Supporting Information for:

EnrichVisBox: An Integrated Web Toolbox for Visualization of Functional Enrichment Analysis Results of Omics Data

Shisheng Wang†, Dan Du†, Meng Gong†, Jingqiu Cheng† and Hao Yang†

†West China-Washington Mitochondria and Metabolism Research Center; Key Lab of Transplant Engineering and Immunology, MOH, West China Hospital, Sichuan University, Chengdu, China

Corresponding Author

Email address: yanghao@scu.edu.cn.

Contents

I. Supplementary Notes

This part contains a detailed description of this software to help users understand it better when they process their own datasets.

1. Data preparation
2. Import data
3. Preview
4. Polar plot
5. Network plot
6. Heatmap
7. Ridgeline
8. Objects–terms link

II. Supplementary Figures and Tables

Figure S1. Data input module and the visualization module in EnrichVisBox.

Figure S2. Table file formats supported in EnrichVisBox.

Figure S3. Bubble plot and upset plot for previewing basic information of users' data.

Figure S4. Network plot with edges linking overlapped objects among each enriched ID/term.

Table S1. The visualization categories implemented in EnrichVisBox compared to other existing tools.

Table S2. The summarized information on the testing of this web tool.

Table S3. The implementation of each plot in EnrichVisBox.

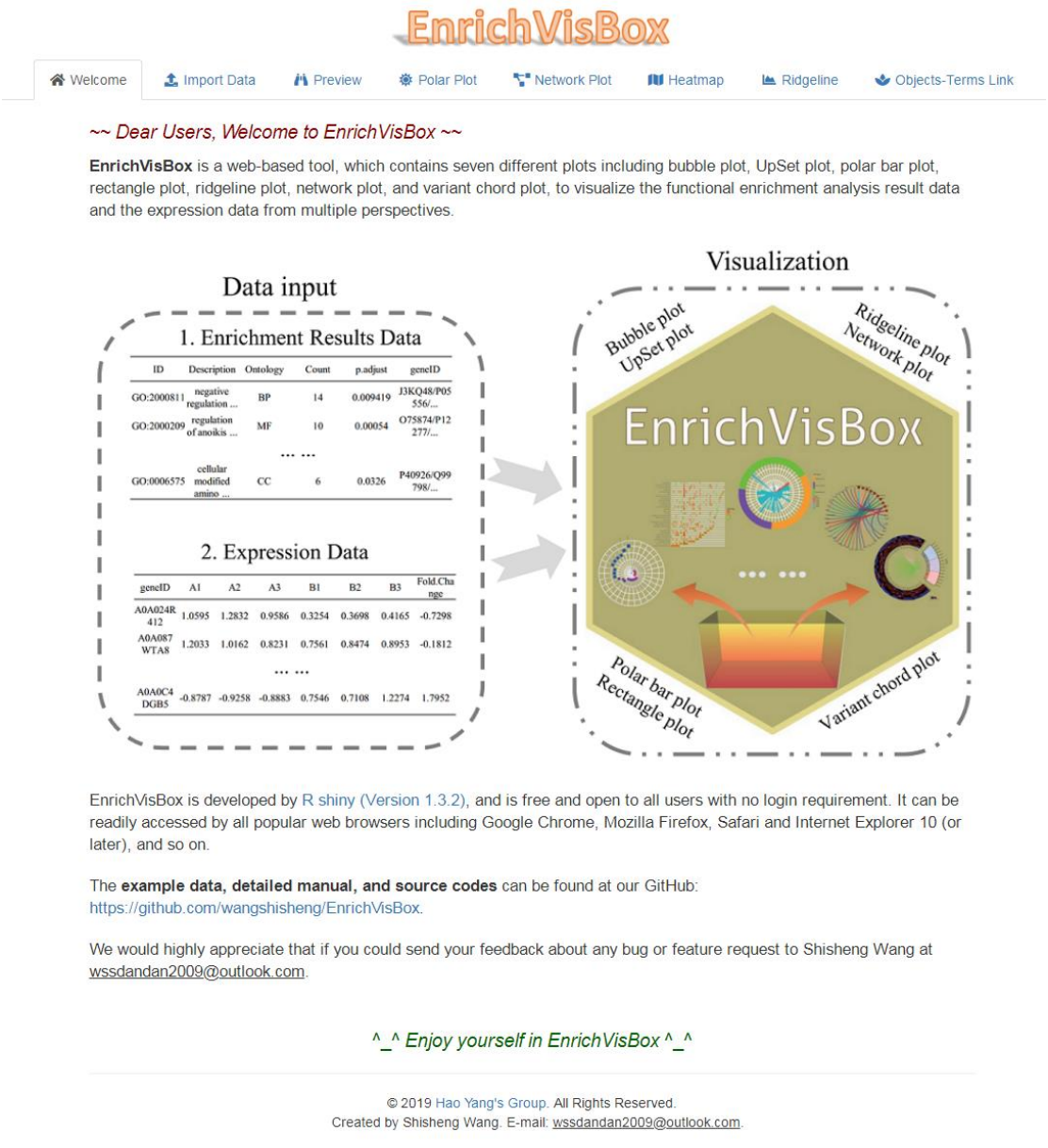
III. References

I. Supplementary Notes

EnrichVisBox, which is powerful and integrated software, can automatically visualize functional enrichment analysis results, such as gene ontology (GO) enrichment analysis results [1] and KEGG pathway enrichment analysis results [2]. Here, the detailed implementation and operation of EnrichVisBox are described. Users can follow this description to visualize their own data freely.

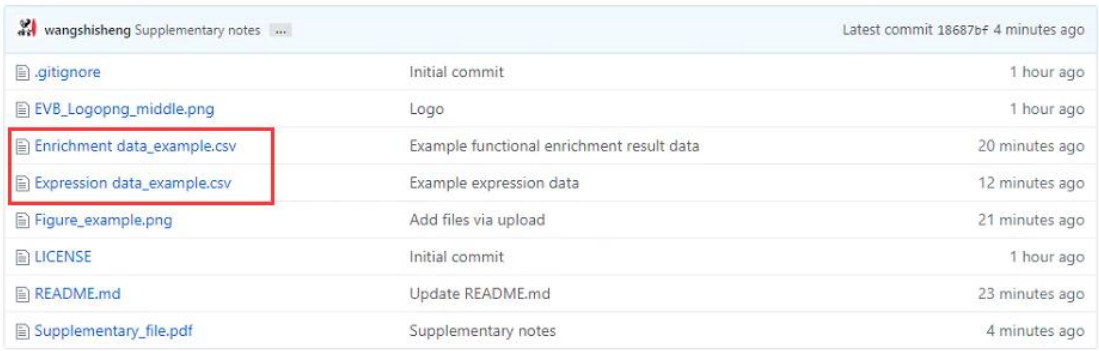
Users can open the site https://www.omicsolution.org/wukong/EnrichVisBox in their browsers. It is recommended that users use Chrome (downloaded from https://chrome.en.softonic.com).

The website homepage is as below:



1. Data preparation

Because EnrichVisBox aims mainly at visualizing functional enrichment analysis results, it is necessary to prepare two types of data here: object (genes/proteins/metabolites) expression data and enrichment analysis resulting data. If users cannot understand what the 'functional enrichment analysis' is, they can read this reference[3] for further learning. Example data can be downloaded from https://github.com/wangshisheng/EnrichVisBox.



The 'Expression data_example.csv' file contains protein expression data, which are extracted from PXD008522[4]. The 'Enrichment data_example.csv' file contains the enrichment analysis results of these proteins.

1.1 Objects (genes/proteins/metabolites) expression data

Different object data need different strategies to obtain the expression data [5-7], but the final formats are usually as shown below.

	A	B	C	D	E	F	G
1	a.	BT549_1	BT549_2	BT549_3	BT549DoxR_1	BT549DoxR_2	BT549DoxR_3
2	A0A024R412	1.059506	1.283269	0.214464	-0.822823392	-0.909236124	-0.82518021
3	A0A087WTA8	1.203369	1.016294	0.423007	-0.951003402	-0.796641649	-0.895024749
4	A0A0A0MRM8	-0.87877	-0.9258	-0.88837	0.754659969	0.710834499	1.227454102
5	A0A0A0MTH3	-0.90821	-0.94375	-0.8573	0.716695613	1.155494034	0.83707639
6	A0A0C4DGB5	-0.76534	-0.97416	-0.87332	1.331006196	0.87486418	0.406952287
7	B4DHE8	-0.7318	-0.92715	-0.92916	0.634944351	1.440765632	0.512403765
8	B7ZL14	0.444225	0.738294	1.414579	-0.794439954	-0.883569709	-0.919087885
9	C9J3D7	-0.85009	-0.96627	-0.89219	0.896350298	0.680054135	1.132144448
10	C9J6P4	1.11088	0.898826	0.702195	-0.828559798	-0.991580309	-0.891760745
11	C9JTE9	0.813344	1.100614	0.793451	-0.719331199	-1.038144629	-0.949933464
12	E7EV99	1.209935	0.820583	0.654708	-0.9219523	-1.003077083	-0.760196609
13	F5H345	-1.02479	-0.74981	-0.95088	0.977722465	0.865618721	0.882141359
14	J3KQ48	-0.96425	-0.91211	-0.7643	0.646871137	0.644714482	1.34907925
15	J3QRU1	1.145576	1.00803	0.525084	-0.887914161	-0.940037309	-0.850738713
16	O00151	-0.78797	-0.90148	-0.98002	0.801803834	1.264754783	0.602911702
17	O00592	1.015921	1.045755	0.649597	-0.881145621	-0.939818539	-0.890308402
18	O00625	-0.88022	-0.84651	-0.80439	1.286553014	0.157593249	1.086980006
19	O14907	0.853003	1.352031	0.400099	-0.765250921	-0.975289783	-0.864592167
20	O15075	0.429311	0.989474	1.226264	-0.859006902	-0.897560576	-0.888482192
21	O43852	0.299224	1.003515	1.288304	-0.883249245	-0.905318909	-0.802474524
22	O43854	-0.81727	-0.81507	-0.80942	0.074459052	0.863518832	1.503782181
23	O60701	1.213452	0.903693	0.561584	-0.905306442	-0.84545998	-0.927962439
24	O75874	1.053606	0.751383	0.909227	-0.836992278	-1.072967803	-0.804256174
25	O95340	1.329013	0.669268	0.658082	-0.897059668	-0.894562498	-0.864740307
26	O95671	1.49785	0.199451	0.798682	-0.82347645	-0.804478886	-0.86802776
27	P05161	1.014788	0.077106	1.384383	-0.851998474	-0.848894233	-0.775384158
28	P05556	0.953843	0.906458	0.873183	-0.863287075	-0.855203781	-1.014813431

b.

c.

There are three main sections:

a. Column name: these are sample names. In the example data, there are two group samples (three biological replicates in each group).
b. Row names: these can be gene/protein/metabolite names or some kind of ID, for instance, the ones here are protein Uniprot IDs.
c. Expression values: these can be original measured values or normalized values, for example, the ones here are normalized peak intensities.

Here, in EnrichVisBox, users must include an additional column, fold change, the variable importance in projection (VIP) value derived from the PLSDA or OPLSDA model [8], or other similar values. Here, the log2 fold change between two groups is added.

