Cogena on GSE30999

Zhilong Jia et al.

2015-12-16

Contents

---

1 Introduction

This report reproduces all the results related with GSE30999. An online verison can be found at https://github.com/zhilongjia/psoriasis

2 Data Preparation

2.1 Check package required

```
# Check package required
packages <- c("knitr", "GEOquery", "MetaDE", "annotate", "hgu133plus2.db",
              "affy", "limma", "STRINGdb", "hgu133plus2.db", "devtools", "cogena")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
    stop(paste("Please install packages:", setdiff(packages, rownames(installed.packages()))) )
}
```

2.2 Download the raw data of GSE30999

```
# Download raw files from GEO and untar them if nothing in ../data/GSE30999_RAW
if (length(dir("../data/GSE30999_RAW", all.files=FALSE)) ==0) {

    download.file("http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE30999&format=file",
              destfile="../data/GSE30999_RAW.tar")
    untar("../data/GSE30999_RAW.tar", exdir="../data/GSE30999_RAW")
    file.remove("../data/GSE30999_RAW.tar")
}
```

2.3 Differential Expression Analysis

```
library(GEOquery)
library(affy)

################################################################################
# Download raw data of GSE30999
GSE30999raw <- ReadAffy(celfile.path="../data/GSE30999_RAW")
sampleNames(GSE30999raw) <- sub("(_|\\.).*CEL\\.gz","", sampleNames(GSE30999raw))

################################################################################
# Sample Label preprocessing
GSE30999series <- getGEO("GSE30999", destdir="../data")
GSE30999label <- pData(GSE30999series$GSE30999_series_matrix.txt.gz)[,c("title", "geo_accession")]
GSE30999label$title <- as.character(GSE30999label$title)

GSE30999label[grep("NL", GSE30999label$title),"state"] = "ct"
GSE30999label[grep("LS", GSE30999label$title),"state"] = "Psoriasis"
GSE30999label$state <- factor(GSE30999label$state, levels=c("ct", "Psoriasis"))
GSE30999label[,"gse_id"] = "GSE30999"
GSE30999label$rep <- sapply(strsplit(GSE30999label$title, "_"), "[", 1)


vmd = data.frame(labelDescription = c("title", "geo_accession", "state", "gse_id", "rep"))
phenoData(GSE30999raw) = new("AnnotatedDataFrame", data = GSE30999label, varMetadata = vmd)
pData(protocolData(GSE30999raw)) <-
```

```
    pData(protocolData(GSE30999raw))[rownames(GSE30999label),,drop=FALSE]

# RMA normalization
GSE30999rma <- rma(GSE30999raw)
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

```
###############################################################################
# Filter the non-informative and non-expressed genes.
library(MetaDE)
library(annotate)
library(hgu133plus2.db)

GSE30999.Explist <- list(GSE30999=list(x = exprs(GSE30999rma),
                         y = ifelse (GSE30999label$state=="ct", 0, 1),
                         symbol = getSYMBOL(rownames(exprs(GSE30999rma)), "hgu133plus2") ))
GSE30999.Explist <- MetaDE.match(GSE30999.Explist, pool.replicate="IQR")
GSE30999.Explist.filtered <- MetaDE.filter(GSE30999.Explist, c(0.2,0.2))
colnames(GSE30999.Explist.filtered$GSE30999$x) <- colnames(exprs(GSE30999rma))

###############################################################################
# DEG analysis via limma
DElimma <- function (Expdata, Explabel){

    library(limma)
    Expdesign <- model.matrix(~as.factor(Explabel$rep) + Explabel$state)
    Expfit1 <- lmFit(Expdata, Expdesign)
    Expfit2 <- eBayes(Expfit1)
    dif_Exp <- topTable(Expfit2, coef=tail(colnames(Expdesign), 1), number=Inf)

    return (dif_Exp)
}

GSE30999.limma <- DElimma(GSE30999.Explist.filtered$GSE30999$x, GSE30999label)
GSE30999.DE <- GSE30999.limma[GSE30999.limma$adj.P.Val<=0.05 & abs(GSE30999.limma$logFC)>=1,]
GSE30999.DEG <- rownames(GSE30999.DE)
GSE30999.DEG.expr <- GSE30999.Explist.filtered$GSE30999$x[GSE30999.DEG,]
```

3 Co-expression Analysis by cogena

```
# Install cogena if none
library(cogena)
if (packageVersion("cogena") < "1.2.0") {
    devtools::install_github("zhilongjia/cogena")
}
# Parameters for funtion coExp
nClust <- 11 # 11 clusters
clMethods <- c("pam") # pam clustering method
```

```
# nClust <- 2:20
# clMethods <- c("hierarchical","kmeans","diana","fanny","som","sota","pam","clara","agnes")
ncore <- 7 # 7 cores

################################################################################
# Co-expression analysis
# "correlation" is used for the distance caculation, "complete" is used for
# the agglomeration (for hclust and agnes clustering methods only).
genecl_result <- coExp(GSE30999.DEG.expr, nClust=nClust, clMethods=clMethods,
                       metric="correlation", method="complete", ncore=ncore,
                       verbose=FALSE)
```

4 Pathway Analysis by cogena

