Supplement: Splice Expression Variation Analysis (SEVA): Variability Analysis to Detect Significant Alternative Splicing Events (TCGA Analysis only)

Bahman Afsari

Sunday, June 12, 2016

1 Loading the real data

First, we load the data as:

```
library('Homo.sapiens')
library('org.Hs.eg.db')
library('GenomicRanges')
library("GSReg")
library(EBSeq)
library(limma)
library('gplots')
library(ggplot2)
library('ROCR')
library(Matrix)
```

2 Cross-study Validation with TCGA

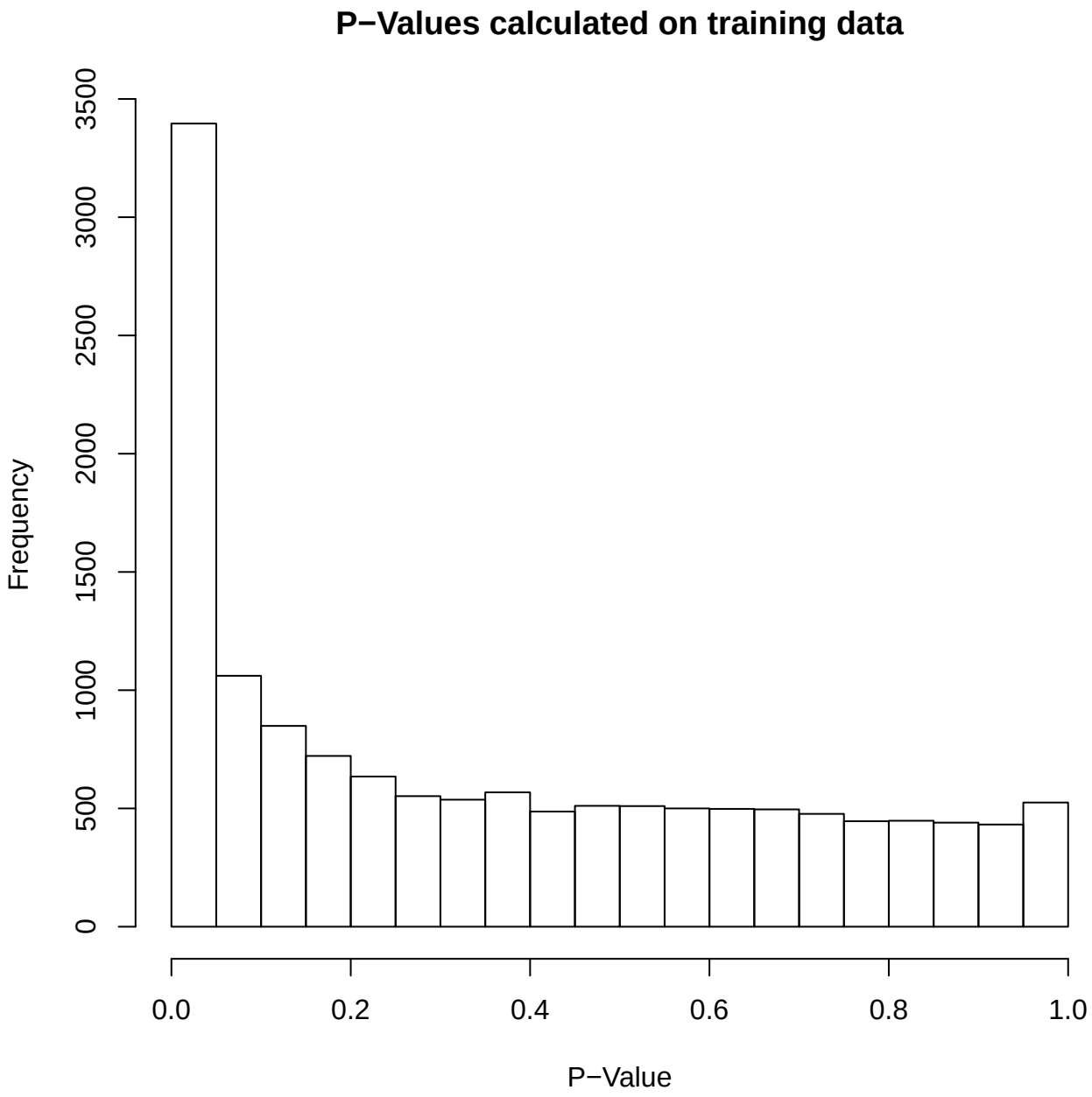
Now, we cross-study the genes we identified using TCGA as the tes-set. (For memory issues we run SEVA on the batches of 1000 genes and augmented the results)

```
#######################################33
### TCGA
##
### loading data
#########################



load("../Data/TCGA/HPVPosTCGAJuncRPM_25Aug2015_Pheno.Rda")
load("../Data/TCGA/TCGA_RSEM_processed_091615.RDa")
load(file = "../Cache/ForTCGAAnalysis.rda")
source("../Scripts/functions.R") #loading the functions for analysis

TCGA.RSEM <- as.matrix(TCGA.RSEM)

#We divide the TCGA data to batches of 1000 genes to make it manageable
junctionPValueTCGAaug <- c()
for( i in 1:14){

  # junctionPValueTCGA <- GSReg.SEVA(juncExprs = junc.RPM.TCGA,
  #                                                                    phenoVect=as.factor(phenoVect.T
```