		BT549_1	BT549_2	BT549_3	BT549DoxR_1	BT549DoxR_2	BT549DoxR_3	Fold.Change
2	A0A024R412	1.059506	1.283269	0.214464	-0.822823392	-0.909236124	-0.82518021	-2.998247456
3	A0A087WTA8	1.203369	1.016294	0.423007	-0.951003402	-0.796641649	-0.895024749	-2.571503195
4	A0A0A0MRM8	-0.87877	-0.9258	-0.88837	0.754659969	0.710834499	1.227454102	4.822678072
5	A0A0A0MTH3	-0.90821	-0.94375	-0.8573	0.716695613	1.155494034	0.83707639	1.435108882
6	A0A0C4DGB5	-0.76534	-0.97416	-0.87332	1.331006196	0.87486418	0.406952287	1.575034541
7	B4DHE8	-0.7318	-0.92715	-0.92916	0.634944351	1.440765632	0.512403765	1.77553431
8	B7ZL14	0.444225	0.738294	1.414579	-0.794439954	-0.883569709	-0.919087885	-2.161046453
9	C9J3D7	-0.85009	-0.96627	-0.89219	0.896350298	0.680054135	1.132144448	4.156198372
10	C9J6P4	1.11088	0.898826	0.702195	-0.828559798	-0.991580309	-0.891760745	-1.179184593
11	C9JTE9	0.813344	1.100614	0.793451	-0.719331199	-1.038144629	-0.949933464	-1.458409586
12	E7EV99	1.209935	0.820583	0.654708	-0.9219523	-1.003077083	-0.760196609	-1.85472371
13	F5H345	-1.02479	-0.74981	-0.95088	0.977722465	0.865618721	0.882141359	1.401252417
14	J3KQ48	-0.96425	-0.91211	-0.7643	0.646871137	0.644714482	1.34907925	1.389542565
15	J3QRU1	1.145576	1.00803	0.525084	-0.887914161	-0.940037309	-0.850738713	-1.310422403
16	O00151	-0.78797	-0.90148	-0.98002	0.801803834	1.264754783	0.602911702	1.507742165
17	O00592	1.015921	1.045755	0.649597	-0.881145621	-0.939818539	-0.890308402	-2.68989838
18	O00625	-0.88022	-0.84651	-0.80439	1.286553014	0.157593249	1.086980006	2.550562206
19	O14907	0.853003	1.352031	0.400099	-0.765250921	-0.975289783	-0.864592167	-1.535320574
20	O15075	0.429311	0.989474	1.226264	-0.859006902	-0.897560576	-0.888482192	-6.177120425
21	O43852	0.299224	1.003515	1.288304	-0.883249245	-0.905318909	-0.802474524	-1.573375864
22	O43854	-0.81727	-0.81507	-0.80942	0.074459052	0.863518832	1.503782181	6.582184599
23	O60701	1.213452	0.903693	0.561584	-0.905306442	-0.84545998	-0.927962439	-1.616736253
24	O75874	1.053606	0.751383	0.909227	-0.836992278	-1.072967803	-0.804256174	-1.759064714
25	O95340	1.329013	0.669268	0.658082	-0.897059668	-0.894562498	-0.864740307	-2.61542742
26	O95671	1.49785	0.199451	0.798682	-0.82347645	-0.804478886	-0.86802776	-3.335527393
27	P05161	1.014788	0.077106	1.384383	-0.851998474	-0.848894233	-0.775384158	-2.611417709
28	P05556	0.953643	0.906458	0.873183	-0.863267075	-0.855203781	-1.014813431	-1.0718328

1.2 Enrichment analysis resulting data

After obtaining the object (gene/protein/metabolite) list, it is generally desirable to obtain the biological interpretation. A standard method of solving this problem is functional enrichment analysis, which summarizes the large object list as a smaller list of more easily interpretable biological functions based on some existing databases, such as the GO database[1] and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway database[2]. The final file format may be like the one below, which is an example result of GO enrichment analysis.

	ID	Description	ONTOLOGY	Count	p.adjust	geneID
1	GO:2000811	negative regulation o	BP	4	0.009419	J3KQ48/P05556/P08648/Q03135
2	GO:2000209	regulation of anoikis	BP	4	0.009419	J3KQ48/P05556/P08648/Q03135
3	GO:0045454	cell redox homeostasi	BP	5	0.009419	P13667/P30101/P48506/Q96HE7/Q9BS26
4	GO:0006099	tricarboxylic acid cy	BP	4	0.009419	O75874/P10515/P40926/Q99798
5	GO:0006575	cellular modified ami	BP	7	0.009419	O75874/P12277/P47712/P48506/P49189/P51649/Q96HE7
6	GO:0006101	citrate metabolic pro	BP	4	0.009707	O75874/P10515/P40926/Q99798
7	GO:0022617	extracellular matrix	BP	5	0.009707	P07711/P16035/P50281/Q07954/Q7Z460
8	GO:0043276	anoikis	BP	4	0.009707	J3KQ48/P05556/P08648/Q03135
9	GO:0033631	cell-cell adhesion me	BP	3	0.009707	O00592/P05556/P08648
10	GO:0005178	integrin binding	MF	6	0.009707	O43854/P05161/P08648/P16035/P20908/P50281
11	GO:0072350	tricarboxylic acid me	BP	4	0.009707	O75874/P10515/P40926/Q99798
12	GO:0016999	antibiotic metabolic	BP	6	0.009707	O75874/P10515/P40926/P51649/Q04828/Q99798
13	GO:0035313	wound healing, spread	BP	3	0.009761	P08648/P20908/Q7Z460
14	GO:1901888	regulation of cell ju	BP	5	0.023368	P17302/P50281/Q03135/Q07954/Q7Z460
15	GO:0016616	oxidoreductase activi	MF	5	0.024569	O60701/O75874/P40926/Q04828/Q53GQ0
16	GO:0001968	fibronectin binding	MF	3	0.026621	P05556/P07711/Q53GQ0
17	GO:0030175	filopodium	CC	5	0.026621	O00592/P05556/P26038/Q13509/Q9Y4F1

ID: GO ID;

Description: A statement about the function of objects;

ONTOLOGY: A formal representation of a body of knowledge within a given domain — the detailed information about ontology can be found here: http://geneontology.org/docs/ontology-documentation;

Count: Number of enriched objects (genes/proteins/metabolites);

p.adjust: Adjusted p values;

geneID: IDs/names of enriched objects (genes/proteins/metabolites).

It is recommended that users prepare their enrichment analysis resulting data with the same columns, but the content of each column can be different.

How to process functional enrichment analysis?

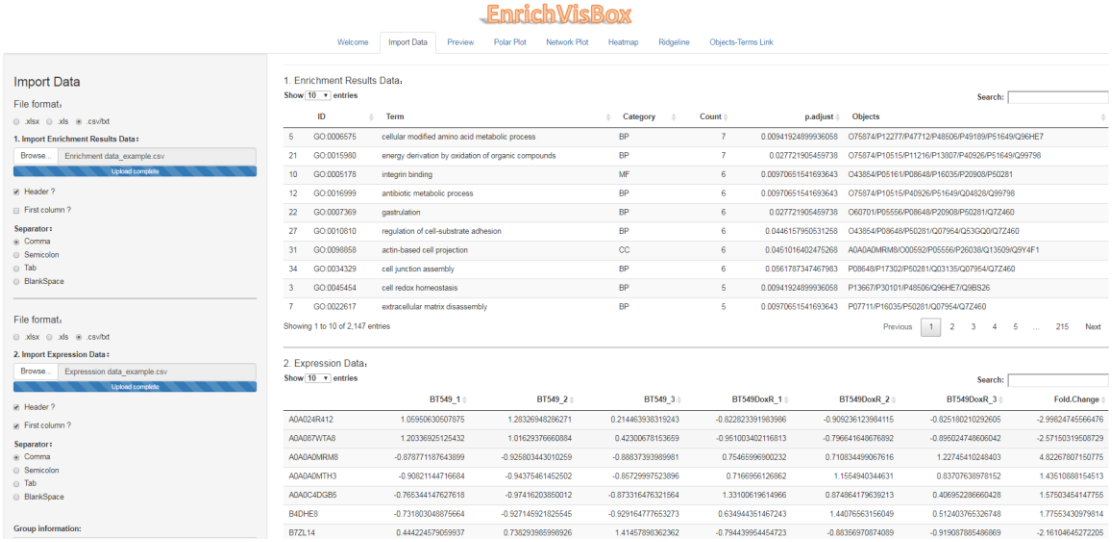
Some users, especially beginners, may have this question. Here, some published software tools are recommended.

Data type	Tool names	Links
Genomics	Enrichr [9]	https://amp.pharm.mssm.edu/Enrichr
	DAVID [10]	https://david.ncifcrf.gov
Proteomics	MixProTool [11]	https://www.omicsolution.org/wukong/MixProTool
	FunRich [12]	http://www.funrich.org
Metabolomics	MetaboAnalyst [13]	https://www.metaboanalyst.ca
	ConsensusPathDB [14]	http://consensuspathdb.org

If users are familiar with the R language, they can choose to use some R packages, such as clusterProfiler [15], and topGO [16]. Whichever tool users choose to use, it is recommended that they assign their enrichment results as above.

2. Import data

When users prepare their data (expression data and enrichment analysis resulting data), they can click ‘Import Data’ and upload the data from here.



Users should choose the right file format based on their data (the example data have been saved as a .csv file, so.csv/.txt is chosen here) and then click the ‘Browse’ button to import the data. To facilitate subsequent analysis, the columns in the enrichment analysis resulting data (Section 1.2) were renamed as: *ID*, *Term*, *Category*, *Count*, *p.adjust*, and *Objects*.

1. Enrichment Results Data:

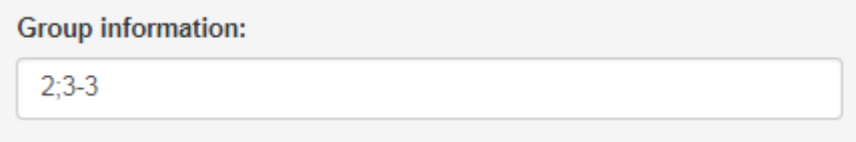
Show 10 entries Search:

	ID	Term	Category	Count	p.adjust	Objects
5	GO:0006575	cellular modified amino acid metabolic process	BP	7	0.00941924899936058	O75874/P12277/P47712/P48506/P49189/P51649/Q96HE7
21	GO:0015980	energy derivation by oxidation of organic compounds	BP	7	0.027721905459738	O75874/P10515/P11216/P13807/P40926/P51649/Q99798
10	GO:0005178	integrin binding	MF	6	0.00970651541693643	O43854/P05161/P08648/P16035/P20908/P50281
12	GO:0016999	antibiotic metabolic process	BP	6	0.00970651541693643	O75874/P10515/P40926/P51649/Q04828/Q99798
22	GO:0007369	gastrulation	BP	6	0.027721905459738	O60701/P05556/P08648/P20908/P50281/Q7Z460
27	GO:0010810	regulation of cell-substrate adhesion	BP	6	0.0446157950531258	O43854/P08648/P50281/Q07954/Q53GQ0/Q7Z460
31	GO:0098858	actin-based cell projection	CC	6	0.0451016402475268	A0A0A0MRM8/O00592/P05556/P26038/Q13509/Q9Y4F1
34	GO:0034329	cell junction assembly	BP	6	0.0561787347467983	P08648/P17302/P50281/Q03135/Q07954/Q7Z460
3	GO:0045454	cell redox homeostasis	BP	5	0.00941924899936058	P13667/P30101/P48506/Q96HE7/Q9BS26
7	GO:0022617	extracellular matrix disassembly	BP	5	0.00970651541693643	P07711/P16035/P50281/Q07954/Q7Z460

Showing 1 to 10 of 2,147 entries Previous 1 2 3 4 5 ... 215 Next

Header: this means whether the first row is column names. If true, you should choose this parameter.
First column: this indicates whether the first column is row names. If true, you should choose this parameter.
Separator: this is the field separator character. Values on each line of the file are separated by this character.

