

Cytoscape

A Tutorial Guide



Yangyang Zhang
18/11/2023

目录

| | |
|--------------------------------|----|
| 简介 | 3 |
| 1. Cytoscape 简介 | 3 |
| 2. Cytoscape 使用 | 4 |
| 2.1 下载和安装 | 4 |
| 2.2 启动 cytoscape | 7 |
| 3. 插件介绍 | 8 |
| 3.1.1 下载插件 | 9 |
| 3.1.2 string 网站用于准备输入文件 | 9 |
| 3.1.3 cytoHubba 实操 | 11 |
| 3.2 MCODE(子网络构建) | 14 |
| 3.2.1 安装 | 14 |
| 3.2.2 MCODE 实操 | 14 |
| 3.3 stringAPP (蛋白互作网络) | 16 |
| 3.3.1 下载 | 17 |
| 3.3.2 stringAPP 实操 | 17 |
| string 网站和 cytoscape 的互动 | 19 |
| 3.4 Centiscape (计算多个中心值) | 20 |
| 3.4.1 安装 | 20 |
| 3.4.2 实操 | 21 |
| 3.5 iRegulon (转录调控) | 22 |
| 3.5.1 安装 | 23 |
| 3.5.2 实操 | 23 |
| 4. 补充 | 27 |
| 4.1 Cytoscape 可视化 | 27 |

| | |
|------------------------------|----|
| 4.1.1 如何绘制蛋白互作“圈圈”网络图? | 27 |
| 4.2 基于 python 的网络图绘制..... | 32 |
| 4.2.1 基于 omicverse | 32 |
| 4.3 基于 R 的网络图绘制 | 39 |
| 4.4 GENEMANIA 网站..... | 43 |
| 4.4.1 实操 | 44 |

Cytoscape

Yangyang Zhang

2023-11-18

简介

本手册使用的 Cytoscape 基于 3.x 的架构构建，开发者 API 和用户控件已启用。这个版本是笔者作为课堂作业的最初版本，在未来版本中，将持续调整和改进软件和文档。

本手册将介绍包括 cytoscape 的安装，十大插件的使用，以及如何使用相关网站和 R 进行网络探索。如果你不喜欢使用 pdf 版本，也可以网页版访问。本手册相关代码和数据已上传：[github](https://github.com/pigudog/cytoscape)(<https://github.com/pigudog/cytoscape>)

尽管本手册详细介绍了 Cytoscape 的安装和使用，以及对于网络图绘制进行了进一步的补充，但是为了更全面的了解 Cytoscape 及其生态系统以及使用最新版本的软件，笔者建议阅读 <https://cytoscape.org> 网站上的 **Welcome Letter**

1. Cytoscape 简介

Cytoscape 是一个广泛用于生物信息学和系统生物学研究的开源软件工具，用于可视化、分析和解释生物网络数据。以下是 Cytoscape 的一些常见用法：

- 网络可视化：** Cytoscape 主要用于可视化生物网络，例如蛋白质相互作用网络、代谢网络、基因调控网络等。用户可以通过导入网络数据文件（如 SIF、XGMML 等格式）来构建和展示网络图。网络中的节点代表生物分子（如基因、蛋白质等），边代表它们之间的关系（如相互作用、调控等）。用户可以自定义节点和边的样式、颜色、标签等，以便更好地展示网络结构和功能。
- 网络分析：** Cytoscape 提供了许多网络分析工具，用于探索网络的拓扑结构、关键节点、社区结构等。用户可以计算节点的度中心性、介数中心性、紧密中心性等指标，以评估节点在网络中的重要性。此外，Cytoscape 还支持网络布局算法，以便在图上更好地分布节点，从而更清晰地展示网络拓扑。
- 数据整合：** 用户可以将其他生物信息学数据集与网络数据集整合，以便在网络上显示附加信息。例如，可以将基因表达数据、蛋白质功能注释等与网络节点关联起来，从而在网络图上展示多维度的信息。

