GSEA Ribo vs polyA

Ariel Hippen

2022-07-28

Contents

Analyzing differential expression results

We have calculated differential expression genes in DE_ribo_vs_polyA.Rmd, now we will try to make sense of them. Our main workhorse will be Gene Set Enrichment Analysis (GSEA) across several reference sets.

```
suppressPackageStartupMessages({
  library(DESeq2)
  library(WebGestaltR)
  library(ggplot2)
  library(rtracklayer)
  library(yaml)
})

params <- read_yaml("../../config.yml")
data_path <- params$data_path
local_data_path <- params$local_data_path
samples <- params$samples
```

```
# Load the DESeq2 object with the original count matrix
deseq_path <- paste(local_data_path, "deseq2_output", sep = "/")
dds <- readRDS(paste(deseq_path, "ribo_vs_polyA_data.rds", sep = "/"))

# Load the DESeqResults object with differentially expressed genes, at FDR 0.1 and 0.05
```

```
res1 <- readRDS(paste(deseq_path, "ribo_vs_polyA_FDR_0.1.rds", sep = "/"))
res05 <- readRDS(paste(deseq_path, "ribo_vs_polyA_FDR_0.05.rds", sep = "/"))
```

Top genes

Let's look at the top 20 most upregulated and downregulated genes and see if we can find a pattern.

```
res1 <- subset(res1, res1$padj < 0.1)
res1 <- res1[order(res1$log2FoldChange), ]
as.data.frame(head(res1, n=20))
```

```
##                baseMean log2FoldChange     lfcSE       stat        pvalue
## RMRP       24000.731646     -11.350487 0.5204542 -21.808808 1.913968e-105
## AL356488.2  6402.189419     -11.192611 0.4557820 -24.556941 3.646291e-133
## AL355075.4 10035.558064     -10.486047 0.3934179 -26.653710 1.620973e-156
## AL627171.2 32028.174113      -8.953906 0.3508589 -25.519965 1.183609e-143
## AC006064.5   800.339233      -8.676798 0.4171055 -20.802409  4.116075e-96
## HIST1H4F     126.165209      -7.584147 0.6718555 -11.288360  1.498057e-29
## TERC         380.580052      -7.565524 0.4326045 -17.488317  1.758762e-68
## HIST1H1E    3875.456278      -7.466552 0.3881176 -19.237858  1.784320e-82
## HIST1H3I      97.949551      -7.450876 0.4741335 -15.714724  1.199025e-55
## HIST1H4L      28.497041      -7.407631 0.7322943 -10.115648  4.708914e-24
## HIST1H4B     195.566155      -7.098757 0.5481861 -12.949537  2.363749e-38
## HIST1H4A     130.716736      -6.862988 0.7576443  -9.058325  1.324685e-19
## HIST1H3F     259.470135      -6.856573 0.4361862 -15.719372  1.114223e-55
## HIST1H1B     658.088415      -6.657982 0.4850095 -13.727530  6.946162e-43
## AC009686.1     9.232829      -6.360311 0.8327379  -7.637830  2.209135e-14
## HIST1H2BI    183.101743      -6.321904 0.5317333 -11.889239  1.346270e-32
## PRKCA-AS1     36.574623      -6.316965 0.7225019  -8.743181  2.266365e-18
## AC023385.1    18.779022      -6.209457 0.6563505  -9.460581  3.062363e-21
## AC051619.7    10.648389      -6.177520 0.8078444  -7.646918  2.058532e-14
## HIST1H4C     319.913332      -6.170677 0.5796527 -10.645473  1.830527e-26
##                     padj
## RMRP       6.277381e-103
## AL356488.2 1.783716e-130
## AL355075.4 1.299570e-153
## AL627171.2 6.832263e-141
## AC006064.5  1.006764e-93
## HIST1H4F    3.336182e-28
## TERC        1.944881e-66
## HIST1H1E    2.976823e-80
## HIST1H3I    8.066728e-54
## HIST1H4L    7.878764e-23
## HIST1H4B    8.131410e-37
## HIST1H4A    1.656545e-18
## HIST1H3F    7.531311e-54
## HIST1H1B    2.935287e-41
## AC009686.1  1.893113e-13
## HIST1H2BI   3.551740e-31
## PRKCA-AS1   2.624872e-17
## AC023385.1  4.276048e-20
## AC051619.7  1.766782e-13
```

```
## HIST1H4C    3.478121e-25
```

