cogena, a tool for co-expressed gene-set enrichment analysis and visualization

Zhilong Jia, Michael Barnes

2015-04-16

cogena can be used as gene set enrichment analysis of up or down regulated genes as well as genes in clusters resulting from various clustering methods. And the gene sets could be Pathway, Gene Ontology or user-defined gene-sets. Accordingly, this tool can be used for pathway enrichment, GO enrichment and so on. See the workflow of cogena in Figure 1.

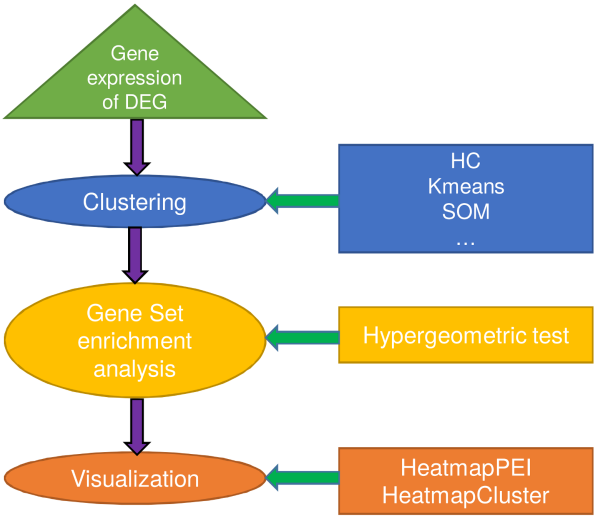

Figure 1: Overview of the cogena workflow

Input data

```
Note: all the gene names should be gene SYMBOL since it is used in the
gene dataset files. Other kind of gene names could be used according to
the kind of gene names of user-defined gene-set files.
```

1. Input data needed.

- 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.

2. Example dataset.

   There is an example data in `cogena` package. This dataset, from GSE20163, is about Parkinson's disease. There are three objects in the PD dataset. See `?PD` for more details.

```
library(cogena)
data(PD)
# Three objects in the PD dataset and one in AllGeneSymbols dataset.
```

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

cogena analysis

1. Make Annotation

   There are a variety of gene sets in the `cogena` package, collected from MSigDB. They are c2.cp.kegg.v4.0.symbols.gmt, c5.bp.v4.0.symbols.gmt. with gene-sets, user can get the genes-specific gene sets matrix as showed in the following example. Here the gene-sets can be user-defined gene-sets formatted gmt, too.

```
annoGMT <- "c2.cp.kegg.v4.0.symbols.gmt"
annofile <- system.file("extdata", annoGMT, package="cogena")
```

2. Run cogena

   Setting some parameters for cogena and running cogena analysis.

```
# the number of clusters. It could be a vector.
nClust <- 2:8
# the number of cores.
ncore <- 2
# the clustering methods
clMethods <- c("hierarchical","kmeans")
# the distance metric
metric <- "correlation"
# the agglomeration method used for hierarchical clustering
#(hierarchical and agnes)
method <- "complete"

# the cogena analysis
cogena_result <- cogena(DEexprs, nClust=nClust, clMethods=clMethods,
    metric=metric, method=method,  annofile=annofile,
    sampleLabel=sampleLabel, ncore=ncore, verbose=TRUE)
```

Analysing results of cogena

1. After completing the cogena analysis, user can use `summary` to see the summary of the result of cogena. Since there are so many options of results, user can `optCluster` caculate wchih clusering methods and the number of clusters are optimum, and `enrichment` caculate the enrichment score of certain clustering methods and certain number of cluster.