4. **模块和通路分析:** Cytoscape 允许用户通过插件扩展功能, 以进行更高级的分析, 如寻找网络中的功能模块、通路分析等。这些插件可以帮助用户识别网络中的相关节点子集, 从而更好地理解生物学过程。
5. **网络互动和分享:** Cytoscape 允许用户对网络图进行交互操作, 如放大、缩小、拖动节点等。用户还可以保存网络图为图像或特定格式的文件, 以便与同事共享研究结果。
6. **插件支持:** Cytoscape 具有丰富的插件生态系统, 用户可以根据需要选择和安装插件, 以扩展 Cytoscape 的功能。这些插件可以提供各种高级分析工具、网络布局算法、数据导入导出功能等。

总之, Cytoscape 是一个强大的工具, 用于探索、可视化和分析生物网络数据, 有助于生物信息学研究人员更好地理解生物体系的复杂性和相互作用。

2. Cytoscape 使用

下载和安装 Cytoscape 有多种方式, 可以参考 <https://cytoscape.org> 网站上的下载页面。 - 用于 Windows, macOS 和 Linux 平台的安装包 (适用于大多数用户)。 - 从发行版压缩包安装 Cytoscape。 - 从源代码构建 Cytoscape。可以从 [Git 仓库](#) 中获取最新代码。

我们以 windows10 为例作为示范

2.1 下载和安装

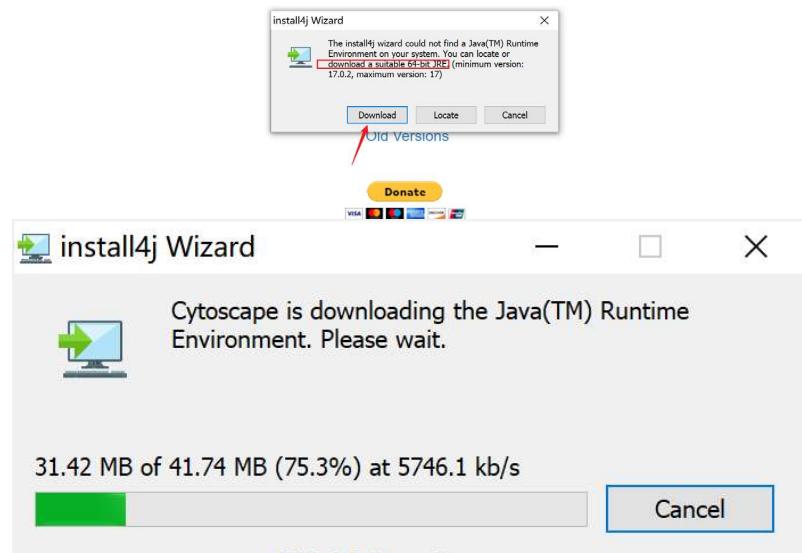
进入 [cytoscape 官网](#) 下载该软件



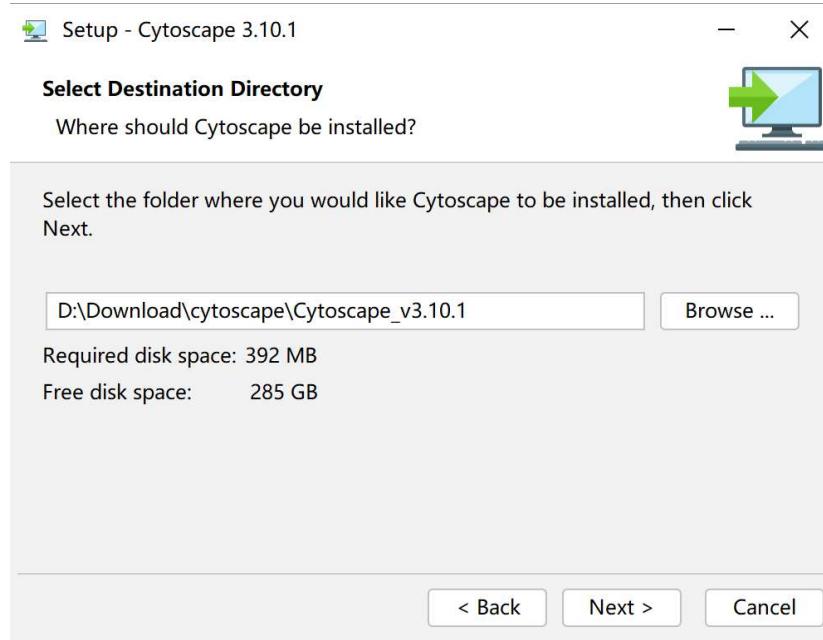
安装包时，会自动询问你是否安装 java17 的依赖，直接选择 download 即可

Java 17 will be automatically installed if not already present. If you experience difficulty with this, manual installers for Java can be downloaded [here](#).

