Homework 3

Lev Mazaev

October 22, 2019

Let's load the data file.

```
load(file = 'GSE23878_norm.dat')
```

We have to extract expression matrix and metadata for each sample/feature.

```
m <- Biobase::exprs(eset) # expression matrix
md <- Biobase::pData(eset) # metadata for each sample
fd <- Biobase::fData(eset) # metadata for each feature (gene)
nrow(m); ncol(m); nrow(md); ncol(md)
```

```
## [1] 20183
```

```
## [1] 58
```

```
## [1] 58
```

```
## [1] 32
```

Let's see if the expression data is normalized.

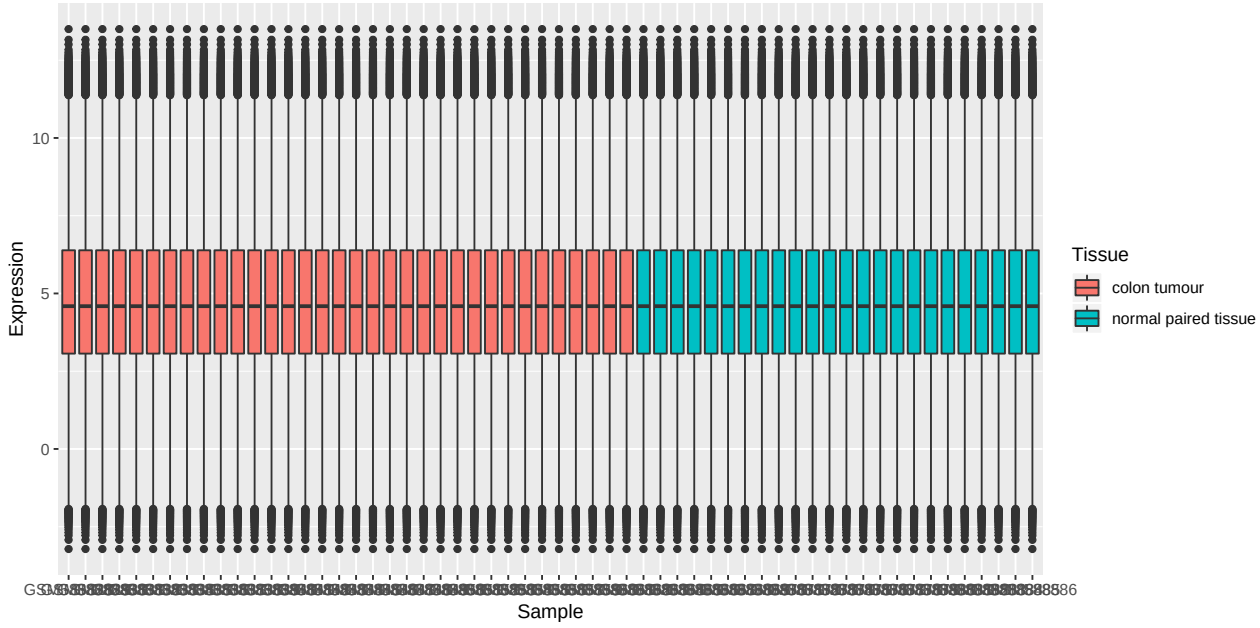
```
colSums(m)
```

```
## GSM588828 GSM588829 GSM588830 GSM588831 GSM588832 GSM588833 GSM588834
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588835 GSM588836 GSM588837 GSM588838 GSM588839 GSM588840 GSM588841
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588842 GSM588843 GSM588844 GSM588845 GSM588846 GSM588847 GSM588848
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588849 GSM588850 GSM588851 GSM588852 GSM588853 GSM588854 GSM588855
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588856 GSM588857 GSM588859 GSM588860 GSM588861 GSM588862 GSM588863
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588864 GSM588865 GSM588866 GSM588867 GSM588868 GSM588869 GSM588870
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588871 GSM588872 GSM588873 GSM588874 GSM588875 GSM588876 GSM588877
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588878 GSM588879 GSM588880 GSM588881 GSM588882 GSM588883 GSM588884
##  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95  95354.95
## GSM588885 GSM588886
##  95354.95  95354.95
```

At least the sum along each column is the same. What would show the boxplots?

```
mt <- as_tibble(m)
mt$gene <- rownames(m)
mt <- mt %>% pivot_longer(cols = starts_with('GSM'),
                    names_to = 'Sample',
                    values_to = 'Expression')
mt <- mt %>% left_join(y = md[, c(2, 8)], by =c('Sample' = 'geo_accession' ))
colnames(mt)[4] <- 'Tissue'
ggplot(data = mt, mapping = aes(x = Sample, y = Expression,
```