   As so many clustering methods and a range of number of clusters available, by setting the threshold of adjusted p-value as 0.05 so as to obtain the number of significant gene sets, we recommend that one can choose the clustering methods and the number of clusters, which is intuitively considered as the optimum, on which the number of significant gene sets are local maximum. In other words, with the cut-off of adjusted p-value, the optimum should have as many as possible gene sets enriched in a local range of number of clusters. The reason why choosing local maximum is largely because when the

number of cluster is large and all clusters have more gene sets enriched, some clusters are generated from cutting a cluster from smaller number of clusters, which are highly enriched with gene sets, into more clusters. We consider the number of significant gene sets with the cut-off 0.05 of p-value as the score. Moreover, for some clustering methods with a certain number of clusters, if over 75% clusters possess a mixed up-regulated and down-regulated genes, the score (except not available (NA)) will be changed to inverse number. In addition, if the number of cluster is 2 and there are mixed up-regulated and down-regulated genes in each clusters, the score (if positive and except not available (NA)) of the other number of cluster for the same clustering method will be changed to inverse number.

```
summary(cogena_result)
```

```
##
## Clustering Methods:
##  hierarchical kmeans
##
## The Number of Clusters:
##  2 3 4 5 6 7 8
##
## Metric of Distance Matrix:
##  correlation
##
## Agglomeration method for hierarchical
##         clustering (hclust and agnes):
##  complete
```

```
score <- optCluster(cogena_result, ncores=1)
#choose optimun based on cluster I.
#score <- optCluster(cogena_result, based="I", ncores=1)
```

	2	3	4	5	6	7	8
hierarchical	25	31	32	33	34	32	32
kmeans	25	31	31	29	32	30	30

```
#Here we consider the "hierarchical" method and 6 clusters as the optimum.
#Always make the number as character, please!
enrichment.table <- enrichment(cogena_result, "hierarchical", "6")
```

2. To show the enrichment graph, `heatmapPEI` can be used. see Figure 2. See ?heatmapPEI for more details.

```
#The values shown in Figure 2. is the -log2 (p-values).
#Always make the number as character, please!
heatmapPEI(cogena_result, "hierarchical", "6", printGS=FALSE)
```

```
#ordered by the Cluster I.
#heatmapPEI(cogena_result, "hierarchical", "6", orderMethod = "I")
```

3. The heatmap of expression profiling with clusters is plotted by `heatmapCluster`. see Figure 3. User can know which cluster contains up-regulated or down-regulated genes.

