GScluster (v1.1.6)

- *User's Manual -*

Author: Sora Yoon <yoonsora1@unist.ac.kr>
Maintainer: Jinhwan Kim <kjh0530@unist.ac.kr>

Last updated: 2019. 2. 28.

1. Introduction

GScluster is a Shiny R package that performs network-weighted gene-set clustering by incorporating both gene-set overlap and protein-protein interaction (PPI) networks. Gene-set enrichment analysis of complex diseases often yields a long list of dysregulated pathways in disease. GScluster can cluster such pathways and help users summarize them. Built on Shiny package, it provides interactive user-interface so that users can easily navigate the gene-set and each cluster's gene network. GScluster provides PPI networks for ten species such as human, mouse, rat, arabidopsis, C.elegans, E.Coli, fly, rice, yeast and zebrafish. The installation and usage of GScluster is described below. PPI networks and gene networks are used interchangeably in this manual, because the network data for GScluster can be any kind of functional interactions between genes.

2. Installation (Do only once)

1) Open R program.
2) Type following commands in R console.

```
>> install.packages('devtools')
>> library(devtools)
>> install_github('unistbig/GScluster')
```

3. Launch

1) Load the GScluster R package by typing following line in R console

```
>> library(GScluster)
```

2) Run GScluster using 'GScluster' function. To run the demo, enter GScluster(). Otherwise, modify following example source codes to fit your data.

```
## example code ##
# 1) Read gene-set analysis result table.
>>
GSAresult=read.delim('https://github.com/unistbig/GScluster/raw/master/sample_geneset.txt', stringsAsFactors=FALSE)
# 2) Read gene score table.
>>
GeneScores=read.delim('https://github.com/unistbig/GScluster/raw/master/sample_genescore.txt', header=F)
# 3) Read PPI matrix.
>> download.file(url='https://github.com/unistbig/GScluster-Data/raw/master/human/string.RData', dest='humanPPI.RData')
>> load('humanPPI.Rdata') # It loads a PPI matrix named 'string'
>> PPI=string
# 4) Run GScluster
>> GScluster(GSAresult = GSAresult, GeneScores = GeneScores, PPI = PPI, Species = 'H', alpha = 1, GsQCutoff = 0.25, GQCutoff = 0.05)
```

As shown in the code above, *GScluster* function takes seven arguments as follows:

① **GSAresult**: A data frame representing gene-set analysis result, consisting of three or four columns as follows:

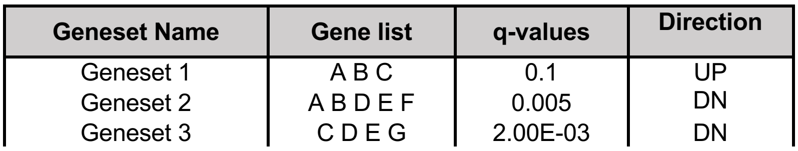
A. **Geneset name:** A character vector of gene-set names

B. **Gene list:** A character vector of gene-set member lists. For each gene-set, genes should be separated by a space (Figure 1).
* Any kind of gene IDs can be used for custom PPI data, but gene symbols should be used for the default STRING.

C. **q-values:** A numeric vector of gene-set q-values (or p-values).

D. **Direction (optional):** A character vector indicating whether each gene-set is up-regulated (UP) or down-regulated (DN).

An example GSA result file is available at:
https://github.com/unistbig/GScluster/raw/master/sample_geneset.txt



Geneset Name	Gene list	q-values	Direction
Geneset 1	A B C	0.1	UP
Geneset 2	A B D E F	0.005	DN
Geneset 3	C D E G	2.00E-03	DN

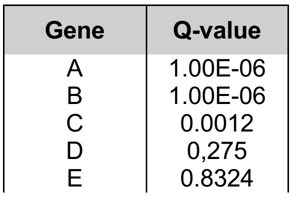
Figure 1. GSA result table format. A header is required.

② **GeneScores:** A data frame of gene scores (e.g., p-value of q-value in differential expression analysis) consisting of two columns as follows:

A. **Gene:** A character vector of gene names.

B. **Scores:** A numeric vector of gene p-values (or q-values). Each value must be between 0 and 1. Exponential formats such as 1E-3 are allowed.

An example gene score file is available at:
https://github.com/unistbig/GScluster/raw/master/sample_genescore.txt

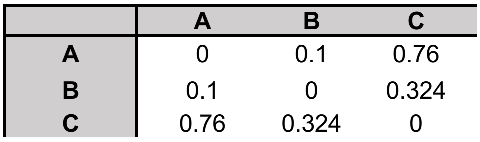


Gene	Q-value
A	1.00E-06
B	1.00E-06
C	0.0012
D	0,275
E	0.8324

Figure 2. Gene score table format. A header is required.

③ **PPI:** A matrix of protein-protein interaction scores consisting of numeric values between 0 and 1. If the '**Species**' parameter is specified as one of the ten species provided by GScluster, the user does not need to specify this option (In this case, STRING PPI network table is automatically loaded).

