RNA-Seq workflow: Thyroid analysis within 3 type of infiltration

Análisis de datos ómicos PEC2

Ana Isabel del Val

7 de junio, 2020

Contents

Abstract

The dataset for the exercise comes from the repository GTEx and has been provided by UOC. This repository contains data of multiple types out of 54 tissues. We focus on RNA-seq expression data of a thyroid analysis, where 3 type of infiltration is compared in 292 samples.

The distribution of the samples is the following: - Not infiltrated tissues (NIT): 236 samples - Small focal infiltrates (SFI): 42 samples - Extensive lymphoid infiltrates (ELI): 14 samples

We already have the preprocessed data in the count matrix, *counts.csv* file.

The distribution of samples are stored in *targets.csv*, where we select 30 samples (10 of each type). These 30 samples will subset the columns of the count matrix, having rows of targets and columns of counts in the same order.

This code is already sync with ***github repository***, which is public.

This is the Bioconductor help.

Objectives

It consists on the differential expression analysis, comparing 3 type of Group samples:

- NIT-SFI
- NIT-ELI
- SFI-ELI

Materials and methods

Data source and experiment design

We count on 292 samples of 3 types out of 54 tissues. We select 30 samples out of 292, equally distributed by group.

- 10 samples of group NIT
- 10 samples of group ELI
- 10 samples of group SFI

Data has 2 molecular data type:

- RNA-Seq (NGS)
- Allele-Specific Expression

If we filter by RNA-Seq (NGS) only, there are only 8 samples of ELI group. Consequently, we have consider the two molecular data types to get the random samples.

Methods and Tools

R and Bioconductor have been the tools selected to follow the RNA-seq pipeline:

- Data gathering
- Data preprocessing
- Data filtering and transformations
- Differential expression analysis
- Results comparisons
  - NIT_SFI
  - NIT_ELI
  - SFI_ELI
- Results annotations
  - NIT_SFI
  - NIT_ELI
  - SFI_ELI
- Remove unwanted variations

Results

The data preparation is crutial for obtaining good results. These are the highlights:

- The order of rows in targets is the same as the order of columns in counts.
- Remove ENSEMBLE version from genes.
- Standardize naming convention for samples (be careful not to change "-" by "." in columns of counts when loading data).
- The seed is fixed to "1234"to ensure results reproducibility.

In the definition of DESeqDataSet object, we specify the variable "Group" to test for its effect in the experiment.

```
ddsMatrix
```

```
## class: DESeqDataSet
## dim: 56202 30
## metadata(1): version
## assays(1): counts
## rownames(56202): ENSG00000223972 ENSG00000227232 ... ENSG00000210195
##   ENSG00000210196
## rowData names(0):
## colnames(30): GTEX-11EQ9-0626-SM-5A5K1 GTEX-11NV4-0626-SM-5N9BR ...
##   GTEX-ZYVF-1126-SM-5E458 GTEX-ZYY3-1926-SM-5GZXS
## colData names(9): Experiment SRA_Sample ... Group ShortName
```

Our count matrix with our DESeqDataSet contains many rows with only zeros, and additionally many rows with only a few fragments total. In order to reduce the size of the object, and to increase the speed of our functions, we can remove the rows that have no or nearly no information about the amount of gene expression. Here we apply the most minimal filtering rule: removing rows of the DESeqDataSet that have no counts, or only a single count across all samples.