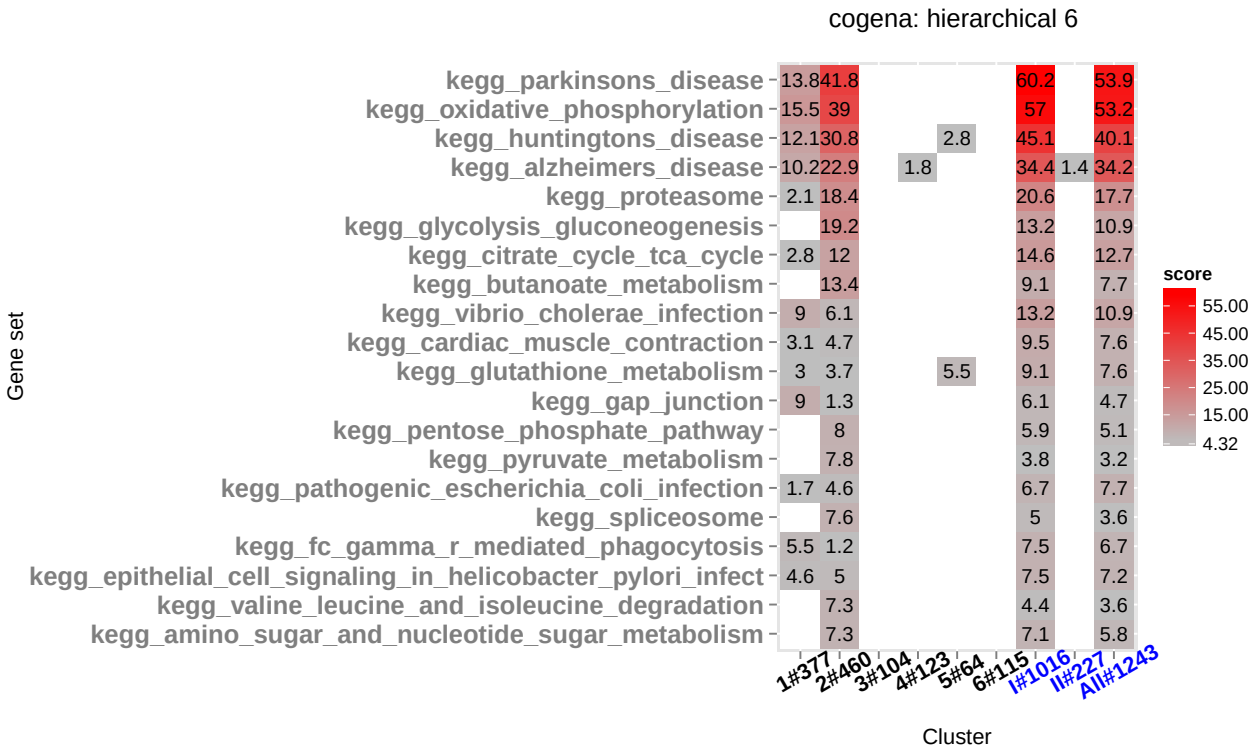

Figure 2: KEGG pathway enrichment

```
#Always make the number as character, please!
heatmapCluster(cogena_result, "hierarchical", "6")
```

```
## The number of genes in each cluster:
## cluster_size2
##    1    2
## 1016  227
## cluster_size
##   1   2   3   4   5   6
## 377 460 104 123  64 115
```

4. If interested genes in a certain cluster, user can obtain the genes in a certain cluster via `geneInCluster`.

```
#Always make the number as character, please!
head(geneInCluster(cogena_result, "hierarchical", "6", "2"))
```

```
## [1] "YWHAZ"    "ALDH1A1"  "NDUFA9"   "ABCA3"    "ATP6V0D1" "HMGB3"
```

5. The gene expression profilings with cluster infromation could be obtained by `geneExpInCluster`. There are two items, `clusterGeneExp` and `label`, in the returned object of `geneExpInCluster`. This can be used for other application.

