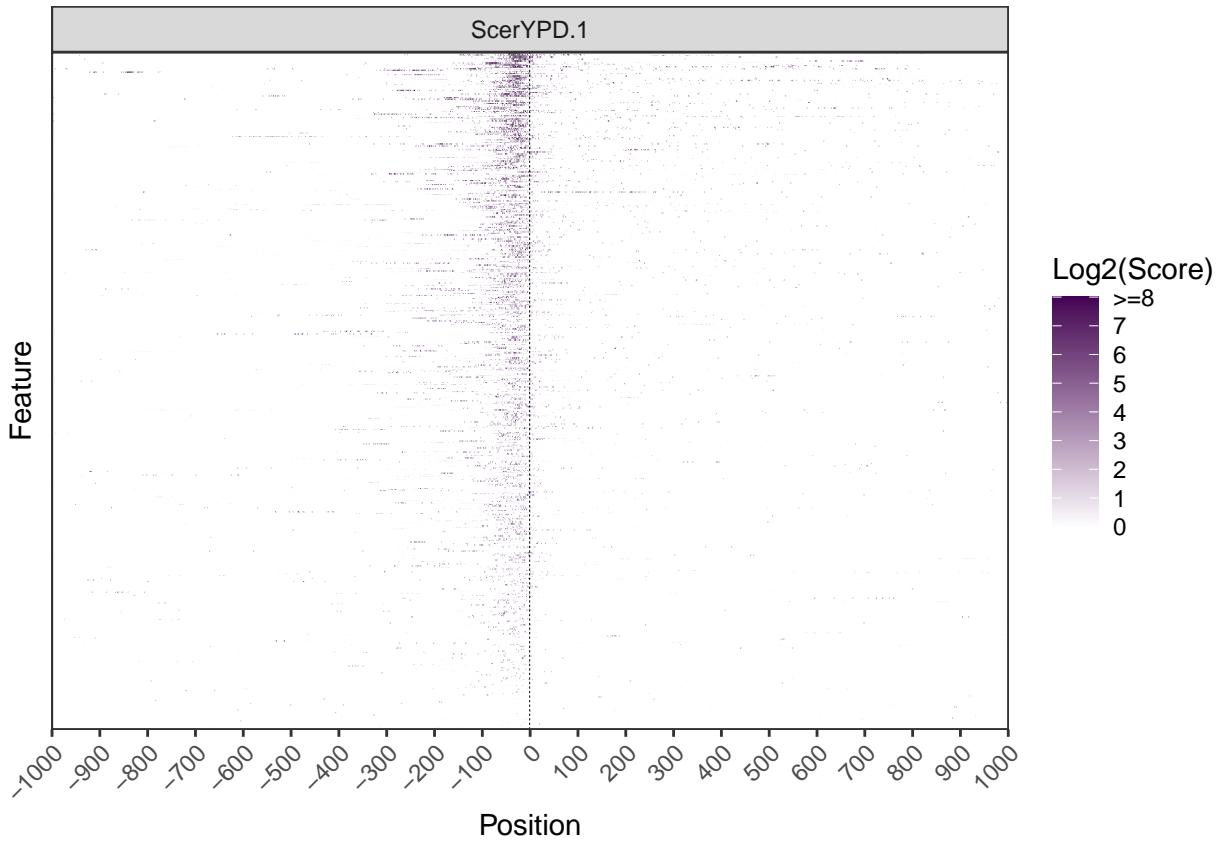
## ScerGal.1.negerGal.1.pos & ScerYPD.1.neg & YPD.1.pos



```
# Density plot (Fig. 1G)
plot_density(exp, data_type = "tss", samples = "ScerYPD.1", threshold = 10) +
  theme(text = element_text(color = "black"))
```



```
# Heatmap (Fig. 1H)
plot_heatmap(exp, data_type = "tss", samples = "ScerYPD.1", threshold = 10,
             log2_transform = TRUE, use_normalized = TRUE, high_color = "#440154FF",
             upstream = 1000, downstream = 1000, max_value = 8) +
  theme(text = element_text(color = "black"))
```