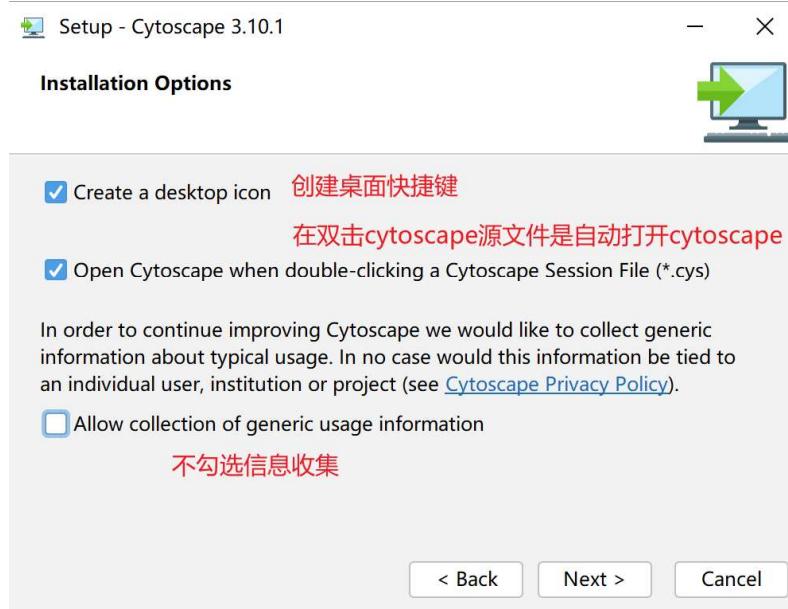
Problems? [Read this page first](#)



