

Differential Features

It has been previously shown that there are pervasive changes in TSSs and TSRs between different developmental conditions, disease states, and in response to the environment. These changes in TSRs may be a functionally important readout for alternative promoter usage, as well as other factors such as transcript isoform.

tsrexplorer allows the discovery of differential TSSs, TSRs, and feature level counts similar to RNA-seq.

Preparing Data

This example will use a set of *S. cerevisiae* TSRs collected using the STRIPE-seq method. One of the TSR sets comes from a treatment group that consisted of a 1 hour exposure to 1.5 mM Diamide prior to collection of total-RNA. Diamide is a reducing agent that damages DNA, and was shown to cause large changes to the transcriptome.

```
library("TSRexploreR")

# Load example TSSs.
TSSs <- system.file("extdata", "S288C_TSSs.RDS", package="TSRexploreR")
TSSs <- readRDS(TSSs)

# Prepare genome annotation.
annotation <- system.file("extdata", "S288C_Annotation.gtf", package="TSRexploreR")

# Prepare sample sheet.
sample_sheet <- data.frame(
  sample_name=c(sprintf("S288C_D_%s", seq_len(3)), sprintf("S288C_WT_%s", seq_len(3))),
  file_1=NA, file_2=NA,
  condition=c(rep("Diamide", 3), rep("Untreated", 3))
)

# Create TSRexploreR object.
exp <- tsr_explorer(TSSs, sample_sheet=sample_sheet, genome_annotation=annotation)
```

```
## Warning in .get_cds_IDX(mcols0$type, mcols0$phase): The "phase" metadata column contains non-NA values
## stop_codon. This information was ignored.
```

Processing of TSSs

The start of TSS processing is similar to that shown in the standard analysis vignette. Details about each step can be found here. A few of the steps are skipped for expediency of the vignette, but all analysis and plots can be generated for the different treatment condition sets.

```
exp <- format_counts(exp, data_type="tss") %>%
  normalize_counts(data_type="tss") %>%
  tss_clustering(threshold=3, max_distance=25)
```