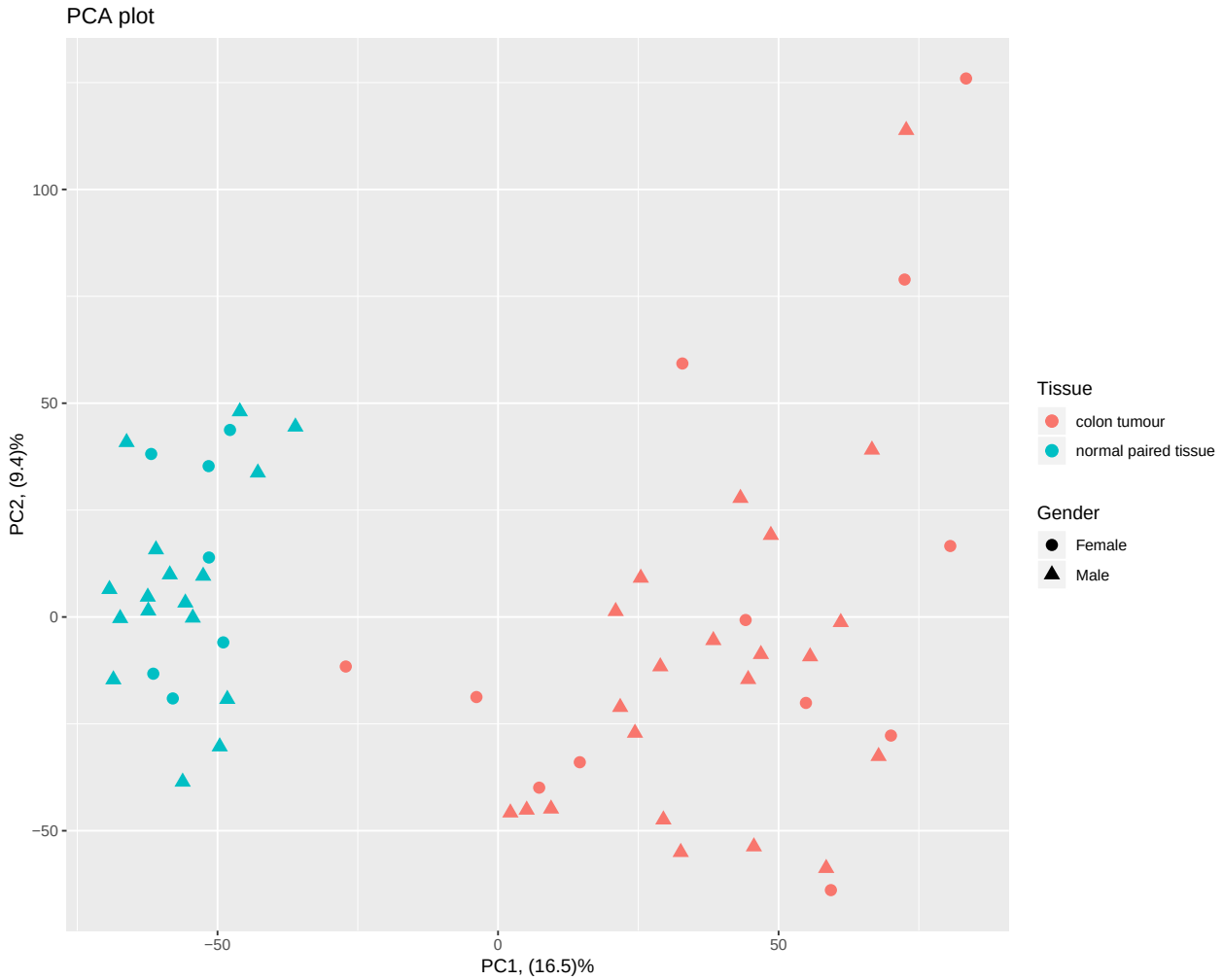
```
                                fill = Tissue)) +
  geom_boxplot()
```



Here are 58 boxplots, one for each sample. As we can see they all are well aligned, so the normalization is OK.