```
# Parameters for funtion clEnrich
annoGMT <- "c2.cp.kegg.v5.0.symbols.gmt.xz" # kegg pathway gene set
annofile <- system.file("extdata", annoGMT, package="cogena")
sampleLabel <- GSE30999label$state
names(sampleLabel) <- rownames(GSE30999label)

################################################################################
# cogena analysis (Pathway analysis)
cogena_result <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)

# Summary the results obtained by cogena
summary(cogena_result)
```

```
##
## Clustering Methods:
##  pam
##
## The Number of Clusters:
##  11
##
## Metric of Distance Matrix:
##  correlation
##
## Agglomeration method for hierarchical clustering (hclust and agnes):
##  complete
##
## Gene set:
##  c2.cp.kegg.v5.0.symbols.gmt.xz
```

4.1 Heatmap with co-expressed genes

```
# Figure 1
heatmapCluster(cogena_result, "pam", "11", maintitle="Psoriasis")
```

```
## The number of genes in each cluster:
## upDownGene
##   1   2
## 722 341
## cluster_size
##   1   2   3   4   5   6   7   8   9  10  11
## 192 107 163  61  87  85 120 112  76  27  33
```

4.2 Table : Co-expressed genes are highly connected

```
# pPPI function: get the PPI summary information about input genes
pPPI <- function(geneC, string_db){
    example1_mapped <- string_db$map(as.data.frame(geneC), "geneC",
                                     removeUnmappedRows = TRUE, quiet=TRUE)
    hits <- example1_mapped$STRING_id
    net_summary <- string_db$get_summary(unique(hits))
    as.numeric( gsub("[^1:9]+\\: |\\)", "", strsplit(net_summary, "\n|\\(")[[1]] ) )
}
# Init table
cluster_ppi <- data.frame(protein=numeric(14), interactions=numeric(14),
                          expected_interactions=numeric(14),
                          p_value=numeric(14), stringsAsFactors=FALSE)
rownames(cluster_ppi) <- c(1:11, "Up", "Down", "All_DE")

# Get PPI information for each cluster.
library(STRINGdb)
suppressWarnings(string_db <- STRINGdb$new(version="10", species=9606,
                                           score_threshold=400,
                                           input_directory="../tmp"))
for (i in 1:11) {
    i <- as.character(i)
    cluster_ppi[i,] <- pPPI(geneInCluster(cogena_result, "pam", "11", i), string_db)
}
cluster_ppi["Up",] <- pPPI(rownames(GSE30999.DE[GSE30999.DE$logFC>0,]), string_db)
cluster_ppi["Down",] <- pPPI(rownames(GSE30999.DE[GSE30999.DE$logFC<0,]), string_db)
cluster_ppi["All_DE",] <- pPPI(rownames(GSE30999.DE), string_db)
cluster_ppi$ratio <- cluster_ppi$interactions / cluster_ppi$expected_interactions

# Table 1
knitr::kable(cluster_ppi, caption="Summary of interactions within clusters")
```

Table 1: Summary of interactions within clusters

	protein	interactions	expected_interactions	p_value	ratio
1	179	212	86	0.0000000	2.465116
2	98	40	6	0.0000000	6.666667
3	154	39	40	0.6275275	0.975000
4	57	279	14	0.0000000	19.928571
5	87	515	49	0.0000000	10.510204
6	81	19	6	0.0000985	3.166667

	protein	interactions	expected_interactions	p_value	ratio
7	114	40	19	0.0000181	2.105263
8	105	33	23	0.0301568	1.434783
9	66	34	13	0.0000016	2.615385
10	27	14	2	0.0000000	7.000000
11	31	10	1	0.0000064	10.000000
Up	680	2393	1136	0.0000000	2.106514
Down	319	347	172	0.0000000	2.017442
All_DE	999	3633	2188	0.0000000	1.660421

4.3 Figure : The result of pathway analysis

```
# Figure 2
heatmapPEI(cogena_result, "pam", "11", printGS=FALSE, maintitle="Psoriasis")
```

4.4 GSEA

This is to get the GSEA results. The result can be obtained from *result/GSEA_output*. See gct and cls file format if needed.

```
# Prepare inputs for GSEA
expData <- as.data.frame(exprs(GSE30999rma))
expData$DESCRIPTION <- NA
expData <- expData[,c("DESCRIPTION", colnames(expData)[1:170])]

###############################################################################

# Generate gct file
write.table(expData, file="../result/GSEA_input/GSE30999_exp.gct", sep="\t", quote=FALSE)
# Add the following 3 lines at the begining of GSE30999_exp.gct
fConn <- file('../result/GSEA_input/GSE30999_exp.gct', 'r+')
Lines <- sub("DESCRIPTION", "NAME\tDESCRIPTION", readLines(fConn))
writeLines(c("#1.2\n54675\t170", Lines ), con = fConn)
close(fConn)

###############################################################################

# Generate cls file
write.table(t(as.character(GSE30999label$state)), file="../result/GSEA_input/GSE30999.cls",quote=FALSE,
fConn1 <- file('../result/GSEA_input/GSE30999.cls', 'r+')
writeLines(c("170 2 1\n#ct Psoriasis", readLines(fConn1) ), con = fConn1)
close(fConn1)