Reminder, these ^ are the ones that are much more expressed in rRNA depletion than poly-A selection. There's a lot of genes to make histone proteins here! The top gene, RMRP, is the RNA component of a endoribonuclease that cleaves mitochondrial RNA. TERC is the RNA component of telomerase. The others are a bunch of lncRNAs that seem pretty unstudied.

```
as.data.frame(tail(res1, n=20))
```

```
##                baseMean log2FoldChange      lfcSE      stat        pvalue
## AC011388.1 5.444677e+00       3.221127 0.77020442  4.182171  2.887382e-05
## AL161431.1 2.220362e+02       3.222972 0.52357710  6.155678  7.475675e-10
## AL137857.1 1.132349e+01       3.228229 0.55671968  5.798662  6.684594e-09
## Z83745.1   6.766042e+01       3.293826 0.31180885 10.563608  4.394409e-26
## MT-CO2     2.573582e+05       3.374991 0.07228102 46.692626  0.000000e+00
## AL713998.1 1.192016e+01       3.441651 0.58969724  5.836302  5.337227e-09
## LINC02604  2.430626e+02       3.534244 0.17856872 19.792064  3.484742e-87
## AC008735.2 1.212162e+02       3.883226 0.19535223 19.878072  6.300941e-88
## AC245060.5 4.607114e+01       4.005727 0.43611374  9.185052  4.113281e-20
## AC087500.2 8.705568e+00       4.629852 0.75710763  6.115183  9.644618e-10
## MUC3A      4.401273e+02       4.762839 0.53305838  8.934929  4.074406e-19
## MT-CYB     2.136425e+05       5.148185 0.19336782 26.623793 3.600591e-156
## MT-ND4L    4.208011e+04       5.264596 0.10811378 48.694955  0.000000e+00
## MT-ATP6    1.609113e+05       5.271319 0.09431666 55.889582  0.000000e+00
## MT-ND5     1.868483e+05       5.633187 0.24955362 22.573051 7.974851e-113
## MT-ND1     2.373681e+05       5.816280 0.09704852 59.931670  0.000000e+00
## MT-ND6     4.081759e+04       5.924393 0.21969695 26.966202 3.683395e-160
## MT-ND4     3.864219e+05       6.108083 0.18138678 33.674356 1.372419e-248
## MT-ATP8    1.979167e+04       6.220806 0.10817253 57.508192  0.000000e+00
## MT-ND2     2.235693e+05       6.714580 0.17078582 39.315792  0.000000e+00
##                     padj
## AC011388.1  9.672757e-05
## AL161431.1  4.323040e-09
## AL137857.1  3.508470e-08
## Z83745.1    8.156362e-25
## MT-CO2      0.000000e+00
## AL713998.1  2.826986e-08
## LINC02604   6.750109e-85
## AC008735.2  1.262901e-85
## AC245060.5  5.333222e-19
## AC087500.2  5.506686e-09
## MUC3A       4.963931e-18
## MT-CYB     2.808655e-153
## MT-ND4L     0.000000e+00
## MT-ATP6     0.000000e+00
## MT-ND5     2.989223e-110
## MT-ND1      0.000000e+00
## MT-ND6     3.221520e-157
## MT-ND4     3.046982e-245
## MT-ATP8     0.000000e+00
## MT-ND2      0.000000e+00
```

These ^ are the genes that are much more expressed in poly-A selection. The obvious jump-out is the

mitochondrial genes. Other things to note are MUC3A, which encodes an epithelial glycoprotein, and a bunch of lncRNAs.

Volcano plot

We'll now filter down to only protein-coding genes, using info from the gtf file downloaded from the Cellranger website.