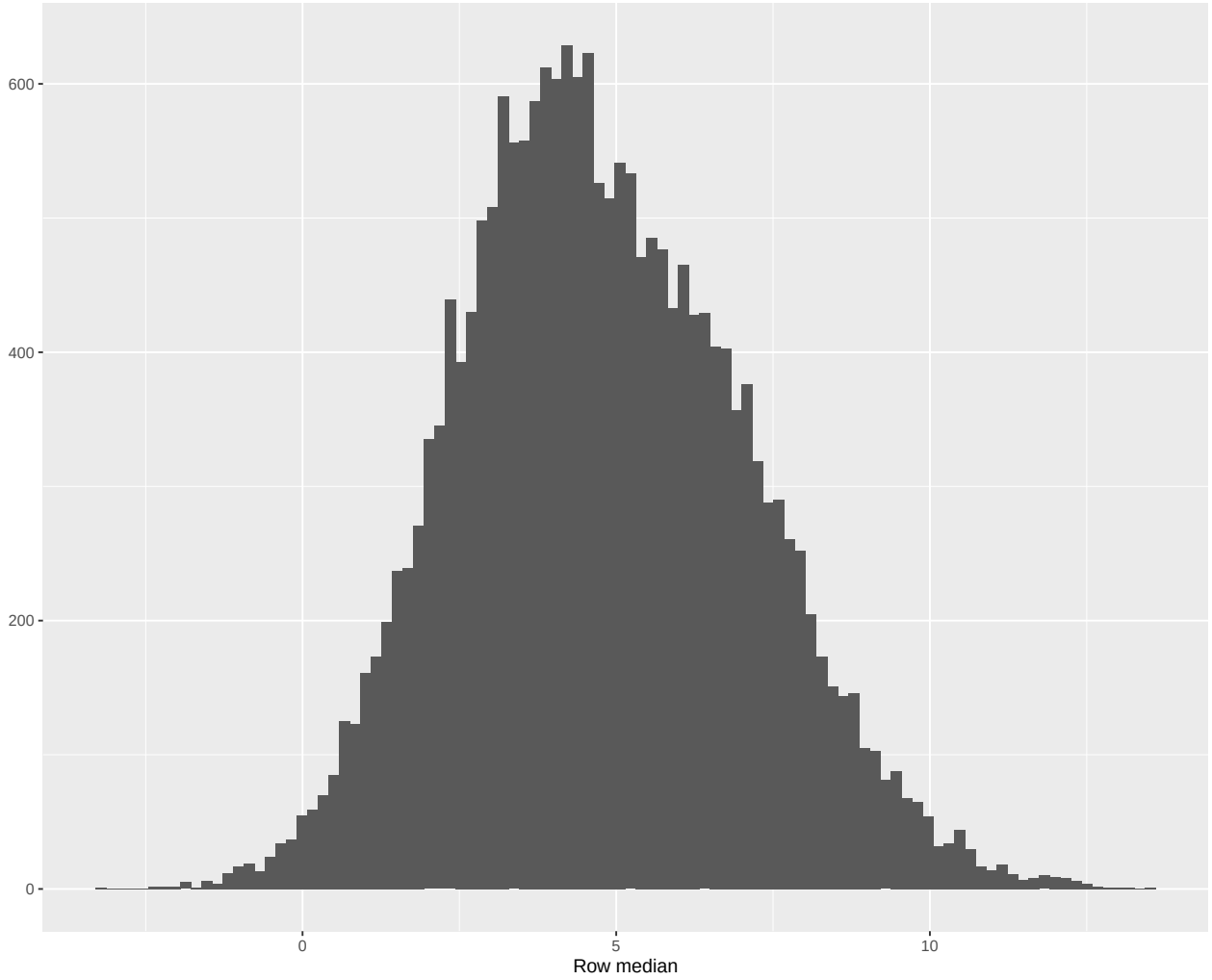
Next we have to find genes differentially expressed between tissue types. In order to see something let's do the PCA at first.

```
pca <- prcomp(t(m), scale. = FALSE)
pv <- round(100 * pca$sdev^2 / sum(pca$sdev^2), 1)
data_pca <- tibble(PC1 = pca$x[, 1], PC2 = pca$x[, 2],
                   Tissue = md$source_name_ch1,
                   Gender = md$`gender:ch1`)
ggplot(data = data_pca, mapping = aes(x = PC1, y = PC2,
                                      color = Tissue, shape = Gender)) +
  geom_point(size = 3) + ggtitle('PCA plot') +
  xlab(paste0('PC1, (', pv[1], ')%')) +
  ylab(paste0('PC2, (', pv[2], ')%'))
```



It's obvious that differential expression between the tissue types is the dominant source of variation. Gender is just for joke, it definitely does not convey any message. Let's see if we need to filter some probes (genes).

```
row_meds <- Biobase::rowMedians(m)
qplot(row_meds, bins = 100, xlab = 'Row median')
```