###############################################################################

# GSEA analysis
if (isTRUE(system("which java", intern=FALSE)==0) & file.exists("gsea2-2.1.0.jar")) {
    system(command="java -cp ./gsea2-2.1.0.jar -Xmx512m xtools.gsea.Gsea -res ../result/GSEA_input/GSE3
} else {
```

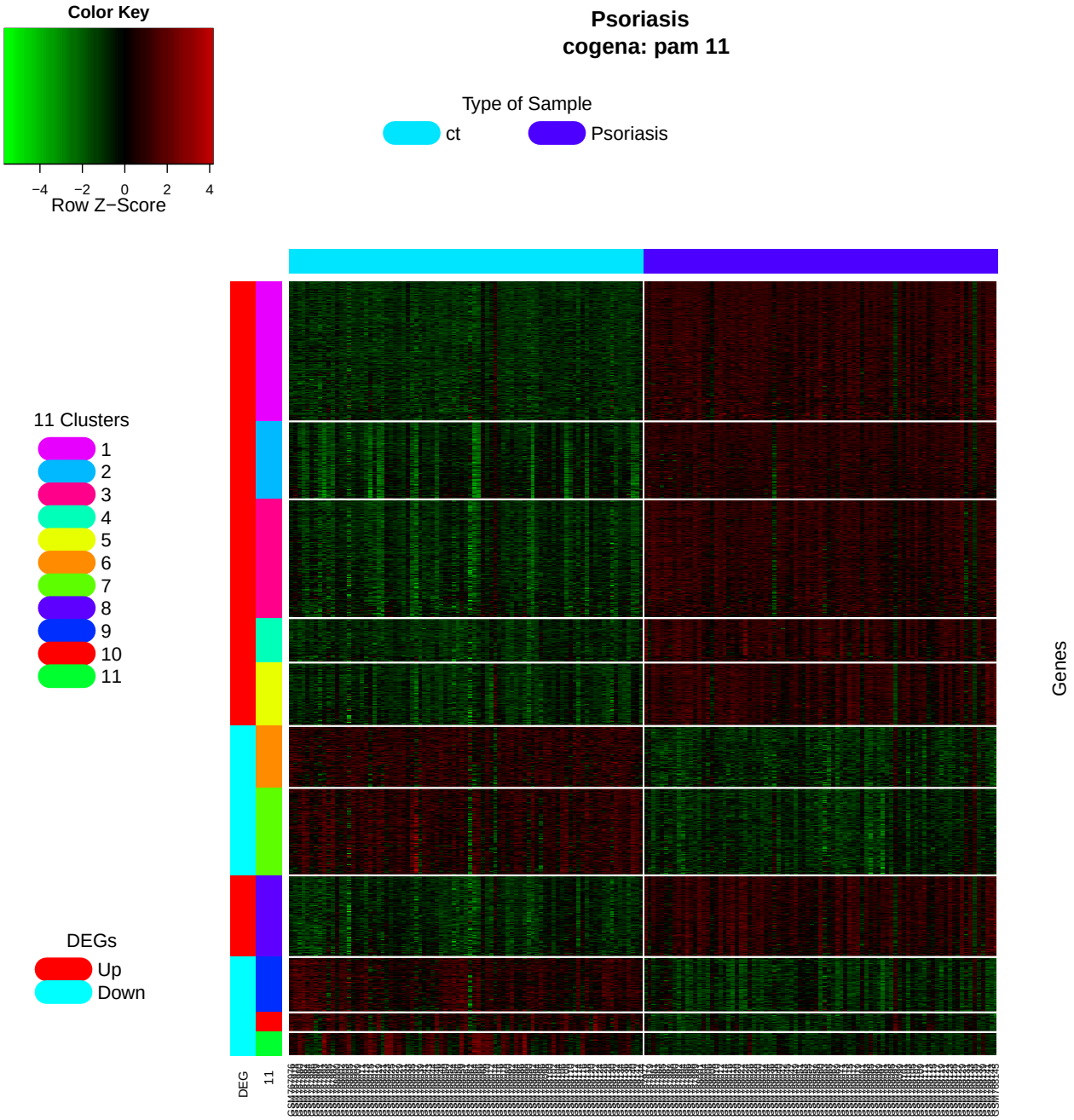

Figure 1: Heatmap with co-expression information

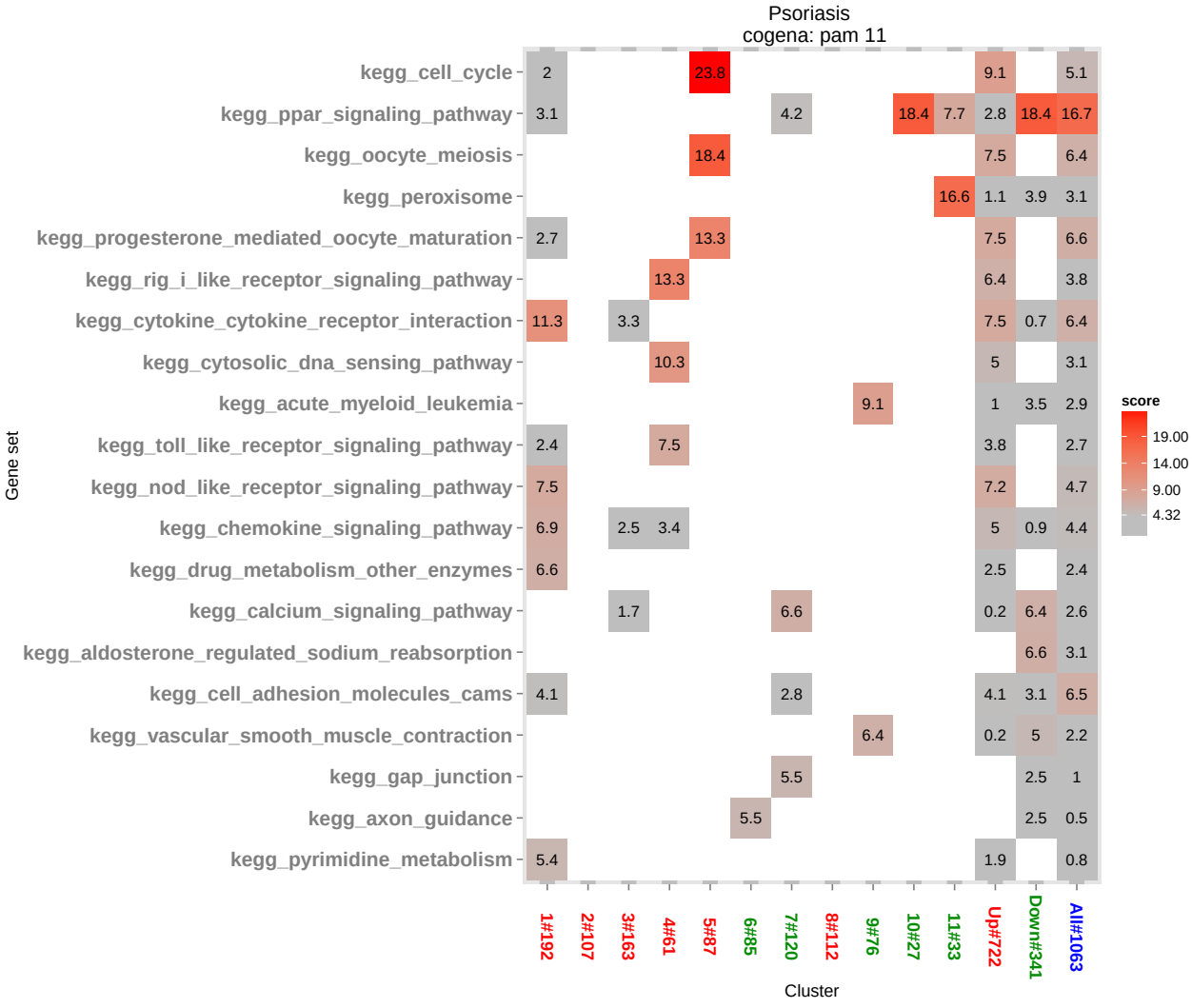

Figure 2: Pathway Analysis