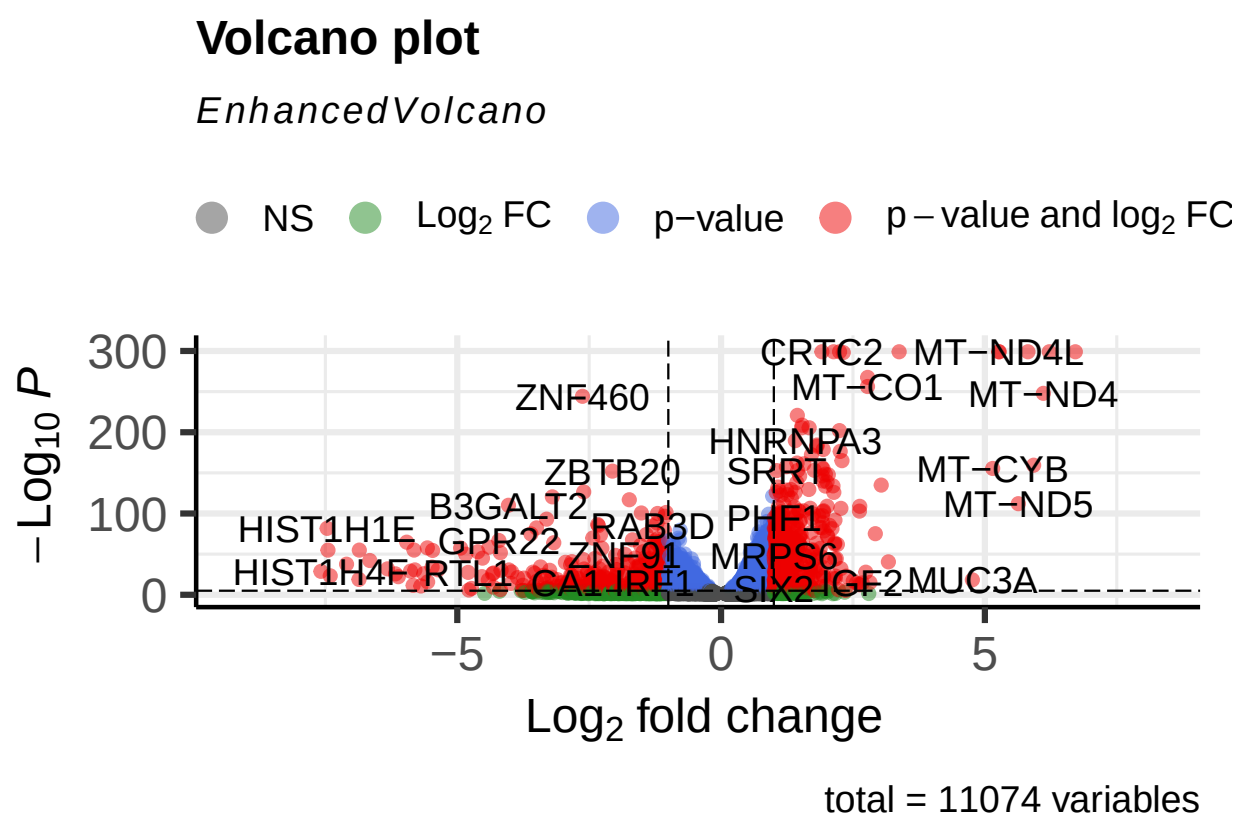
```
genefile <- paste(data_path,"index/refdata-gex-GRCh38-2020-A/genes/genes.gtf",sep = "/")
gff <- readGFF(genefile)
protein_coding <- subset(gff, gff$gene_type=="protein_coding")

res1 <- subset(res1, rownames(res1) %in% protein_coding$gene_name)
```

```
library(EnhancedVolcano)
```

```
## Loading required package: ggrepel
```