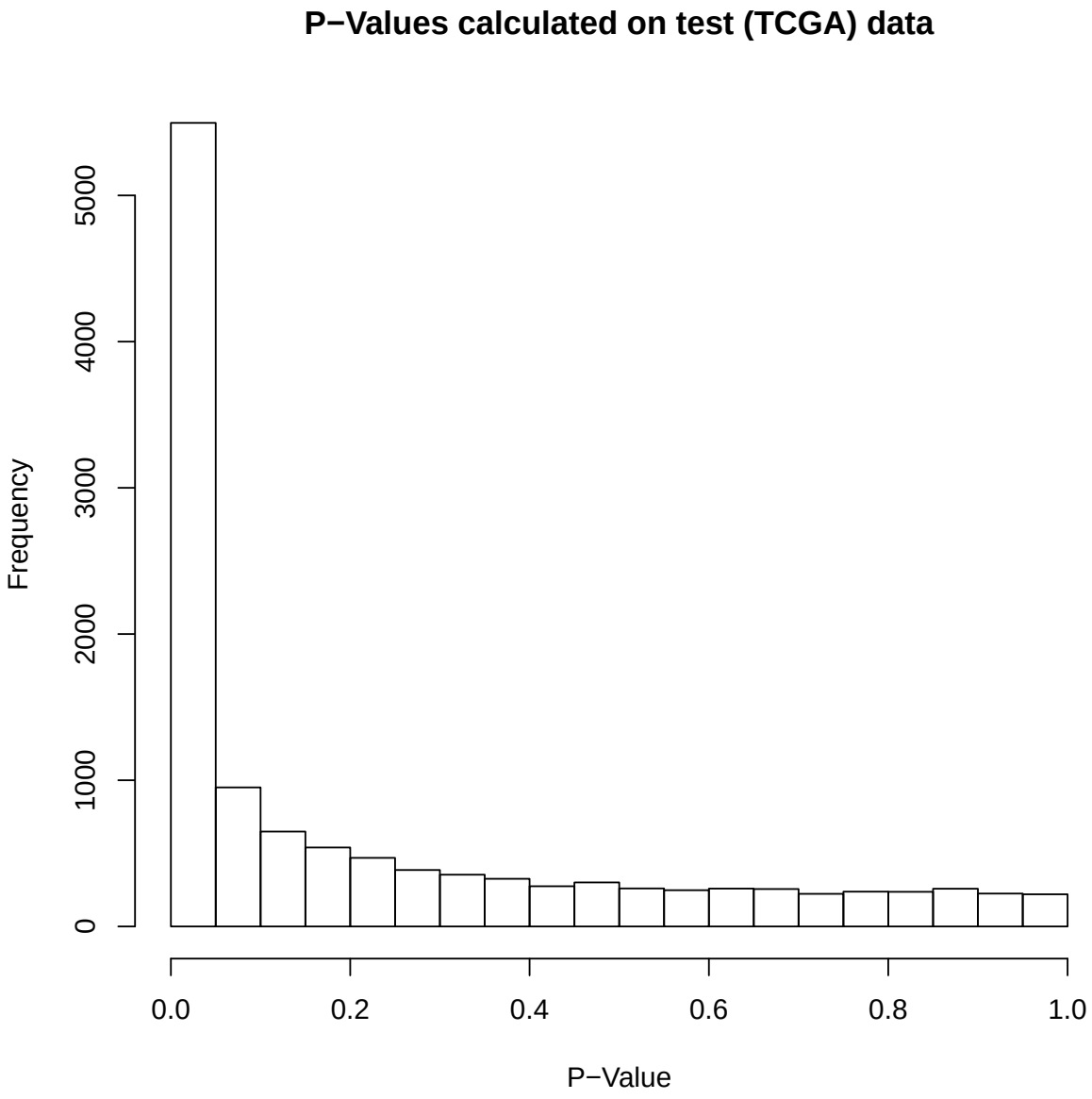
```
  #                                     sparse = T,
  #                                     verbose = F,
  #                                     geneexpr=TCGA.RSEM,
  #                                     minmeanloggeneexp= 3,
  #                                     GenestoStudy =
  #                                             as.vector(na.omit(intersect(names(junctionPValue),
  #                                                                         rownames(TCGA.RSEM))
  #                                                                   [(1:1000)+i*1000])))

  junctionPValueTCGA <- SEVA.meangeneFilter(junc.RPM =   junc.RPM.TCGA,
                                      phenoVect=as.factor(phenoVect.TCGA),
                                     geneexpr=TCGA.RSEM,
                                     minmeanloggeneexp= 3,
                                     GenestoStudy =
                                             as.vector(na.omit(intersect(names(junctionPValue),
                                                                         rownames(TCGA.RSEM))
                                                                   [(1:1000)+i*1000])))

  gc()
  junctionPValueTCGAaug <- c(junctionPValueTCGAaug,junctionPValueTCGA)
}


save(list=c("junctionPValueTCGA","junctionPValueTCGAaug"),
     file = "../Cache/junctionPValueTCGA.rda")
```