```
    warning("Java is not found! GSEA was not run.")
}
#Show the gsea code above
# java -cp ./gsea2-2.1.0.jar -Xmx512m xtools.gsea.Gsea
# -res ../result/GSEA_input/GSE30999_exp.gct -cls ../result/GSEA_input/GSE30999.cls
# -gmx ../result/GSEA_input/c2.cp.kegg.v5.0.symbols.gmt
# -collapse true -mode Max_probe -norm meandiv -nperm 1000 -permute phenotype
# -rnd_type no_balance -scoring_scheme weighted -rpt_label GSE30999
# -metric Signal2Noise -sort real -order descending
# -chip ../result/GSEA_input/HG_U133_Plus_2.chip -include_only_symbols true
# -make_sets false -median false -num 100 -plot_top_x 20 -rnd_seed 149
# -save_rnd_lists false -set_max 500 -set_min 15 -zip_report false
# -out ../result/GSEA_output -gui false
```

5 Drug repositioning by cogena

```
# Drug repositioning based on CmapDn100 gene set
cmapDn100_cogena_result <- clEnrich_one(genecl_result, "pam", "11",
        annofile=system.file("extdata", "CmapDn100.gmt.xz", package="cogena"),
        sampleLabel=sampleLabel)

# Drug repositioning based on CmapUp100 gene set
cmapUp100_cogena_result <- clEnrich_one(genecl_result, method="pam", nCluster="11",
        annofile=system.file("extdata", "CmapUp100.gmt.xz", package="cogena"),
        sampleLabel=sampleLabel)
```

5.1 Figure : Drug repositioning for cluster 1

```
# Figure 3
heatmapPEI(cmapDn100_cogena_result, "pam", "11", printGS=FALSE,
           orderMethod = "1", maintitle="Psoriasis")
```

5.2 Figure : Drug repositioning for cluster 4

```
# Figure 4
heatmapPEI(cmapDn100_cogena_result, "pam", "11", printGS=FALSE,
           orderMethod = "4", maintitle="Psoriasis")
```

5.3 Figure : Drug repositioning for cluster 5