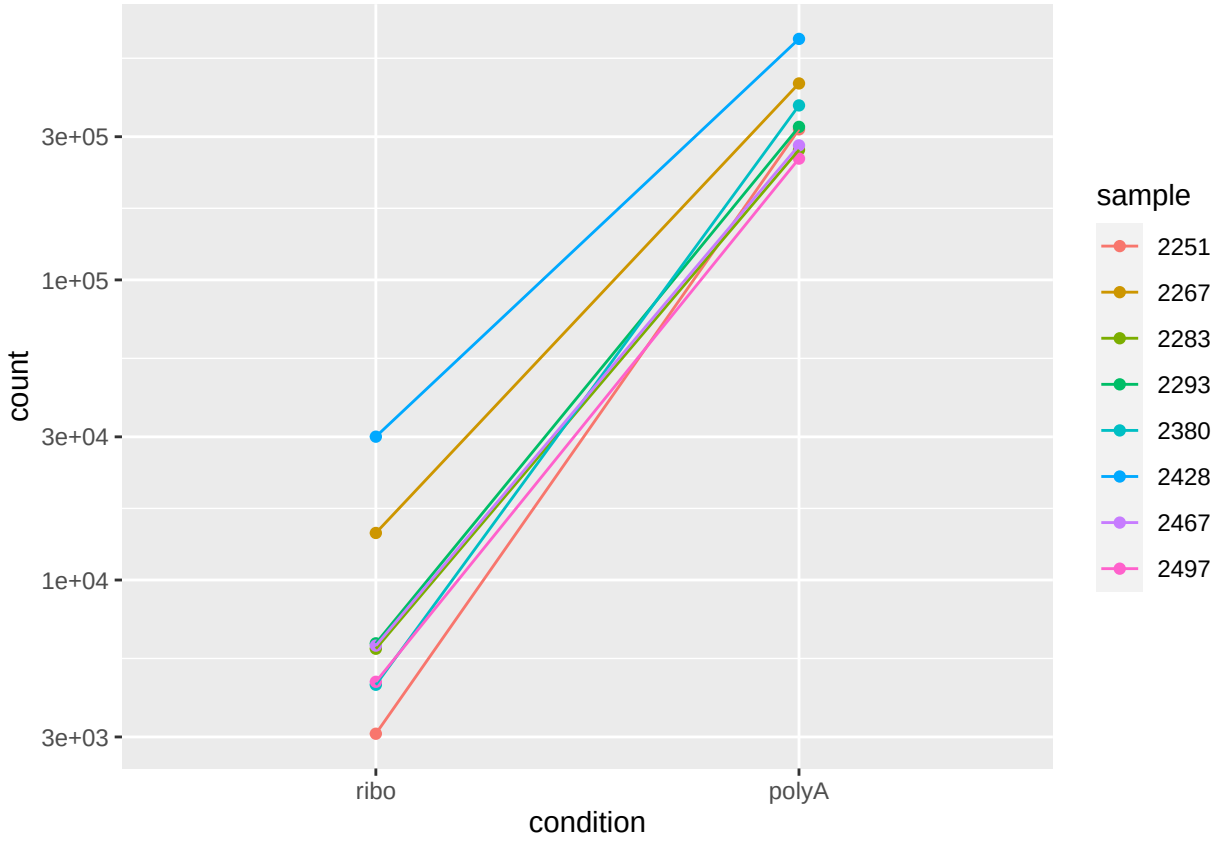
```
EnhancedVolcano(res1, lab = rownames(res1), x = 'log2FoldChange', y = 'pvalue')
```

```
## Warning: One or more p-values is 0. Converting to 10^-1 * current lowest non-
## zero p-value...
```



Plot counts

```
gene <- "MT-ND5"
d <- plotCounts(dds, gene=gene, intgroup=c("condition","sample"), returnData=TRUE)
ggplot(d, aes(x=condition, y=count, group=sample, color=sample)) +
  geom_point() + scale_y_log10() + geom_line()
```



Let's try with all mito genes

```
mt_genes <- grep("MT-", rownames(dds), value = T)
mito_expr <- as.data.frame(colSums(assay(dds[mt_genes,])))
colnames(mito_expr) <- "counts"
mito_expr$id <- colData(dds)$id
```

Let's also try with all histone genes

```
hist_genes <- grep("HIST", rownames(dds), value = T)
hist_expr <- as.data.frame(colSums(assay(dds[hist_genes,])))
colnames(hist_expr) <- "counts"
hist_expr$id <- colData(dds)$id
```

GSEA

WebGestaltR expects a data frame with two columns, gene name and fold change.

```
res1$gene <- rownames(res1); rownames(res1) <- NULL
res1 <- subset(res1, select=c("gene","log2FoldChange"))
res1 <- as.data.frame(res1)
nrow(res1)
```

```
## [1] 11074
```

```
res05 <- subset(res05, res05$padj < 0.05)
res05$gene <- rownames(res05); rownames(res05) <- NULL
res05 <- subset(res05, select=c("gene","log2FoldChange"))
res05 <- as.data.frame(res05)
nrow(res05)
```

```
## [1] 14972
```