### TSR detection

```
# Cluster TSSs
exp <- tss_clustering(exp, max_distance = 25, max_width = 250,
                      threshold = 10, n_samples = 1)

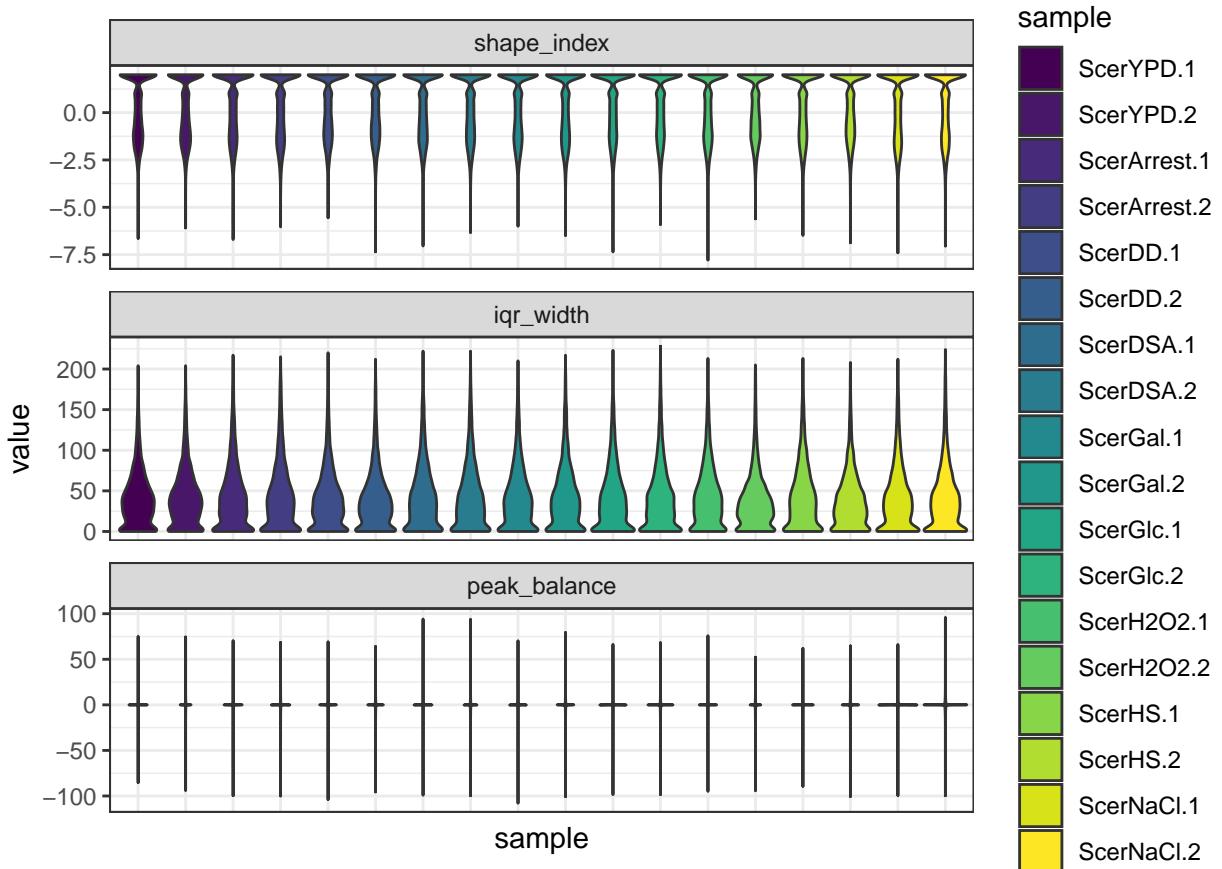
# Associate TSSs with TSRs
exp <- associate_with_tsr(exp)

# Mark dominant TSS per TSR
exp <- mark_dominant(exp, data_type = "tss", threshold = 10)
```

### TSR metrics

```
exp <- tsr_metrics(exp)

plot_tsr_metric(exp, tsr_metrics = c("shape_index", "iqr_width", "peak_balance"),
                log2_transform = FALSE, ncol = 1, samples = samples_ordered,
                threshold = 10) +
  theme(text = element_text(color = "black"))
```



### Sequence analysis

```
# Sequence logos (Fig. 2B-C)
plot_sequence_logo(
  exp, dominant = TRUE, samples = "ScerYPD.1", data_conditions = conditionals(
    data_quantiling = quantiling(score, 5), data_filters = tsr_width > 1))

# Color map (Fig. 2D)
plot_sequence_colormap(exp, samples = "ScerYPD.1", dominant = TRUE,
                       data_conditions = conditionals(
                         data_filters = tsr_width > 1)) +
  theme(text = element_text(color = "black", size = 12))
```