```
# Figure 5
heatmapPEI(cmapDn100_cogena_result, "pam", "11", printGS=FALSE,
           orderMethod = "5", maintitle="Psoriasis")
```

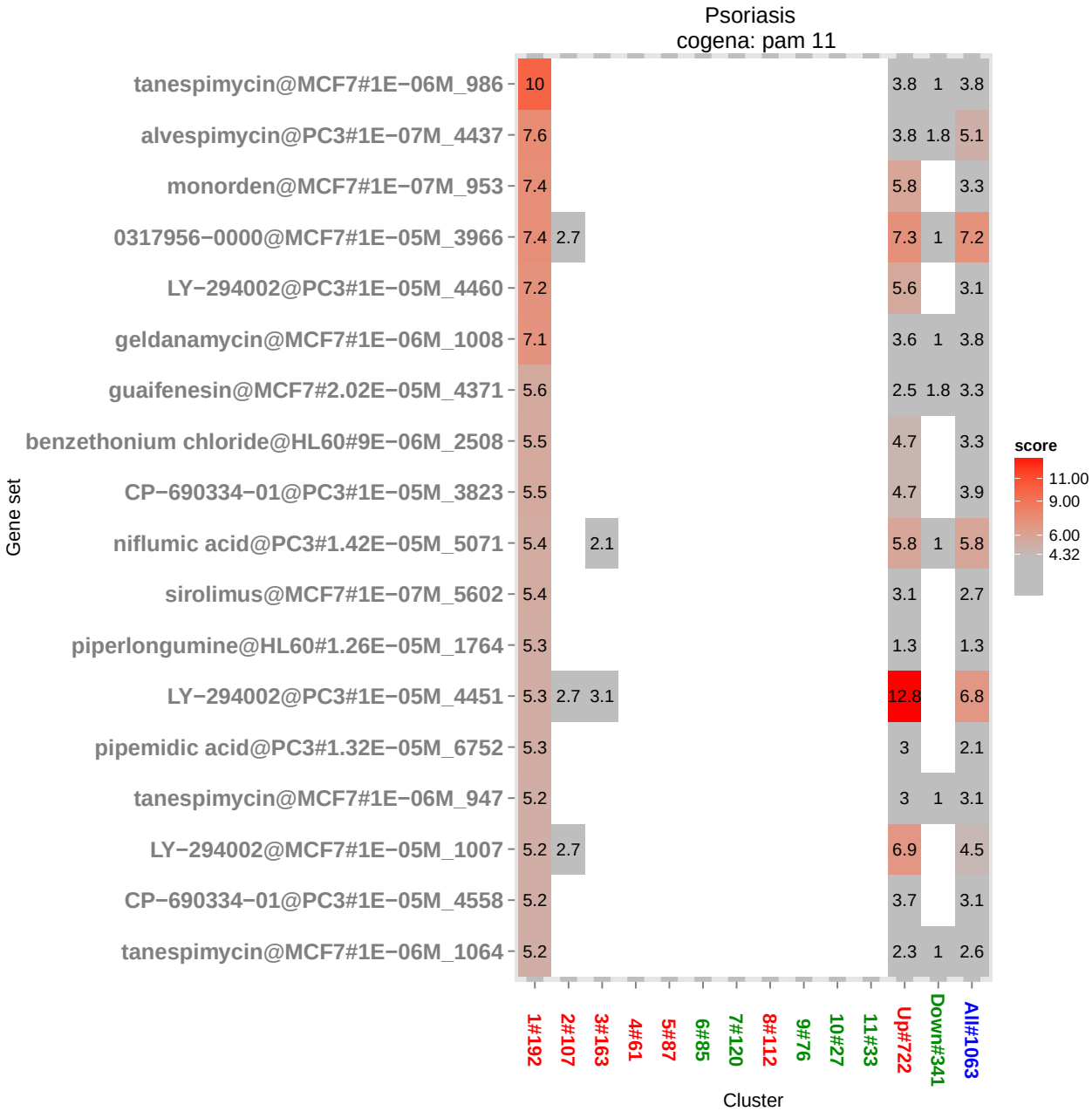

Figure 3: Drug Repositioning for cluster 1

