`permPATH`: Permutation Based Gene Expression Pathway Analysis

Ivo D. Shterev * Kouros Owzar Gregory D. Sempowski

July 6, 2016

1 Introduction

This vignette describes the R extension package `permPATH` for performing permutation based gene expression pathway analysis. The package works by computing a score for each group (pathway) of genes. The score is a function of the individual gene test statistics involved in the pathway. Currently, the package computes as the score the mean of the test statistics, the mean of the absolute values of the test statistics and the so called maxmean score [1]. The individual test statistics that the package currently supports are the t-test statistic, the Wilcoxon, Pearson, the Spearman and the Jonckheere-Terpstra (JT) test statistics.

2 Adjusting for Multiple Comparisons

In addition to computing individual test statistics and scores, the package also computes raw permutation p-values, false discovery (FDR) adjusted p-values, Bonferroni corrected p-values, as well as family wise error (FWER) adjusted two sided permutation p-values.

3 Data Format

The R package `permPATH` assumes that the gene expression data is in the form of a `K`×`n` matrix, where `K` is the number of genes and `n` is the number of samples. The row names of the data frame should be the gene symbols. The phenotype data should be in the form of a vector of length `n`. The user also needs to provide a list of pre-defined pathways with each list element containing the gene symbols associated with the pathway. The name of each list element should be the pathway name.

4 Input Parameters

The code requires that the user also specifies the type of local test statistic for each gene, the global test statistic used to compute the score and the number of random permutations. The user can specify the minimum number of genes that a pathway should contain, thus filtering out pathways with smaller number of genes. Likewise, the user can specify the maximum number of genes that a pathway should contain, thus filtering out pathways with

*i.shterev@duke.edu

larger number of genes. In case of missing values, the user can specify a value that can be imputed in the gene expression data. The package also allows for the user to specify a transformation to be applied to the gene expression data prior to the analysis.

5 Output

The output of `permPATH` is in the form of a list with the following elements:

- `res`: Data frame consisting of the pathway names (Pathway), the genes involved in each pathway (Genes), the number of genes in each pathway (Size), the score for each pathway (Score), the permutation raw p-value (pval), the FWER-adjusted permutation p-value (pfwer), the FDR-adjusted permutation p-value, the Bonferroni-adjusted permutation p-value (bonferroni). If specified by the user, annotation (anno) for each pathway.
- `stats`: The individual test statistic for each gene.
- `scores`: A matrix of scores. The matrix is of dimension `(B+1)`$\times$`M`, where `M` is the number of pathways. The first column contains the unpermuted scores, the remaining `B` columns contain the scores computed after each permutation.

The results can be sorted according to decreasing order of absolute score values or according to increasing order of raw p-values. This can be specified by the user.