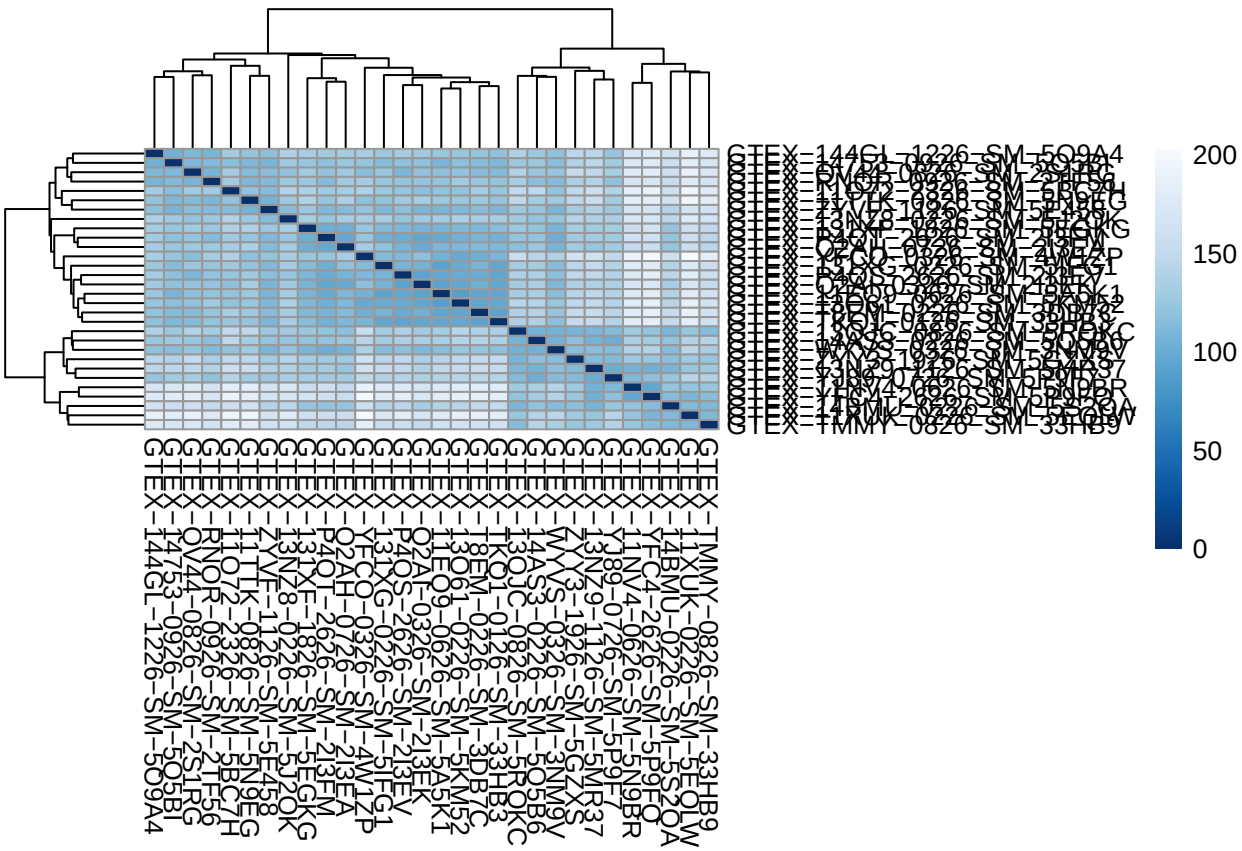
```
ddsMatrix <- ddsMatrix[ rowSums(counts(ddsMatrix)) > 1, ]
nrow(ddsMatrix)
```

```
## [1] 44154
```

DESeq2 offers two transformations for count data that stabilize the variance across the mean: the regularized-logarithm transformation or rlog and the variance stabilizing transformation (VST) for negative binomial data with a dispersion-mean trend. In this case we choose rlog because the number of samples is 30, and rlog tends to work well with small datasets.

Now we assess the sample distances, which is the overall similarity between samples. The distance function expect samples in rows and genes in columns, that's why we traspose the matrix coming from rlog transformation. We use the heatmap plot to see the sample distances.

```
sampleDists <- dist(t(assay(rldM)))
sampleDistMatrix <- as.matrix(sampleDists)
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```



When performing the differential expression analysis and later comparisons between groups, we consider a fraction of 10% false positives acceptable so we can consider all genes with an adjusted p value below 10% = 0.1 as significant.

These are the significant genes for the 3 comparisons, corresponding to NIT-SFI, NIT-ELI and SFI-ELI, respectively. There is a huge difference between the first comparison and the other 2.