Here, one should also type in the ‘Group information’:



As described above, there are two group samples (three replicates in each group) in the example

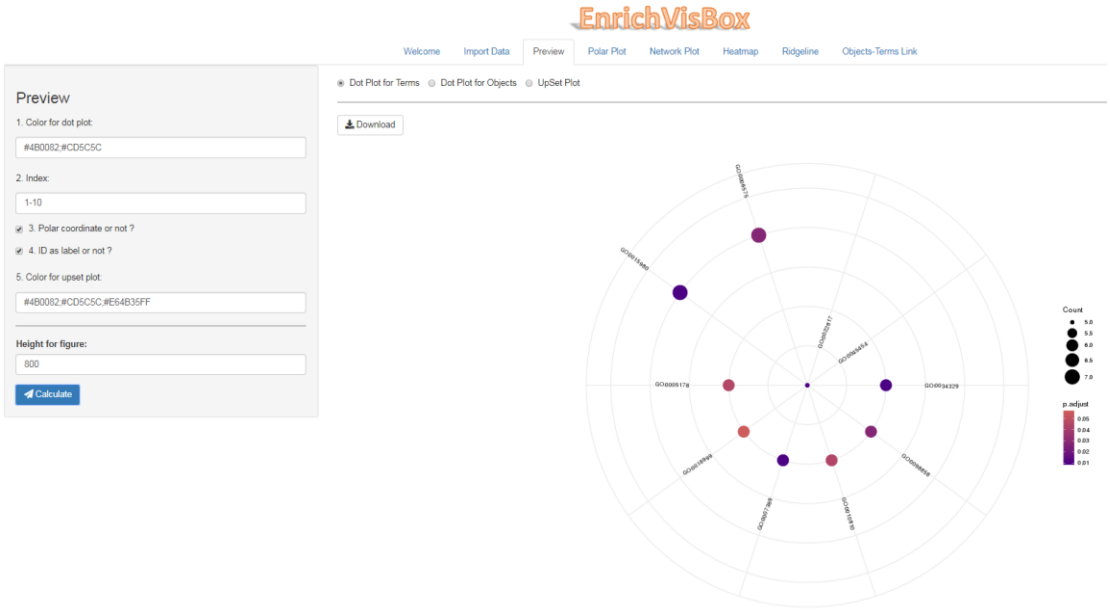
data, so here one should input ‘2;3-3’. Please note this format. One more example is given here for users to understand this parameter better.
If your data have five groups and six replicates in each group, you should input ‘5;6-6-6-6-6’.

If the data have been uploaded successfully, the results will be shown simultaneously on the right panel.

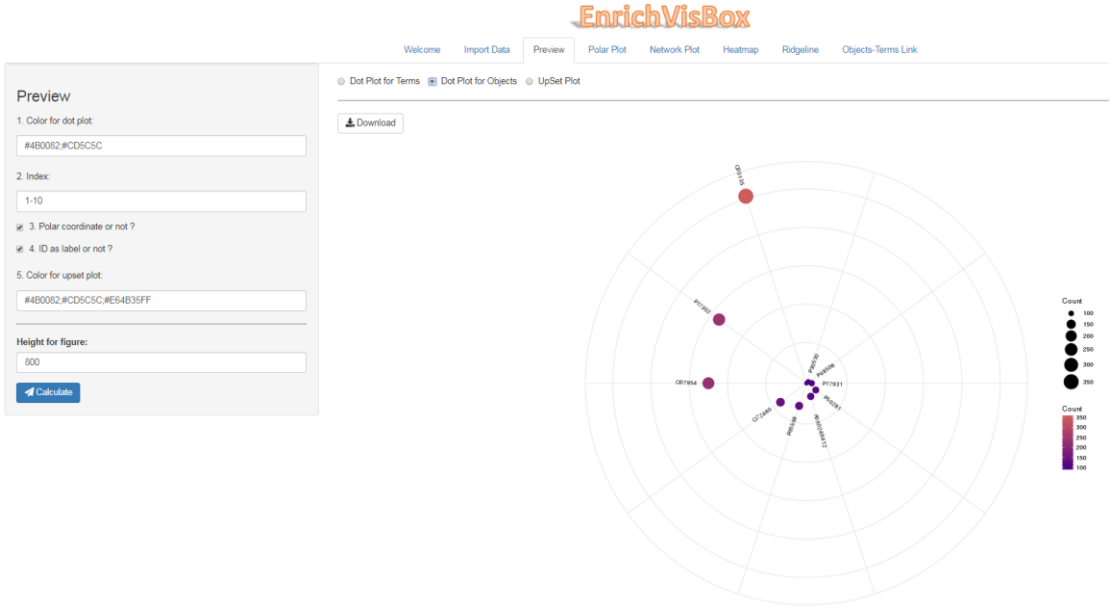
3. Preview

After importing data with the right format, users can click 'Preview', which means users can visualize some basic information about their data in this step. There are three parts in this module.

a. *Dot Plot for Terms*: to show the terms with bubbles. The count of one term is larger, the bubble size is bigger, and the colour of each bubble corresponds to the adjusted p value. This plot was implemented with geom_point and coord_polar functions in the ggplot2 package [17].

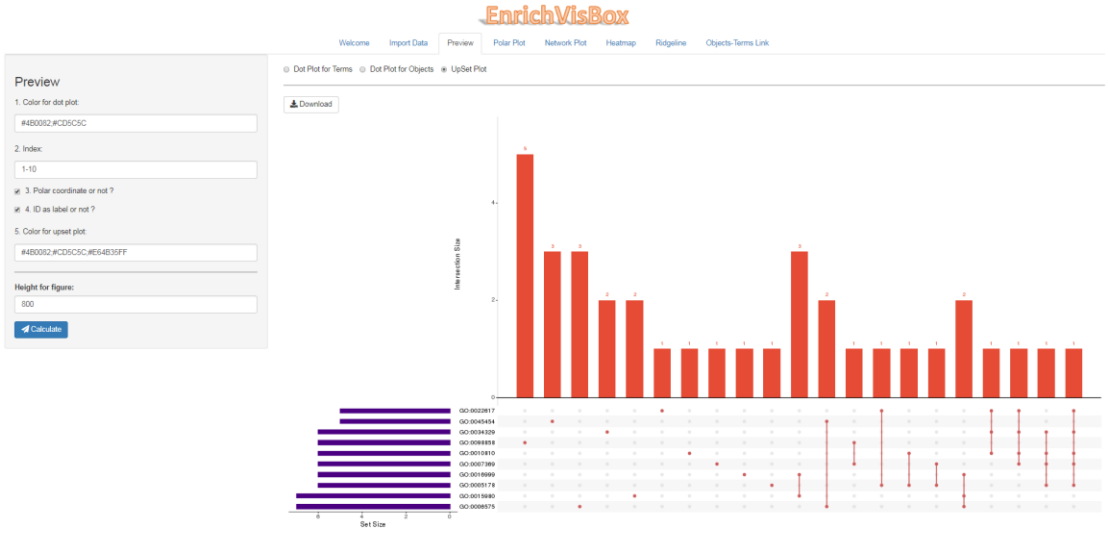


b. *Dot Plot for Objects*: to show the objects (genes/proteins/metabolites) with bubbles. Because most of objects may be overrepresented in more than one term, this part can count how many terms every object is assigned to. The number is larger, the bubble size is bigger, and the colour of each bubble here corresponds to the number. This plot was implemented with geom_ point and coord_polar functions in the ggplot2 package [17].

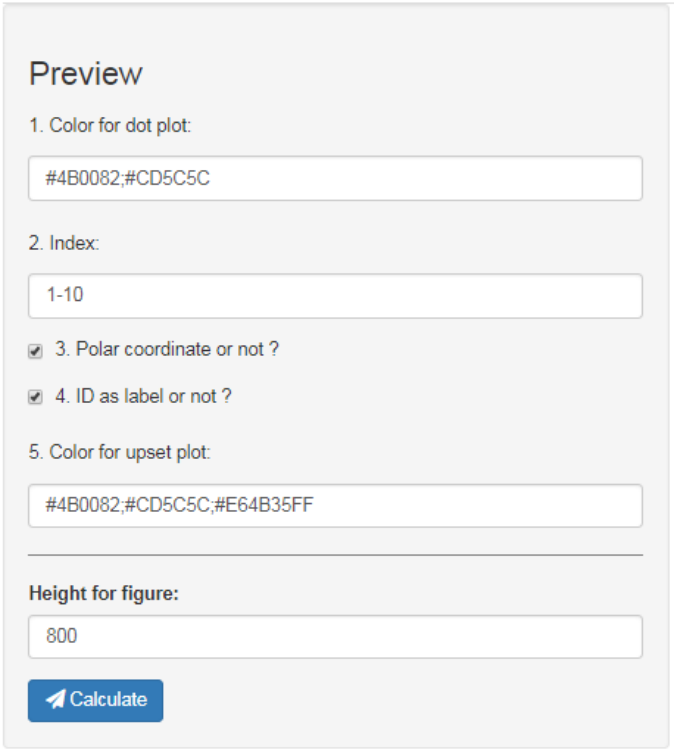


c. *UpSet Plot*: to show interactions among the terms. Because there are the same or different

objects between each two terms, this part can show their overlap. This plot was implemented with the upset function in the UpSetR package [18].



Parameters in this module:

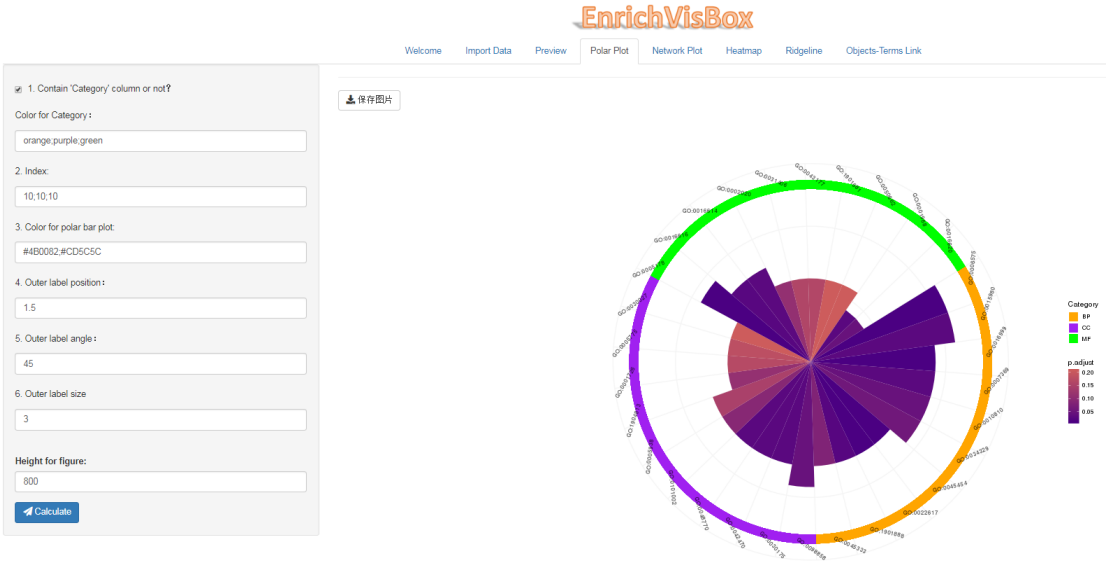


Color for dot plot: to change bubble colour. Users can type in two colour names with a semicolon. The first one is for the lowest p.adjust, and the second one is for the highest p.adjust.
Index: which terms or objects will be plotted. The default '1-10' means the top 10 (1 to 10) terms or objects are shown. If users type in '10-21', it means it shows the 10th to 21st terms or objects (total of 12 terms or objects). If users input '1,10,21', this means it will show the 1st, the 10th, and the 21st terms or objects (total three terms or objects).
Polar coordinate or not: if the dot plot is shown in a circle or not, the default is true, otherwise,

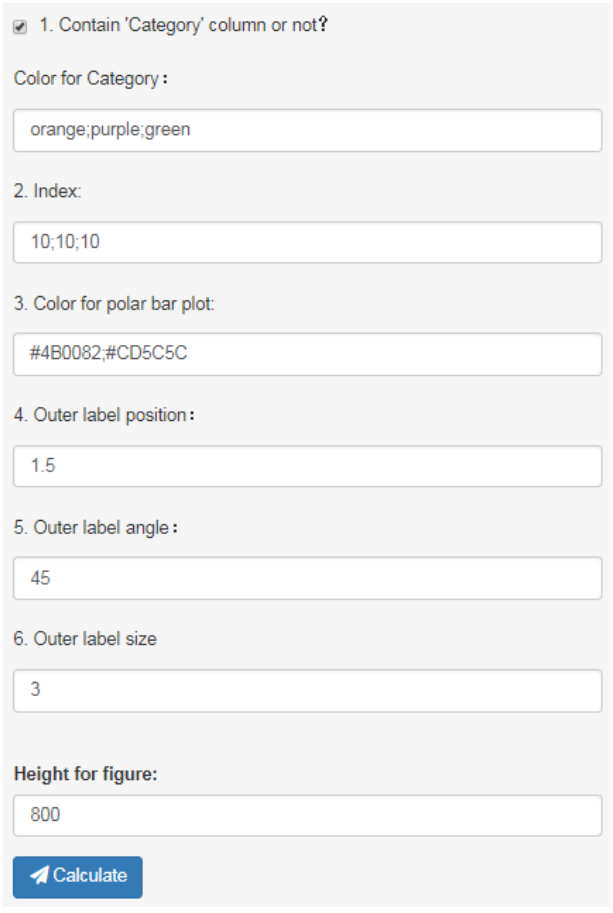
the dot plot is shown in a normal rectangular coordinate system.
ID as label or not: the label in the dot plot is shown with IDs or Terms, the default is with IDs.
Color for upset plot: to change colours in the upset plot. Users can type in three colour names with two semicolons. The first one is the colour of the set size bar plot, the second one is the colour of the intersection points, and the third one is the colour of the main bar plot.
Height for figure: adjusting the height of the figure.