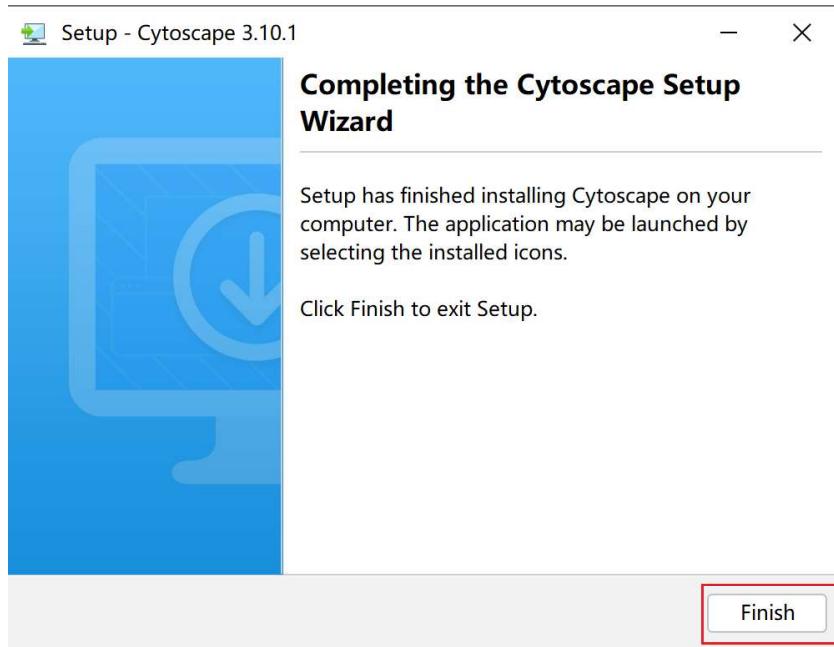
选择安装的目录



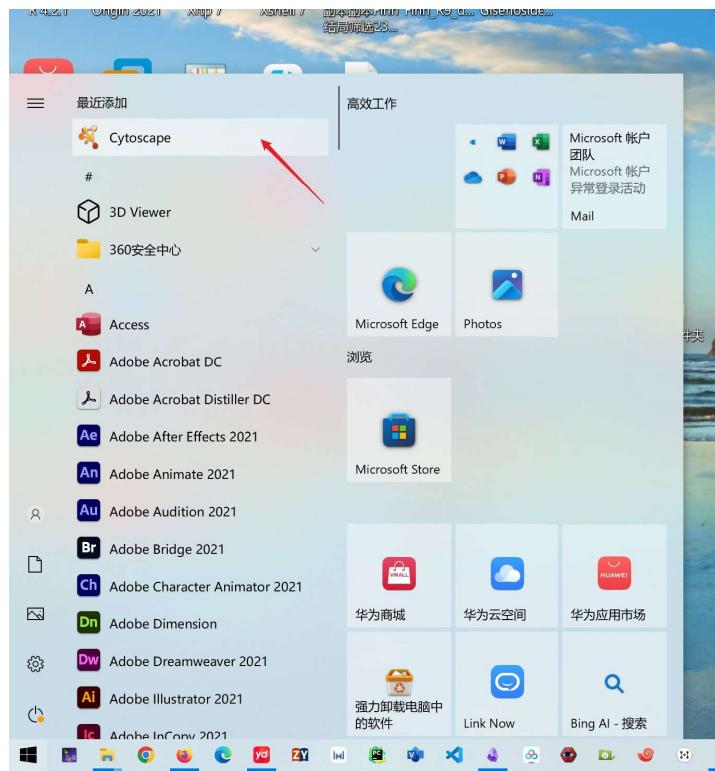
勾选相关条件：



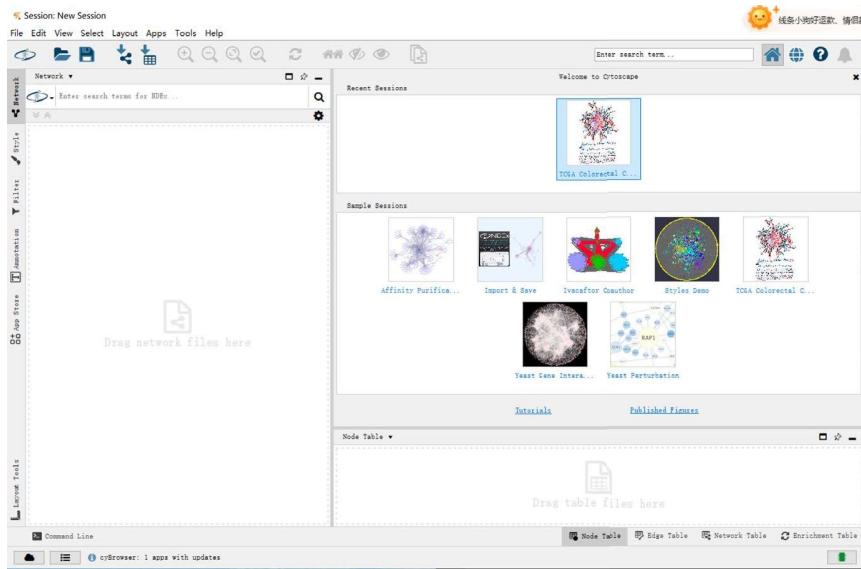
继续点击 next，即可安装完成



2.2 启动 cytoscape



启动 Cytoscape 后，将出现一个如下所示的窗口：



如果你的 Cytoscape 窗口与此不同，可能需要进一步配置。请参考 <https://cytoscape.org> 网站上的 **Release Notes**。

3. 插件介绍

我们需要先准备一个基因集（以线粒体基因与多囊卵巢综合征差异基因的交集 38 个基因为例，以上传在 [mito_hub.txt](#)

(关键基因) 目的：先用 STRING 网站构建 PPI 网络，当存在上百个基因的对应关系时，就需要再利用插件通过拓扑网络算法给每个基因赋值，**排序发现其关键基因 (hub gene)**和子网络

- cytoHubba 根据 nodes 在网络中的属性进行排名。它提供了 **11 种拓扑分析方法**，包括：
 - Degree
 - Edge Percolated component
 - Maximum neighborhood component
 - Density of Maximum Neighborhood Component
 - Maximal Clique Centrality
 - six centralities(Bottleneck,Eccentricity,Closeness, Radiality,Betweenness, Stress)

文章 [cytoHubba: identifying hub objects and sub-networks from complex interactome](#)