```
sum(resNIT_SFI$padj < 0.1, na.rm=TRUE)
```

```
## [1] 103
```

```
sum(resNIT_ELI$padj < 0.1, na.rm=TRUE)
```

```
## [1] 7181
```

```
sum(resSFI_ELI$padj < 0.1, na.rm=TRUE)
```

```
## [1] 7722
```

We proceed with the results annotation. The columns $symbol and $entrez could lead us to NA values. The most justified reason is that there could be transcripts not associated with any specific gene.

```
head(resNIT_SFIOrdered)
```

```
##                  baseMean log2FoldChange     lfcSE      stat       pvalue         padj
## ENSG00000240382  735.4841      -4.634872 0.8381213 -5.530073 3.200981e-08 0.0003618729
## ENSG00000211976  227.8900      -5.277783 0.9579072 -5.509701 3.594428e-08 0.0003618729
```

```
## ENSG00000211619  104.0940      -7.675897 1.4032325 -5.470153 4.496461e-08 0.0003618729
## ENSG00000211611  246.9306      -6.434839 1.1818633 -5.444656 5.190561e-08 0.0003618729
## ENSG00000211640 1404.3847      -4.689035 0.8958123 -5.234395 1.655264e-07 0.0009232072
## ENSG00000244116 1106.2802      -3.799348 0.7562107 -5.024192 5.055556e-07 0.0023497381
##                 symbol entrez
## ENSG00000240382   <NA>   <NA>
## ENSG00000211976   <NA>   <NA>
## ENSG00000211619   <NA>   <NA>
## ENSG00000211611   <NA>   <NA>
## ENSG00000211640   <NA>   <NA>
## ENSG00000244116   <NA>   <NA>
```

```
head(resNIT_ELIOrdered)
```

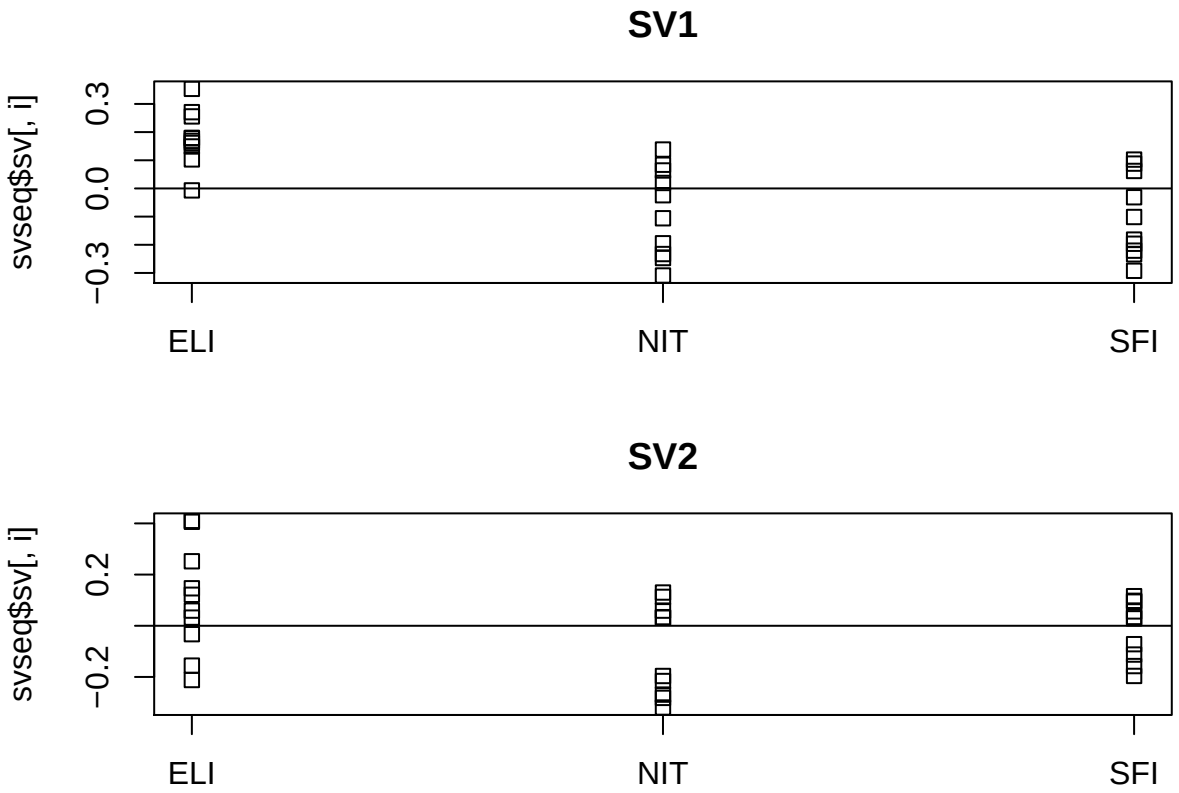
```
##                 baseMean log2FoldChange     lfcSE      stat       pvalue         padj
## ENSG00000177455 1431.298      -7.787651 0.4982691 -15.62941 4.590494e-55 1.476624e-50
## ENSG00000156738 6438.926      -7.746587 0.5091559 -15.21457 2.830843e-52 4.552986e-48
## ENSG00000167483 1050.943      -7.221992 0.5053272 -14.29172 2.464642e-46 2.642672e-42
## ENSG00000173200 2126.504      -4.614987 0.3270762 -14.10982 3.304027e-45 2.657016e-41
## ENSG00000247982 2020.987      -4.135274 0.2965381 -13.94517 3.366477e-44 2.165789e-40
## ENSG00000104894 3273.531      -4.444171 0.3204725 -13.86756 9.960959e-44 5.340236e-40
##                    symbol entrez
## ENSG00000177455      CD19    930
## ENSG00000156738     MS4A1    931
## ENSG00000167483    NIBAN3 199786
## ENSG00000173200    PARP15 165631
## ENSG00000247982 LINC00926 283663
## ENSG00000104894      CD37    951
```