6 Examples

6.1 Synthetic Data

In this section we demonstrate the use of `permPATH` on synthetically generated data.

```
# Generate toy phenotype and gene expression data sets
# This example consists of 40 genes grouped into 5 pathways and 100 patients
# grp is a binary trait (e.g., case vs control)
# bp is a continuous trait (e.g., blood pressure)
set.seed(1234)
library(permPATH)


## Loading required package:  R2HTML
## Loading required package:  xtable

n = 100
K = 40
grp = rep(1:0,each=n/2)
bp = rnorm(n)

pdat = data.frame(grp, bp)
rm(grp, bp)
expdat = matrix(rnorm(K*n),K,n)

## Assign marker names g1,...,gK to the expression data set and
## patient ids id1,...,idn to the expression and phenotype data
gnames = paste("g",1:K,sep="")
rownames(expdat) = gnames
patid = paste("id",1:n,sep="")
rownames(pdat) = patid
colnames(expdat) = patid

# Group the K genes into M pathways of sizes n1,...,nM
M = 5
p = runif(M)
p = p/sum(p)
nM = rmultinom(1, size=K, prob=p)
gset = lapply(nM, function(x){gnames[sample(x)]})
names(gset) = paste("pathway",1:M,sep="")
names(gset)
```

```
## [1] "pathway1" "pathway2" "pathway3" "pathway4" "pathway5"

# Carry out permutation analysis with grp as the outcome
# using the two-sample Wilcoxon test with B=100 random permutations.
# The score is the maxmean test statistic
res = perm.path(expdat, y=pdat[["grp"]], local.test="wilcoxon",
           global.test="maxmean", B=100, gset=gset, min.num=2,
           max.num=50, sort="score")

# Output results for top pathways
head(res[["res"]])

##           Pathway                              Genes Size      Score pval
## pathway4 pathway4                              g2;g1    2 -0.9754755 0.43
## pathway1 pathway1 g5;g3;g8;g4;g10;g6;g1;g9;g2;g7;g11   11 -0.8134707 0.68
## pathway2 pathway2 g3;g9;g10;g2;g7;g6;g8;g5;g4;g11;g1   11 -0.8134707 0.68
## pathway3 pathway3           g7;g1;g8;g5;g3;g6;g4;g2    8 -0.6783519 0.85
## pathway5 pathway5           g6;g5;g4;g8;g2;g3;g1;g7    8 -0.6783519 0.85
##          pfwer  fdr bonferroni
## pathway4  0.65 0.85          1
## pathway1  0.83 0.85          1
## pathway2  0.83 0.85          1
## pathway3  0.95 0.85          1
## pathway5  0.95 0.85          1

# Output individual test statistics
res[["stats"]]

##          g5          g3          g8          g4         g10          g6
##  0.28264661 -1.12369264  0.08961966 -0.30332807  1.06854208 -0.01378764
##          g1          g9          g2          g7         g11
## -0.42741683 -1.46838363 -1.52353419  1.17884320 -0.83415220

# Carry out permutation analysis with bp as the outcome
# using the Spearman test with B=100 random permutations.
# The score is the maxmean test statistic
res = perm.path(expdat, y=pdat[["bp"]], local.test="spearman",
           global.test="maxmean", B=100, gset=gset, min.num=2,
           max.num=50, sort="score")

# Output results for top pathways
head(res[["res"]])

##           Pathway                              Genes Size       Score pval
## pathway3 pathway3           g7;g1;g8;g5;g3;g6;g4;g2    8 -0.09945395 0.42
## pathway5 pathway5           g6;g5;g4;g8;g2;g3;g1;g7    8 -0.09945395 0.42
## pathway1 pathway1 g5;g3;g8;g4;g10;g6;g1;g9;g2;g7;g11   11 -0.07216322 0.78
## pathway2 pathway2 g3;g9;g10;g2;g7;g6;g8;g5;g4;g11;g1   11 -0.07216322 0.78
## pathway4 pathway4                              g2;g1    2  0.02052205 0.97
##          pfwer  fdr bonferroni
## pathway3  0.62 0.97          1
## pathway5  0.62 0.97          1
## pathway1  0.84 0.97          1
## pathway2  0.84 0.97          1
## pathway4  1.00 0.97          1

# Output individual test statistics
res[["stats"]]

##          g5          g3          g8          g4         g10          g6
##  0.10024602  0.02095410  0.06522652 -0.09945395  0.02682268  0.03545155
##          g1          g9          g2          g7         g11
##  0.02653465 -0.04487249  0.01450945  0.06544254  0.11585959
```