Now, checking if the genes identified on the original data generates enriched p-values on the TCGA data.

```
load(file = "../Cache/junctionPValueTCGA.rda")
load(file = "../Cache/SEVAJoe.rda")


#Pvalues based on  the original data (Training data a.k.a. Joe's Data)
originaldatapval <- sapply(junctionPValue,function(x) x$pvalue) #All P-values on the Training set
hist(x = originaldatapval,
     xlab="P-Value",main=paste("P-Values calculated on training data"))
```




```
SEVATCGAGenes <- intersect(names(junctionPValueTCGAaug),
                                      SEVAGenesPure)


tcgaallpvals <- sapply(X = junctionPValueTCGAaug,FUN = function(x) x$pvalue)

hist(x = tcgaallpvals,
     xlab="P-Value",main=paste("P-Values calculated on test (TCGA) data"))
```

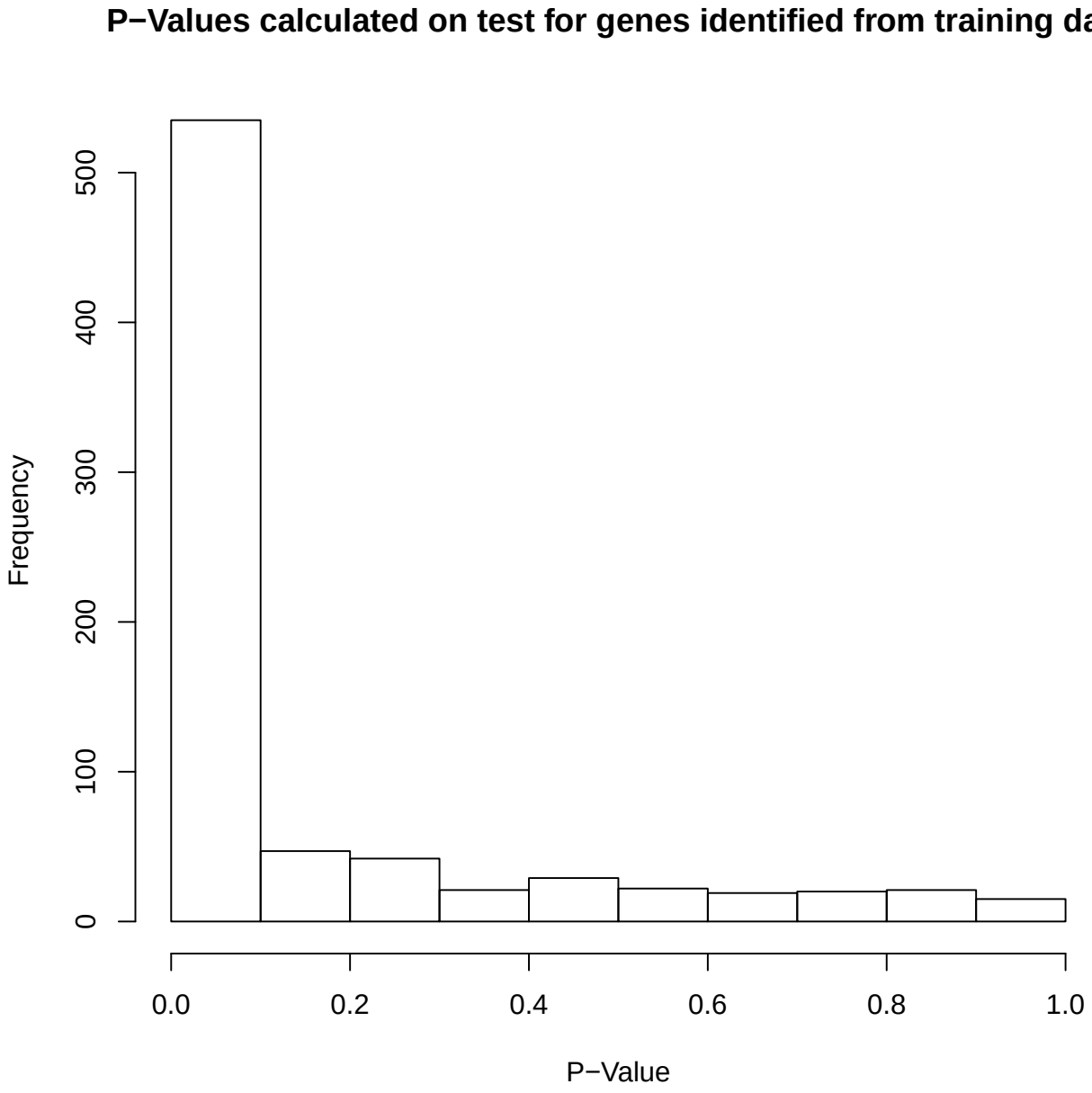



```
print(cor.test(tcgaallpvals,originaldatapval[names(tcgaallpvals)],method = "spearman"))
```

```
##
##  Spearman's rank correlation rho
##
## data:  tcgaallpvals and originaldatapval[names(tcgaallpvals)]
## S = 2.5598e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1114068
```

```
tcgapval <- sapply(junctionPValueTCGAaug[SEVATCGAGenes],function(x) x$pvalue)
```

```
hist(x = tcgapval[SEVATCGAGenes],
     xlab="P-Value",main=
       paste("P-Values calculated on test for genes identified from training data"))
```




```
cat("#Genes survived on the Gou data and were expressed in TCGA",
    length(tcgapval))
```

```
## #Genes survived on the Gou data and were expressed in TCGA 771
```

```
cat("#Genes did not survive on the Gou data and were expressed in TCGA",length(setdiff(SEVAGenesPure,nam
)
```

```
## #Genes did not survive on the Gou data and were expressed in TCGA 214
```

```
cat("Number of Genes that survived on test",
    sum(p.adjust(tcgapval,"BH")<0.01))
```

```
## Number of Genes that survived on test 352
```

```
cat("percentage that survived on test",
    mean(p.adjust(tcgapval,"BH")<0.01))
```

```
## percentage that survived on test 0.4565499
```

```
cat("Quatile of the p-value distribution SEVA genes using TCGA data")
```

```
## Quatile of the p-value distribution SEVA genes using TCGA data
```

```
print(quantile(tcgapval))
```

```
##           0%          25%          50%          75%         100%
## 0.000000e+00 3.097819e-05 9.754261e-03 1.892338e-01 9.904528e-01
```