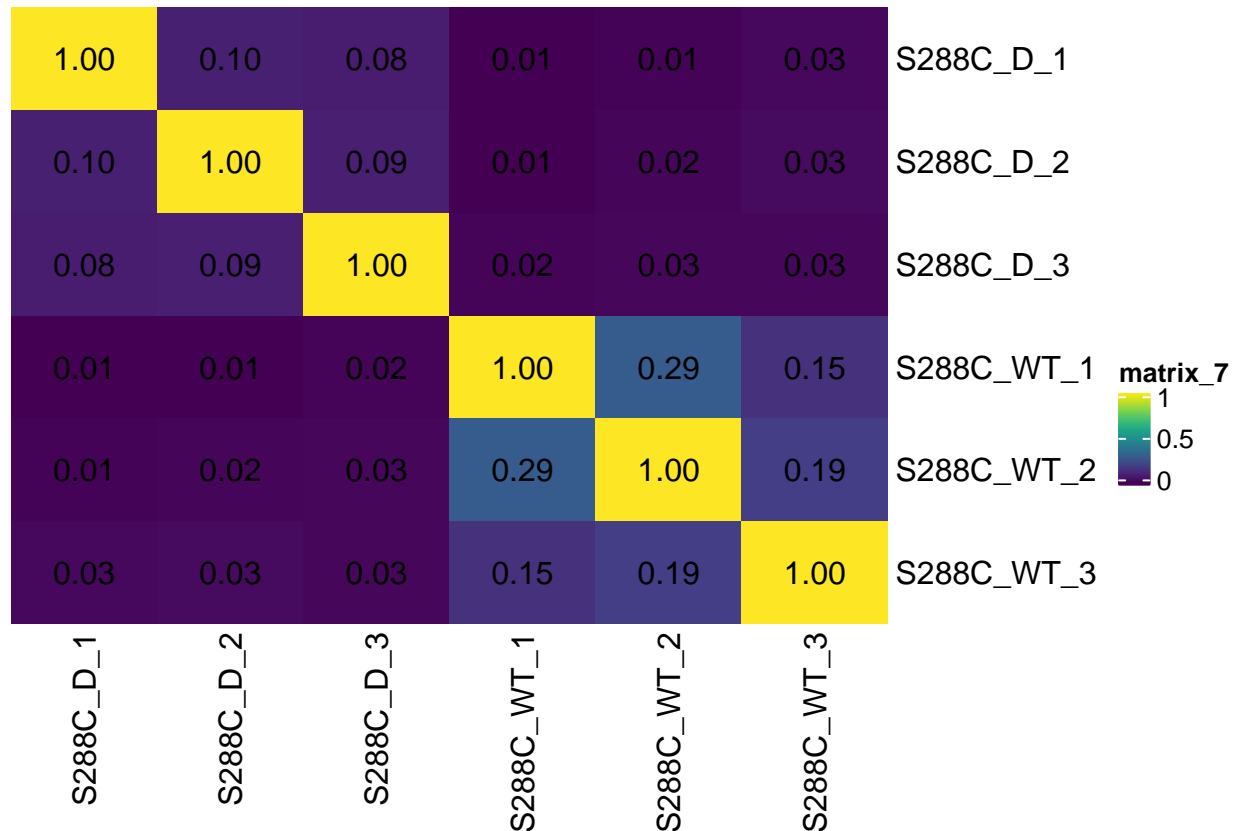
TSR Correlation and Dimension Reduction

Correlation analysis and dimensionality reduction is a good first step to assess overall differences between samples.

Correlation Matrix Plots

After ‘TMM’ normalizing between the samples, various correlation plots can be generated by tsrexplorer. An example is shown here, in which half the plot is a scatter plot, and the other half a heatmap. Already one can start to see a difference between the untreated and treated samples.

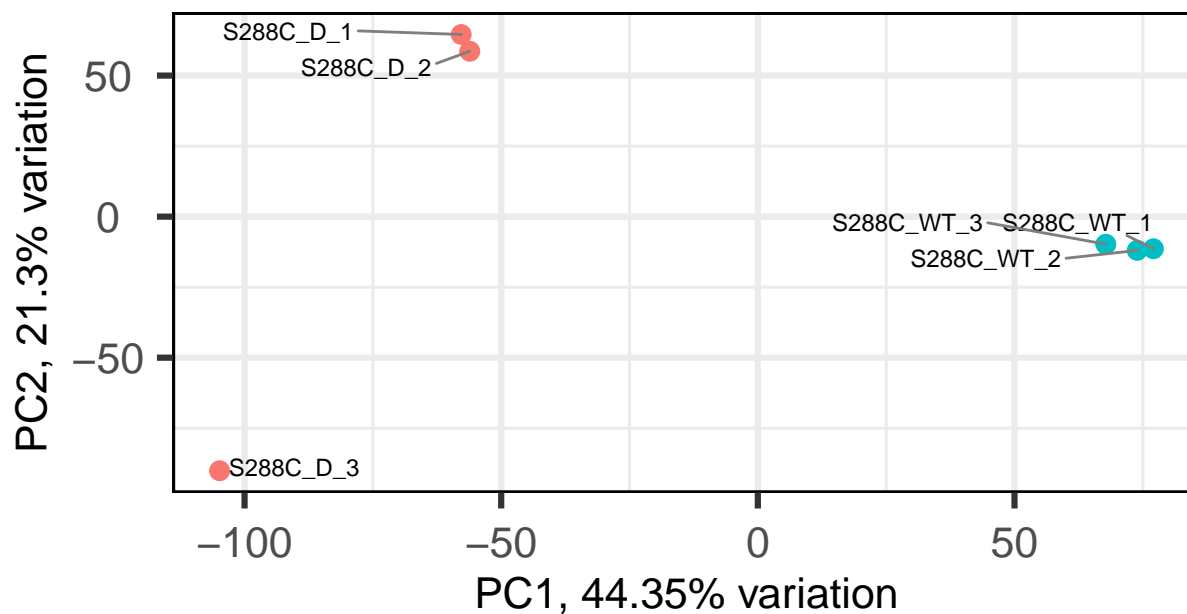
```
plot_correlation(  
  exp, data_type="tsr", font_size=12,  
  heatmap_colors=viridis::viridis(100)  
)
```



Dimensionality Reduction

Another useful way to check differences between samples (and look at replicate concordance) is to generate a dimensionality reduction plot. The preferred plot type for this is UMAP for its general robustness, stability, and often better ability to separate points.

```
plot_reduction(exp, data_type="tsr", remove_var=0.25, colby="condition")
```



Differential TSRs

Between different developmental or environmental conditions, there may be differential usage of TSSs and TSRs. `tsrexplorer` takes advantage of `edgeR` to allow the discovery of these potential differential features.