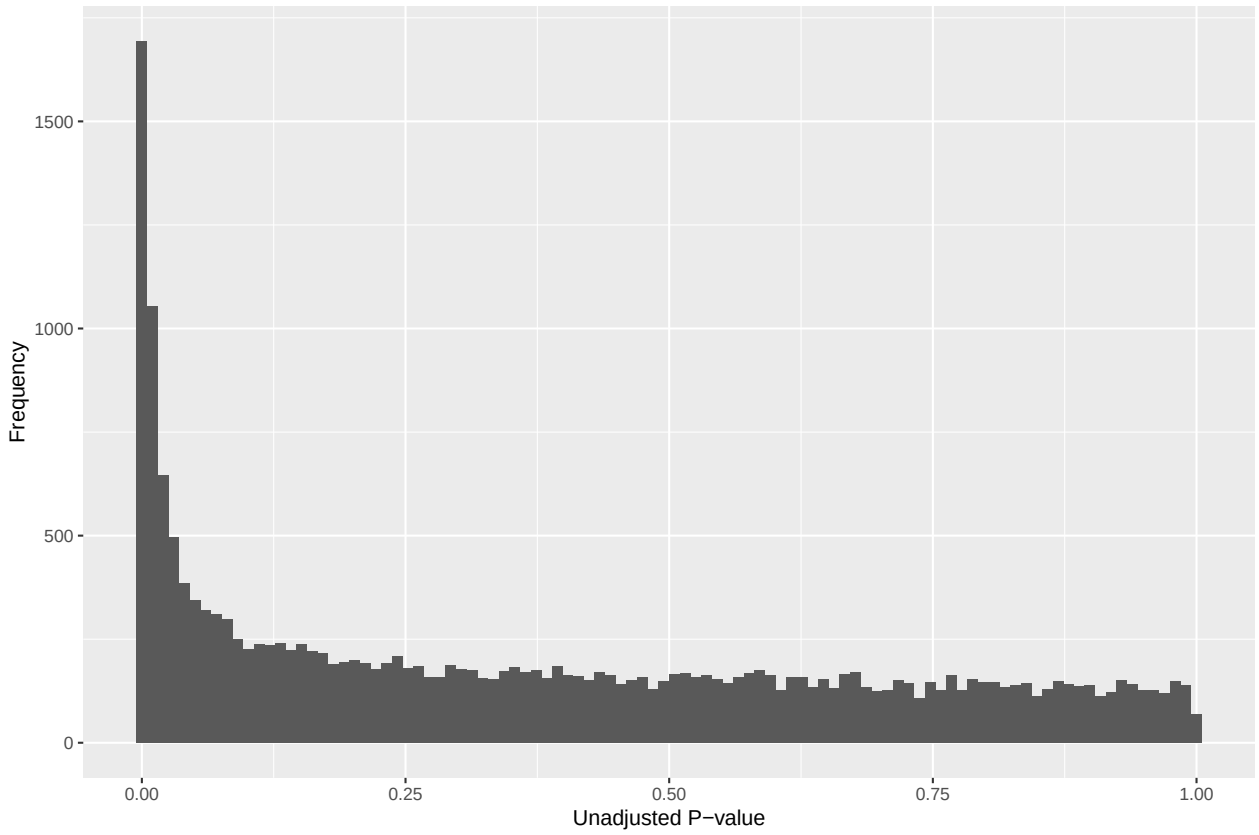


I think we don't need any filtering. Now let's use GLM with negative-binomial family to find some differentially expressed genes (I used *limma* at first, but it strangely categorized all genes as DE).

```
library(MASS)
glm.control(maxit = 100)
```

```
## $epsilon
## [1] 1e-08
##
## $maxit
## [1] 100
##
## $trace
## [1] FALSE
```

```
transM <- as_tibble(t(m))
tum <- startsWith(as.vector(md$source_name_ch1), 'colon')
pvals <- sapply(transM, function(x) return(anova(glm.nb(tum ~ x))$`Pr(>Chi)`[2]))
qplot(pvals, xlab = 'Unadjusted P-value', ylab = 'Frequency', bins = 100)
```



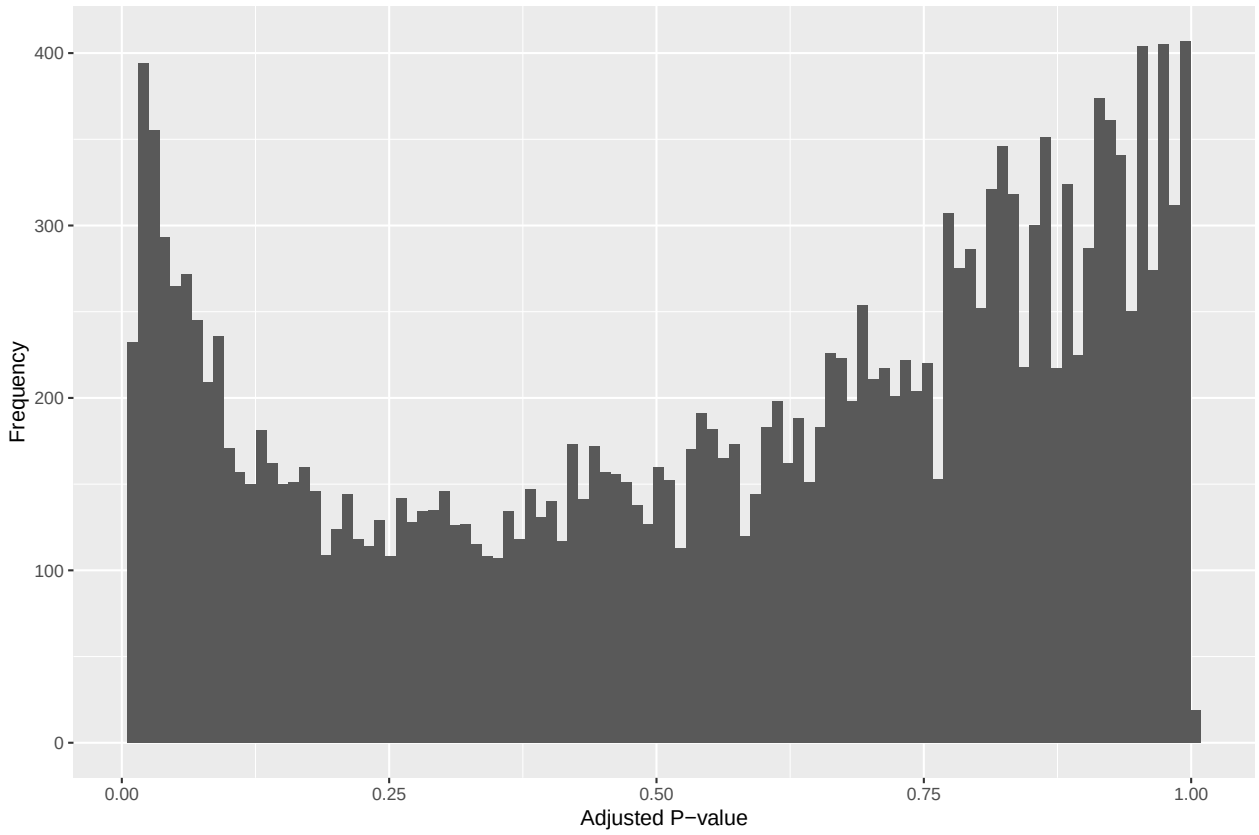
top10 genes:

```
head(fd$`Gene Symbol`[order(pvals)], n = 10)
```

```
##  [1] "CDH3"  "ETV4"  "LGI1"  "ESM1"  "PLP1"  "FOXQ1" "CEMIP" "EPHX4"
##  [9] "MYOT"  "CA7"
```