An example PPI matrix file (an object named 'string' in the .Rdata file) is available at:
https://github.com/unistbig/GScluster-Data/raw/master/human/string.RData



	A	B	C
A	0	0.1	0.76
B	0.1	0	0.324
C	0.76	0.324	0

Figure 3. PPI table format. It is a symmetric matrix with diagonal value of zero.

④ **Species:** One uppercase alphabet representing the species of input data. Default value is "H" to indicate 'human'. Possible values are "A" (Arabidopsis), "C" (C.elegans), "E" (E.coli), "F" (Fly), "H" (Human), "I" (RIce), "M" (Mouse), "R" (Rat), "Y" (Yeast) and "Z" (Zebrafish).

⑤ **alpha:** A network weight for the pMM distance, in the range 0 to 1. The default value is 1. If Alpha is 0, pMM and MM have the same value.

⑥ **GsQCutoff:** A numeric value indicating gene-set q-value cutoff (0~1). The gene-sets with a q-value greater than the cutoff is excluded from clustering.

Default value is 0.25.

⑦ **GQCutoff:** A numeric value indicating gene q-value cutoff (0~1). The genes with a q-value greater than this cutoff is excluded from the graph.

- **TIP**

■ We have developed gene-set analysis tools for

1) GWAS summary data
   (**GSA-SNP2**, download site: https://sourceforge.net/projects/gsasnp2/) and,
2) Gene expression data (RNA-seq or microarray)
   (**GSAseq**, available at http://gsaseq.appex.kr/gsaseq/)

The gene-set analysis result file obtained from GSA-SNP2 or GSAseq can be directly applied to *GScluster* function using **'GetGSASNP2Data'** and **'GetGSASeqData'** functions, respectively. Example codes are as follows:

```
# Case 1: using GSA-SNP2 result file
# 1) Transform GSA-SNP2 result file. 'filename' is specified as the location of the file.
>>
data=GetGSASNP2Data(filename='https://github.com/unistbig/GScluster/raw/master/inst/GScluster/GSASNP2_DIAGRAM.txt')
# Here, the 'data' object is a list of two tables including GSA result ('GSAresult') and gene score table ('GeneScores')
# 2) Get GSA result table.
>> GSAresult = data$GSAresult
# 3) Get Gene score table
>> GeneScores = data$GeneScores
# 4) Run GScluster function
>> GScluster(GSAresult = GSAresult, GeneScores = GeneScores, Species = 'H', alpha = 1, GsQCutoff = 0.25, GQCutoff = 0.05)
```

The GSAseq result file can be applied in a similar way.

```
# Case 2: using GSAseq result file
# 1) Transform the GSAseq result file.
>>
data=GetGSASeqData(filename=https://github.com/unistbig/GScluster/blob/master/inst/GScluster/GSAseq_GSE4107.txt?raw=true')
```

```
# 2) Get GSA result table.
>> GSAresult = data$GSAresult
# 3) Get Gene score table
>> GeneScores = data$GeneScores
# 4) Run GScluster function
>> GScluster(GSAresult = GSAresult, GeneScores = GeneScores, Species = 'H',
alpha = 1, GsQCutoff = 0.01, GQCutoff = 0.01)
```

4. Exploring Networks

After running the GScluster function, the Shiny app will run in a few seconds or minutes and display the gene-set networks (Figure 4). Since the program was developed under Chrome environment, we recommend that the user set Chrome as default browser.

The default options for gene-set clustering are

1) Distance type: pMM (PPI-weighted Meet/Min distance)
2) Distance cutoff: Top x% pMM value. (x: Percentage value corresponding to MM = 0.5)
3) Seed cluster size: 3 (required for fuzzy clustering)

For more information on pMM, please refer to the manuscript.