Discovering Differential TSRs

The first step is to build an `edgeR` model that contains the samples and their (replicate) groups. This can be done at the TSS, TSR, or feature level. This example will be looking at differential TSRs.

```
exp <- fit_de_model(exp, data_type="tsr", formula= ~ condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

After building the `edgeR` model, the model can then be used to call differential TSRs.

```
exp <- differential_expression(
  exp, data_type="tsr",
  comparison_name="Diamide_vs_Untreated",
  comparison_type="name",
  comparison="condition_Untreated_vs_Diamide"
)
```

```
## Warning in .local(x, row.names, optional, ...): Arguments in '...' ignored
```

Annotating Differential Features

The next step is to annotate the differential features.

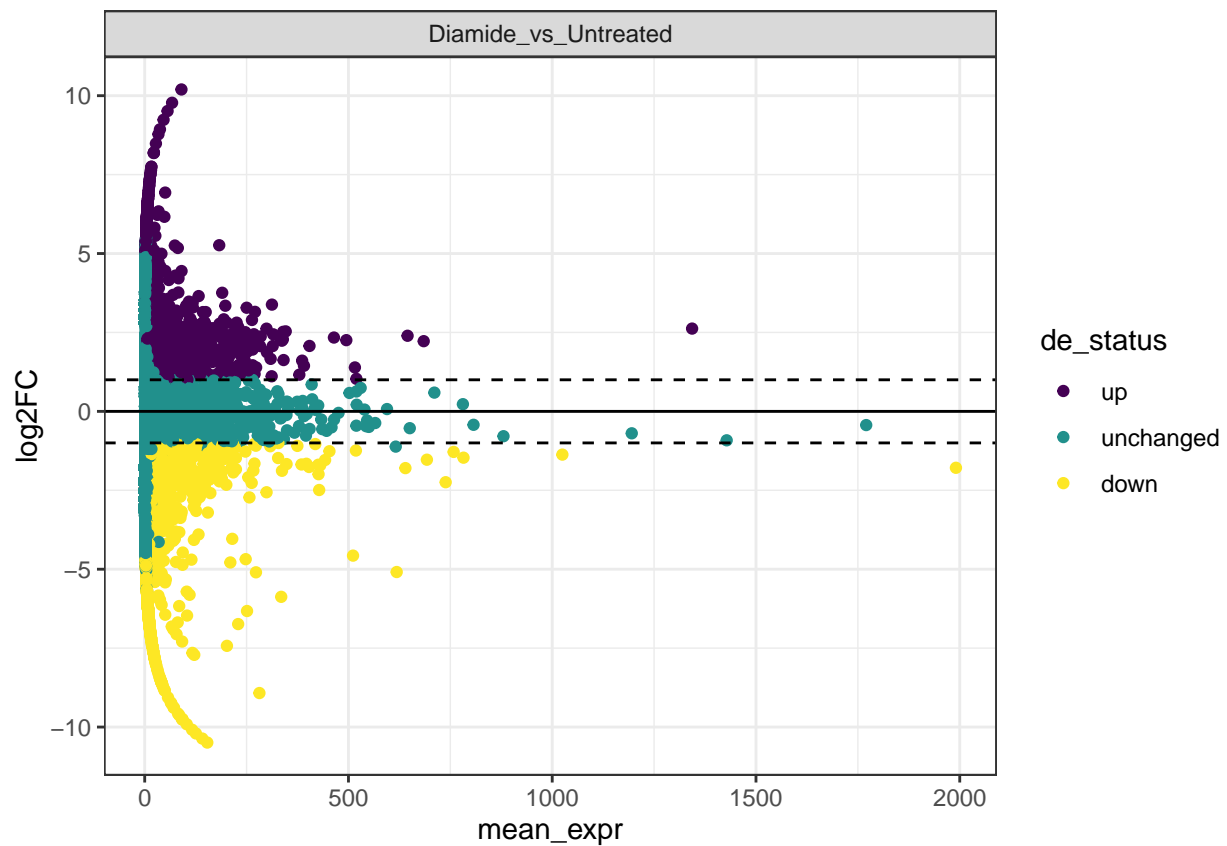
```
exp <- annotate_features(exp, data_type="tsr_diff", feature_type="transcript")
```