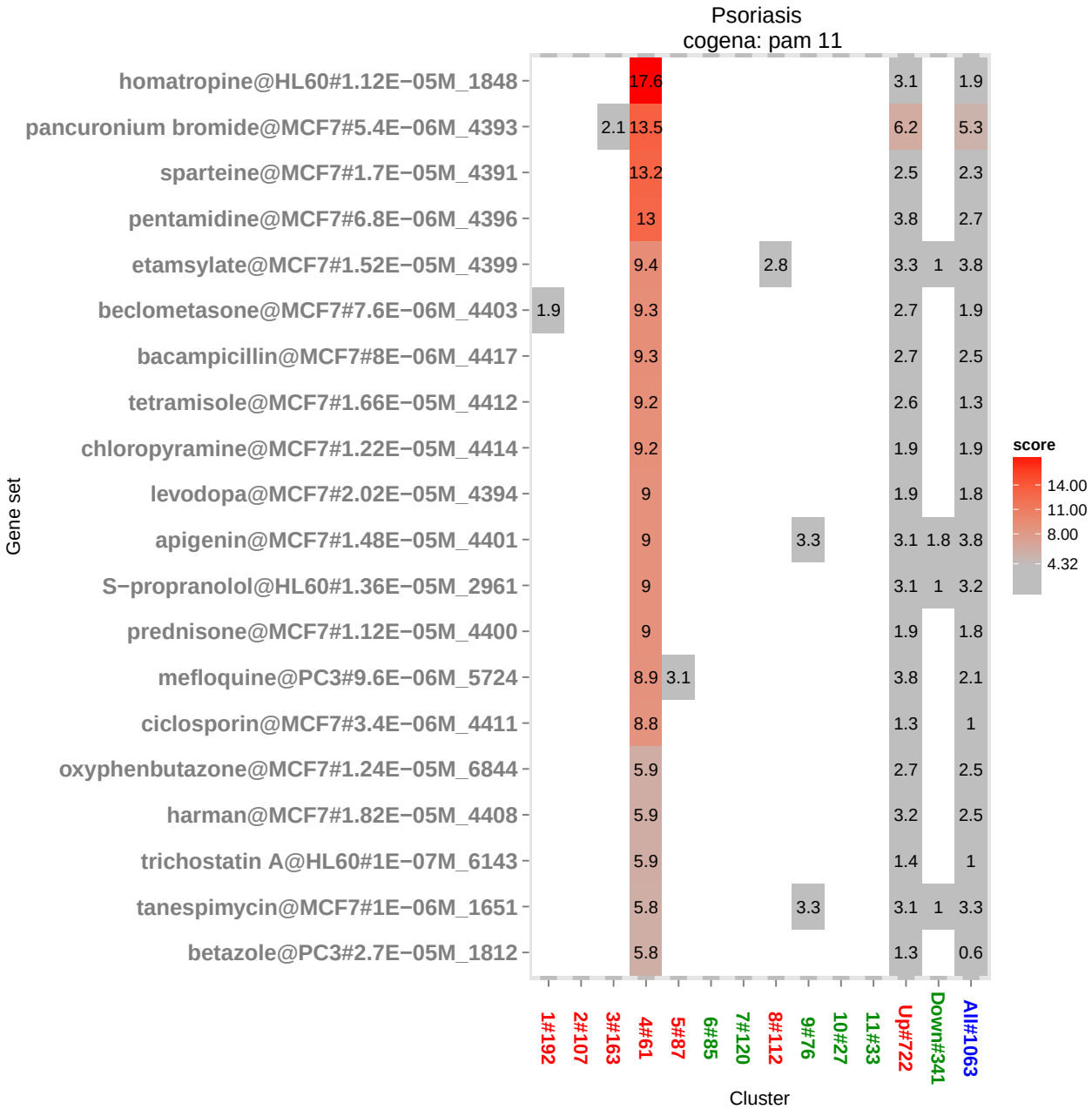

Figure 4: Drug Repositioning for cluster 4

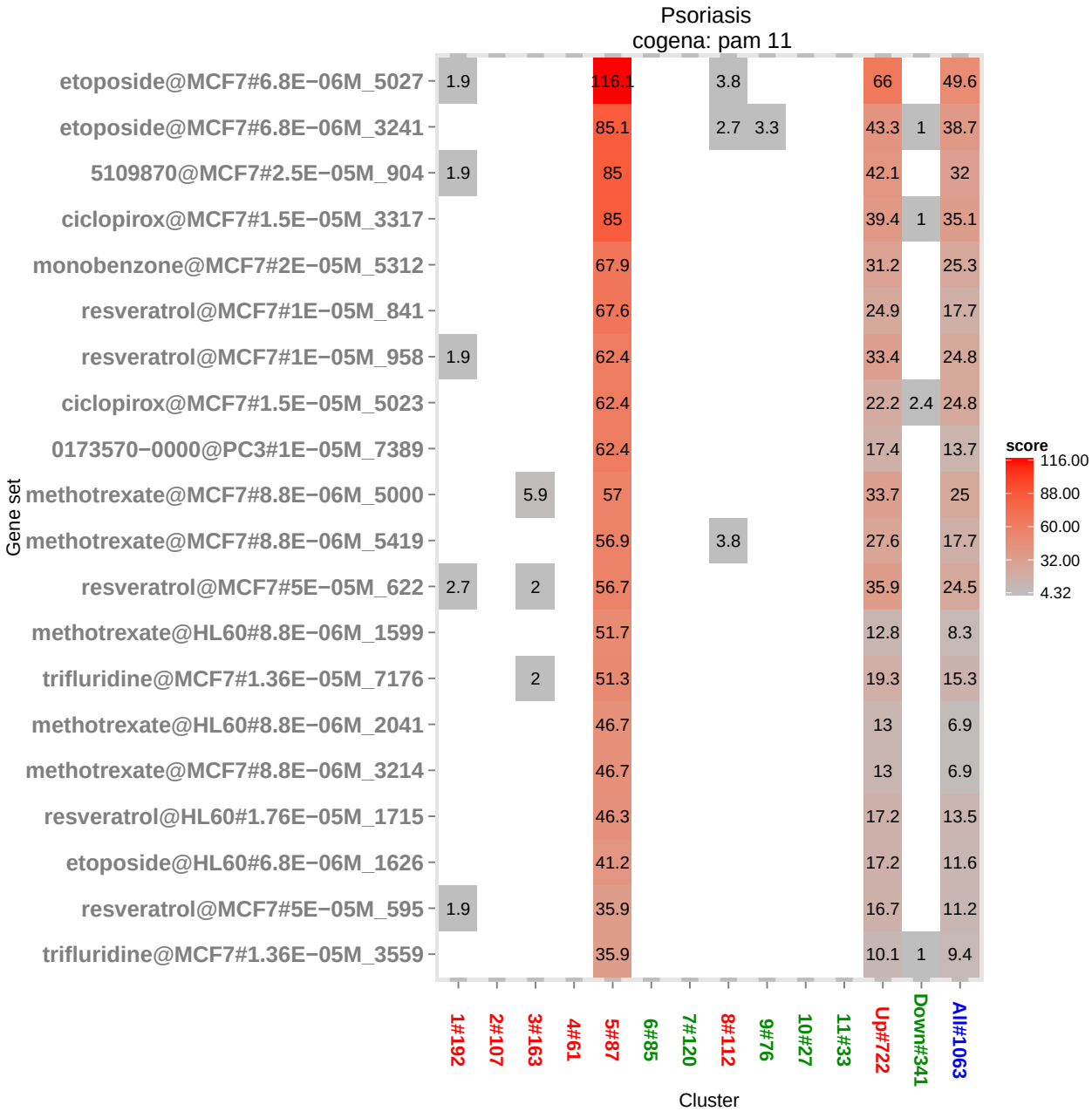

Figure 5: Drug Repositioning for cluster 5

5.4 Figure : Drug repositioning for cluster 10