Now let's adjust p-values for multiple hypothesis testing.

```
apvals <- p.adjust(pvals, method = 'BH')
qplot(apvals, xlab = 'Adjusted P-value', ylab = 'Frequency', bins = 100)
```




```
alpha <- 0.05
```

Number of genes passing the threshold $FDR < 0.05$ is 1421. Top 10 according to adjusted p-values:

```
head(fd$`Gene Symbol`[order(apvals)], n = 10)
```

```
##  [1] "CDH3"   "ETV4"   "LGI1"   "ESM1"   "PLP1"   "EPHX4"  "MAMDC2"
##  [8] "HILPDA" "CEMIP"  "CA7"
```

```
geneList <- tibble(
  ProbeID = fd$ID,
  GeneSymbol = fd$`Gene Symbol`,
  EntrezID = fd$ENTREZ_GENE_ID,
  Pvalue = pvals,
  AdjPvalue = apvals
)
siGenes <- geneList[geneList$AdjPvalue < alpha, ]
knitr::kable(head(arrange(siGenes, AdjPvalue), n=10))
```

ProbeID	GeneSymbol	EntrezID	Pvalue	AdjPvalue
203256_at	CDH3	1001	3.0e-07	0.0054964
211603_s_at	ETV4	2118	8.0e-07	0.0058503
206349_at	LGI1	9211	9.0e-07	0.0058503
208394_x_at	ESM1	11082	1.2e-06	0.0062634
210198_s_at	PLP1	5354	1.6e-06	0.0063831
239579_at	EPHX4	253152	3.5e-06	0.0079901
228885_at	MAMDC2	256691	4.4e-06	0.0079901
218507_at	HILPDA	29923	4.8e-06	0.0079901

ProbeID	GeneSymbol	EntrezID	Pvalue	AdjPvalue
212942_s_at	CEMIP	57214	3.3e-06	0.0079901
207504_at	CA7	766	4.3e-06	0.0079901

Performing enrichment analysis.

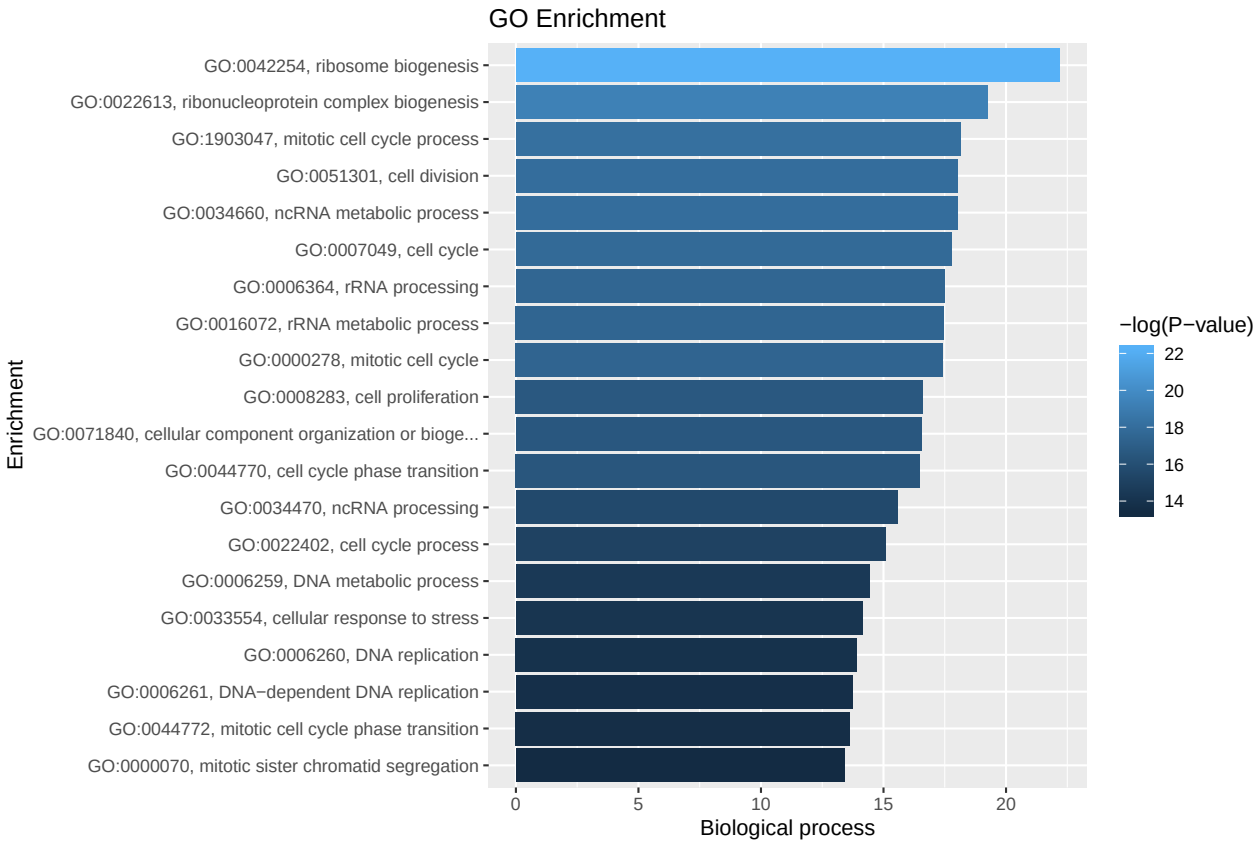
```
library(topGO)
gl <- apvals
names(gl) <- fd$ENTREZ_GENE_ID
setGO <- new('topGOdata', description = 'new session', ontology = 'BP',
             allGenes = gl, geneSel = function(x) return(x < alpha),
             nodeSize = 10,
             annot = annFUN.org,
             mapping = 'org.Hs.eg.db',
             ID = 'entrez')
# 3 different test
resultKS.elim <- runTest(setGO, algorithm = 'elim', statistic = 'ks')
resultKS.classic <- runTest(setGO, algorithm = 'classic', statistic = 'ks')
resultFisher <- runTest(setGO, algorithm = "classic", statistic = "fisher")
```

```
goEnrichment <- GenTable(setGO, KS=resultKS.classic, orderBy = 'KS',
                         topNodes = 20)
knitr::kable(goEnrichment)
```