```
## >> preparing features information...    2020-11-17 01:05:30 PM
## >> identifying nearest features...      2020-11-17 01:05:30 PM
## >> calculating distance from peak to TSS... 2020-11-17 01:05:30 PM
## >> assigning genomic annotation...      2020-11-17 01:05:30 PM
## >> assigning chromosome lengths        2020-11-17 01:05:31 PM
## >> done...                            2020-11-17 01:05:31 PM
```

MA Plots

MA plots provide a good overview of the distribution of differentially expressed genes relative to average gene expression.

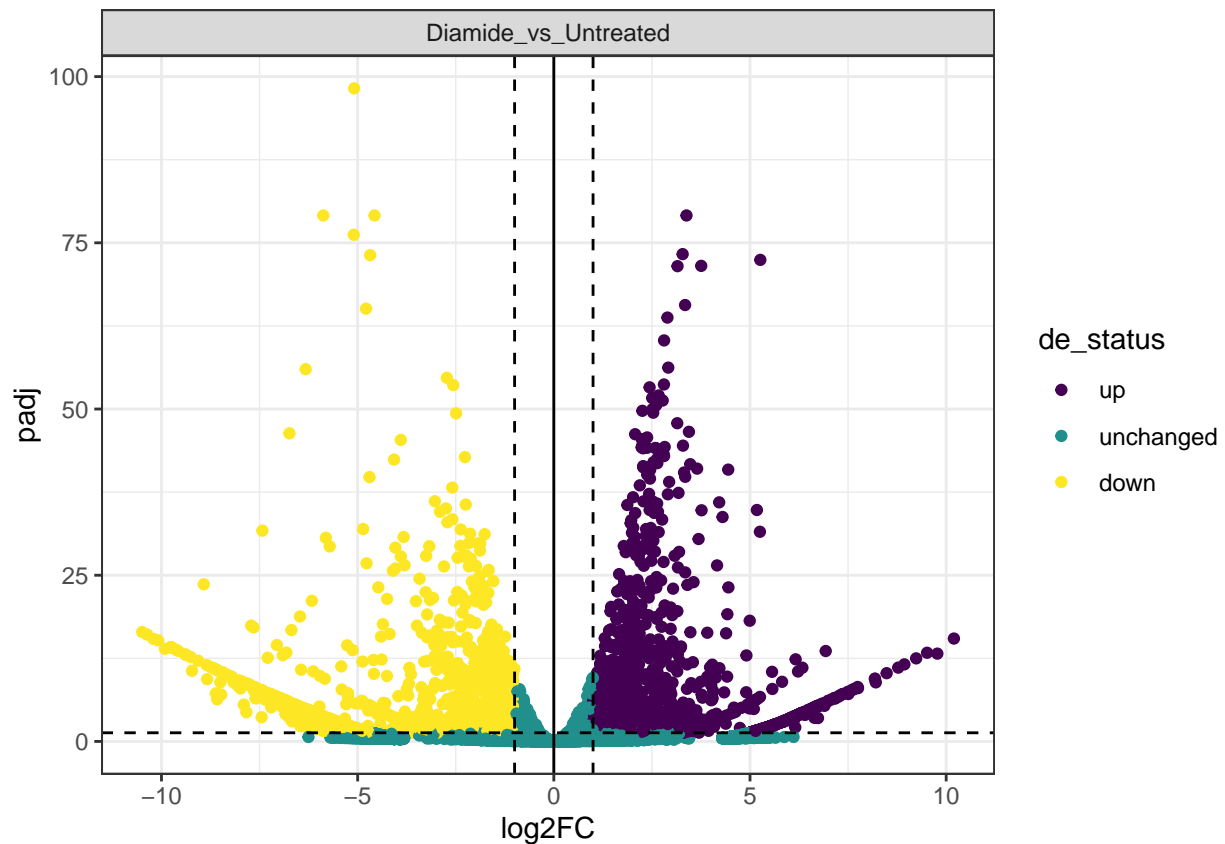
```
plot_ma(exp, data_type="tsr") +  
  scale_color_viridis_d() +  
  theme_bw()
```



Volcano Plots

```
plot_volcano(exp, data_type="tsr") +  
  scale_color_viridis_d() +  
  theme_bw()
```

```
## Warning: Removed 10925 rows containing missing values (geom_point).
```



Number of DE TSRs

A convenient plot to visualize the number of DE TSRs between the different comparisons.