4. Polar plot

This module shows data with the polar bar plot, which involves in *ID (Term)*, *Category*, *Count*, and *p.adjust* columns. Different *Categories* show discrete colours in the outermost circle, and the bar colours are filled with continuous colours corresponding to adjusted p values. This plot was implemented with the geom_bar, geom_segment and coord_polar functions in the ggplot2 package [17].

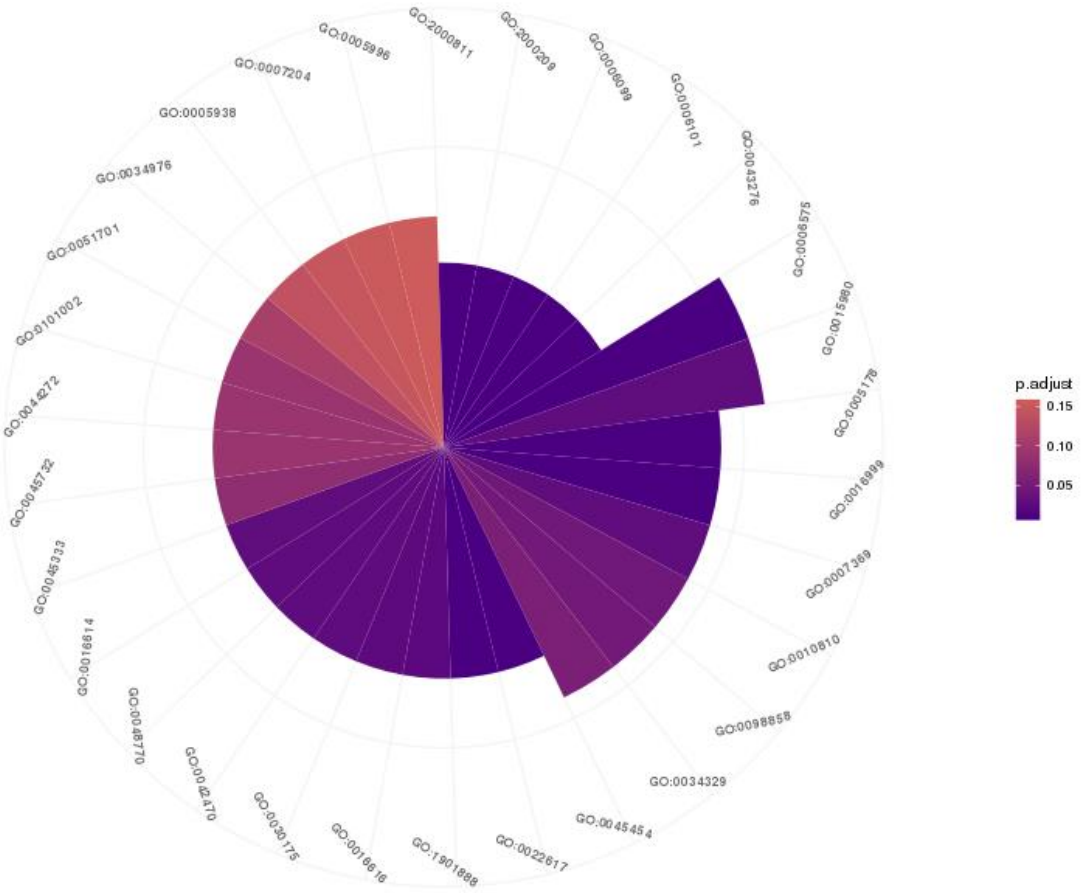


Parameters in this module:



Contain 'Category' column or not: if there is this '*Category*' column in the enrichment analysis resulting data, users should select this parameter and input colour names in the *Color for category* parameter. Otherwise, unselect this parameter. The number should be same, which means there are three categories, so there should be three colours here.

Index: which terms will be plotted. If users select the *Contain 'Category' column or not* parameter, the index number should be the same as the category number and linked with semicolons. For instance, there are three categories in the example data, and, to show top 10 (1 to 10) terms in each category, '10;10;10' is input here. If users unselect the *Contain 'Category' column or not* parameter, the number here will sum up, which means the plot will show the top 30 ('10;10;10' is typed in here) terms in all the data, as below.



Color for polar bar plot: to change the bar colour. Users can type in two colour names with a semicolon. The first one is for lowest p.adjust, and the second one is for highest p.adjust.

Outer label position: to change the position of outer labels.

Outer label angle: to change the angle of outer labels.

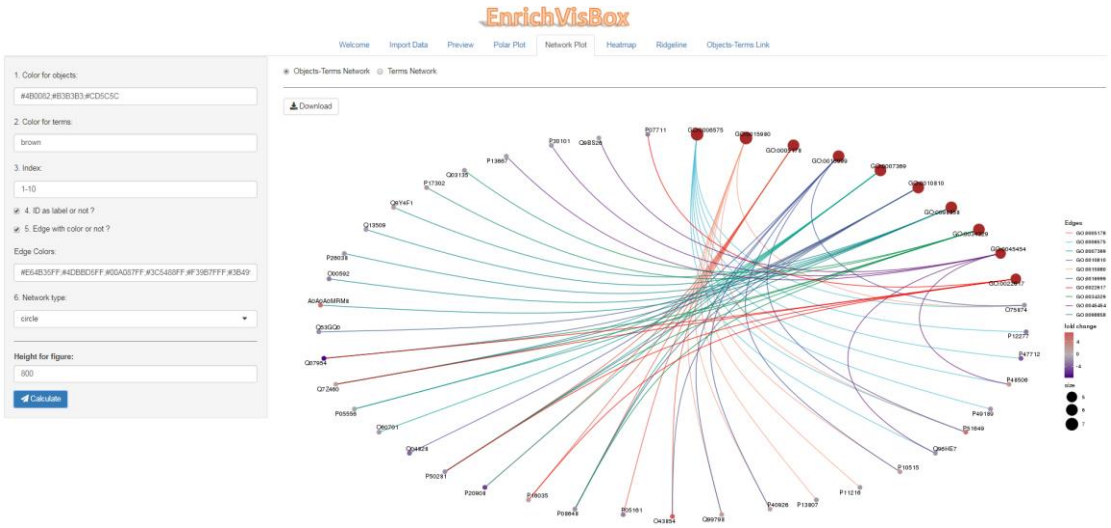
Outer label size: to change the size of outer labels.

Height for figure: adjusting the height of the figure.

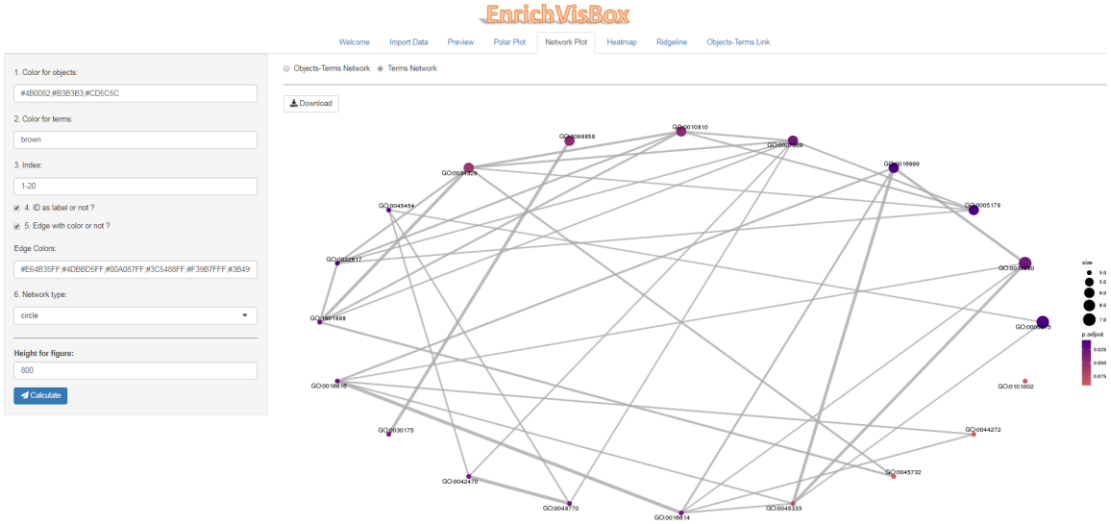
5. Network plot

There are two types in this module. The first one shows the interactions between each object and ID/term, and the second one shows the interactions between every two IDs/terms. This plot was implemented with the cnetplot and emapplot functions in the enrichplot package [15].