GO.ID	Term	Annotated	Significant	Expected	KS
GO:0042254	ribosome biogenesis	189	37	15.96	2.3e-10
GO:0022613	ribonucleoprotein complex biogenesis	320	51	27.02	4.4e-09
GO:1903047	mitotic cell cycle process	744	125	62.83	1.3e-08
GO:0034660	ncRNA metabolic process	437	70	36.90	1.5e-08
GO:0051301	cell division	545	100	46.02	1.5e-08
GO:0007049	cell cycle	1585	204	133.84	1.9e-08
GO:0006364	rRNA processing	142	30	11.99	2.5e-08
GO:0016072	rRNA metabolic process	176	36	14.86	2.6e-08
GO:0000278	mitotic cell cycle	854	133	72.11	2.7e-08
GO:0008283	cell proliferation	1755	200	148.20	6.1e-08
GO:0071840	cellular component organization or bioge...	5505	518	464.86	6.4e-08
GO:0044770	cell cycle phase transition	548	93	46.27	6.9e-08
GO:0034470	ncRNA processing	273	51	23.05	1.7e-07
GO:0022402	cell cycle process	1178	165	99.47	2.8e-07
GO:0006259	DNA metabolic process	893	114	75.41	5.4e-07
GO:0033554	cellular response to stress	1774	177	149.80	7.3e-07
GO:0006260	DNA replication	247	44	20.86	9.0e-07
GO:0006261	DNA-dependent DNA replication	123	28	10.39	1.1e-06
GO:0044772	mitotic cell cycle phase transition	508	91	42.90	1.2e-06
GO:0000070	mitotic sister chromatid segregation	128	34	10.81	1.5e-06

```
goEnrichment$ExtTerm <- paste(goEnrichment$GO.ID, goEnrichment$Term, sep = ', ')
goEnrichment$KS <- as.numeric(gsub(',', '.', goEnrichment$KS))
ggplot(data = goEnrichment, aes(x = reorder(ExtTerm, -KS), y = -log(KS))) +
  geom_col(aes(fill = -log(KS))) + coord_flip() +
  scale_fill_gradient() +
  xlab('Enrichment') +
```