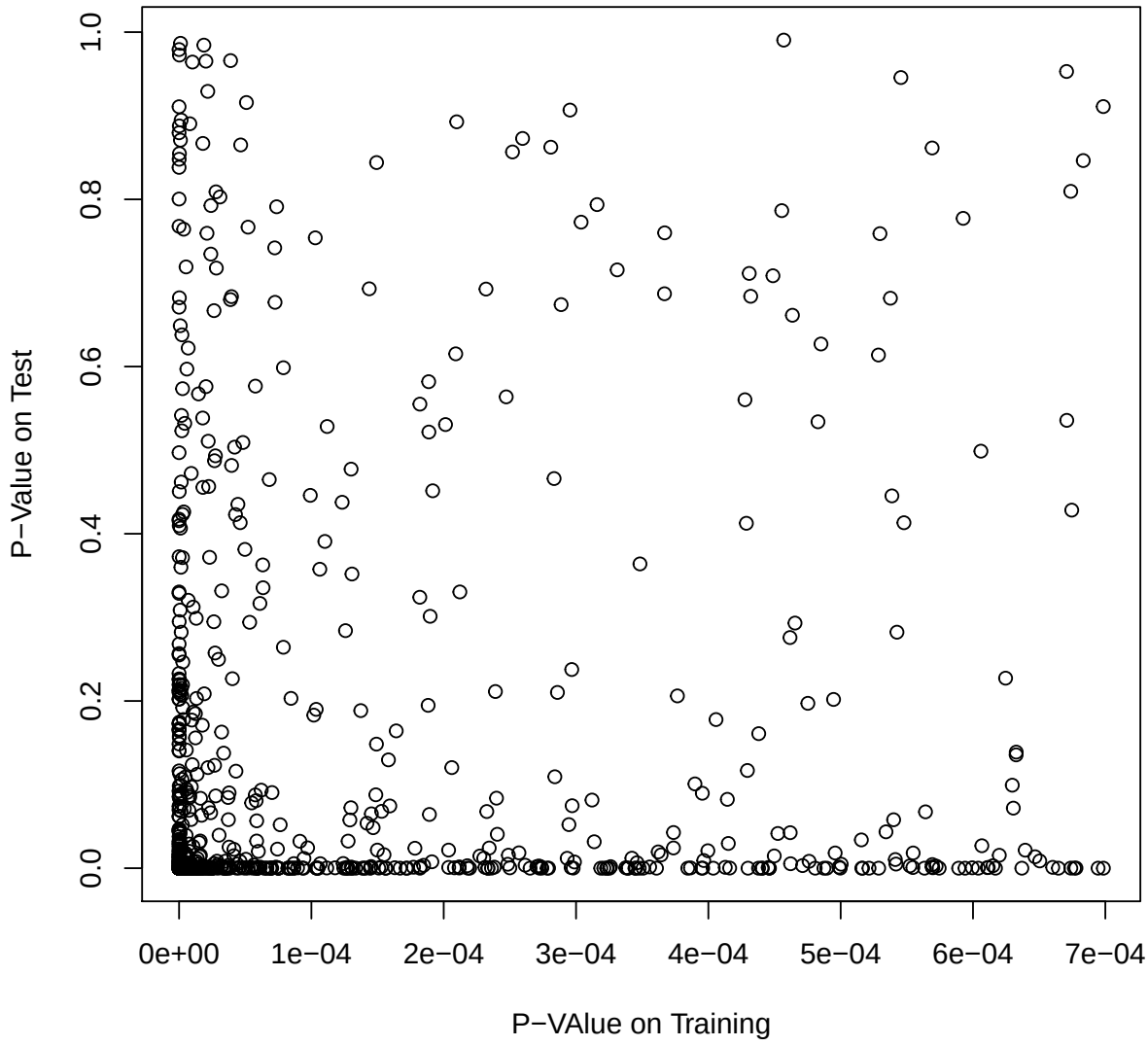
```
tcgaallpvals <- sapply(X = junctionPValueTCGAaug,FUN = function(x) x$pvalue)

cat("Enrichment of the p-values on the test data for the genes identified from training.")
```

```
## Enrichment of the p-values on the test data for the genes identified from training.
```

```
wilcox.test(index=SEVATCGAGenes,
            sapply(junctionPValueTCGAaug,function(x) abs(x$zscore)),
            alternative = "greater")
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  sapply(junctionPValueTCGAaug, function(x) abs(x$zscore))
## V = 74073000, p-value < 2.2e-16
## alternative hypothesis: true location is greater than 0
```

```
plot(originaldatapval[names(tcgapval)],
     tcgapval,
     ylab = "P-Value on Test",
     xlab = "P-VAlue on Training",
     main ="Cross-study P-Values")
```




```
cat("Quatile of the p-value distribution random genes using TCGA data")
```

```
## Quatile of the p-value distribution random genes using TCGA data
```

```
print(quantile(tcgaallpvals))
```

```
##           0%          25%          50%          75%         100%
## 0.000000000 0.002977169 0.077212061 0.392145388 0.999967804
```

```
#print(wilcox.test(x=tcgapval,y=tcgaallpvals,alternative = "less",conf.int = T,conf.level = 0.95))


save(list=ls(),file = "../Cache/SEVATCGA.rda")
```