cogena, a workflow for co-expressed gene-set enrichment analysis

Zhilong Jia, Michael R. Barnes

2015-10-13

Contents

1 Abstract

Co-expressed **g**ene-set **en**richment **a**nalysis, cogena, is a workflow for gene set enrichment analysis of co-expressed genes. The co-expression analysis is made based on various clustering methods. And The gene set analysis can be pathway analysis, GO analysis, drug repositioning and so on. The type of analysis cogena can is dependent on the gene set used in cogena. Lots kinds of gene sets are integrated into cogena from Msigdb and Connectivity Map. See the workflow of cogena in Figure 1.

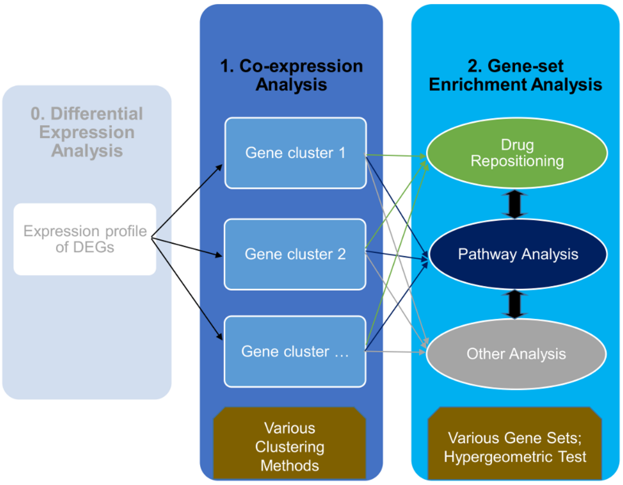
0. Differential Expression Analysis
Expression profile of DEGs
1. Co-expression Analysis
Gene cluster 1
Gene cluster 2
Gene cluster …
Various Clustering Methods
2. Gene-set Enrichment Analysis
Drug Repositioning
Pathway Analysis
Other Analysis
Various Gene Sets; Hypergeometric Test

Figure 1: Overview of the cogena workflow

2 A quick start

See examples in `?cogena` please.

3 Data Input

```
Note: all the gene names should be gene SYMBOLs since they are used in the
gene set files. Other kind of gene names can be used according to
the kind of gene names of user-defined gene-set files.
```

3.1 Input data required

- the gene expression profiling of differentially expressed genes. It can be matrix with gene in row and sample in column, data.frame or ExpressionSet object.
- the sample labels indicating the labels, like control and disease, of each sample. A vector with sample names.

3.2 Example dataset

There is an example dataset in `cogena` package. This dataset, from GSE13355, is about psoriasis. There are two objects in the Psoriasis dataset. See `?Psoriasis` for more details.

```
library(cogena)
data(Psoriasis)
# objects in the Psoriasis dataset.
# Note: lable of interest should be behind of control label as this will affect
# the direction of gene regulation. For instance.
# Use factor(c("Normal", "Cancer", "Normal"), levels=c("Normal", "Cancer")),
# instead of factor(c("Normal", "Cancer","Normal")) since "Cancer" is the
# lable of interest.
```

```
## [1] "DEexprs"     "sampleLabel"
```

4 Various Analyses

4.1 What kind of analysis can be done?

The gene set used determines the type of analysis.

There are a variety of gene sets in the `cogena` package, partly collected from MSigDB and CMap. They are ***CmapDn100.gmt.xz***, ***CmapUp100.gmt.xz***, ***c2.cp.biocarta.v5.0.symbols.gmt.xz***, ***c2.cp.kegg.v5.0.symbols.gmt.xz***, ***c2.cp.reactome.v5.0.symbols.gmt.xz***, ***c2.cp.v5.0.symbols.gmt.xz***, ***c5.bp.v5.0.symbols.gmt.xz***, ***c5.mf.v5.0.symbols.gmt.xz***. Here the gene-sets can be user-defined gene-sets formatted gmt and compressed by xz, too (such as c2.cp.kegg.v5.0.symbols.gmt.xz). Copy it into the extdata directory in the installation directory of cogena, such as ~/R/x86_64-pc-linux-gnu-library/3.2/cogena/extdata in Linux), or kindly send to the author of cogena to share with others.

4.2 Types of analyses

- Pahtway Analysis
- GO Analysis
- Drug repositioning
- ...

5 Pathway Analysis

KEGG Pathway Analysis, as an example, will be done to show the utility of cogena.