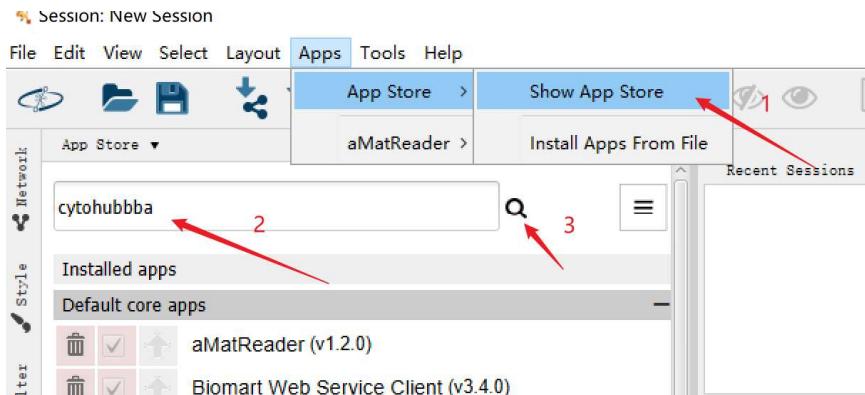
Conclusions

In this study, we implement our network scoring methods, MCC, MNC and DMNC, and eight other popular methods into a Cytoscape plugin, *cytoHubba*. Through the extendable, flexible and modulated properties of Cytoscape, *cytoHubba* can work together with other plugins. The computing processes had been optimized and can complete all eleven analysis on a common desktop/ notebook in a reasonable time cost. We also improve the network retrieving function in *cytoHubba* control panel. Therefore, users can utilize a PPIs network from public domain and extract sub-networks based on users' domain-knowledge.

Among the 11 methods, the newly proposed method MCC performs better than the others. MCC captures more essential proteins in the top ranked list in both high-degree and low-degree proteins. Another method, DMNC, catches different set of essential proteins suggesting it scores the network in different way. Since the biological network is heterogeneous, it is reasonable to use more than one method for catching essential proteins. We hope this handy tool can serve as good starting points to new therapies and novel insights in understanding basic mechanisms controlling normal cellular processes and disease pathologies.

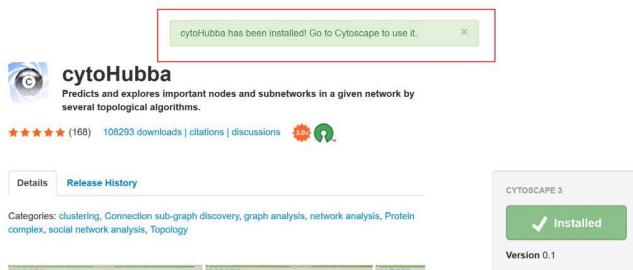
结果：可以看出MCC方法得出来的结论是最佳的，能得到更多且排名靠前的蛋白质。所以运用这个方法得到我们的关键基因说服力是相对较强的！

3.1.1 下载插件



- 首先点击 Apps-App Store-Show App Store 打开插件商店
- 在 Search 中输入所需要的插件: cytohubba

这样就会自动跳转到官网, 点击下载, install 成功后, 官网会显示 cytoHubba has been installed!, 我们回到 Cytoscape 软件中使用



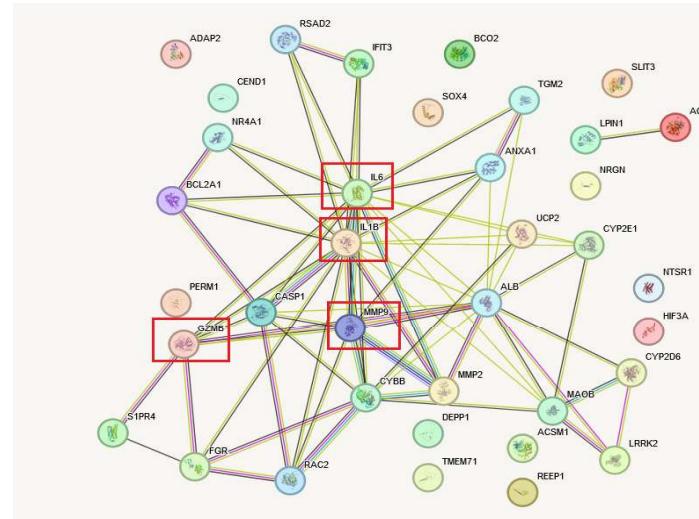
可以直接看到 cytoHubba 已经安装成功



3.1.2 string 网站用于准备输入文件

STRING (<https://string-db.org>) 是一个非常全面的蛋白互作网络数据库, 里面存储了非常多物种和基因的相互作用关系。我们只要把基因名字提交上去, 就能够判定他们之间时候有互作关系了。