```
# Figure 6
heatmapPEI(cmapUp100_cogena_result, "pam", "11", printGS=FALSE,
           orderMethod = "10", maintitle="Psoriasis")
```

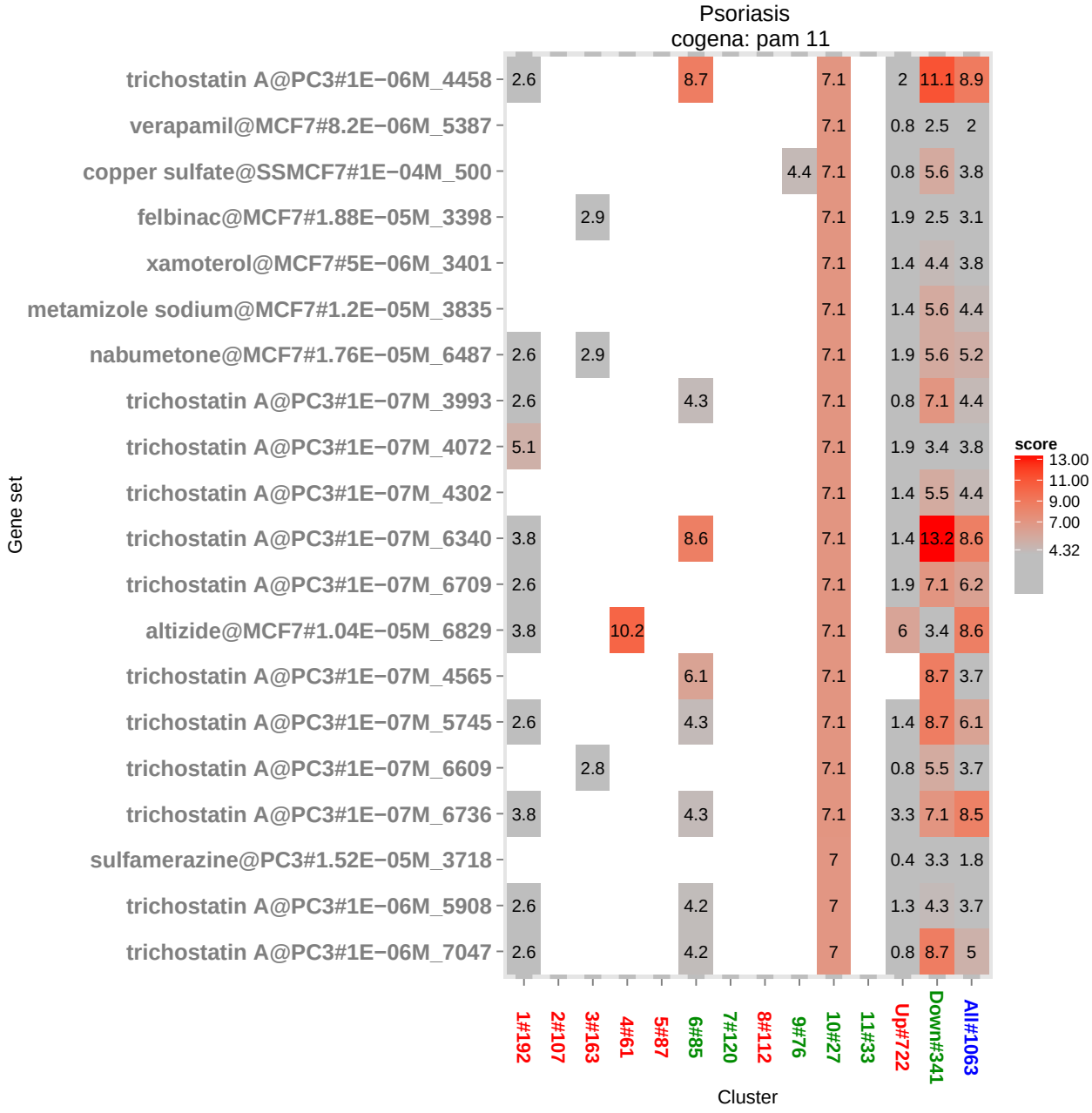

Figure 6: Drug Repositioning for cluster 10

5.5 Output DEGs for CMAP and NFFinder Analysis

The input files for CMap and NFFinder , outputed by this chunk, are in *result/CMAP_input/* and *result/NFFinder_input/* respectively. Please visit CMap and NFFinder to get the final results (*Table S2*) by yourself. Or you can find the results in *result/CMAP_output/* and *result/NFFinder_output/* respectively