5.1 Parameter setting

```
# KEGG Pathway gene set
annoGMT <- "c2.cp.kegg.v5.0.symbols.gmt.xz"
# GO biological process gene set
# annoGMT <- "c5.bp.v5.0.symbols.gmt.xz"
```

```r
annofile <- system.file("extdata", annoGMT, package="cogena")

# the number of clusters. It can be a vector.
# nClust <- 2:20
nClust <- 10

# the number of cores.
# ncore <- 8
ncore <- 2

# the clustering methods
# clMethods <- c("hierarchical","kmeans","diana","fanny","som","model",
# "sota","pam","clara","agnes") # All the methods can be used together.
clMethods <- c("hierarchical","pam")

# the distance metric
metric <- "correlation"

# the agglomeration method used for hierarchical clustering
# (hierarchical and agnes)
method <- "complete"
```

5.2 Cogena running

```r
# Co-expression Analysis
genecl_result <- coExp(DEexprs, nClust=nClust, clMethods=clMethods,
                       metric=metric, method=method, ncore=ncore, verbose=TRUE)

# Enrichment (Pathway) analysis for the co-expressed genes
clen_res <- clEnrich(genecl_result, annofile=annofile, sampleLabel=sampleLabel)
```

5.3 Results of pathway analysis

5.3.1 Summary of cogena result

After completing the cogena analysis, user can use `summary` to see the summary of the result of cogena. And `enrichment` caculate the enrichment score of certain clustering methods and certain number of cluster.

Which clustering method and the number of cluster should be choose? There is no an automatical way. Here we show some principles:

- Different clusters should account for different gene sets.
- A gene set should enirched only in one cluster but not two or more.
- The number of gene in a gene set enriched cluster should be as smaller as it can when the enrichment score is high.

```r
summary(clen_res)
```

```
##
## Clustering Methods:
```

```
##  hierarchical pam
##
## The Number of Clusters:
##  10
##
## Metric of Distance Matrix:
##  correlation
##
## Agglomeration method for hierarchical clustering (hclust and agnes):
##  complete
##
## Gene set:
##  c2.cp.kegg.v5.0.symbols.gmt.xz
```

```
# Here we consider the "pam" method and 10 clusters.
# Always make the number as character, please!
enrichment.table <- enrichment(clen_res, "pam", "10")
```

5.3.2 Heatmap of expression profiling with clusters

`heatmapCluster` is developed to show it. Based on Figure 2, it is obvious to know which cluster contains up-regulated or down-regulated genes.

```
# Always make the number as character, please!
heatmapCluster(clen_res, "pam", "10", maintitle="Psoriasis")
```

```
## The number of genes in each cluster:
## upDownGene
##   1   2
## 468 238
## cluster_size
##   1   2   3   4   5   6   7   8   9  10
## 158  65  38  92  50  67  63  94  61  18
```

5.3.3 Enrichment heatmap of co-expressed genes

`heatmapPEI` can be used to show the enrichment graph. See Figure 3. See `?heatmapPEI` for more details.

```
# The enrichment score for 10 clusters, together with Down-regulated,
# Up-regualted and All DE genes. The values shown in Figure 2 is the -log2(FDR).
#
# Always make the number as character, please!
heatmapPEI(clen_res, "pam", "10", printGS=FALSE, maintitle="Pathway analysis for Psoriasis")
```

5.4 Others

5.4.1 Querying genes in a certain cluster

User can obtain the genes in a certain cluster via `geneInCluster`.