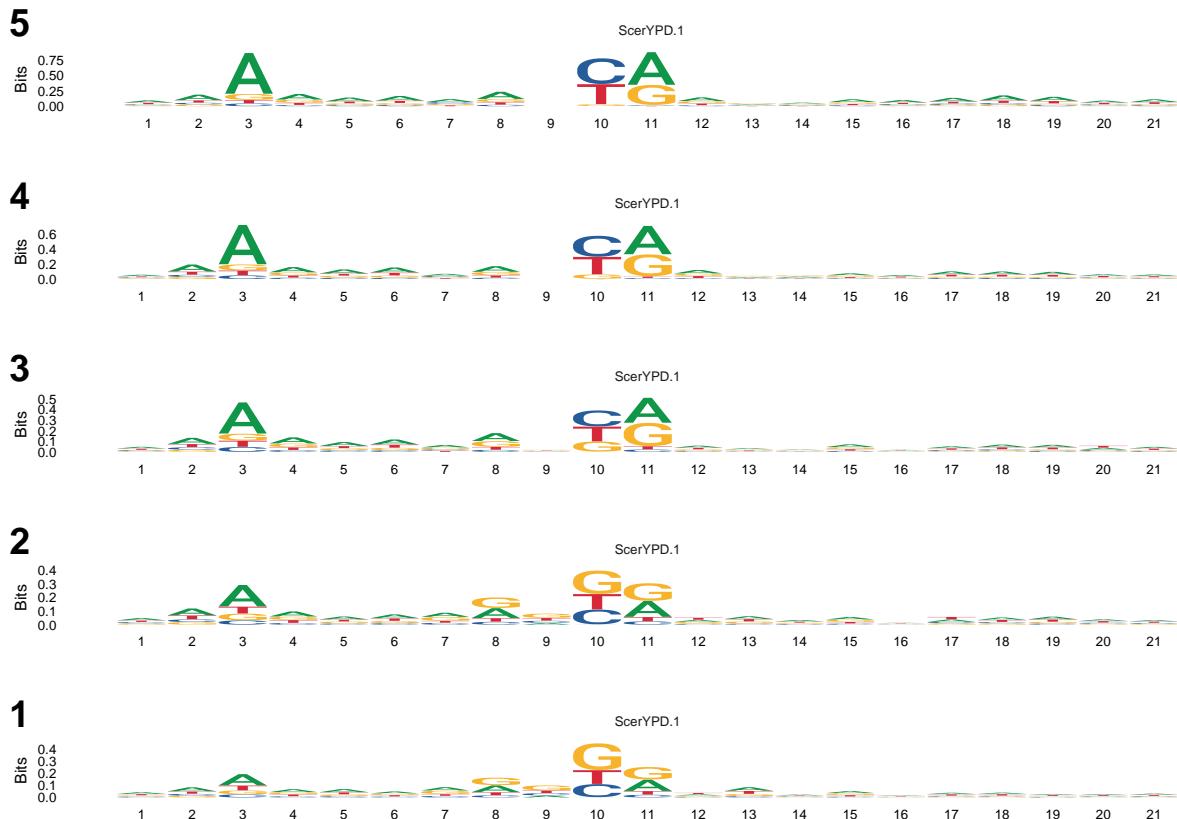