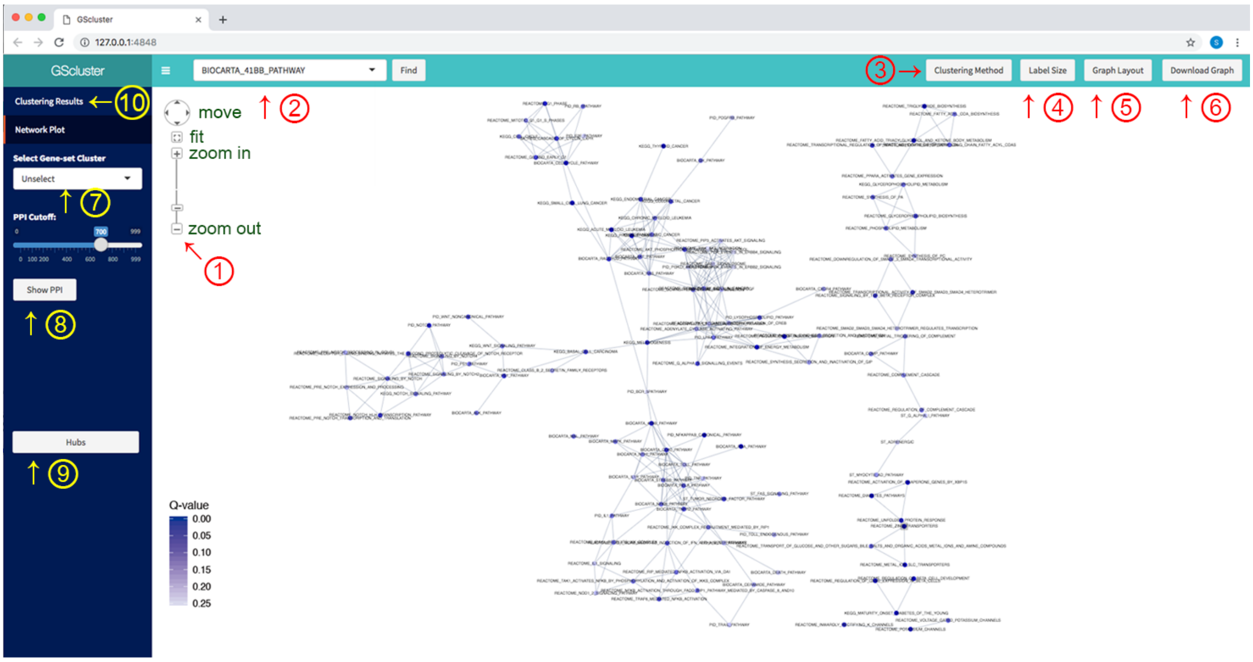


Figure 4. Gene-set networks. The functions or detailed information are described below (numbers 1 – 10).

Useful Functions in GScluster

It provides a number of functions to explore both gene-set and gene networks.

1) **Graph control panel:** By controlling mouse or graph control panel (① in Fig. 4), the user can zoom in and out of the graph, adjust it to fit the screen, or move it.
   Using mouse drag, the user can 1) select multiple nodes on the graph or 2) move the entire graph by holding down mouse button for 2 seconds.

2) **Node (Gene-set or gene) search:** To locate a particular node (gene-set or gene) on the graph, type the keyword in the upper left box (② in Fig. 4) and select the item you want. Then, the node will be highlighted and blink for a few seconds in magenta color on the graph. In case of gene-set node, corresponding cluster number will also appear (Fig. 5).

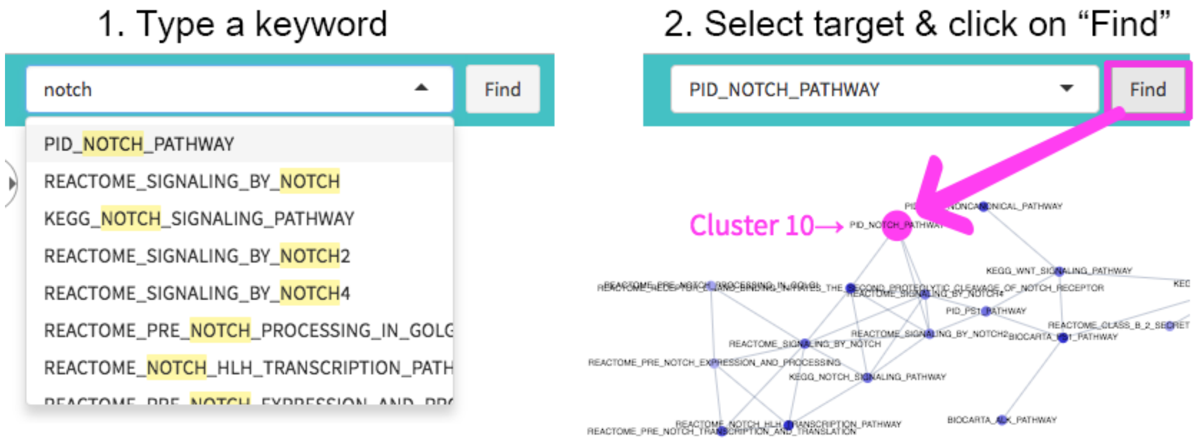

1. Type a keyword
notch
Find
PID_NOTCH_PATHWAY
REACTOME_SIGNALING_BY_NOTCH
KEGG_NOTCH_SIGNALING_PATHWAY
REACTOME_SIGNALING_BY_NOTCH2
REACTOME_SIGNALING_BY_NOTCH4
REACTOME_PRE_NOTCH_PROCESSING_IN_GOLG
REACTOME_NOTCH_HLH_TRANSCRIPTION_PATH
2. Select target & click on "Find"
PID_NOTCH_PATHWAY
Find
Cluster 10⟶
PID_NOTCH_PATHWAY
NONCANONICAL_PATHWAY
KEGG_WNT_SIGNALING_PATHWAY
REACTOME_SIGNALING_BY_NOTCH4
PID_PS1_PATHWAY
REACTOME_CLASS_B_2_SECRET
REACTOME_SIGNALING_BY_NOTCH2
BIOCARTA_PS1_PATHWAY
REACTOME_SIGNALING_BY_NOTCH
REACTOME_PRE_NOTCH_EXPRESSION_AND_PROCESSING
KEGG_NOTCH_SIGNALING_PATHWAY
REACTOME_NOTCH_HLH_TRANSCRIPTION_PATHWAY
REACTOME_PRE_NOTCH_TRANSCRIPTION_AND_TRANSLATION
BIOCARTA_ALK_PATHWAY