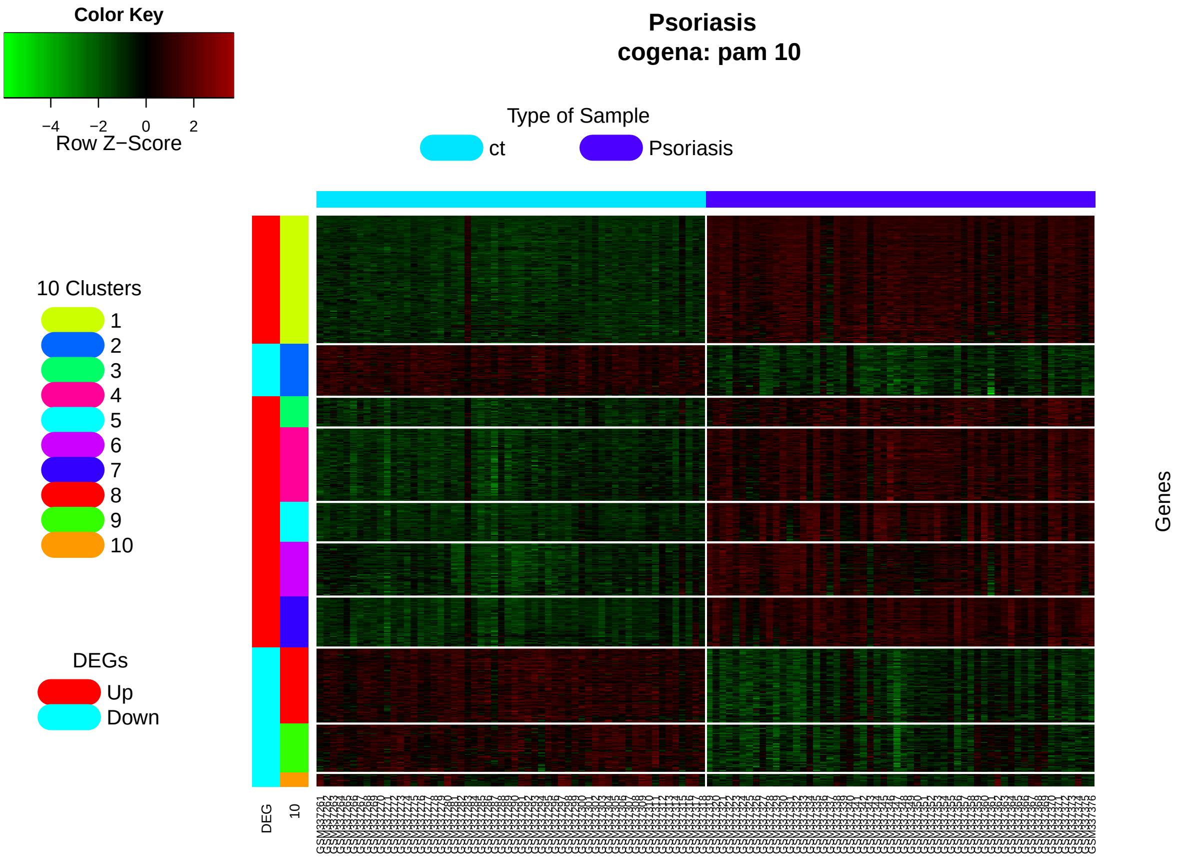

Figure 2: Heatmap of expression profiling with clusters

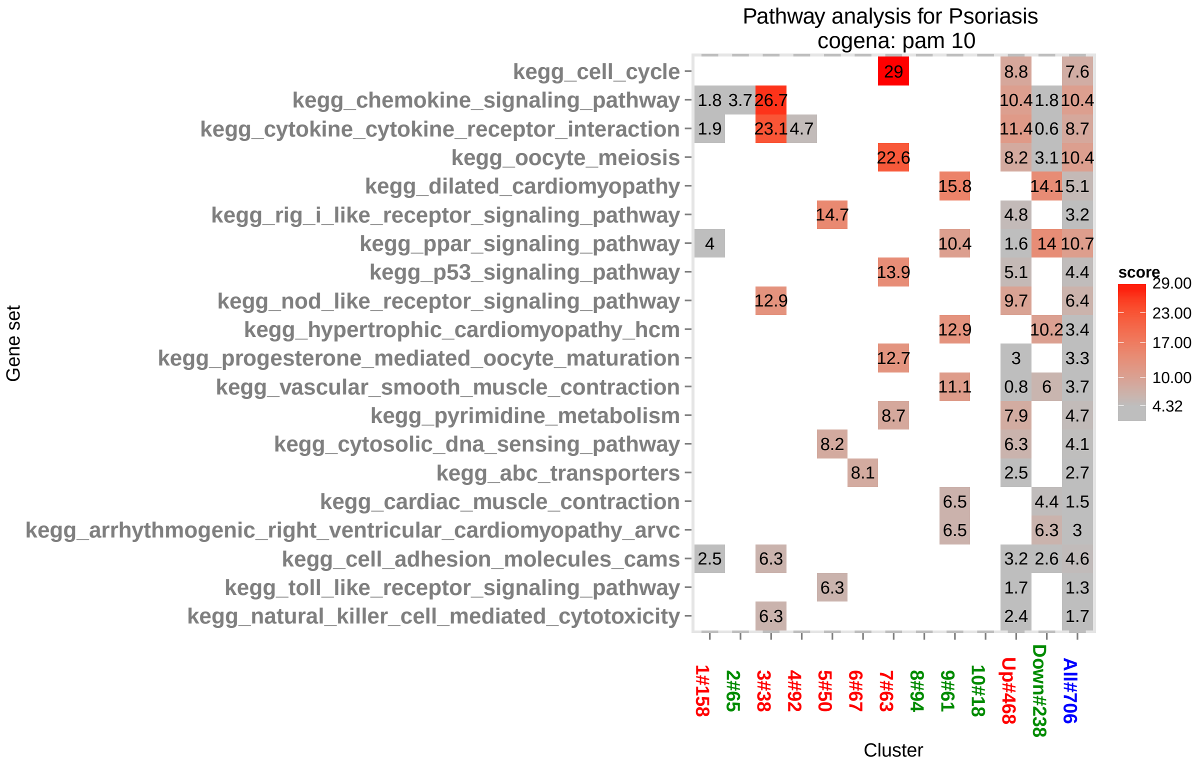

Figure 3: KEGG pathway enrichment

```
# Always make the number as character, please!
geneC <- geneInCluster(clen_res, "pam", "10", "4")
head(geneC)
```

```
## [1] "CD47"      "SERPINB13" "PNP"       "MPZL2"     "KCNJ15"    "SOX7"
```

5.4.2 Gene expression profiling with cluster infromation

It can be obtained by `geneExpInCluster`. There are two items, `clusterGeneExp` and `label`, in the returned object of `geneExpInCluster`. It can be used for other application.

```
# Always make the number as character, please!
gec <- geneExpInCluster(clen_res, "pam", "10")
gec$clusterGeneExp[1:3, 1:4]
```

```
##           cluster_id GSM337261 GSM337262 GSM337263
## PI3                1  6.556556  6.040479  7.033708
## S100A7A            1  4.989918  4.971686  5.677227
## S100A12            1  4.873823  5.168421  5.255036
```

```
gec$label[1:4]
```

```
## GSM337261 GSM337262 GSM337263 GSM337264
##        ct         ct         ct         ct
## Levels: ct Psoriasis
```

5.4.3 The gene correlation in a cluster

The correlation among a cluster can be checked and shown by `corInCluster`. See Figure 4.

```
# Always make the number as character, please!
corInCluster(clen_res, "pam", "10", "10")
```