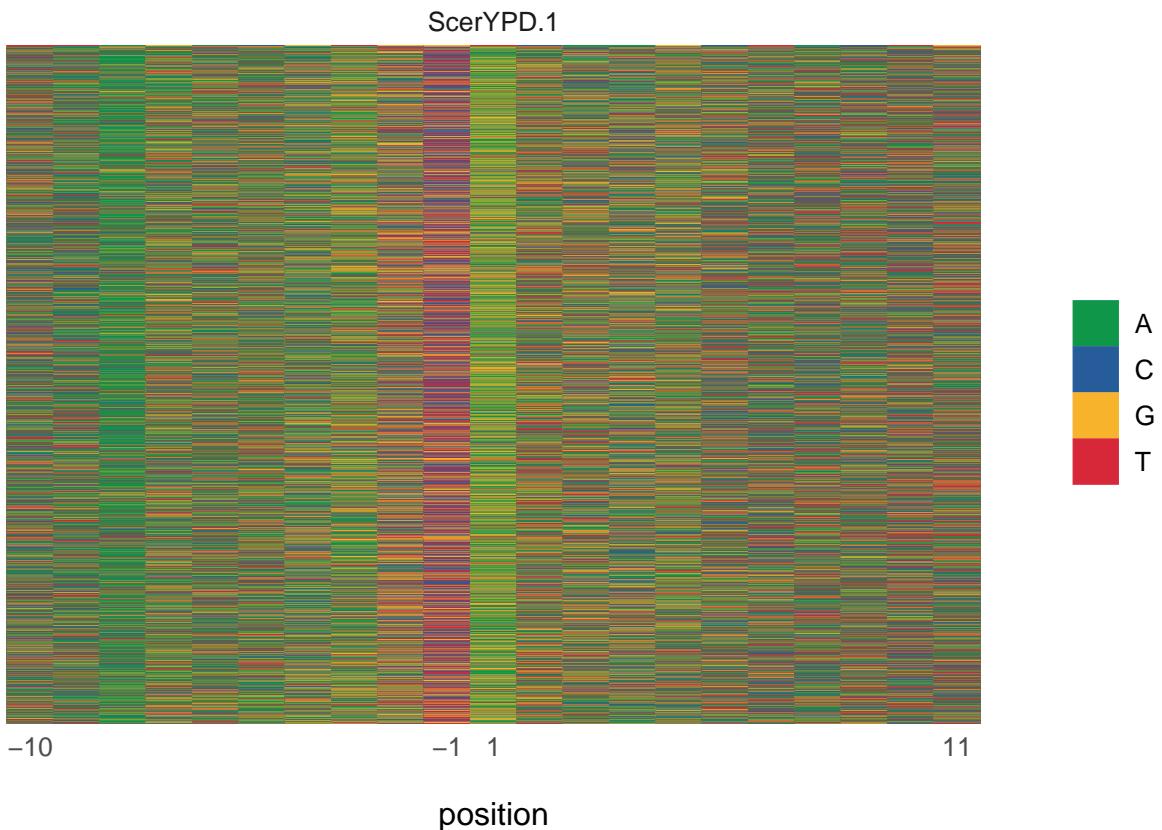