6.2 Incorporating Annotation

This subsection describes the use of `permPATH` with real gene symbols that can be mapped to a gene pathway data base supported by Broad Institute. The user can also create pathways on the bases of files from the Molecular Signatures Database[2].

```
# Generate gene symbols
set.seed(1234)
library(permPATH)

gnames = c("CCL13", "CCL19", "CCL2", "CCL3", "CCL3L1", "CCL4",
          "CCL5", "CCL7", "CCL8", "CCR1", "CCR2", "CCR3", "CCR5",
          "CD14", "CD180", "CD2", "CD209", "CD40", "CD44", "CD80",
          "CD86", "CD8A", "CDC42", "CEBPA", "CSF2", "CXCL1", "CXCL10",
          "CXCR4", "EIF2AK2", "ELK1", "ERBB2", "FCAR", "HLAA",
          "HLADQA1", "HLADQB1", "HSPA1A", "IFIT3", "IFNA1", "IFNB1",
```

```
          "IFNG", "IL10", "IL12A", "IL12B", "IL16", "IL1A", "IL1B",
          "IL2", "IL6", "IL8", "INHBA", "IRF1", "IRF3", "ITGAM",
          "LTA", "LYN", "MAP3K7", "MAP4K4", "MAPK8", "MAPK8IP3",
          "MYD88", "NFKB1", "NFKBIA", "NFKBIL1", "NFRKB", "PELI1",
          "PTGS2", "REL", "RELA", "RIPK2", "SARM1", "STK4", "TAP2",
          "TGFB1", "TIRAP", "TLR1", "TLR10", "TLR2", "TLR3", "TLR4",
          "TLR5", "TLR6", "TLR7", "TLR8", "TLR9", "TNF", "UBE2N", "B2M",
          "RPL13A", "ACTB", "HGDC", "RTC1", "RTC2", "RTC3", "PPC1", "PPC2", "PPC3")

# extract pathways available at "http://software.broadinstitute.org/gsea/resources/msigdb/4.0/c2.cp.reactome.v4.0.symbols.gmt"
xx = readLines("c2.cp.reactome.v4.0.symbols.gmt")
pnames = as.character(sapply(xx, function(x){unlist(strsplit(x, "\t", fixed=TRUE))[1]}))
anno = as.character(sapply(xx, function(x){unlist(strsplit(x, "\t", fixed=TRUE))[2]}))
gset = lapply(xx, function(x){unlist(strsplit(x, "\t", fixed=TRUE))[-c(1,2)]})
names(gset) = pnames
gset = list(gset, pnames, anno)

#intersect gene nsymbols with gene symbols from pathways
ind = unlist(lapply(gset[[1]], function(x){ifelse(length(intersect(x,gnames))>1, TRUE, FALSE)}))
gset[[1]] = gset[[1]][ind]
gset[[2]] = gset[[2]][ind]
gset[[3]] = gset[[3]][ind]
gset[[1]] = lapply(gset[[1]], function(x){intersect(x, gnames)})
names(gset[[1]]) = gset[[2]]
names(gset[[3]]) = gset[[2]]

#create gene expression data
n = 220
K = length(gnames)
expdat = matrix(abs(rnorm(K*n)), K, n)
rownames(expdat) = gnames
patid = paste("id",1:n,sep="")
colnames(expdat) = patid

grp = rep(1:0,each=n/2)
bp = abs(rnorm(n))
pdat = data.frame(grp, bp)
rm(grp, bp)

# Carry out permutation analysis with grp as the outcome
# using the two-sample Wilcoxon test with B=10000 random permutations.
# The score is the maxmean test statistic
res = perm.path(expdat, y=pdat[["grp"]], local.test="wilcoxon",
            global.test="maxmean", B=10^4, gset=gset[[1]], min.num=2,
            max.num=50, sort="score", anno=gset[[3]])

# Output results for top pathways
head(res[["res"]])
```

```
##                                                                Pathway
## REACTOME_DEFENSINS                                       REACTOME_DEFENSINS
## REACTOME_BETA_DEFENSINS                                REACTOME_BETA_DEFENSINS
## REACTOME_DIABETES_PATHWAYS                          REACTOME_DIABETES_PATHWAYS
## REACTOME_PERK_REGULATED_GENE_EXPRESSION REACTOME_PERK_REGULATED_GENE_EXPRESSION
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4       REACTOME_ACTIVATION_OF_GENES_BY_ATF4
## REACTOME_UNFOLDED_PROTEIN_RESPONSE           REACTOME_UNFOLDED_PROTEIN_RESPONSE
##                                          Genes Size     Score
## REACTOME_DEFENSINS                    CCR2;TLR1;TLR2    3 -1.806970
## REACTOME_BETA_DEFENSINS               CCR2;TLR1;TLR2    3 -1.806970
## REACTOME_DIABETES_PATHWAYS                 IL8;CCL2    2 -1.652329
## REACTOME_PERK_REGULATED_GENE_EXPRESSION    IL8;CCL2    2 -1.652329
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4       IL8;CCL2    2 -1.652329
## REACTOME_UNFOLDED_PROTEIN_RESPONSE         IL8;CCL2    2 -1.652329
##                                           pval pfwer  fdr bonferroni
## REACTOME_DEFENSINS                       0.082 0.834 0.73          1
## REACTOME_BETA_DEFENSINS                  0.082 0.834 0.73          1
## REACTOME_DIABETES_PATHWAYS               0.109 0.932 0.73          1
## REACTOME_PERK_REGULATED_GENE_EXPRESSION  0.109 0.932 0.73          1
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4     0.109 0.932 0.73          1
## REACTOME_UNFOLDED_PROTEIN_RESPONSE       0.109 0.932 0.73          1
##                                                                                                                anno
## REACTOME_DEFENSINS                                       http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_DEFENSINS
## REACTOME_BETA_DEFENSINS                                http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_BETA_DEFENSINS
## REACTOME_DIABETES_PATHWAYS                          http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_DIABETES_PATHWAYS
## REACTOME_PERK_REGULATED_GENE_EXPRESSION http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_PERK_REGULATED_GENE_EXPRESSION
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4       http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_ACTIVATION_OF_GENES_BY_ATF4
## REACTOME_UNFOLDED_PROTEIN_RESPONSE           http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_UNFOLDED_PROTEIN_RESPONSE
```

```
# Carry out permutation analysis with bp as the outcome
# using the Spearman test with B=10000 random permutations.
# The score is the maxmean test statistic
res = perm.path(expdat, y=pdat[["grp"]], local.test="spearman",
            global.test="maxmean", B=10^4, gset=gset[[1]], min.num=2,
            max.num=50, sort="score", anno=gset[[3]])

# Output results for top pathways
head(res[["res"]])
```

```
##                                                   Pathway
## REACTOME_DEFENSINS                          REACTOME_DEFENSINS
```

```
## REACTOME_BETA_DEFENSINS                                       REACTOME_BETA_DEFENSINS
## REACTOME_DIABETES_PATHWAYS                                 REACTOME_DIABETES_PATHWAYS
## REACTOME_PERK_REGULATED_GENE_EXPRESSION REACTOME_PERK_REGULATED_GENE_EXPRESSION
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4       REACTOME_ACTIVATION_OF_GENES_BY_ATF4
## REACTOME_UNFOLDED_PROTEIN_RESPONSE           REACTOME_UNFOLDED_PROTEIN_RESPONSE
##                                                   Genes Size      Score
## REACTOME_DEFENSINS                      CCR2;TLR1;TLR2    3 -0.1221037
## REACTOME_BETA_DEFENSINS                 CCR2;TLR1;TLR2    3 -0.1221037
## REACTOME_DIABETES_PATHWAYS                    IL8;CCL2    2 -0.1116540
## REACTOME_PERK_REGULATED_GENE_EXPRESSION       IL8;CCL2    2 -0.1116540
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4          IL8;CCL2    2 -0.1116540
## REACTOME_UNFOLDED_PROTEIN_RESPONSE            IL8;CCL2    2 -0.1116540
##                                          pval pfwer   fdr bonferroni
## REACTOME_DEFENSINS                      0.072 0.825 0.689          1
## REACTOME_BETA_DEFENSINS                 0.072 0.825 0.689          1
## REACTOME_DIABETES_PATHWAYS              0.107 0.927 0.689          1
## REACTOME_PERK_REGULATED_GENE_EXPRESSION 0.107 0.927 0.689          1
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4    0.107 0.927 0.689          1
## REACTOME_UNFOLDED_PROTEIN_RESPONSE      0.107 0.927 0.689          1
##                                                                                                               anno
## REACTOME_DEFENSINS                                   http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_DEFENSINS
## REACTOME_BETA_DEFENSINS                         http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_BETA_DEFENSINS
## REACTOME_DIABETES_PATHWAYS                   http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_DIABETES_PATHWAYS
## REACTOME_PERK_REGULATED_GENE_EXPRESSION http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_PERK_REGULATED_GENE_EXPRESSION
## REACTOME_ACTIVATION_OF_GENES_BY_ATF4       http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_ACTIVATION_OF_GENES_BY_ATF4
## REACTOME_UNFOLDED_PROTEIN_RESPONSE           http://www.broadinstitute.org/gsea/msigdb/cards/REACTOME_UNFOLDED_PROTEIN_RESPONSE
```