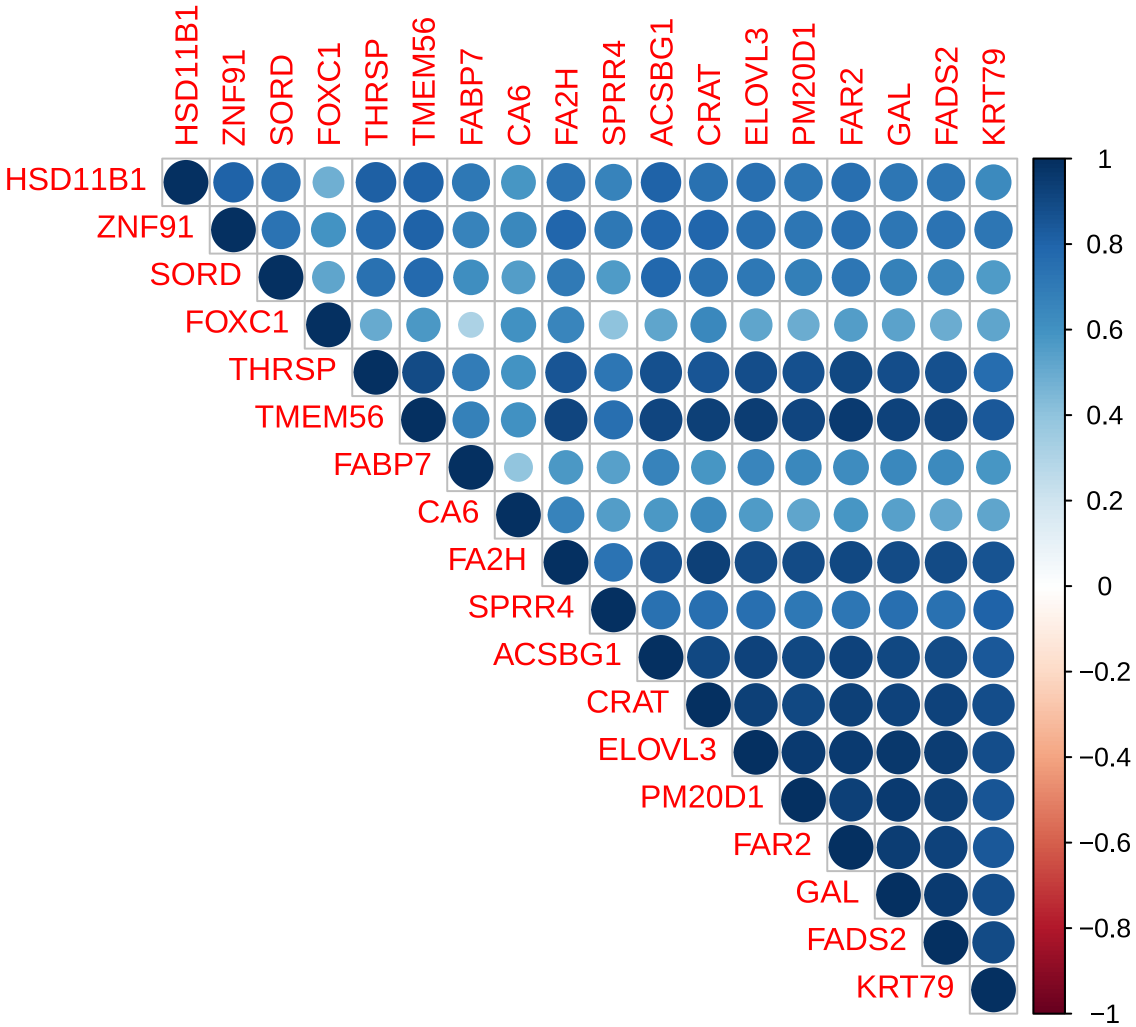

Figure 4: Correlation of genes in a cluster

6 Drug repositioning

Drug repositioning is made based on co-expressed genes instead of all the differentially expressed genes. If the input of cogena is a disease related data, the drugs enriched should recovery the gene expression changed by the disease, while if the input is drug related, the enriched drugs should show similar gene expression changes caused by the drug studied. Here we show drugs for treating psoriasis, an autoimmune disease.

6.1 Drug repositioning analysis running

The drug repositioning gene set choice of *CmapDn100.gmt.xz* or *CmapUp100.gmt.xz* should be made based on the regulation direction of clusters. For example, as the 7th cluster contains up-regulated genes for psoriasis, the *CmapDn100.gmt.xz* is choosen for drug repositioning of psoriasis to recovery the gene expression changes caused by the disease.

```
# A comprehensive way
# cmapDn100_cogena_result <- clEnrich(genecl_result,
# annofile=system.file("extdata", "CmapDn100.gmt.xz", package="cogena"),
# sampleLabel=sampleLabel)

# A quick way
# Based on the pathway analysis results
cmapDn100_cogena_result <- clEnrich_one(genecl_result, "pam", "10",
    annofile=system.file("extdata", "CmapDn100.gmt.xz", package="cogena"),
    sampleLabel=sampleLabel)
```

6.2 Original result of drug repositioning

`Showing the results ordered by the 7th cluster in Figure 5.`

```
heatmapPEI(cmapDn100_cogena_result, "pam", "10", printGS=FALSE,
           orderMethod = "7", maintitle="Drug repositioning for Psoriasis")
```

```
# Results based on cluster 5.
# heatmapPEI(cmapDn100_cogena_result, "pam", "10", printGS=FALSE,
#           orderMethod = "5", maintitle="Drug repositioning for Psoriasis")

# Results based on cluster 9, containing down-regulated genes.
# heatmapPEI(cmapUp100_cogena_result, "pam", "10", printGS=FALSE,
#           orderMethod = "9", maintitle="Drug repositioning for Psoriasis")
```

6.3 Multi-instance merged result of drug repositioning

Usually there are more than 1 instances for a drug with different doses or time-points in the Cmap gene set. `heatmapCmap` can merge the multi-instance results based on parameter *mergeMethod* (“mean” or “max”). Figure 6 shows the multi-instance merged results ordered by the 7th cluster.

```
heatmapCmap(cmapDn100_cogena_result, "pam", "10", printGS=FALSE,
            orderMethod = "7", maintitle="Drug repositioning for Psoriasis")
```