Figure 5. Gene-set node search

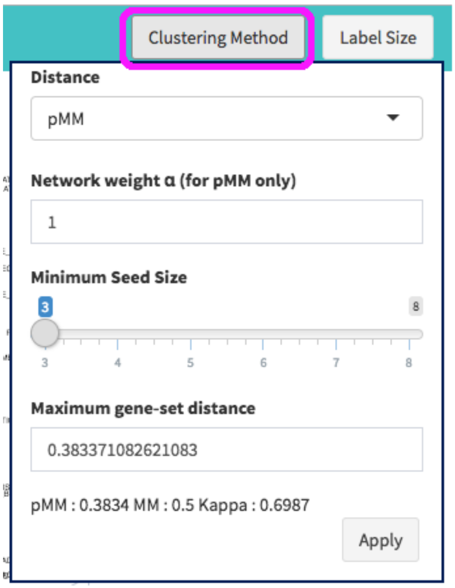
3) **Clustering Method:**

   ① The distance type (meet/min distance (MM), PPI-weighted meet/min distance (pMM) or Cohen's Kappa distance (Kappa))

   ② minimum seed size (= minimum gene-set cluster size in fuzzy clustering) and

   ③ maximum distance between connected gene-sets

   can be set in 'Clustering Method' (③ in fig 4). After setting these parameters, click on **'APPLY' button**.



Distance

pMM

Network weight α (for pMM only)

1

Minimum Seed Size

3 — 8

3 4 5 6 7 8

Maximum gene-set distance

0.383371082621083

pMM : 0.3834 MM : 0.5 Kappa : 0.6987

Figure 6. Clustering options

4) **Label size:** It adjusts the node label size (④ in Figure 4, Figure 7).

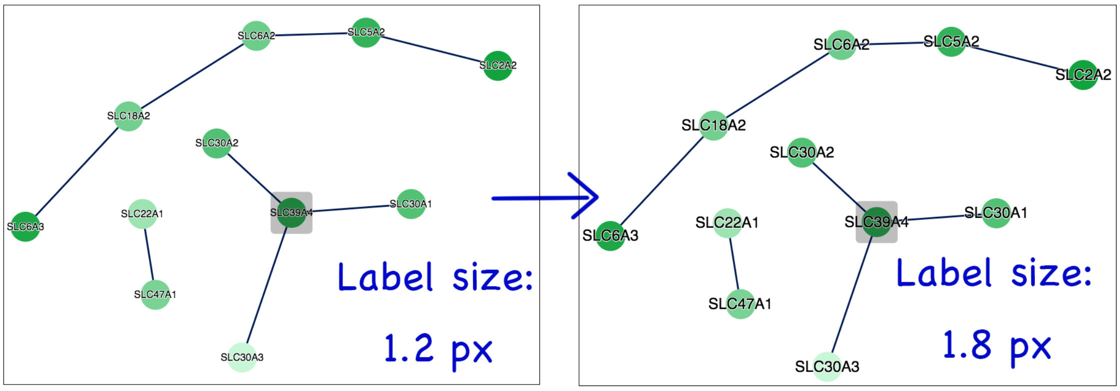


Figure 7. Node label size adjustment

5) **Graph Layout:** Click on the 'Graph layout' button (⑤ in Figure 4) to change the graph layout between 'circle' and 'cola' (Figure 8).
***TIP.** If the clusters do not appear to be fully isolated, click on ‘Graph Layout’ and ‘Cola’ button.

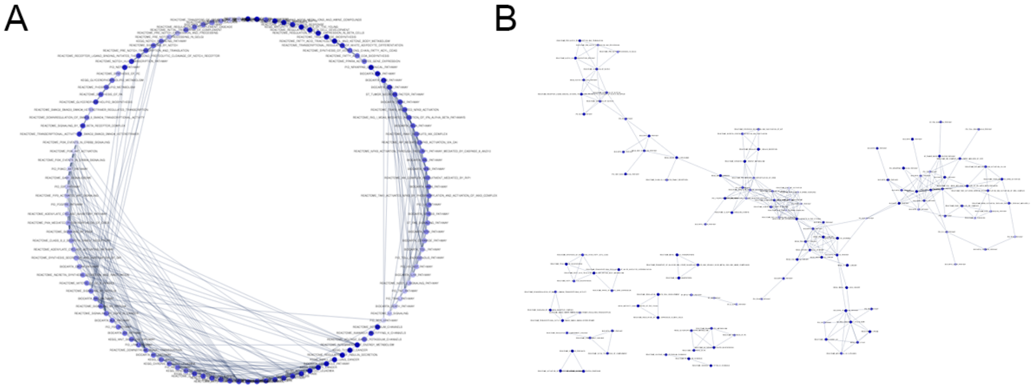


Figure 8. Graph layout. (A) Circle and (B) Cola layout

6) **Download Graph:** The user can download a vector image file (.SVG) for current plot by clicking on the 'Download Graph' button (⑥ in Figure 4).