7 Exporting Results to HTML File

The user has the option to export the results of `permPATH` to an HTML file via the function `permPATH2HTML`. This option is useful when pathways have large number of genes and allows for improved readability of the results.

```
library(permPATH)
set.seed(1234)
n = 100
K = 40
grp = rep(1:0,each=n/2)
bp = rnorm(n)

pdat = data.frame(grp, bp)
rm(grp, bp)
expdat = matrix(rnorm(K*n),K,n)

## Assign marker names g1,...,gK to the expression data set and
## patient ids id1,...,idn to the expression and phenotype data
gnames = paste("g",1:K,sep="")
rownames(expdat) = gnames
patid = paste("id",1:n,sep="")
rownames(pdat) = patid
colnames(expdat) = patid

#Group the K genes into M pathways of sizes n1,...,nM
M = 5
p = runif(M)
p = p/sum(p)
nM = rmultinom(1, size=K, prob=p)
gset = lapply(nM, function(x){gnames[sample(x)]})
names(gset) = paste("pathway",1:M,sep="")

## Carry out permutation analysis with grp as the outcome
## using the two-sample Wilcoxon with B=100 random permutations
res = perm.path(expdat, y=pdat[["grp"]], local.test="wilcoxon", global.test="maxmean", B=100, gset=gset,
           min.num=2, max.num=50, sort="score")

# create an html file
#epermPATH2HTML(rstab, dir="/dir/", fname="tophits")

sessionInfo()


## R version 3.3.1 (2016-06-21)
## Platform: i686-pc-linux-gnu (32-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] permPATH_0.7 xtable_1.8-2 R2HTML_2.3.2
##
## loaded via a namespace (and not attached):
## [1] magrittr_1.5  formatR_1.4   tools_3.3.1   stringi_1.1.1 highr_0.6
## [6] knitr_1.13    stringr_1.0.0 evaluate_0.9
```

8 Acknowledgement

This work was supported by a National Institute of Allergy and Infectious Disease/National Institutes of Health contract (No. HHSN272200900059C).

References

[1] B. Efron and R. Tibshirani. On testing the significance of sets of genes. *The Annals of Applied Statistics*, 1(1):107–129, 2007.

[2] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102(43):15545–15550, 2005.