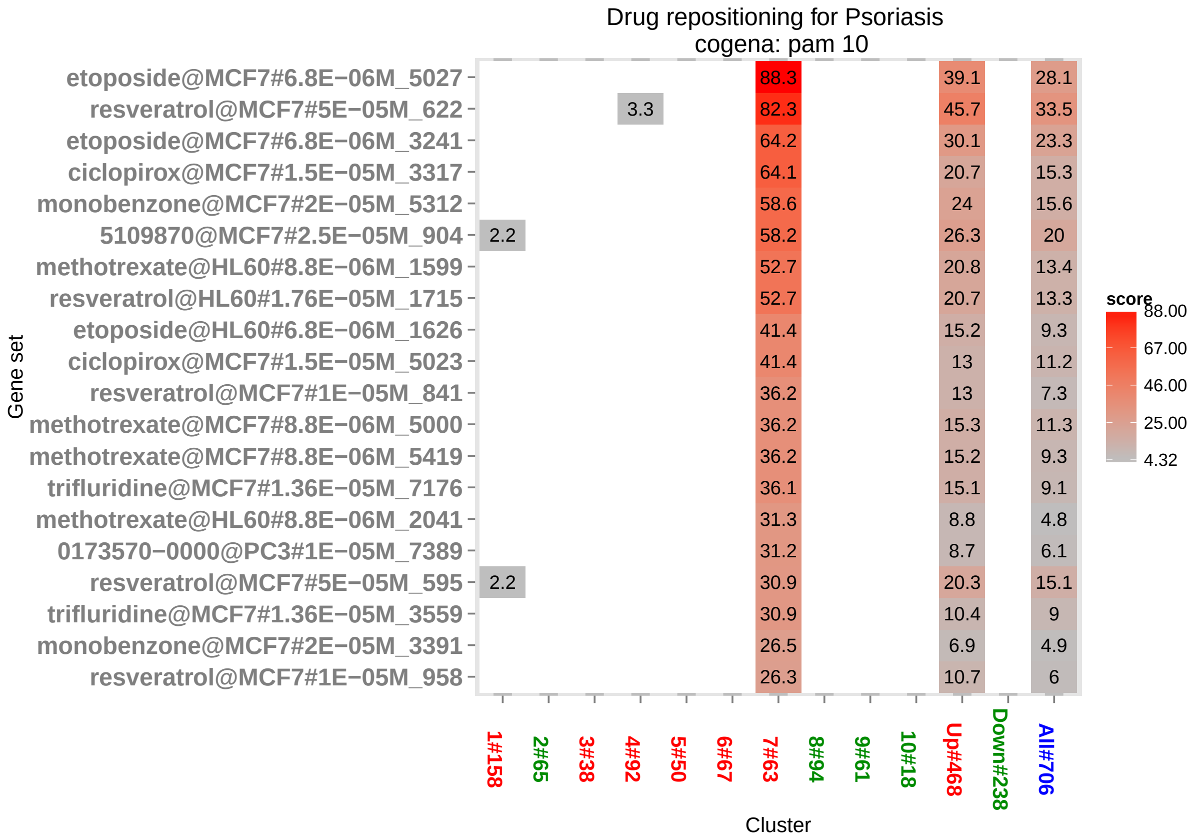

Figure 5: Drug repositioning

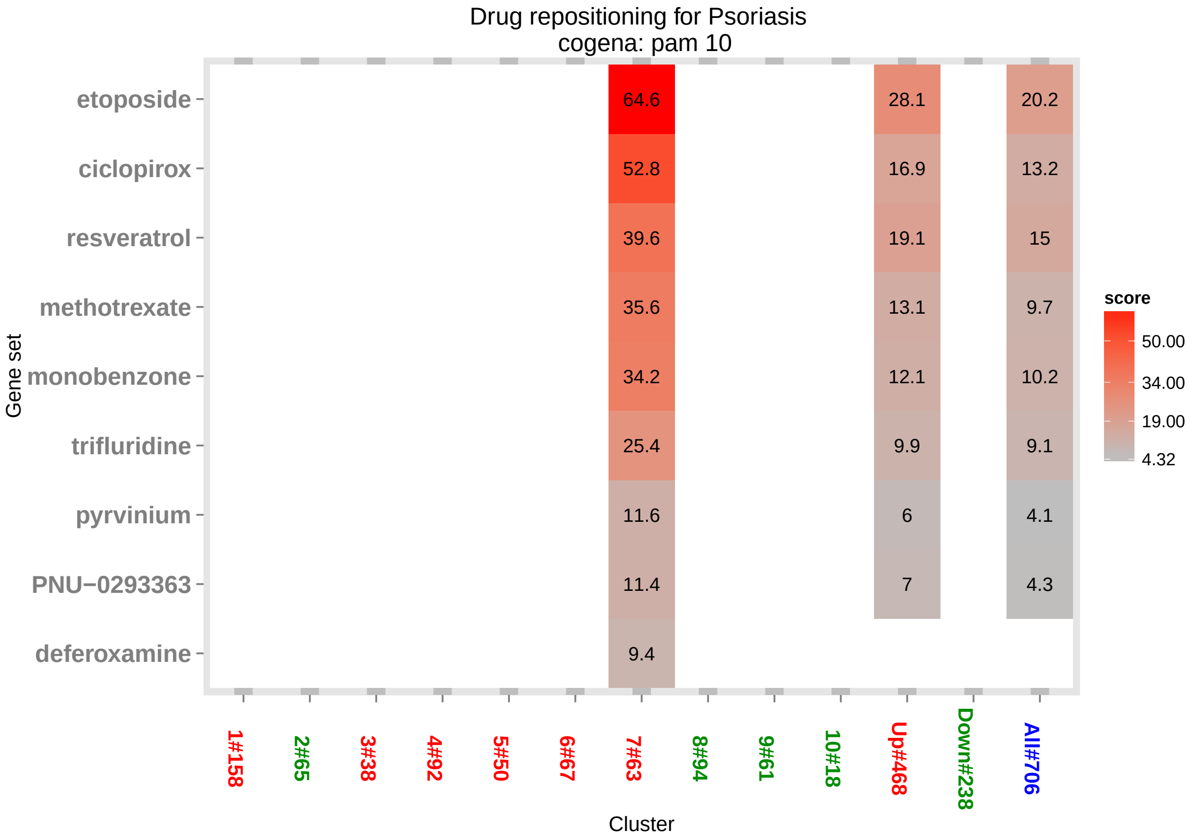

Figure 6: Drug repositioning (multi-instance merged)

7 Bug Report

https://github.com/zhilongjia/cogena/issues

8 Citation

Jia Z. et al. *Cogena, a tool for co-expressed gene-set enrichment analysis and visualization.*

9 Other Information

System info

```
## R version 3.2.2 Patched (2015-08-16 r69094)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server 2012 R2 x64 (build 9600)
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] cogena_1.4.0   kohonen_2.0.19 MASS_7.3-44    class_7.3-14
## [5] ggplot2_1.0.1  cluster_2.0.3
##
## loaded via a namespace (and not attached):
##  [1] fastcluster_1.1.16  gtools_3.5.0        reshape2_1.4.1
##  [4] corrplot_0.73       lattice_0.20-33     pcaPP_1.9-60
##  [7] colorspace_1.2-6    amap_0.8-14         htmltools_0.2.6
## [10] stats4_3.2.2        