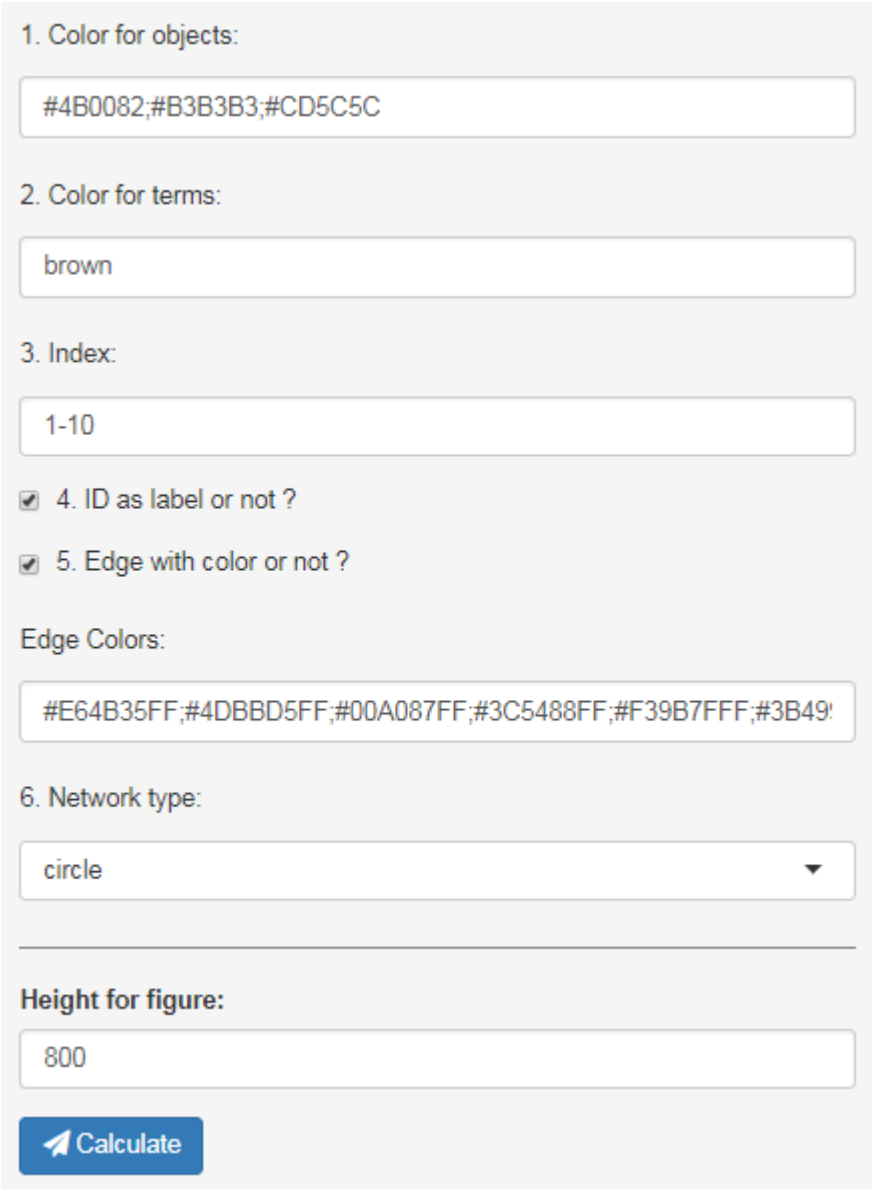
a. *Objects-Terms Network:* the objects are linked via coloured lines to their assigned IDs/terms. The line colours mean different IDs/terms, the colour of each bubble corresponds to fold change (log2 here), and, the larger the count of one term, the bigger the bubble size.



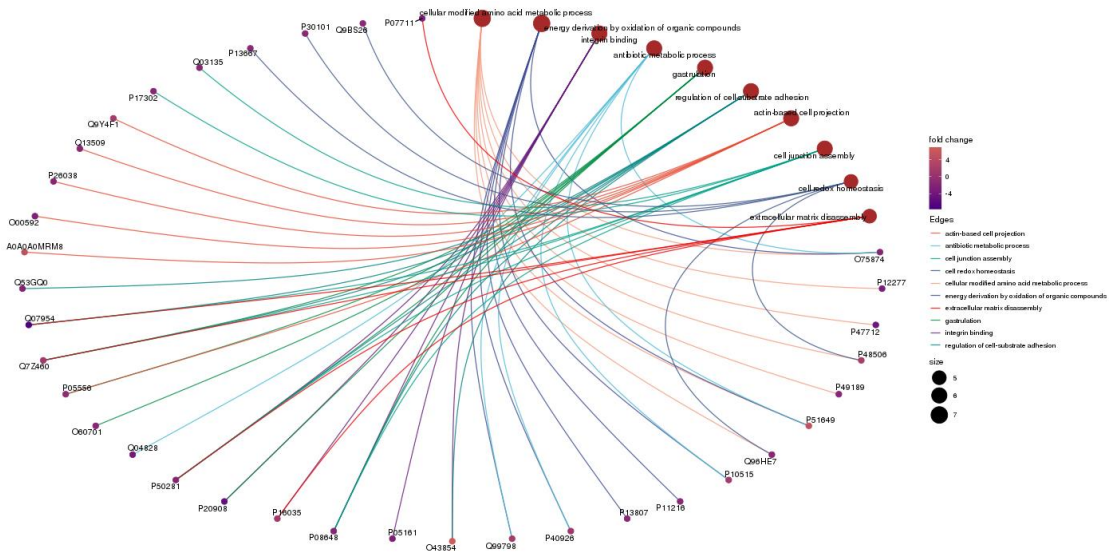
b. *Terms Network:* this plot shows each ID/term with edges linking overlapped objects between every two enriched IDs/terms. When the number of overlapped objects between two IDs/terms is larger, the line size is bigger. Colours of points correspond to adjusted p values.



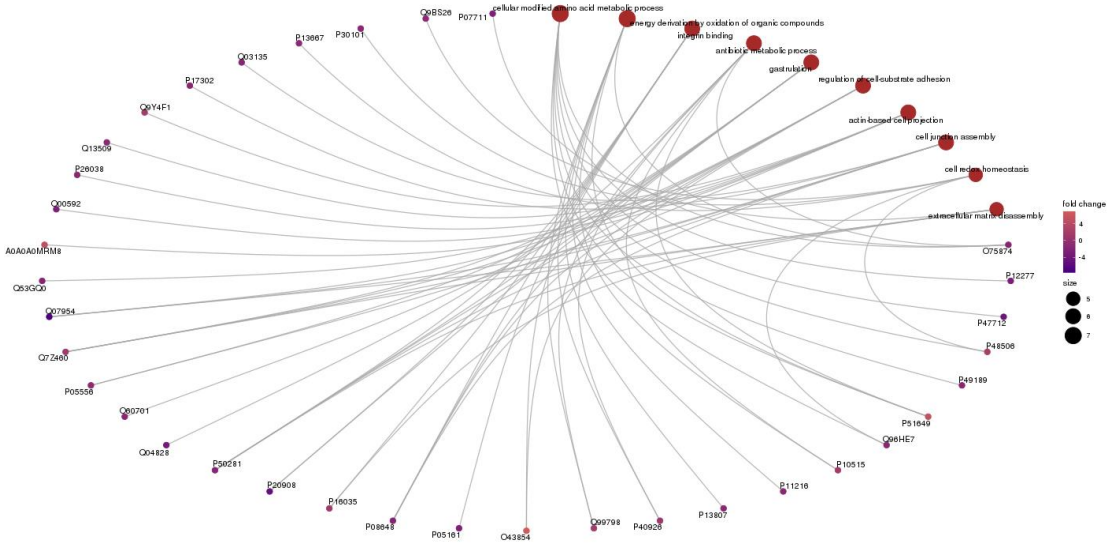
Parameters in this module:



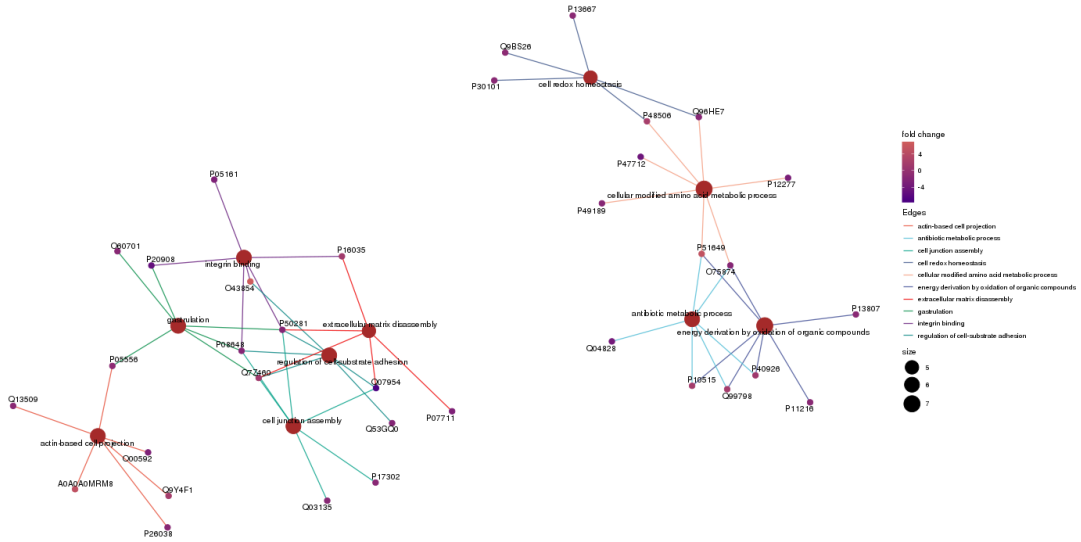
Color for objects: to change point (object) colour. Users can type in two or three colour names with a/two semicolon(s). The first one is for the lowest fold change value (log2 here), and the last one is for the highest fold change value (log2 here).
Color for terms: to change point (ID/term) colour.
Index: which terms or objects will be plotted. The default '1-10' means the top 10 (1 to 10) terms or objects are shown. If users type in '10-21', it means it shows the 10th to 21st terms or objects (total of 12 terms or objects). If users input '1,10,21', this means it will show the 1st, 10th, and 21st terms or objects (total of three terms or objects).
ID as label or not: if selected, the IDs will be shown; otherwise, the terms will be shown, as below.



Edge with color or not: if selected, users should type in the same number of colours in the *Edge colors* parameter. For example, if '1-10' is input in the *index* parameter, here one should type in 10 colour names for each ID/term. If it is not selected, the line colours will become grey, as below.



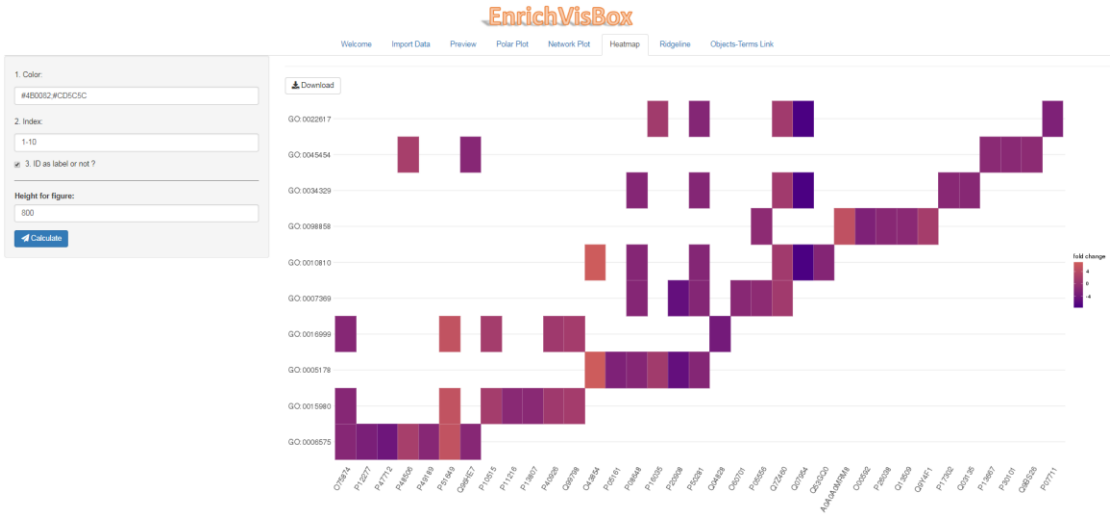
Network type: to determine how the vertices are placed on the plot. There are two types here: circle and kk. Circle means showing the plot with a circular layout, and kk means placing the vertices on the plane based on the Kamada–Kawai layout algorithm [19].



