
DESeq:

```
DESeq <- function(exprsFile,classFile,output){
  exprs <- as.matrix(read.csv(exprsFile, header=TRUE, sep = ",",row.names = 1,as.is=TRUE))
  storage.mode(exprs) = "integer"
  classcond <- as.data.frame(read.csv(classFile, header=TRUE, sep = ",",
                                     #row.names = 1,
                                     as.is=TRUE))

  DESeq.cds = newCountDataSet(countData = exprs, conditions = classcond$Condition)
  DESeq.cds = estimateSizeFactors(DESeq.cds)
  DESeq.cds = estimateDispersions(DESeq.cds, sharingMode = "maximum", method = "pooled",
fitType = "local")
  DESeq.test = nbinomTest(DESeq.cds, "Tumor", "Normal")
  DESeq.pvalues = DESeq.test$pval
  DESeq.adj.pvalues = p.adjust(DESeq.pvalues, method = "BH")
  tstout = cbind(row.names(exprs),DESeq.adj.pvalues)
  write.table(tstout,output, sep = ",", col.names = F, row.names = F)
}
```

edgeR:

```
edgeR <- function(exprsFile,classFile,output){
  exprs <- as.matrix(read.csv(exprsFile, header=TRUE, sep = ",",row.names = 1,as.is=TRUE))
  classcond <- as.data.frame(read.csv(classFile, header=TRUE, sep = ",",
                                     #row.names = 1,
                                     as.is=TRUE))

  edgeR.dgelist = DGEList(counts=exprs, group = classcond$Condition)
  edgeR.dgelist = calcNormFactors(edgeR.dgelist, method = "TMM")
  edgeR.dgelist = estimateCommonDisp(edgeR.dgelist)
  edgeR.dgelist = estimateTagwiseDisp(edgeR.dgelist, trend = "movingave")
  edgeR.test = exactTest(edgeR.dgelist)
  edgeR.pvalues = edgeR.test$table$PValue
  edgeR.adj.pvalues = p.adjust(edgeR.pvalues, method = "BH")
  edgeR.logFC = edgeR.test$table$logFC
  tstout = cbind(row.names(exprs),edgeR.adj.pvalues,edgeR.logFC)
  write.table(tstout, output, sep = ",", col.names = F, row.names = F)
}
```

voom:

```
voom <- function(exprsFile,classFile,output){
  library(edgeR)
  library(limma)
  exprs <- as.matrix(read.csv(exprsFile, header=TRUE, sep = ",",
                             row.names = 1,
                             as.is=TRUE))
  classcond <- as.data.frame(read.csv(classFile, header=TRUE, sep = ",",
                                     #row.names = 1,
                                     as.is=TRUE))

  nf = calcNormFactors(exprs, method = "TMM")#TMM normalization method
  voom.data = voom(exprs, design = model.matrix(~classcond$Condition), lib.size =
colSums(exprs) * nf)
  voom.data$genes = row.names(exprs)
```

```

voom.fitlimma = lmFit(voom.data, design = model.matrix(~classcond$Condition))
voom.fitbayes = eBayes(voom.fitlimma)
voom.pvalues = voom.fitbayes$p.value[, 2]
voom.adjppvalues = p.adjust(voom.pvalues, method = "BH")

tstout = cbind(rownames(exprs),voom.adjppvalues)
write.table(tstout,paste(output,"voom.csv",sep=""), sep = ",", col.names = F, row.names = F)
}

```

VST:

```

vst <-function(exprs,classFile,output){
  exprs <- as.matrix(read.csv(exprs, header=TRUE, sep = ",",row.names = 1,as.is=TRUE))
  storage.mode(exprs) = "integer"
  classcond <- as.data.frame(read.csv(classFile, header=TRUE, sep = ",",
                                     #row.names = 1,
                                     as.is=TRUE))
  DESeq.cds = newCountDataSet(countData = exprs, conditions = classcond$Condition)
  DESeq.cds = estimateSizeFactors(DESeq.cds)
  DESeq.cds = estimateDispersions(DESeq.cds, method = "blind", fitType = "local")
  DESeq.vst = getVarianceStabilizedData(DESeq.cds)
  DESeq.vst.fitlimma = lmFit(DESeq.vst, design = model.matrix(~classcond$Condition))
  DESeq.vst.fitbayes = eBayes(DESeq.vst.fitlimma)
  DESeq.vst.pvalues = DESeq.vst.fitbayes$p.value[, 2]
  DESeq.vst.adjppvalues = p.adjust(DESeq.vst.pvalues, method = "BH")

  tstout = cbind(rownames(exprs),DESeq.vst.adjppvalues)
  write.table(tstout, output, sep = ",", col.names = F, row.names = F)
}

```