yaml_2.1.13         DBI_0.3.1
## [13] BiocGenerics_0.16.0 biwt_1.0            foreach_1.4.3
## [16] plyr_1.8.3          robustbase_0.92-5   stringr_1.0.0
## [19] munsell_0.4.2       gtable_0.1.2        devtools_1.9.1
## [22] caTools_1.17.1      mvtnorm_1.0-3       codetools_0.2-14
## [25] memoise_0.2.1       evaluate_0.8        Biobase_2.30.0
## [28] knitr_1.11          doParallel_1.0.8    parallel_3.2.2
## [31] DEoptimR_1.0-3      proto_0.3-10        Rcpp_0.12.1
## [34] KernSmooth_2.23-15  scales_0.3.0        formatR_1.2.1
## [37] gdata_2.17.0        apcluster_1.4.1     gplots_2.17.0
## [40] digest_0.6.8        stringi_0.5-5       dplyr_0.4.3
## [43] grid_3.2.2          tools_3.2.2         bitops_1.0-6
## [46] magrittr_1.5        lazyeval_0.1.10     rrcov_1.3-8
## [49] Matrix_1.2-2        assertthat_0.1      rmarkdown_0.8.1
## [52] iterators_1.0.8     R6_2.1.1            mclust_5.0.2
## [55] compiler_3.2.2
```