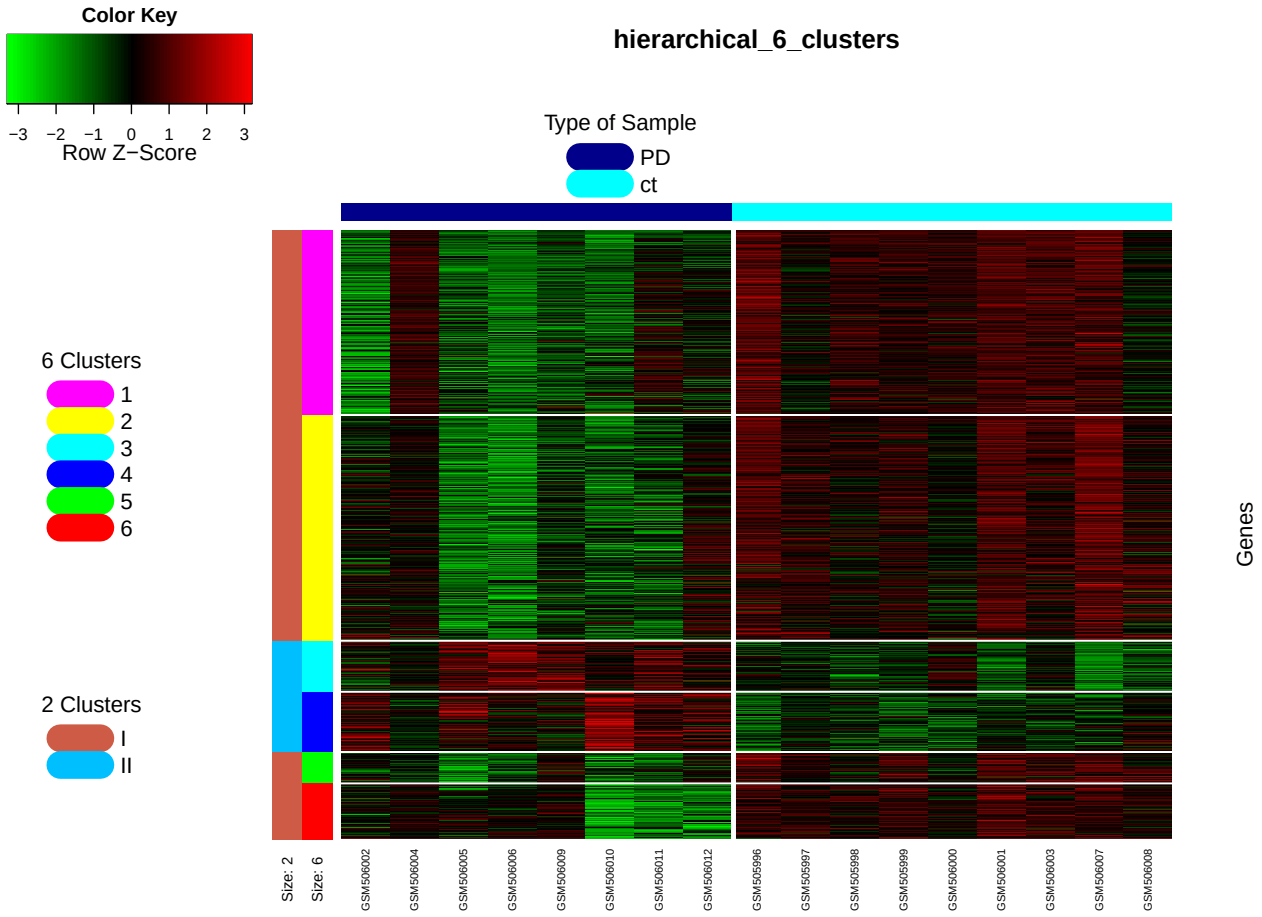

Figure 3: Heatmap of expression profiling with clusters

```
#Always make the number as character, please!
gec <- geneExpInCluster(cogena_result, "hierarchical", "6")
head(gec$clusterGeneExp, 3)
```

```
##        cluster_id GSM505996 GSM505997 GSM505998 GSM505999 GSM506000
## SV2C            1  12.75816  12.85066  12.58651  12.93088  12.57228
## AGTR1           1  11.78429  10.88479  12.23727  11.43515  10.93398
## SLC6A3          1  13.15431  13.14876  13.24905  13.26366  13.08076
##        GSM506001 GSM506002 GSM506003 GSM506004 GSM506005 GSM506006
## SV2C    13.94984  9.466624  12.86536  12.56857 10.768014  9.479922
## AGTR1   11.93646  9.863329  12.08112  11.12530  9.816477  9.802056
## SLC6A3  13.40420 10.578950  13.45746  13.07213 11.612758 10.185610
##        GSM506007 GSM506008 GSM506009 GSM506010 GSM506011 GSM506012
## SV2C    13.64374  12.41889 10.763702  9.104793  9.488126  9.584942
## AGTR1   12.87570  11.09241  9.891531 10.207929  9.706585 10.328377
## SLC6A3  13.79151  13.07574 11.863556 11.196894 10.847975 12.368082
```

```
gec$label
```

```
## GSM505996 GSM505997 GSM505998 GSM505999 GSM506000 GSM506001 GSM506002
##        ct        ct        ct        ct        ct        ct        PD
## GSM506003 GSM506004 GSM506005 GSM506006 GSM506007 GSM506008 GSM506009
##        ct        PD        PD        PD        ct        ct        PD
## GSM506010 GSM506011 GSM506012
##        PD        PD        PD
## Levels: PD ct
```

6. The correlation among one cluster could be shown via `corInCluster`. see Figure 4.

```
#Always make the number as character, please!
corInCluster(cogena_result, "hierarchical", "8", "8")
```

Bug Report

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

Citation

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

Other Information

System info

```
## R version 3.2.0 (2015-04-16)
## Platform: x86_64-unknown-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.2 LTS
##
```