```
# Convert gene symbols to probes in HGU133a.
symbol2Probe <- function(gs){
    library(hgu133a.db)
    p <- AnnotationDbi::select(hgu133a.db, gs, "PROBEID", "SYMBOL")$PROBEID
    p <- unique(p[which(!is.na(p))])
}

upGene <- rownames(GSE30999.limma[GSE30999.limma$logFC>= 1 & GSE30999.limma$adj.P.Val<=0.05,])
dnGene <- rownames(GSE30999.limma[GSE30999.limma$logFC<= -1 & GSE30999.limma$adj.P.Val<=0.05,])
upProbe <- symbol2Probe(upGene)
dnProbe <- symbol2Probe(dnGene)

# 1000 probe limitation of CMap
upProbe <- upProbe[1:(1000-length(dnProbe))]

################################################################################
# Output files for CMap and NFFinder
write.table(upProbe, file=paste0("../result/CMAP_input/", "GSE30999_Up.grp"),
            quote=F, col.names = F, row.names = F)
write.table(dnProbe, file=paste0("../result/CMAP_input/", "GSE30999_Dn.grp"),
            quote=F, col.names = F, row.names = F)
write.table(upGene, file=paste0("../result/NFFinder_input/", "GSE30999_Up.txt"),
            quote=F, col.names = F, row.names = F)
write.table(dnGene, file=paste0("../result/NFFinder_input/", "GSE30999_Dn.txt"),
            quote=F, col.names = F, row.names = F)
################################################################################

# save.image(file="../result/cogena_GSE30999.RData")
################################################################################
```

6 Website, BugReports and System Info

- Website: https://github.com/zhilongjia/psoriasis
- BugReports: https://github.com/zhilongjia/psoriasis/issues

```
sessionInfo()
```

```
## R version 3.2.0 (2015-04-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux jessie/sid
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
```

```
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] stats4    tools     parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] hgu133a.db_3.2.2      STRINGdb_1.10.0       hash_2.2.6
##  [4] gplots_2.17.0         RColorBrewer_1.1-2    plotrix_3.6
##  [7] RCurl_1.95-4.7        bitops_1.0-6          igraph_1.0.1
## [10] plyr_1.8.3            sqldf_0.4-10          gsubfn_0.6-6
## [13] proto_0.3-10          png_0.1-7             cogena_1.5.1
## [16] kohonen_2.0.19        MASS_7.3-45           class_7.3-14
## [19] ggplot2_1.0.1         cluster_2.0.3         limma_3.26.3
## [22] hgu133plus2.db_3.2.2  org.Hs.eg.db_3.2.3    RSQLite_1.0.9000
## [25] DBI_0.3.1.9008        annotate_1.48.0       XML_3.98-1.3
## [28] AnnotationDbi_1.32.0  GenomeInfoDb_1.6.1    IRanges_2.4.4
## [31] S4Vectors_0.8.3       MetaDE_1.0.5          combinat_0.0-8
## [34] impute_1.44.0         survival_2.38-3       hgu133plus2cdf_2.18.0
## [37] affy_1.48.0           GEOquery_2.36.0       Biobase_2.30.0
## [40] BiocGenerics_0.16.1
##
## loaded via a namespace (and not attached):
##  [1] devtools_1.9.1        doParallel_1.0.10     R6_2.1.1
##  [4] affyio_1.40.0         KernSmooth_2.23-15    lazyeval_0.1.10.9000
##  [7] colorspace_1.2-6      compiler_3.2.0        preprocessCore_1.32.0
## [10] chron_2.3-47          biwt_1.0              formatR_1.2.1
## [13] caTools_1.17.1        scales_0.3.0          DEoptimR_1.0-4
## [16] mvtnorm_1.0-3         robustbase_0.92-5     stringr_1.0.0
## [19] apcluster_1.4.1       digest_0.6.8          rmarkdown_0.8.1
## [22] rrcov_1.3-8           htmltools_0.2.6       highr_0.5.1
## [25] BiocInstaller_1.20.1  mclust_5.1            gtools_3.5.0
## [28] dplyr_0.4.3.9000      magrittr_1.5          Matrix_1.2-3
## [31] Rcpp_0.12.2           munsell_0.4.2         stringi_1.0-1
## [34] yaml_2.1.13           zlibbioc_1.16.0       grid_3.2.0
## [37] gdata_2.17.0          lattice_0.20-33       splines_3.2.0
## [40] knitr_1.11            tcltk_3.2.0           fastcluster_1.1.16
## [43] reshape2_1.4.1        codetools_0.2-14      evaluate_0.8
## [46] foreach_1.4.3         gtable_0.1.2          amap_0.8-14
## [49] assertthat_0.1        xtable_1.8-0          pcaPP_1.9-60
## [52] iterators_1.0.8       memoise_0.2.1         corrplot_0.73
```

Thank you!