- 可以看到 38 个基因所组成的网络，包括 MMP9 相关的基质重构和黏附，IL6 介导的相关脂肪分解和消除胰岛素抵抗等，这些基因和其表达的蛋白均和多囊卵巢综合征密切相关，但是谁是真正的关键基因（蛋白）？



我们选择 cytoscape 相关格式进行下载，我们会得到一个叫 [string_interactions_short.tsv](#) 的文件

| File Format | Description |
|---|--|
| ... as bitmap image | download file format: PNG: portable network graphic |
| ... as a high-resolution bitmap | download same PNG format, but at higher resolution |
| ... as a vector graphic | download SVG: scalable vector graphic - can be opened and edited in Illustrator, CorelDraw, Dia, etc. |
| ... as short tabular text output | download "TSV: tab-separated values - can be opened in Excel and Cytoscape (lists only one-way edges: A-B) |
| ... as tabular text output | download "TSV: tab-separated values - can be opened in Excel (reciprocal edges: A-B) |
| ... as XML | download XML: standard XML file |
| ... protein node degrees | download node degree of proteins in your network (given the current score cut-off) |
| ... network coordinates | download a flat-file format describing the coordinates and colors of nodes in the network |
| ... protein sequences | download MFA: multi-fasta format - containing the aminoacid sequences in the network |
| ... protein annotations | download a tab-delimited file describing the names, domains and descriptions of proteins in your network |
| ... functional annotations | download a tab-delimited file containing all known functional terms of proteins in your network |

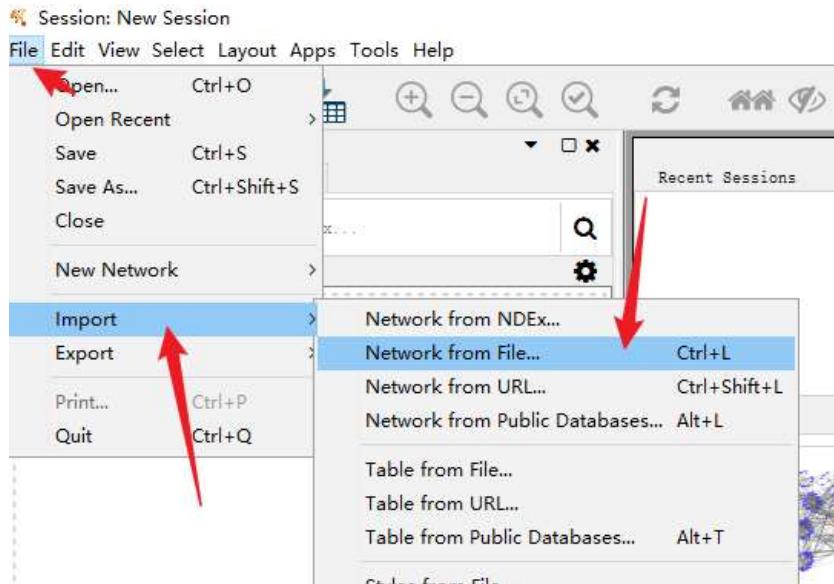
文件格式如下，可以看到左侧的基因对基因是一个长数据框格式的文件，右边最核心的 value 是 combined_score，代表着两个基因的相关性