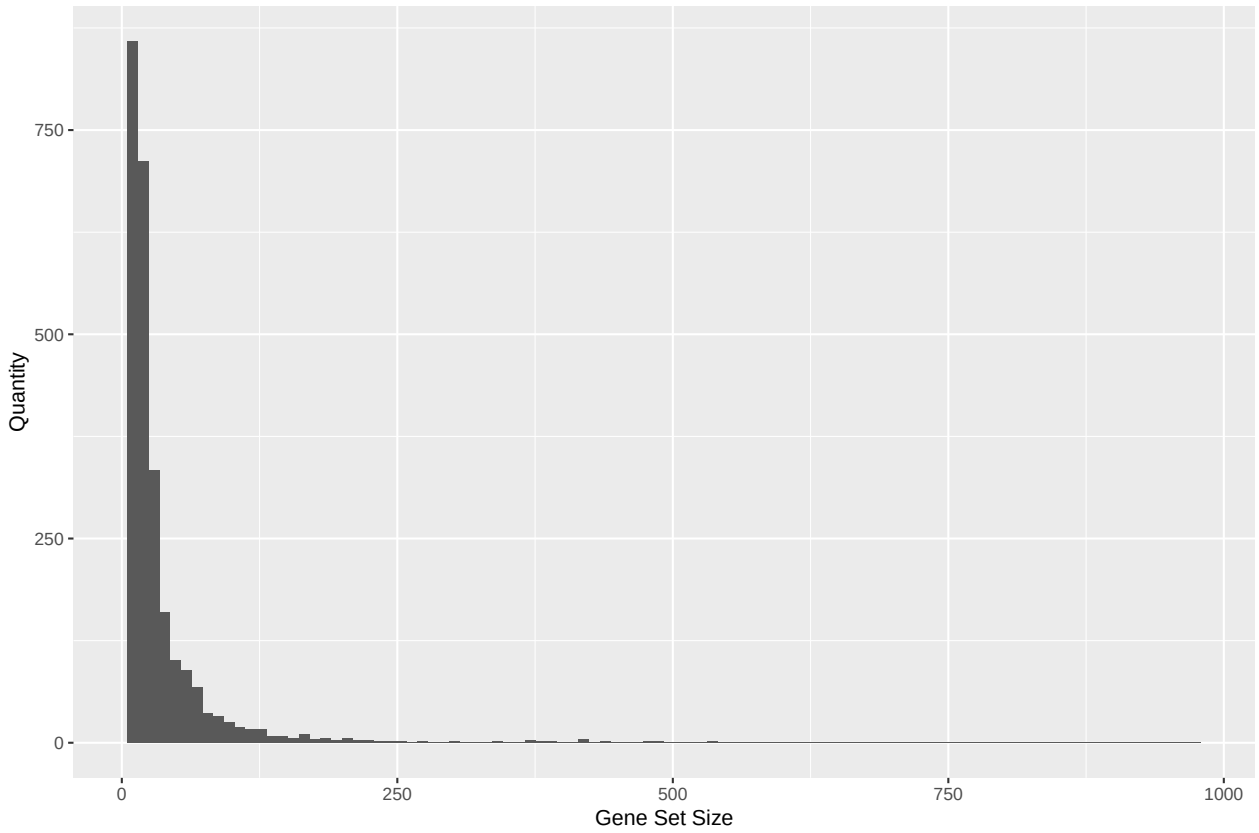
```
  ylab('Biological process') +
  ggtitle('GO Enrichment') +
  labs(fill = '-log(P-value)')
```



Overrepresentation analysis.

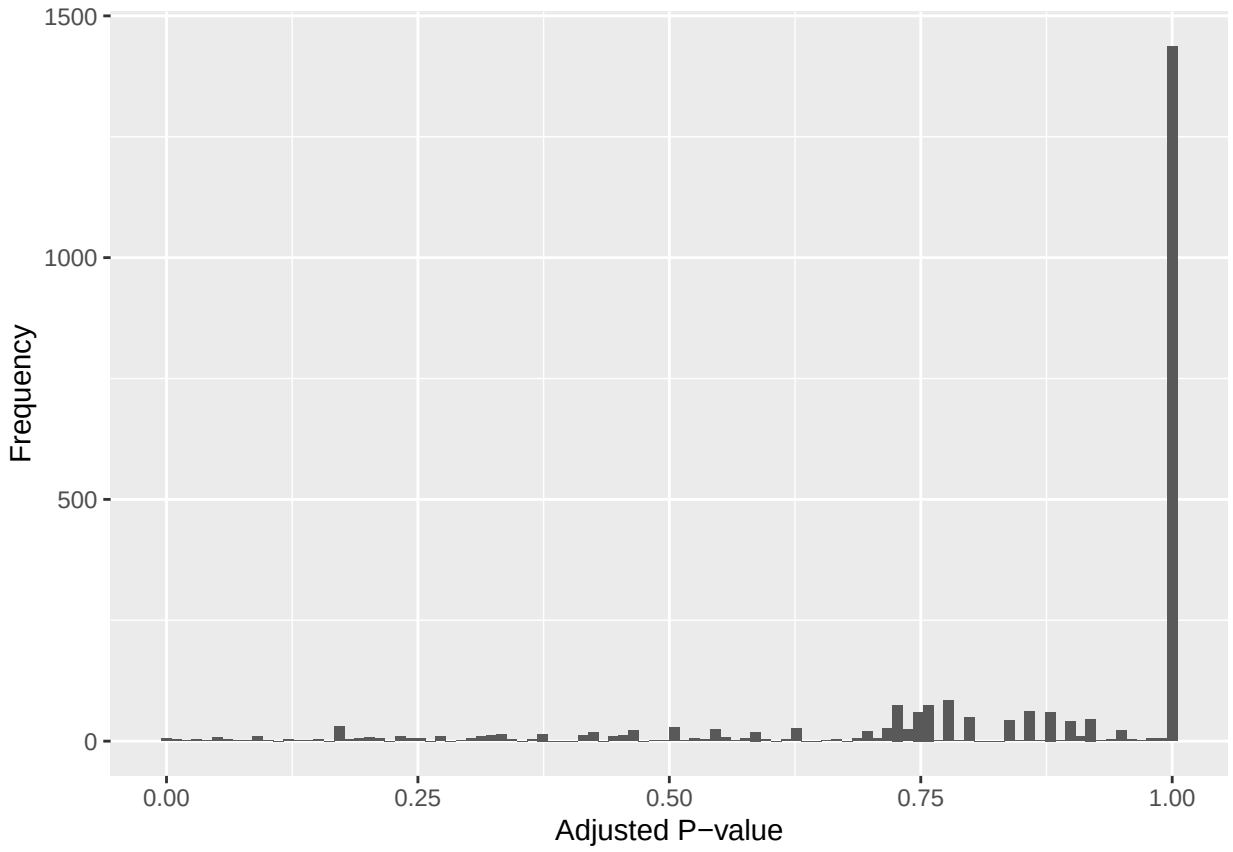
Let's filted filter out small (<10) gene sets. Also let's remove all genes not represented in at least one gene set. In parallel I'm doing overrepresentation analysis using WebGestalt (using **affy hg u133 plus 2**). Later I'll compare it with manual Fisher Test.

```
interest <- siGenes$EntrezID
geneSets <- annFUN.org("BP", mapping = "org.Hs.eg.db", ID = "entrez")
geneSets <- geneSets[lengths(geneSets) >= 10]
genePool <- purrr::reduce(geneSets, union)
interest <- interest[interest %in% genePool]
noninterest <- genePool[!genePool %in% interest]
qplot(lengths(geneSets), bins = 100, xlab = 'Gene Set Size', ylab = 'Quantity')
```




```
write.table(interest, 'genes.txt', quote = FALSE, row.names = FALSE, col.names = FALSE)
```

```
fpvals <- sapply(geneSets, function(gs) {
  gs <- unlist(gs)
  a11 <- length(intersect(interest, gs))
  a12 <- length(setdiff(interest, gs))
  a21 <- length(intersect(noninterest, gs))
  a22 <- length(setdiff(noninterest, gs))
  # print(cbind(c(a11, a21), c(a12, a22)))
  return(fisher.test(cbind(c(a11, a21), c(a12, a22)), alternative = 'greater')$p.value)
})
afpvals <- p.adjust(fpvals, method = 'BH')
qplot(afpvals, xlab = 'Adjusted P-value', ylab = 'Frequency', bins = 100)
```




```
localFisherGS <- names(afpvals[afpvals < alpha])
webGestalt <- read.table(file = 'enrichment_results_wg_result1572007005.txt',
                         header = TRUE, sep = '\t')
webGestaltGS <- webGestalt$geneSet
ksGS <- goEnrichment$GO.ID
```