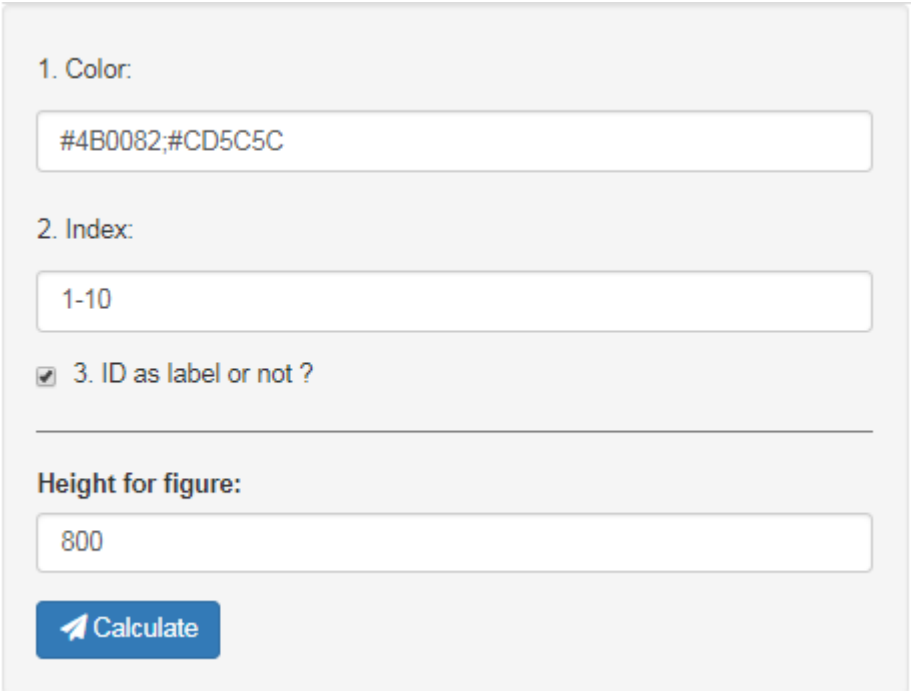
Height for figure: adjusting the height of the figure.

6. Heatmap

This module shows a heatmap-like plot. There is a coloured rectangle if one object is mapped in one ID/term, and the colour corresponds to the fold change (log2 here); otherwise, there is nothing in the position. This plot was implemented with the geom_tile function in ggplot2 package [17].



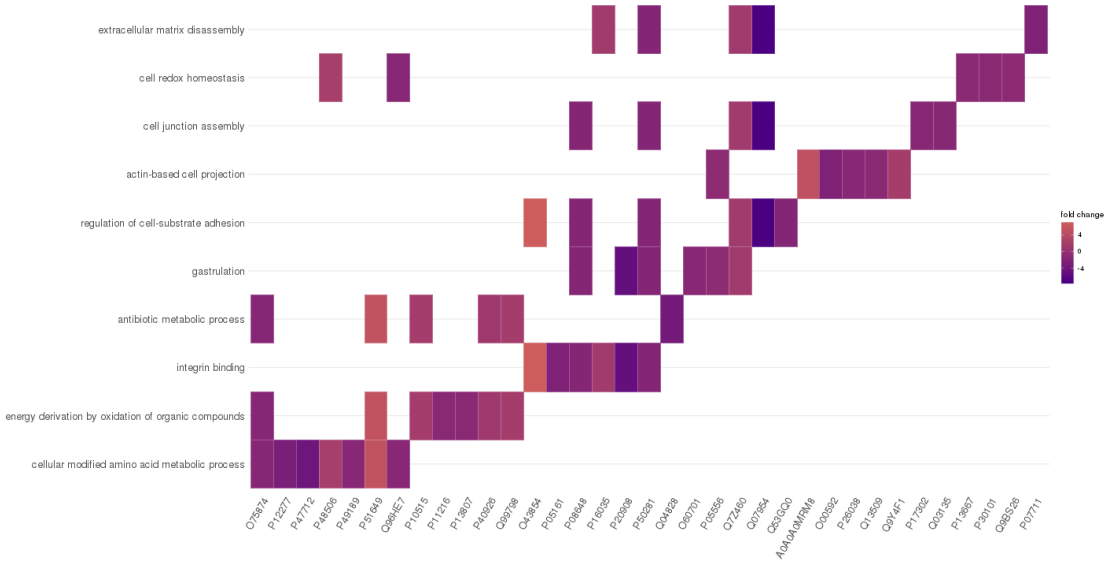
Parameters in this module:



Color: to change rectangles colour. Users can type in two colour names with a semicolon. The first one is for the lowest fold change value (log2 here), and the second one is for the highest fold change value (log2 here).

Index: which terms or objects will be plotted. The default '1-10' means the top 10 (1 to 10) terms or objects are shown. If users type in '10-21', it means it shows the 10th to 21st terms or objects (total of 12 terms or objects). If users input '1,10,21', this means it will show the 1st, 10th, and 21st terms or objects (total of three terms or objects).

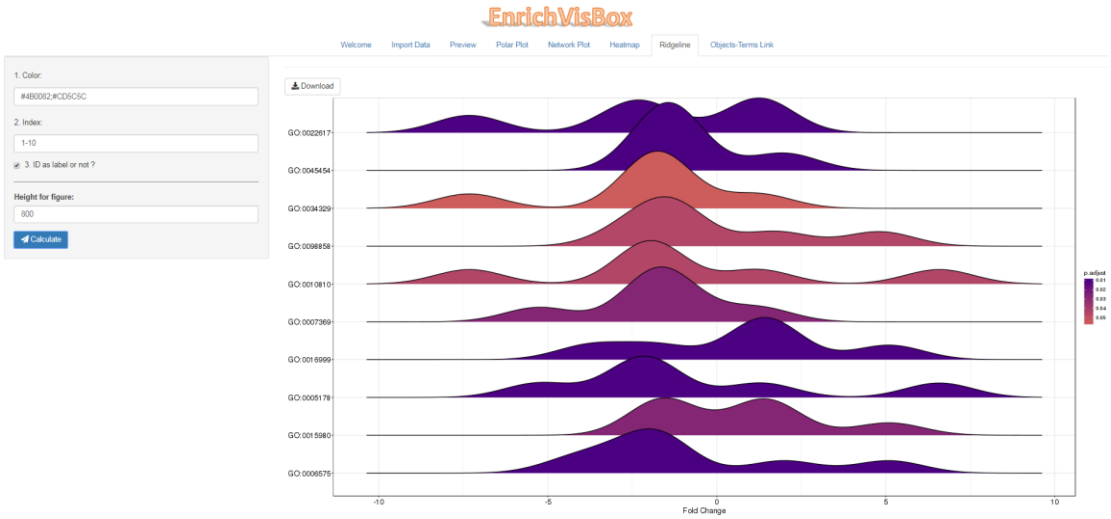
ID as label or not: if selected, the IDs will be shown; otherwise, the terms will be shown, as below.



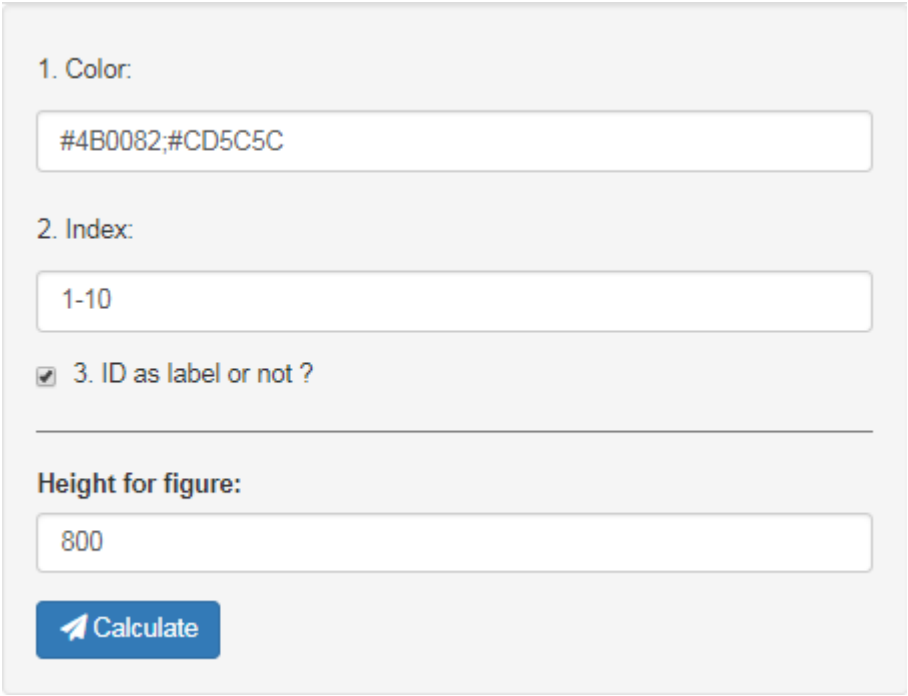
Height for figure: adjusting the height of the figure.

7. Ridgeline

This module shows the density distribution of all object fold changes (log2 here) across each ID/term, and colour corresponds to adjusted p values. This plot was implemented with the ridgeplot function in the enrichplot package[15].



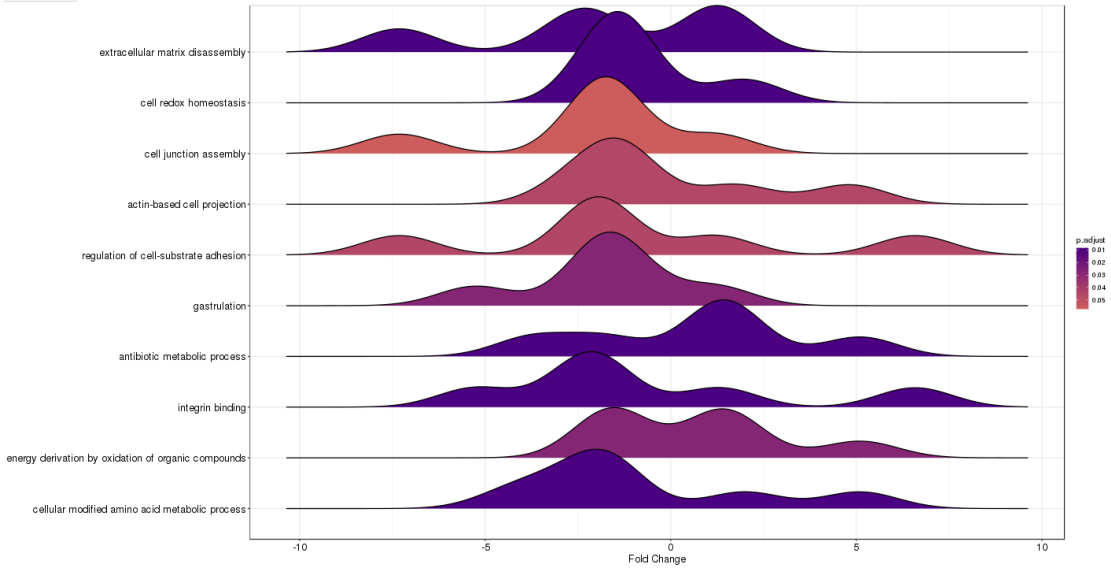
Parameters in this module:



Color: to change the ridgeline colour. Users can type in two colour names with a semicolon. The first one is for the lowest adjusted p value, and the second one is for the highest adjusted p value.

Index: which terms or objects will be plotted. The default '1-10' means the top 10 (1 to 10) terms or objects are shown. If users type in '10-21', it means it shows the 10th to 21st terms or objects (total of 12 terms or objects). If users input '1,10,21', this means it will show the 1st, the 10th, and the 21st terms or objects (total of three terms or objects).