```
p <- plot_num_de(exp, data_type="tsr", de_comparisons="Diamide_vs_Untreated") +  
  scale_fill_viridis_d() +  
  theme_bw()
```

Gene Ontology.

Gene ontology enrichment is a good way to uncover the higher order processes that are being changed. The preferred tool for enrichment analysis in R is the clusterProfiler library. tsrexplorer provides a convenient function to export DE features to a format ready for analysis with clusterProfiler.

```

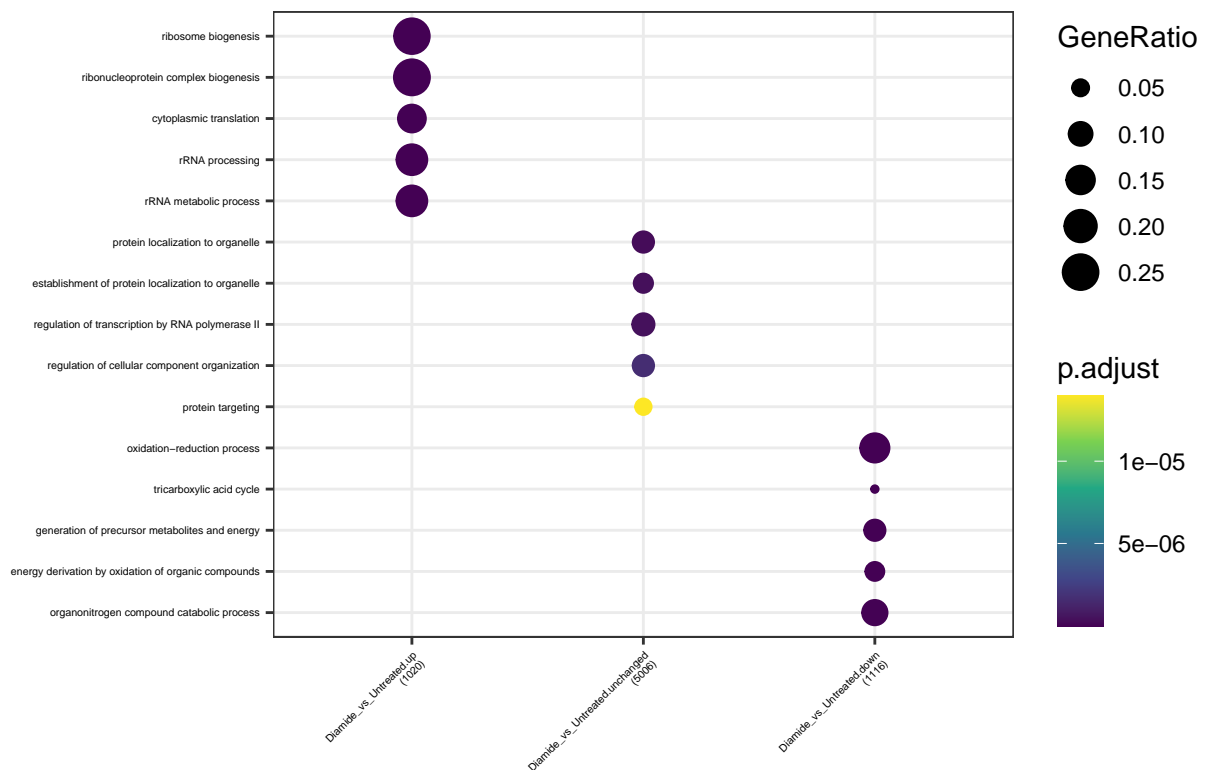
enrichment_data <- export_for_enrichment(exp, data_type="tsr")

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

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=4) +
  scale_color_viridis_c() +
  theme(axis.text.x=element_text(angle=45, hjust=1))

```



Gene Tracks.

You can make gene tracks for genes with differential TSRs.

```

gene_tracks(
  exp, feature_name="YKR076W",
  samples=c(
    TSS="S288C_WT_1", TSR="S288C_WT_1",
    TSS="S288C_D_1", TSR="S288C_D_1"
  ),
  tss_colors=viridis::viridis(2),
  tsr_colors=viridis::viridis(2)
)

```