Let's draw Venn diagram.

```
library(VennDiagram)
venn.diagram(
  x = list(localFisherGS, webGestaltGS, ksGS),
  category.names = c('Manual Fisher Test',
                     'WebGestalt Service',
                     'KS Test from TopGO'),
  filename = 'venn.png',
  output = TRUE,
          imagetype="png",
          height = 800,
          width = 800,
          resolution = 300,
          compression = "lzw",
          lwd = 1,
          col=c("#440154ff", '#21908dff', '#fde725ff'),
          fill = c(alpha("#440154ff",0.3), alpha('#21908dff',0.3), alpha('#fde725ff',0.3)),
          cex = 0.5,
          fontfamily = "sans",
          cat.cex = 0.3,
```

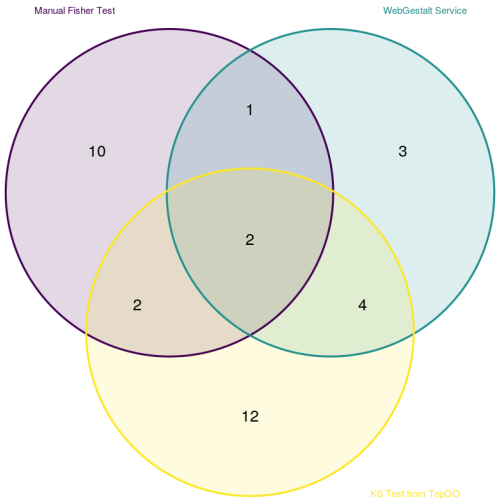


Figure 1: A caption

```
          cat.default.pos = "outer",
          cat.pos = c(-27, 27, 135),
          cat.dist = c(0.055, 0.055, 0.085),
          cat.fontfamily = "sans",
          cat.col = c("#440154ff", '#21908dff', '#fde725ff'),
          rotation = 1
)
```

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

```
knitr::include_graphics('venn.png')
```

And one more pic.

```
my_set <- tibble(
  GO_ID = names(afpvals[afpvals < alpha]),
  ManualFisherTest = afpvals[afpvals < alpha]
)
wg <- tibble(
  GO_ID = as.vector(webGestalt$geneSet),
  WebGestalt = webGestalt$pValue + 1e-16,
)
df <- my_set %>% full_join(wg, by = 'GO_ID')
df <- df %>% pivot_longer(cols = 2:3, names_to = 'Test', values_to = 'Pval')
ggplot(df, mapping = aes(x = reorder(GO_ID, -log(Pval)), y = -log(Pval), fill = Test)) +
  geom_col(position = 'dodge', na.rm = TRUE) + coord_flip() +
  xlab('Enrichment') + ylab('Gene Ontology Category') +
  theme_bw()
```