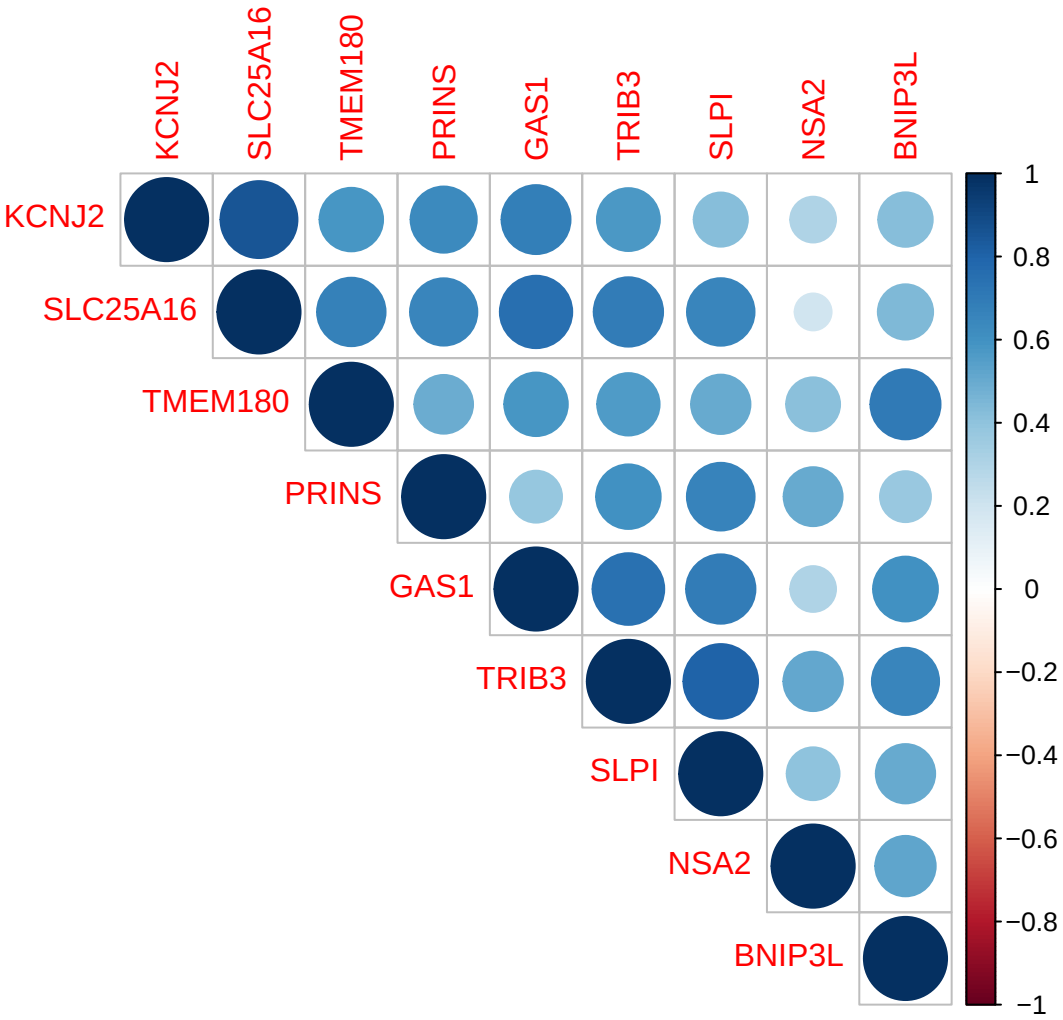

Figure 4: Correlation

```
## 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] cogena_1.0.0  amap_0.8-14   gplots_2.16.0 ggplot2_1.0.1 cluster_2.0.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.11.5        knitr_1.9          devtools_1.7.0
##  [4] MASS_7.3-40        doParallel_1.0.8   munsell_0.4.2
##  [7] colorspace_1.2-6   foreach_1.4.2      fastcluster_1.1.16
## [10] stringr_0.6.2      plyr_1.8.1         caTools_1.17.1
## [13] tools_3.2.0        corrplot_0.73      parallel_3.2.0
## [16] grid_3.2.0         gtable_0.1.2       KernSmooth_2.23-14
## [19] iterators_1.0.7    class_7.3-12       htmltools_0.2.6
## [22] gtools_3.4.2       yaml_2.1.13        digest_0.6.8
## [25] reshape2_1.4.1     formatR_1.1        codetools_0.2-11
## [28] bitops_1.0-6       evaluate_0.6       rmarkdown_0.5.1
## [31] gdata_2.13.3       compiler_3.2.0     scales_0.2.4
## [34] proto_0.3-10
```

The END