Figure 1: Sequence logos



```
# Dinucleotide frequencies (Fig. 2E)
plot_dinucleotide_frequencies(exp, samples = "ScerYPD.1", dominant = TRUE, threshold = 10,
                               data_conditions = conditionals(data_filters = iqr_width > 10)) +
  scale_fill_viridis_c() +
  theme(text = element_text(color = "black"))
```

### Differential TSR analysis

```
# Build model
exp <- fit_de_model(exp, data_type = "tsr", formula = ~condition, method = "deseq2")

# Generate list of treatments for loop
treatments <- filter(exp@meta_data$sample_sheet, condition != "ScerYPD") %>%
  dplyr::select(condition) %>%
  distinct() %>%
  unlist()

# Loop to compare all treatments to YPD
for(i in treatments) {
  exp <- differential_expression(
    exp, data_type = "tsr",
    comparison_name = str_c(i, "_vs_YPD"),
    comparison_type = "contrast",
    comparison = c("condition", i, "ScerYPD"))
}
```

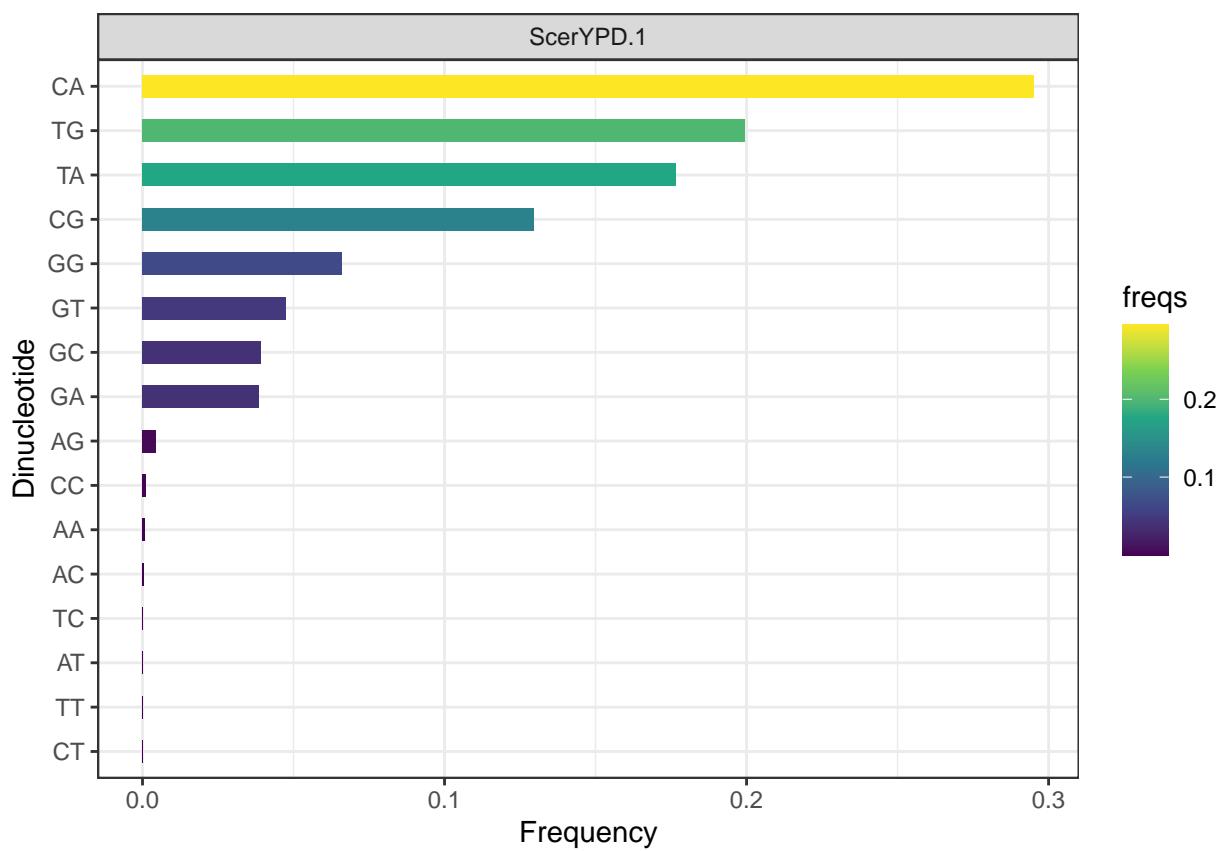


Figure 2: Dinucleotide frequencies for YPD replicate 1

```

plot_num_de(exp, data_type = "tsr", de_comparisons = "all",
            log2fc_cutoff = 1, fdr_cutoff = 0.05,
            keep_unchanged = FALSE) +
  theme_bw() +
  scale_fill_viridis_d(end = 0.5) +
  scale_y_continuous(expand = c(0,0), limits = c(0,8000)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        text = element_text(color = "black", size = 12))