```
head(resSFI_ELIOrdered)
```

```
##                  baseMean log2FoldChange     lfcSE      stat       pvalue         padj
## ENSG00000177455 1431.2980      -7.138178 0.4942964 -14.44109 2.853040e-47 9.177375e-43
## ENSG00000156738 6438.9261      -6.992646 0.5082089 -13.75939 4.472699e-43 7.193665e-39
## ENSG00000104894 3273.5312      -4.344282 0.3204215 -13.55803 7.102263e-42 7.615283e-38
## ENSG00000172794  513.3941      -3.770751 0.2837426 -13.28934 2.669318e-40 2.146599e-36
## ENSG00000136573 1338.9039      -6.208734 0.4702047 -13.20432 8.284410e-40 5.329693e-36
## ENSG00000173200 2126.5039      -4.284518 0.3268023 -13.11043 2.869692e-39 1.538490e-35
##                 symbol entrez
## ENSG00000177455   CD19    930
## ENSG00000156738  MS4A1    931
## ENSG00000104894   CD37    951
## ENSG00000172794  RAB37 326624
## ENSG00000136573    BLK    640
## ENSG00000173200 PARP15 165631
```

It's time to remove hidden batch effects causing unwanted variations. This will detect the source of variation correlated with Group.

```
par(mfrow = c(2, 1), mar = c(3,5,3,1))
for (i in 1:2) {
  stripchart(svseq$sv[, i] ~ ddsMatrix$Group, vertical = TRUE, main = paste0("SV", i))
  abline(h = 0)
}
```



We could then produce results by running DESeq with the new design, incorporating the surrogate variables.

Discussion

I have found a limitation with the first comparison: NIT_SFI. When getting the significant genes, we only get around 1.5% from Comparison 1 with respecto to Groups 2 and 3. So the results for the first comparison could be less accurate.

We could have plotted a lot of visualizations, but as we discussed in the class forum, there is no point in representing the same with different methods. This way, only heatmap has been chosen for sample distances.