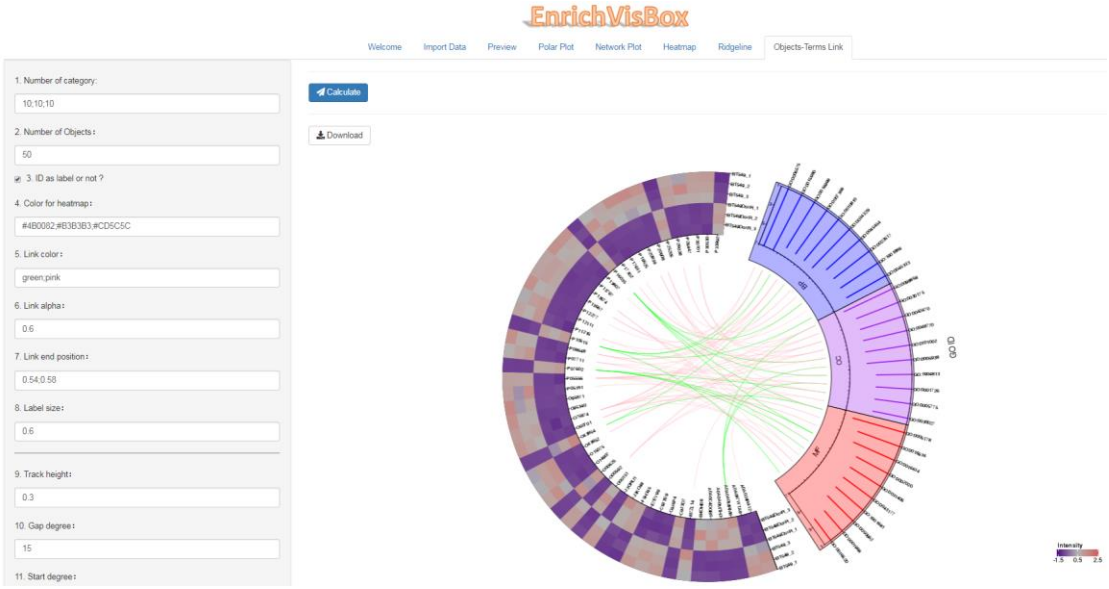
ID as label or not: if selected, the IDs will be shown; otherwise, the terms will be shown, as below.



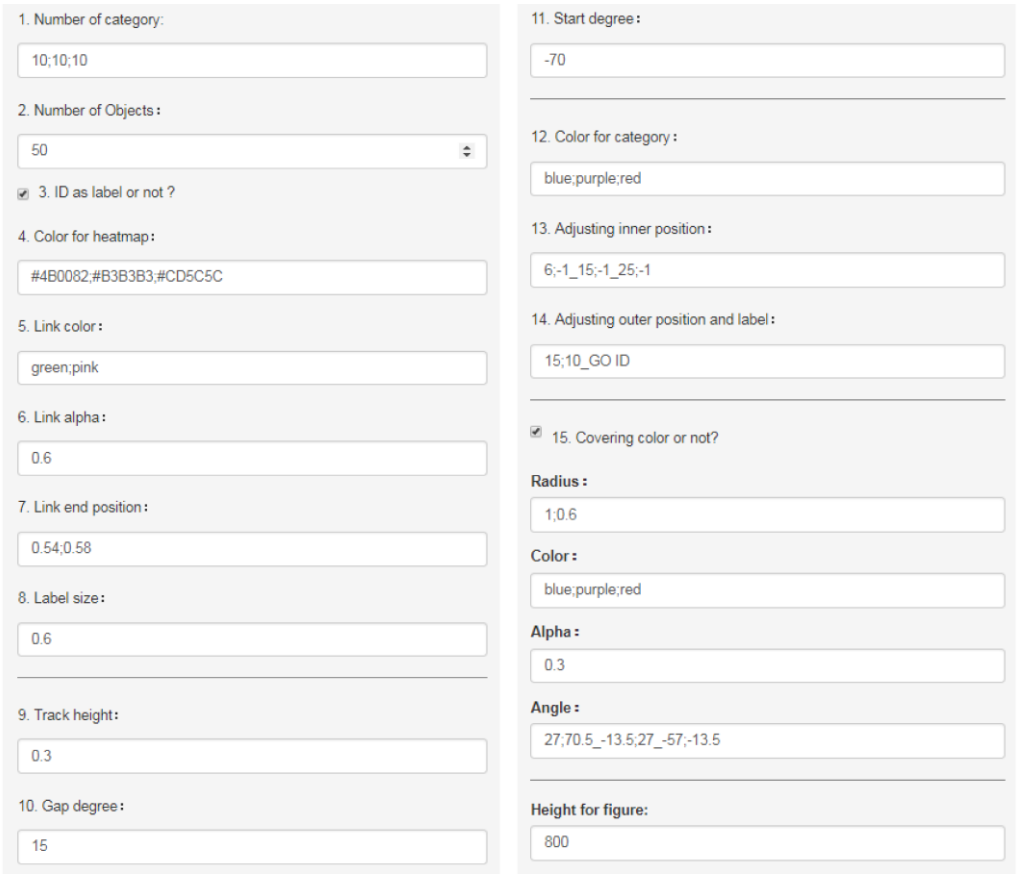
Height for figure: adjusting the height of the figure.

8. Objects–Terms Link

This is a kind of variant chord plot. The left part is the intensity heatmap of proteins in every sample, and the right part is a bar plot of each ID/term; the two parts are linked via coloured lines (green means up-regulated, brown means down-regulated, and colours can be adjusted in the corresponding parameter). This plot was implemented with the circos.trackPlotRegion, circos.rect, circos.lines, and draw.sector functions in the circlize package [20].



Parameters in this module:



Warm tips:

Because this module is somewhat complicated, users need to be careful when analysing their own data, especially when adjusting these parameters.

1. Number of category: which terms will be plotted. The index number should be same as the category number and linked with semicolons. For instance, there are three categories in the example data, and, to show the top 10 (1 to 10) terms in each category, one inputs '10;10;10' here.

2. Number of objects: how many objects in the expression data users want to show. For instance, if the user types in '50' here, it means the first 50 proteins in the expression data will be placed on the plot.

3. ID as label or not: if selected, the IDs will be shown; otherwise, the terms will be shown.

4. Color for heatmap: to change the heatmap colour, which corresponds to the intensity value in the expression data. Users should input three colour names here. The first one is for the lowest intensity value, the second one is for the middle intensity value, and the third one is for the highest intensity value.

5. Link color: to change the line colour. Users should type in two colour names with a semicolon. The first one indicates the negative fold change value (log2 here), and the second one corresponds to the positive fold change value (log2 here).

6. Link alpha: to change line colour transparency. The value should be in (0, 1).

7. Link end position: to change the line position to objects or IDs/terms. Users should type in two values here. The first one is for adjusting the line end distance to objects, and the second one is for adjusting the line end distance to IDs/terms. When the value is larger, the line end is further from the objects or IDs/terms.

8. Label size: to change the size of labels.

9. Track height: to change the height of tracks. It is the percentage according to the radius of the unit circle. The height includes the top and bottom cell paddings but not the margins.

10. Gap degree: to change the gap between two neighbour sectors.

11. Start degree: to change the starting degree from which the circle begins to draw.

12. Color for category: to change the category colour. The colour names and category number should be the same, for instance, in the example data, there are three categories, so there should be three colours here, which are linked with semicolons.

13. Adjusting inner position: to change the inner category title position. The position contains the values on the *x*-axis and *y*-axis linked with a semicolon. Every two positions are linked with an underline. For example, if '6;-1_15;-1_25;-1' is input here, it means that '6;-1' is for the first category ('BP') position ('6' is *x*-axis value, '-1' is *y*-axis value), '15;-1' is for the second category ('CC') position ('15' is *x*-axis value, '-1' is *y*-axis value), and '25;-1' is for the third category ('MF') position ('25' is *x*-axis value, '-1' is *y*-axis value).

14. Adjusting outer position and label: to change the outer title position and title text, they are connected with an underline. For example, if '15;10_GO ID' is input here, '15;10' is for the title position ('15' is *x*-axis value, '10' is *y*-axis value), and 'GO ID' is the title.

15. Covering color or not: whether to change the sector colours for every category. If selected, users will adjust the parameters, including *Radius* (radii for the outer arc and the inner arc in the sector, linked with a semicolon), *Color* (colours for sectors), *Alpha* (colour transparency), *Angle* (start and end degrees for every sector in a counterclockwise direction, linked with a semicolon, while the

angles between every two sectors are linked with an underline — for instance, if '27;70.5_-13.5;27_-57;-13.5' is input here, '27;70.5' is for the first sector ('27' is the start degree, '70.5' is the end degree), '-13.5;27' is for the second sector ('-13.5' is the start degree, '27' is the end degree), and '-57;-13.5' is for the third sector ('-57' is the start degree, '-13.5' is the end degree)).
16. Height for figure: adjusting the height of the figure.

If users set the correct parameters based on their own data, and then click the 'Calculate' button in each module, the figure will be placed on the right panel. They can also click the 'Download' button to save the results as a pdf file on their local computer.

II. Supplementary Figures

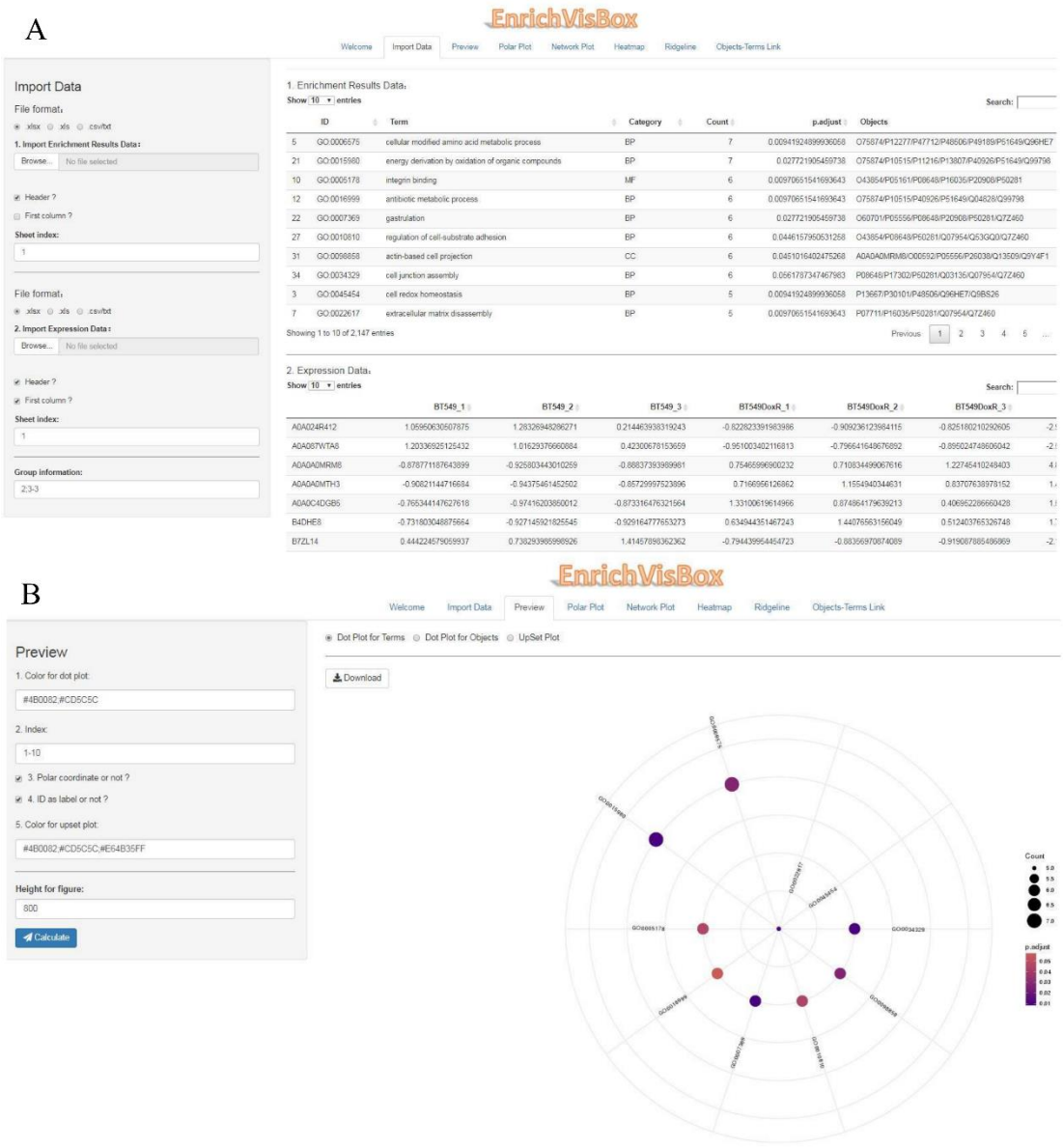


Figure S1. The data input module and the visualization module in EnrichVisBox. A. The data input module. Users can click 'Import Data' in the navigation bar and input their own data by clicking the 'Browse' button in this part. B. The visualization module. This part is mainly used to plot the uploaded data. The 'Preview', 'Polar plot', 'Network Plot', 'Heatmap', 'Ridgeline', and 'Objects-Terms Link' belong to this module, and the detailed description about each function can be found in the supplementary notes.