```

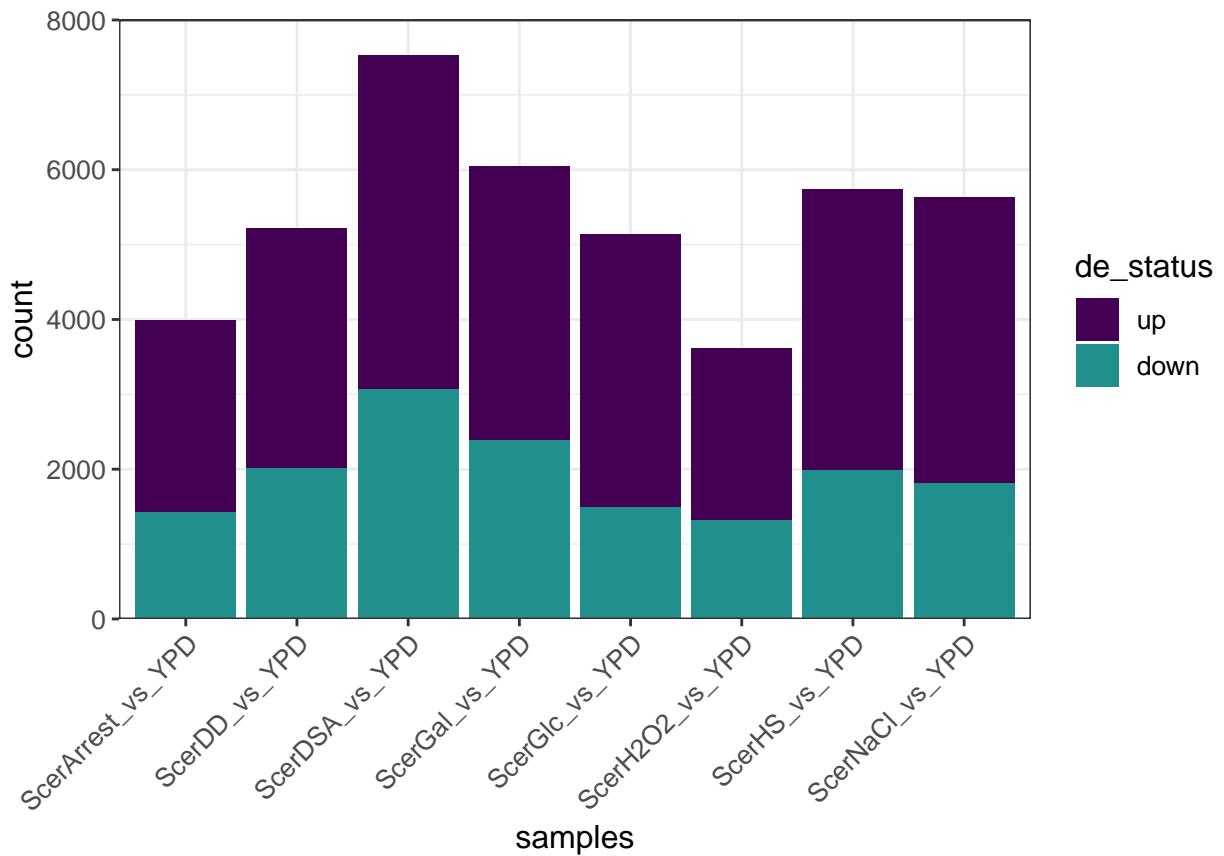


Figure 3: Number of differentially expressed TSRs in each comparison

```

plot_ma(exp, data_type = "tsr", de_comparisons = "ScerGal_vs_YPD", size = 1) +
  scale_color_viridis_d(end = 0.6, direction = -1) +
  theme_bw() +
  theme(text = element_text(color = "black", size = 12))

```

```

plot_volcano(exp, data_type = "tsr", de_comparisons = "ScerGal_vs_YPD", size = 1) +
  theme_bw() +
  scale_color_viridis_d(end = 0.6, direction = -1) +
  theme(text = element_text(color = "black", size = 12))

```

### Functional annotation of differential TSRs

```

# Annotate DE TSRs
exp <- annotate_features(exp, data_type = "tsr_diff", feature_type = "transcript",
                         upstream = 500, downstream = 100)