GO Biological process

Our first try at GSEA will use the same reference set we used for overrepresentation analysis in the single-cell data, GO Biological process.

```
GO_bp <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                          enrichDatabase = "geneontology_Biological_Process_noRedundant",
                          interestGene = res1,
                          interestGeneType = "genesymbol",
                          isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(GO_bp)
```

```
## [1] 7
```

```
GO_bp <- GO_bp[order(GO_bp$normalizedEnrichmentScore, decreasing = TRUE),]
subset(GO_bp, select=c("geneSet","description","normalizedEnrichmentScore","pValue","FDR","size"))
```

```
##      geneSet                                              description
## 7 GO:0098742 cell-cell adhesion via plasma-membrane adhesion molecules
## 1 GO:0071824                  protein-DNA complex subunit organization
## 2 GO:0071103                                 DNA conformation change
## 3 GO:0006333                       chromatin assembly or disassembly
## 4 GO:0050906      detection of stimulus involved in sensory perception
## 5 GO:0007606                   sensory perception of chemical stimulus
## 6 GO:0009593                             detection of chemical stimulus
##   normalizedEnrichmentScore pValue         FDR size
## 7                 -2.056518      0 0.005633303  165
## 1                 -2.457891      0 0.000000000  166
```

```
## 2                 -2.476920      0 0.000000000  162
## 3                 -2.718720      0 0.000000000  109
## 4                 -3.130926      0 0.000000000   84
## 5                 -3.137225      0 0.000000000   86
## 6                 -3.146241      0 0.000000000   79
```

```
GO_bp_05 <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                            enrichDatabase = "geneontology_Biological_Process_noRedundant",
                            interestGene = res05,
                            interestGeneType = "genesymbol",
                            isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(GO_bp_05)
```

```
## [1] 6
```

```
GO_bp_05 <- GO_bp_05[order(GO_bp_05$normalizedEnrichmentScore, decreasing = TRUE),]
subset(GO_bp_05, select=c("geneSet","description","normalizedEnrichmentScore","pValue","FDR","size"))
```

```
##      geneSet                                          description
## 6 GO:0071824             protein-DNA complex subunit organization
## 5 GO:0071103                               DNA conformation change
## 4 GO:0006333                     chromatin assembly or disassembly
## 1 GO:0050906 detection of stimulus involved in sensory perception
## 2 GO:0007606              sensory perception of chemical stimulus
## 3 GO:0009593                        detection of chemical stimulus
##   normalizedEnrichmentScore pValue          FDR size
## 6                 -1.870329      0 7.689687e-03  159
## 5                 -1.891369      0 6.782698e-03  152
## 4                 -2.157511      0 4.929287e-05  102
## 1                 -2.465208      0 0.000000e+00   73
## 2                 -2.491641      0 0.000000e+00   72
## 3                 -2.524485      0 0.000000e+00   67
```

GO Cellular Component

Ooh for this one, looking at cell components might actually be useful.

```
GO_cc <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                            enrichDatabase = "geneontology_Cellular_Component_noRedundant",
                            interestGene = res05,
                            interestGeneType = "genesymbol",
                            isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(GO_cc)
```

```
## [1] 3
```

```
GO_cc <- GO_cc[order(GO_cc$normalizedEnrichmentScore, decreasing = TRUE),]
subset(GO_cc, select=c("geneSet","description","normalizedEnrichmentScore","pValue","FDR","size"))
```

```
##      geneSet           description normalizedEnrichmentScore      pValue
## 3 GO:0000791           euchromatin                -1.751000 0.005277045
## 1 GO:0032993   protein-DNA complex                -2.416045 0.000000000
## 2 GO:0044815 DNA packaging complex                -2.688547 0.000000000
##          FDR size
## 3 0.02616279   20
## 1 0.00000000  103
## 2 0.00000000   47
```

```
GO_cc_05 <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                            enrichDatabase = "geneontology_Cellular_Component_noRedundant",
                            interestGene = res05,
                            interestGeneType = "genesymbol",
                            isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(GO_cc_05)
```

```
## [1] 3
```

```
GO_cc_05 <- GO_cc_05[order(GO_cc_05$normalizedEnrichmentScore, decreasing = TRUE),]
subset(GO_cc_05, select=c("geneSet","description","normalizedEnrichmentScore","pValue","FDR","size"))
```

```
##      geneSet           description normalizedEnrichmentScore      pValue
## 3 GO:0000791           euchromatin                -1.747878 0.008097166
## 1 GO:0032993   protein-DNA complex                -2.430567 0.000000000
## 2 GO:0044815 DNA packaging complex                -2.703285 0.000000000
##          FDR size
## 3 0.02584806   20
## 1 0.00000000  103
## 2 0.00000000   47
```

So there's more ribosomal-associated transcripts in the poly-A selected cells (that seems correct and encouraging?), whereas the rRNA depleted cells have transcripts with more to do with DNA packaging (i.e. histones).