7) **Highlight gene-set cluster**

The user can highlight a gene-sets cluster by selecting a cluster number from 'Select Gene-set Cluster' menu (⑦ in Figure 4). Selecting 'Unselected' turns off highlighting.

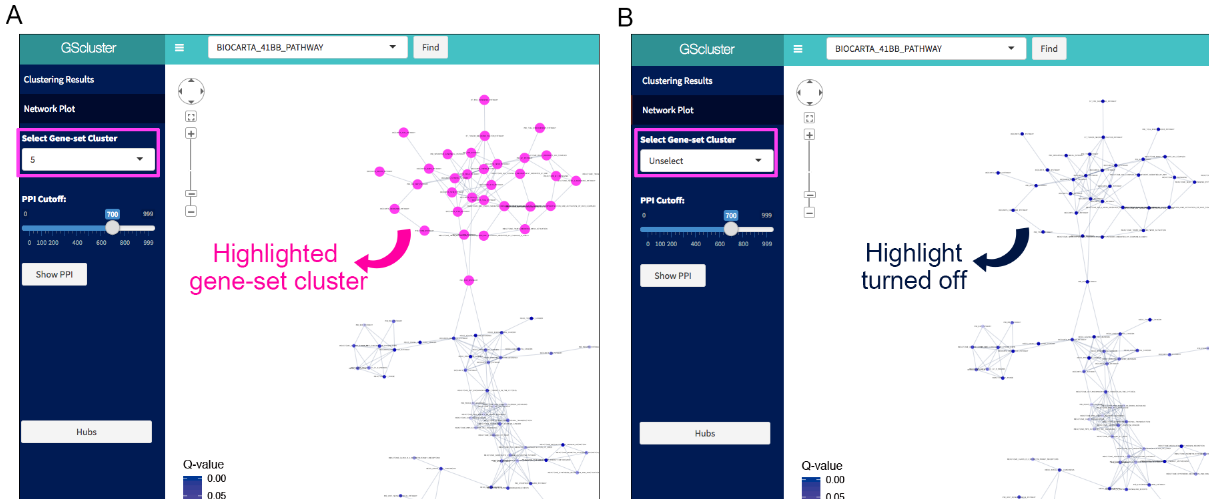


Figure 9. Highlighting gene-set cluster. (A) A gene-set cluster is highlighted. (B) Highlighting is turned off by selecting 'Unselect'.

8) **Gene network**

GScluster also visualize the gene networks of each gene-set cluster based on STRING network (PPI) data (Custom network data is also applicable). For example, the user can see the gene networks in cluster 7, as follows:

A. Choose the cluster number ('7' in this case) from **'Select Gene-set Cluster'** box.
B. Set the **PPI cutoff** (default=700, value is in the range 0 - 1000 where 1000 is the strongest).
C. Click on **'Show PPI'** button (⑧ in Figure 4). Then, it will show the network for genes in cluster 7. To go back to the gene-set network mode, please click on the **'Go back'** button.
D. Each gene node is hyperlinked to the corresponding *GeneCards* site (https://www.genecards.org).
E. In gene networks, two additional functions are provided (**PPI Evidence** (① in Fig 10, Figure 11) and **WordCloud** (② in Fig 10, Figure 12, human only))

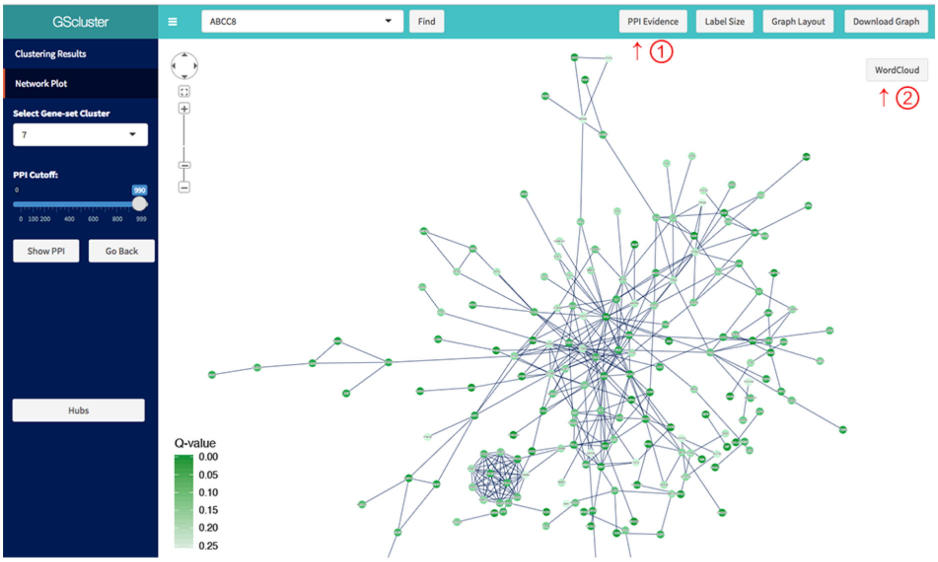


Figure 10. Gene network for a specific cluster (cluster 7)