```

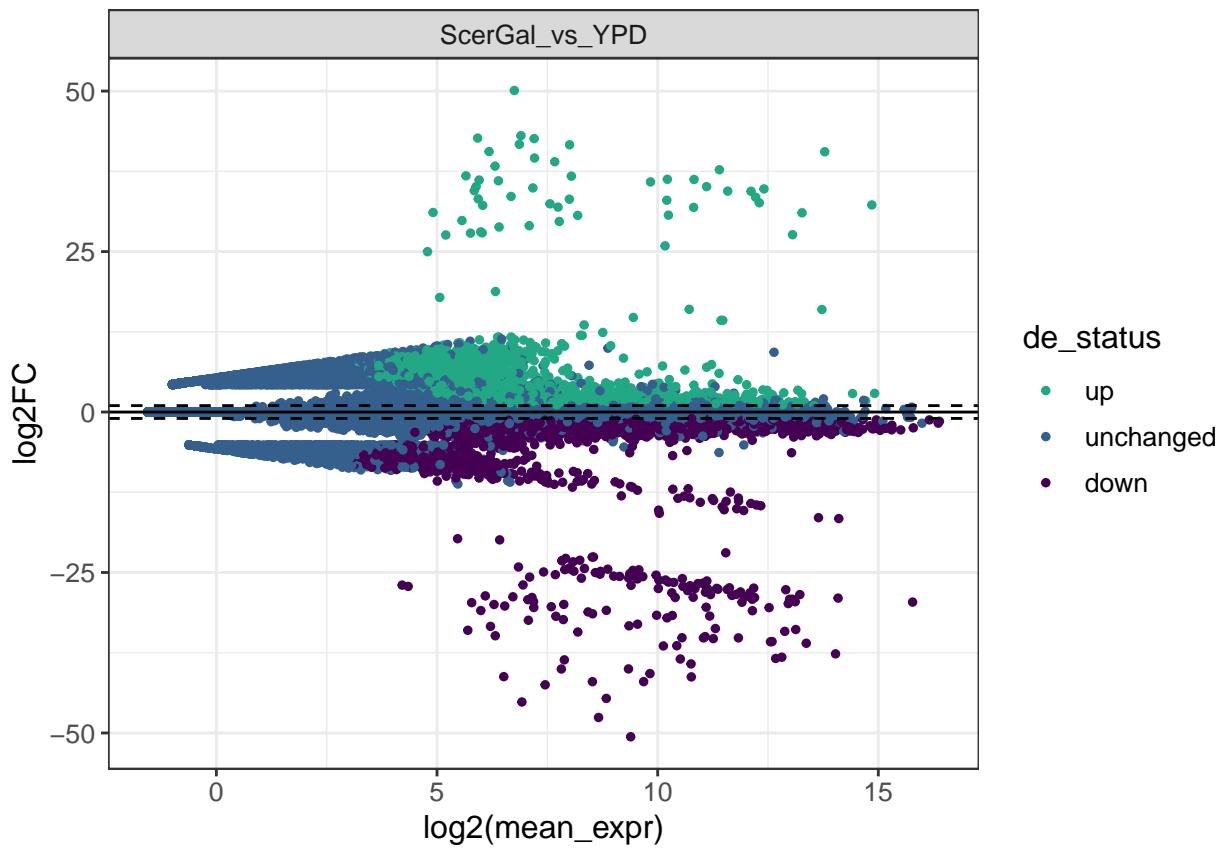


Figure 4: MA plot for YPG vs. YPD comparison

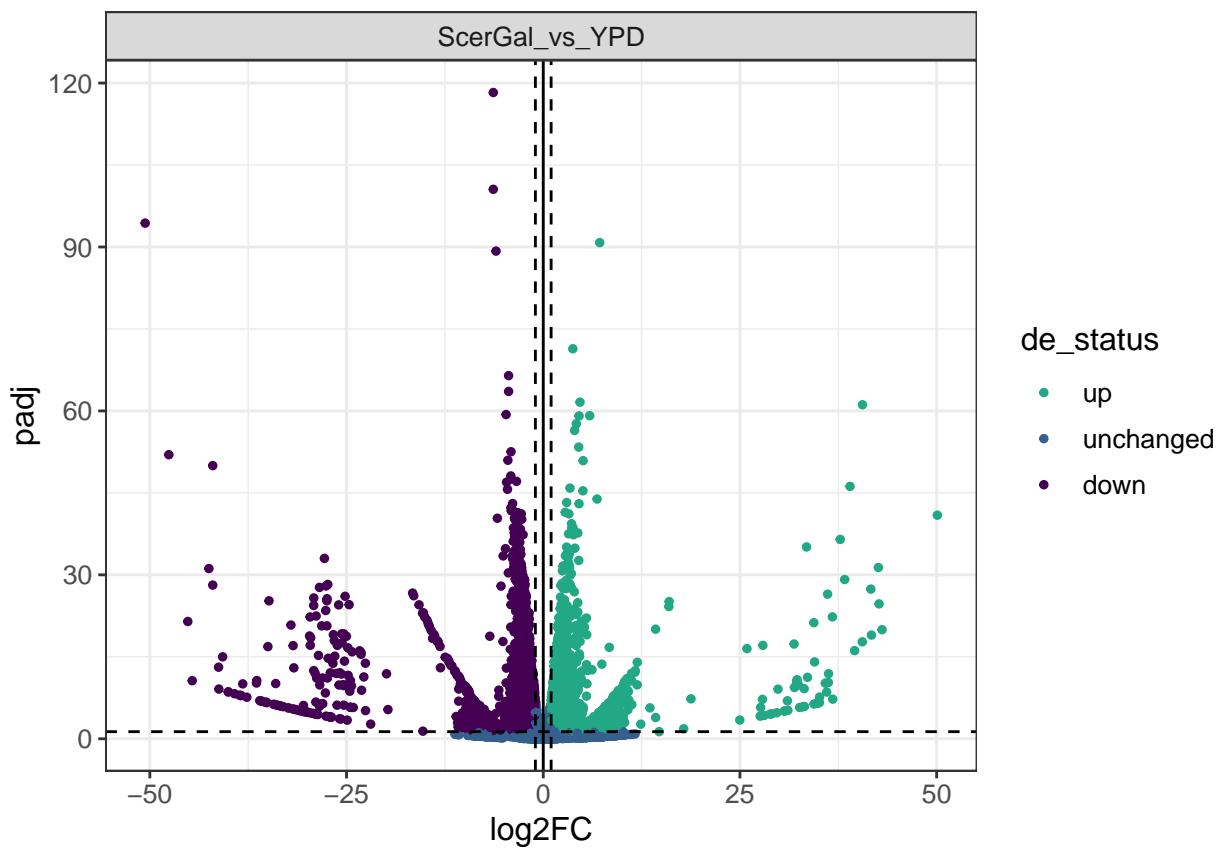


Figure 5: Volcano plot for YPG vs. YPD comparison

```

# GO analysis (Fig. 2I)
enrichment_data <- export_for_enrichment(exp, data_type = "tsr",
                                         de_comparisons = "ScerGal_vs_YPD",
                                         keep_unchanged = FALSE,
                                         anno_categories = "Promoter")

library("clusterProfiler")
library("org.Sc.sgd.db")