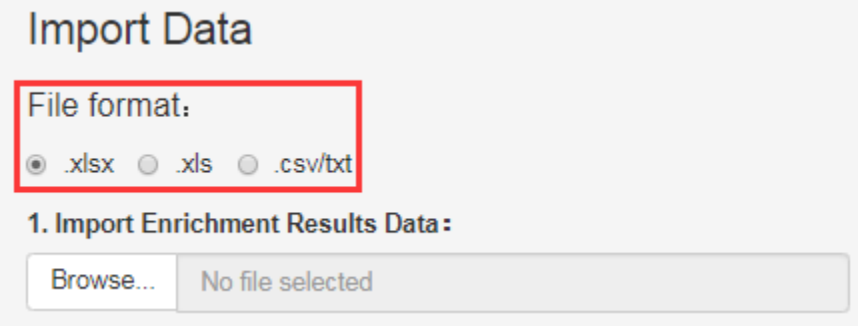


Figure S2. Table file formats supported in EnrichVisBox. The common table file formats (.xlsx, .xls. .csv, and .txt) are all supported. Users should select the correct file format based on their own data.

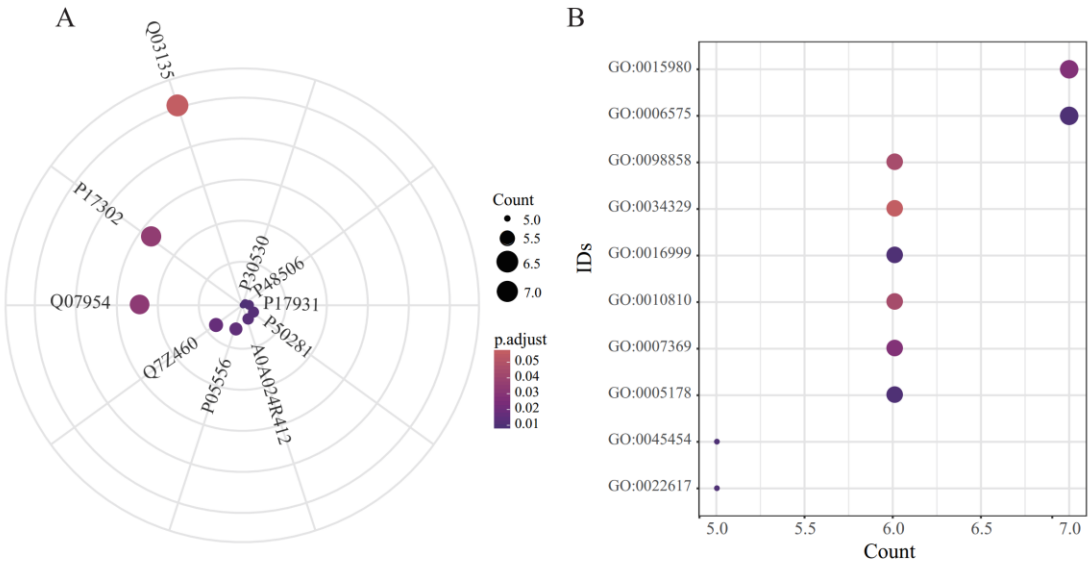


Figure S3. Bubble plot and upset plot for previewing basic information of users' data. (A) Polar bubble plot for objects; here, objects are gene/protein/metabolite IDs or names. (B) Normal bubble plot for IDs/terms. Colours correspond to adjusted p values, and the count corresponds to the number of enriched IDs/terms/objects.

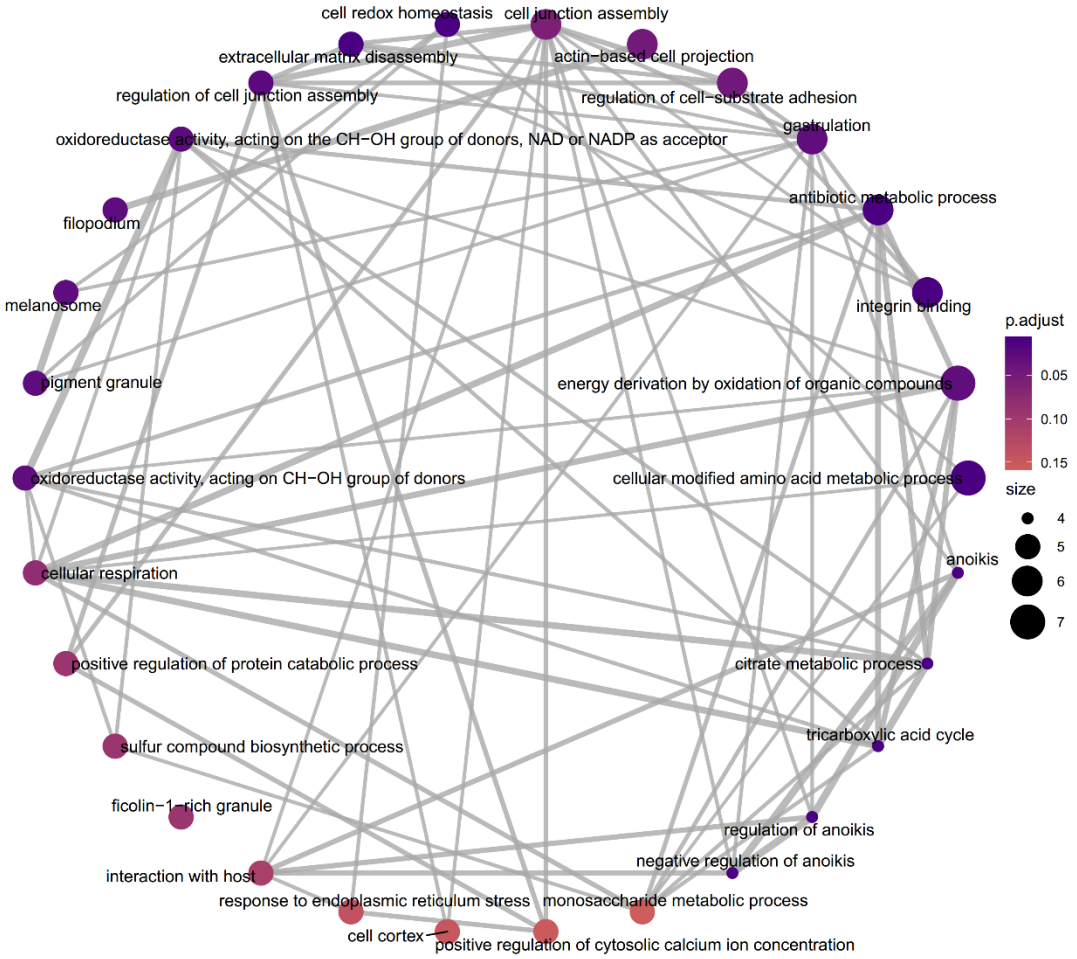


Figure S4. Network plot with edges linking overlapped objects among the enriched ID/terms. Objects here are gene/protein/metabolite IDs or names. If the number of overlapped objects between two IDs/terms is larger, the line size is bigger. Colours of points correspond to adjusted p values.

Table S1. The visualization categories implemented in EnrichVisBox compared with other tools.

Tool names	GOplot[21]	clusterProfiler[15]	pathview[22]	EnrichVisBox
GUI	×	×	×	√
Bubble plot	√	√	×	√
Upset plot	×	√	×	√
Polar bar plot	×	×	×	√
Rectangle plot	√	√	√	√
Ridgeline plot	×	√	×	√
Network plot	×	√	×	√
Chord plot	√	×	×	√

Symbols used for visualization category evaluations with "√" for present, "×" for absent.

Table S2. The summarized information on the testing of this web tool using different operation systems and browsers.

Operation System	Version	Chrome	Firefox	Safari
Windows	7	68.0.3440.106	63.0.3	not tested
Linux	CentOS 7	not tested	52.8.0	not tested
MacOS	HighSierra	70.0.3538.110	not tested	12.0.1

III. References

1. Carbon S, Ireland A, Mungall CJ, et al. AmiGO: online access to ontology and annotation data. Bioinformatics. Jan 15 2009;25(2):288-289.
2. Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic acids research. 2000;28(1):27-30.
3. Reimand J, Isserlin R, Voisin V, et al. Pathway enrichment analysis and visualization of omics data using g: Profiler, GSEA, Cytoscape and EnrichmentMap. Nat Protoc. 2019;14(2):482.
4. Das CK, Linder B, Bonn F, et al. BAG3 Overexpression and Cytoprotective Autophagy Mediate Apoptosis Resistance in Chemoresistant Breast Cancer Cells. Neoplasia. Mar 2018;20(3):263-279.
5. Xia J, Wishart DS. MetPA: a web-based metabolomics tool for pathway analysis and visualization. Bioinformatics. 2010;26(18):2342-2344.
6. Zhang Z, Bast RC, Yu Y, et al. Three biomarkers identified from serum proteomic analysis for the detection of early stage ovarian cancer. Cancer research. 2004;64(16):5882-5890.
7. Brazma A, Vilo J. Gene expression data analysis. Febs Lett. 2000;480(1):17-24.
8. Thevenot E, Thevenot EA, Roux A, Xu Y, Ezan E, Junot C. Multivariate analysis with PCA and (O) PLS (-DA).
9. Chen EY, Tan CM, Kou Y, et al. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. Bmc Bioinformatics. 2013;14(1):128.
10. Dennis G, Sherman BT, Hosack DA, et al. DAVID: database for annotation, visualization, and integrated discovery. Genome biology. 2003;4(9):R60.
11. Wang S, Zheng W, Hu L, Gong M, Yang H. MixProTool: A Powerful and Comprehensive Web Tool for Analyzing and Visualizing Multigroup Proteomics Data. Journal of Computational Biology. 2018;25(10):1123-1127.
12. Pathan M, Keerthikumar S, Ang CS, et al. FunRich: An open access standalone functional enrichment and interaction network analysis tool. Proteomics. 2015;15(15):2597-2601.
13. Xia J, Psychogios N, Young N, Wishart DS. MetaboAnalyst: a web server for metabolomic data analysis and interpretation. Nucleic acids research. 2009;37(suppl_2):W652-W660.
14. Kamburov A, Wierling C, Lehrach H, Herwig R. ConsensusPathDB—a database for integrating human functional interaction networks. Nucleic acids research. 2008;37(suppl_1):D623-D628.
15. Yu G, Wang L-G, Han Y, He Q-Y. clusterProfiler: an R package for comparing biological themes among gene clusters. Omics: a journal of integrative biology. 2012;16(5):284-287.
16. Alexa A, Rahnenfuhrer J. topGO: enrichment analysis for gene ontology. R package version. 2010;2(0):2010.
17. Wickham H. ggplot2: elegant graphics for data analysis. Springer; 2016.
18. Conway JR, Lex A, Gehlenborg N. UpSetR: an R package for the visualization of intersecting sets and their properties. Bioinformatics. 2017;33(18):2938-2940.
19. Kamada T, Kawai S. An algorithm for drawing general undirected graphs. Information processing letters. 1989;31(1):7-15.
20. Gu Z, Gu L, Eils R, Schlesner M, Brors B. circlize implements and enhances circular visualization in R. Bioinformatics. 2014;30(19):2811-2812.
21. Walter W, Sánchez-Cabo F, Ricote M. GOplot: an R package for visually combining

expression data with functional analysis. Bioinformatics. 2015;31(17):2912-2914.
22. Luo W, Brouwer C. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. Bioinformatics. 2013;29(14):1830-1831.