F. PPI Evidence

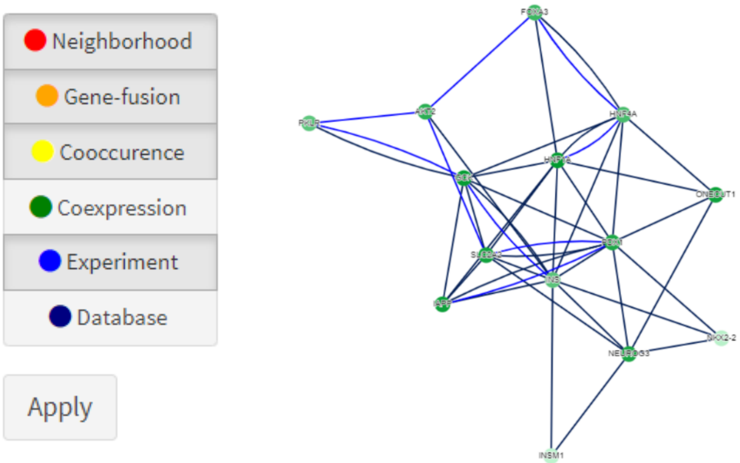


Figure 11. Six different networks from STRING are visualized. The text-mining evidence is excluded.

PPI Evidence shows detailed network information**.** Six edge types such as Neighborhood, Gene fusion, Co-occurrence, Co-expression, Experiments, and Databases are visualized. Detailed explanation for each PPI evidence type is described in STRING web page (https://string-db.org/cgi/help.pl). Default edges represent the combined scores.

F. Wordcloud: GScluster supports disease information obtained from DisGeNET (www.disgenet.org), which is available only for human. It shows wordcloud of any common disease (frequency > 1) for the input genes (Figure 12).

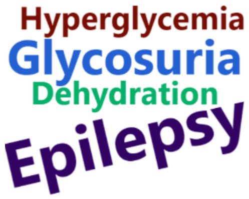
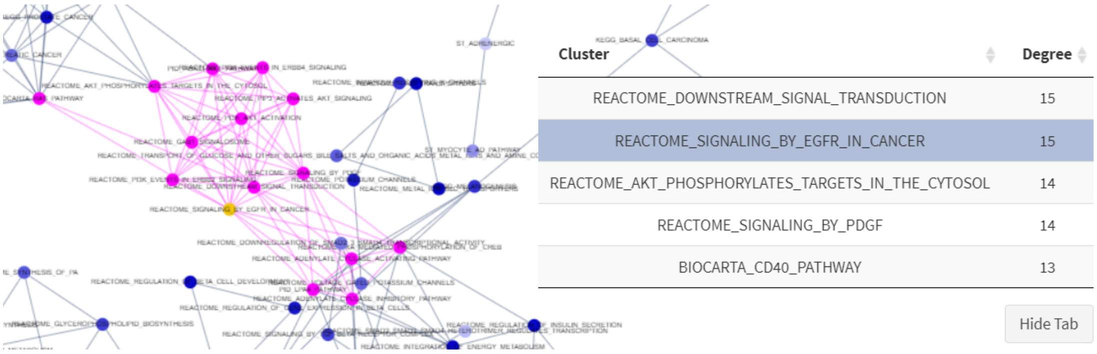


Figure 12. Wordcloud generated from a specific cluster

9) Hub

GScluster supports three types of hubs. (⑨ in Fig. 4. Hub means highly connected node).

① 'Geneset Hub' shows top 5 hub gene-sets in Gene-set network. By clicking on them in table, that hub (yellow) and its neighbors (magenta) will be highlighted (Figure 13). Clicking on the network background the highlighting is canceled. It is available from the gene-set network mode.

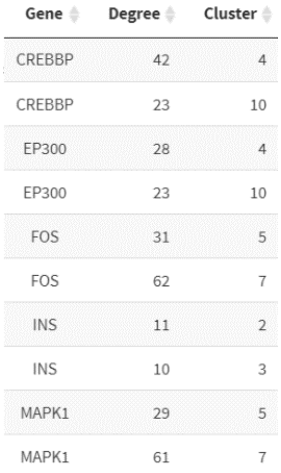


Cluster	Degree
REACTOME_DOWNSTREAM_SIGNAL_TRANSDUCTION	15
REACTOME_SIGNALING_BY_EGFR_IN_CANCER	15
REACTOME_AKT_PHOSPHORYLATES_TARGETS_IN_THE_CYTOSOL	14
REACTOME_SIGNALING_BY_PDGF	14
BIOCARTA_CD40_PATHWAY	13

Figure 13. Gene-set hub and its neighbor nodes are highlighted

② 'Gene hubs' are visualized in the same way (Available from the gene network mode).