# Perform GO enrichment
go_enrichment <- compareCluster(
  geneId ~ sample + de_status,
  data = enrichment_data,
  fun = "enrichGO",
  OrgDb = "org.Sc.sgd.db",
  pAdjustMethod = "fdr",
  ont = "BP",
  keyType="ENSEMBL"
)

dotplot(go_enrichment, font.size = 12, showCategory = 10) +
  scale_color_viridis_c() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 22),
        text = element_text(color = "black", size = 22))

```

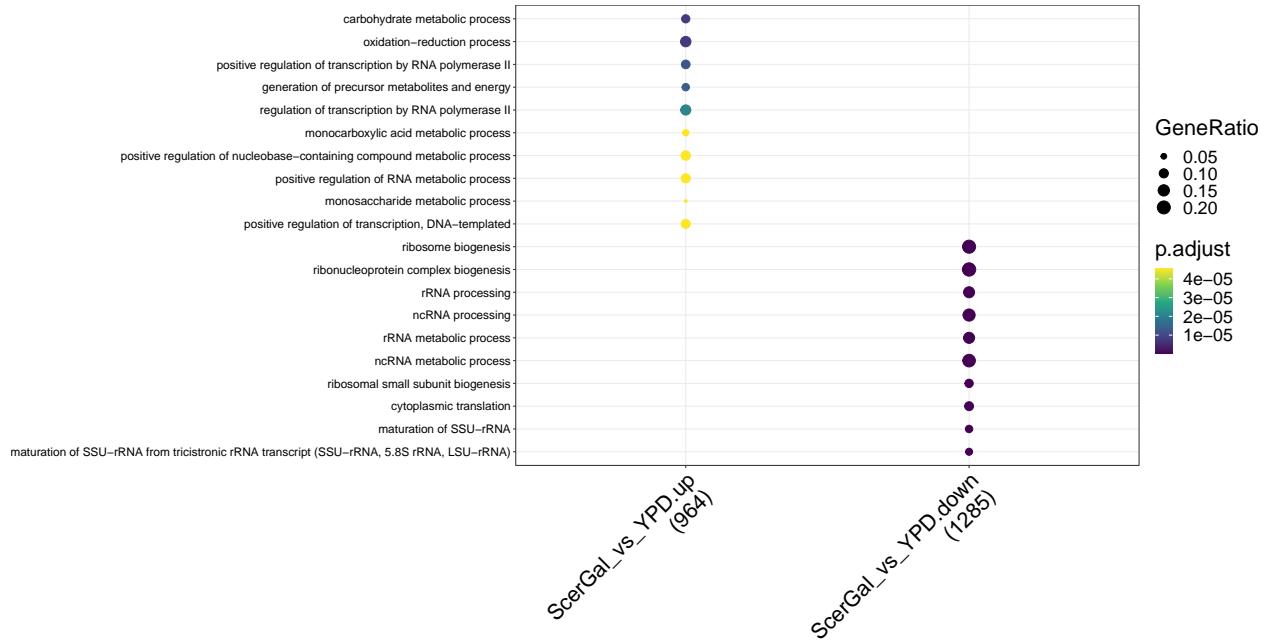


Figure 6: Dot plot of GO biological process enrichment in upregulated and downregulated TSRs in for YPG vs. YPD comparison