Cell types

Let's try a custom set for cell types, as curated by the folks at http://www.gsea-msigdb.org/

```
C8 <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                        enrichDatabaseFile = "GSEA_custom_sets/c8.all.v7.5.1.symbols.gmt",
                        enrichDatabaseType = "genesymbol",
                        interestGene = res1,
                        interestGeneType = "genesymbol",
                        isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(C8)
```

```
## [1] 95
```

```
C8 <- C8[order(C8$normalizedEnrichmentScore, decreasing = TRUE),]
head(subset(C8, select=c("geneSet","normalizedEnrichmentScore","pValue","FDR","size")))
```

```
##                                                       geneSet
## 1                          FAN_OVARY_CL14_MATURE_SMOOTH_MUSCLE_CELL
## 2 DURANTE_ADULT_OLFACTORY_NEUROEPITHELIUM_FIBROBLASTS_STROMAL_CELLS
## 5                          GAUTAM_EYE_IRIS_CILIARY_BODY_FIBROBLASTS
## 4                               AIZARANI_LIVER_C21_STELLATE_CELLS_1
## 3                       BUSSLINGER_ESOPHAGEAL_LATE_SUPRABASAL_CELLS
## 6                        RUBENSTEIN_SKELETAL_MUSCLE_SATELLITE_CELLS
##   normalizedEnrichmentScore pValue          FDR size
## 1                  2.358992      0 0.0000000000  241
## 2                  2.348244      0 0.0000000000   61
## 5                  2.274747      0 0.0007788294   70
## 4                  2.238980      0 0.0005841220  147
## 3                  2.237084      0 0.0004672976   88
## 6                  2.221168      0 0.0007788294  254
```

Note that this one only has cell type pathways upregulated in the polyA, no cell types that are more upregulated in the ribo-depleted.

```
C8_05 <- suppressWarnings(WebGestaltR(enrichMethod = "GSEA",
                        enrichDatabaseFile = "GSEA_custom_sets/c8.all.v7.5.1.symbols.gmt",
                        enrichDatabaseType = "genesymbol",
                        interestGene = res05,
                        interestGeneType = "genesymbol",
                        isOutput = FALSE))
```

```
## Loading the functional categories...
## Loading the ID list...
## Performing the enrichment analysis...
## 1000 permutations of score complete...
```

```
nrow(C8_05)
```

```
## [1] 89
```

```
C8_05 <- C8_05[order(C8_05$normalizedEnrichmentScore, decreasing = TRUE),]
head(subset(C8_05, select=c("geneSet","normalizedEnrichmentScore","pValue","FDR","size")))
```

```
##                                                                geneSet
## 1                                TRAVAGLINI_LUNG_PROXIMAL_BASAL_CELL
## 2 DURANTE_ADULT_OLFACTORY_NEUROEPITHELIUM_FIBROBLASTS_STROMAL_CELLS
## 3                             GAUTAM_EYE_IRIS_CILIARY_BODY_FIBROBLASTS
## 4                          BUSSLINGER_ESOPHAGEAL_LATE_SUPRABASAL_CELLS
## 5                           RUBENSTEIN_SKELETAL_MUSCLE_SATELLITE_CELLS
## 6                                RUBENSTEIN_SKELETAL_MUSCLE_NK_CELLS
##   normalizedEnrichmentScore pValue FDR size
## 1                  2.499236      0   0  442
## 2                  2.458719      0   0   59
## 3                  2.403456      0   0   68
## 4                  2.331475      0   0   85
## 5                  2.320020      0   0  249
## 6                  2.319095      0   0  170
```

Same here, a lot of smooth muscle cells and fibroblast-looking things. Not sure how confident I am in this gene set though.

Conclusions

The genes that are most upregulated in the rRNA depletion are the genes encoding histones and "RNAs with jobs", like telomerase RNA and endoribonucleases. I'm not totally sure why, my quick literature review seems to indicate they should still be polyadenylated, but perhaps not.

The genes that are most upregulated in the poly-A capture are mitochondrial genes (still need to look into why that is), and ribosomal-associated genes (not surprising).