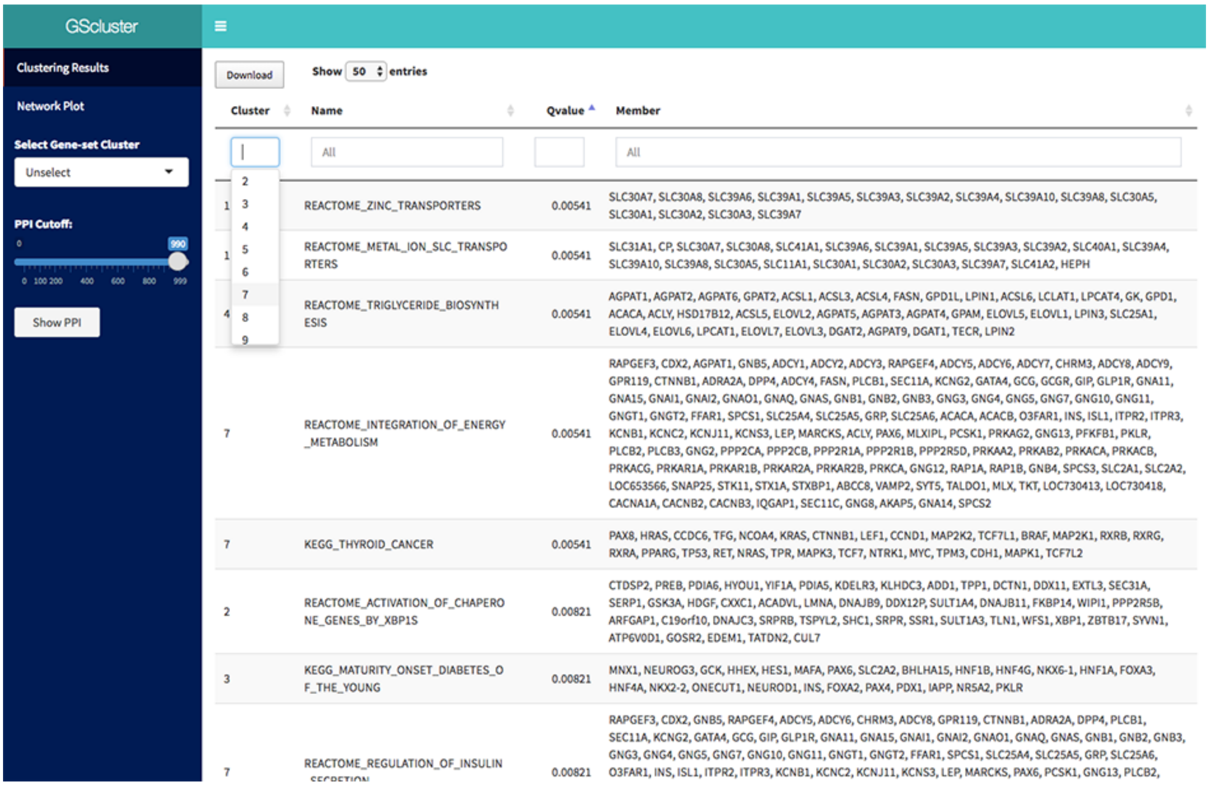
③ 'Multi-cluster Hub' shows gene hubs included in at least two clusters. For example, CREBBP is a hub gene in both cluster 4 and cluster 10 with 23 and 42 neighbors, respectively (Available from the gene-set network mode, Figure 14).



Gene	Degree	Cluster
CREBBP	42	4
CREBBP	23	10
EP300	28	4
EP300	23	10
FOS	31	5
FOS	62	7
INS	11	2
INS	10	3
MAPK1	29	5
MAPK1	61	7

Figure 14. Example of Multi-Cluster Hubs

10) Clustering Results.



Select Gene-set Cluster: Unselect

PPI Cutoff: 990

Show 50 entries

Cluster	Name	Qvalue	Member
	All		All
1	REACTOME_ZINC_TRANSPORTERS	0.00541	SLC30A7, SLC30A8, SLC39A6, SLC39A1, SLC39A5, SLC39A3, SLC39A2, SLC39A4, SLC39A10, SLC39A8, SLC30A5, SLC30A1, SLC30A2, SLC30A3, SLC39A7
1	REACTOME_METAL_ION_SLC_TRANSPORTERS	0.00541	SLC31A1, CP, SLC30A7, SLC30A8, SLC41A1, SLC39A6, SLC39A1, SLC39A5, SLC39A3, SLC39A2, SLC40A1, SLC39A4, SLC39A10, SLC39A8, SLC30A5, SLC11A1, SLC30A1, SLC30A2, SLC30A3, SLC39A7, SLC41A2, HEPH
4	REACTOME_TRIGLYCERIDE_BIOSYNTHESIS	0.00541	AGPAT1, AGPAT2, AGPAT6, GPAT2, ACSL1, ACSL3, ACSL4, FASN, GPD1L, LPIN1, ACSL6, LCLAT1, LPCAT4, GK, GPD1, ACACA, ACLY, HSD17B12, ACSL5, ELOVL2, AGPAT5, AGPAT3, AGPAT4, GPAM, ELOVL5, ELOVL1, LPIN3, SLC25A1, ELOVL4, ELOVL6, LPCAT1, ELOVL7, ELOVL3, DGAT2, AGPAT9, DGAT1, TECR, LPIN2
7	REACTOME_INTEGRATION_OF_ENERGY_METABOLISM	0.00541	RAPGEF3, CDX2, AGPAT1, GNB5, ADCY1, ADCY2, ADCY3, RAPGEF4, ADCY5, ADCY6, ADCY7, CHRM3, ADCY8, ADCY9, GPR119, CTNNB1, ADRA2A, DPP4, ADCY4, FASN, PLCB1, SEC11A, KCNG2, GATA4, GCG, GCGR, GIP, GLP1R, GNA11, GNA15, GNAI1, GNAI2, GNAO1, GNAQ, GNAS, GNB1, GNB2, GNB3, GNG3, GNG4, GNG5, GNG7, GNG10, GNG11, GNGT1, GNGT2, FFAR1, SPCS1, SLC25A4, SLC25A5, GRP, SLC25A6, ACACA, ACACB, O3FAR1, INS, ISL1, ITPR2, ITPR3, KCNB1, KCNC2, KCNJ11, KCNS3, LEP, MARCKS, ACLY, PAX6, MLXIPL, PCSK1, PRKAG2, GNG13, PFKFB1, PKLR, PLCB2, PLCB3, GNG2, PPP2CA, PPP2CB, PPP2R1A, PPP2R1B, PPP2R5D, PRKAA2, PRKAB2, PRKACA, PRKACB, PRKACG, PRKAR1A, PRKAR1B, PRKAR2A, PRKAR2B, PRKCA, GNG12, RAP1A, RAP1B, GNB4, SPCS3, SLC2A1, SLC2A2, LOC653566, SNAP25, STK11, STX1A, STXBP1, ABCC8, VAMP2, SYT5, TALDO1, MLX, TKT, LOC730413, LOC730418, CACNA1A, CACNB2, CACNB3, IQGAP1, SEC11C, GNG8, AKAP5, GNA14, SPCS2
7	KEGG_THYROID_CANCER	0.00541	PAX8, HRAS, CCDC6, TFG, NCOA4, KRAS, CTNNB1, LEF1, CCND1, MAP2K2, TCF7L1, BRAF, MAP2K1, RXRB, RXRG, RXRA, PPARG, TP53, RET, NRAS, TPR, MAPK3, TCF7, NTRK1, MYC, TPM3, CDH1, MAPK1, TCF7L2
2	REACTOME_ACTIVATION_OF_CHAPERONE_GENES_BY_XBP1S	0.00821	CTDSP2, PREB, PDIA6, HYOU1, YIF1A, PDIA5, KDELR3, KLHDC3, ADD1, TPP1, DCTN1, DDX11, EXTL3, SEC31A, SERP1, GSK3A, HDGF, CXXC1, ACADVL, LMNA, DNAJB9, DDX12P, SULT1A4, DNAJB11, FKBP14, WIPI1, PPP2R5B, ARFGAP1, C19orf10, DNAJC3, SRPRB, TSPYL2, SHC1, SRPR, SSR1, SULT1A3, TLN1, WFS1, XBP1, ZBTB17, SYVN1, ATP6V0D1, GOSR2, EDEM1, TATDN2, CUL7
3	KEGG_MATURITY_ONSET_DIABETES_OF_THE_YOUNG	0.00821	MNX1, NEUROG3, GCK, HHEX, HES1, MAFA, PAX6, SLC2A2, BHLHA15, HNF1B, HNF4G, NKX6-1, HNF1A, FOXA3, HNF4A, NKX2-2, ONECUT1, NEUROD1, INS, FOXA2, PAX4, PDX1, IAPP, NR5A2, PKLR
7	REACTOME_REGULATION_OF_INSULIN_SECRETION	0.00821	RAPGEF3, CDX2, GNB5, RAPGEF4, ADCY5, ADCY6, CHRM3, ADCY8, GPR119, CTNNB1, ADRA2A, DPP4, PLCB1, SEC11A, KCNG2, GATA4, GCG, GIP, GLP1R, GNA11, GNA15, GNAI1, GNAI2, GNAO1, GNAQ, GNAS, GNB1, GNB2, GNB3, GNG3, GNG4, GNG5, GNG7, GNG10, GNG11, GNGT1, GNGT2, FFAR1, SPCS1, SLC25A4, SLC25A5, GRP, SLC25A6, O3FAR1, INS, ISL1, ITPR2, ITPR3, KCNB1, KCNC2, KCNJ11, KCNS3, LEP, MARCKS, PAX6, PCSK1, GNG13, PLCB2,

Figure 15. Clustering result table

Clicking on the **'Clustering Results'** button (10 in Figure 4) will show the table for gene-set clustering results. The result table can be filtered based on cluster number, Gene-set name or Gene-set q-value. The user can copy, download (as .csv file) and print the table.