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|----|--------|---------|----------------------|-----------------------|----------------------------|-------------|---------------------------|----------|--------------|--|--------------------|----------------------|----------------|
| | anode1 | node2 | node1_align_id | node2_align_id | neighborhood_on_chromosome | gene_fusion | phylogenetic_cooccurrence | homology | coexpression | experimentally_determined_interactions | database_annotated | automated_annotation | combined_score |
| 1 | ANXA1 | ANXA1 | 9050.ENSP0000033882 | 9050.ENSP00000337958 | 0 | 0 | 0 | 0 | 0.059 | 0.084 | 0 | 0.445 | 0.479 |
| 2 | ANXA1 | C21orf8 | 9050.ENSP0000033882 | 9050.ENSP00000337958 | 0 | 0 | 0 | 0 | 0.073 | 0.069 | 0 | 0.616 | 0.64 |
| 3 | ALB | MNP2 | 9050.ENSP0000033887 | 9006.ENSP00000321970 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.890 | 0.895 |
| 4 | ALB | ANXA1 | 9050.ENSP0000033887 | 9050.ENSP0000033882 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.239 | 0.431 |
| 5 | ALB | LRRK2 | 9050.ENSP0000033887 | 9050.ENSP000003288910 | 0 | 0 | 0 | 0 | 0 | 0.292 | 0 | 0.423 | 0.434 |
| 6 | ALB | MAD2 | 9050.ENSP0000033887 | 9050.ENSP00000337958 | 0 | 0 | 0 | 0 | 0.06 | 0 | 0 | 0.443 | 0.445 |
| 7 | ALB | UCP2 | 9050.ENSP0000033887 | 9050.ENSP00000337958 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.447 | 0.447 |
| 8 | ALB | TGM2 | 9050.ENSP0000033887 | 9050.ENSP0000033830 | 0 | 0 | 0 | 0 | 0.088 | 0 | 0 | 0.485 | 0.509 |
| 9 | ALB | CYP2D6 | 9050.ENSP0000033887 | 9050.ENSP00000349150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.553 | 0.553 |
| 10 | ALB | CYP2C19 | 9050.ENSP0000033887 | 9050.ENSP00000349150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.611 | 0.611 |
| 11 | ALB | CASP1 | 9050.ENSP0000033887 | 9050.ENSP00000343138 | 0 | 0 | 0 | 0 | 0.049 | 0 | 0 | 0.095 | 0.097 |
| 12 | ALB | CYP2C11 | 9050.ENSP0000033887 | 9050.ENSP00000344269 | 0 | 0 | 0 | 0 | 0.049 | 0 | 0 | 0.639 | 0.638 |
| 13 | ANXA1 | MNP2 | 9050.ENSP0000033887 | 9006.ENSP00000321970 | 0 | 0 | 0 | 0 | 0.053 | 0.069 | 0 | 0.734 | 0.745 |
| 14 | ALB | MNP2 | 9050.ENSP0000033887 | 9050.ENSP0000033405 | 0 | 0 | 0 | 0 | 0.053 | 0.069 | 0 | 0.892 | 0.892 |
| 15 | ALB | MW9 | 9050.ENSP0000033887 | 9050.ENSP0000033405 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.571 | 0.602 |
| 16 | ANXA1 | LRRK2 | 9050.ENSP0000033887 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.108 | 0 | 0 | 0.987 | 0.982 |
| 17 | ANXA1 | TGM2 | 9050.ENSP0000033887 | 9050.ENSP0000033830 | 0 | 0 | 0 | 0 | 0.093 | 0 | 0 | 0.632 | 0.632 |
| 18 | ANXA1 | UCP2 | 9050.ENSP0000033887 | 9050.ENSP0000033830 | 0 | 0 | 0 | 0 | 0.098 | 0 | 0 | 0.641 | 0.641 |
| 19 | ANXA1 | I6P | 9050.ENSP00000338159 | 9050.ENSP00000338675 | 0 | 0 | 0 | 0 | 0.108 | 0 | 0 | 0.522 | 0.556 |
| 20 | BCXL1 | ANXA1 | 9050.ENSP00000338159 | 9050.ENSP000003291341 | 0 | 0 | 0 | 0 | 0.355 | 0 | 0 | 0.449 | 0.459 |
| 21 | BCXL1 | CYP2C19 | 9050.ENSP00000338159 | 9050.ENSP000003291341 | 0 | 0 | 0 | 0 | 0.127 | 0 | 0 | 0.449 | 0.459 |
| 22 | BCXL1 | CASP1 | 9050.ENSP00000338159 | 9050.ENSP00000343138 | 0 | 0 | 0 | 0 | 0.251 | 0.125 | 0 | 0.34 | 0.53 |
| 23 | BCXL1 | CYP2D6 | 9050.ENSP00000338159 | 9050.ENSP00000349150 | 0 | 0 | 0 | 0 | 0.093 | 0.292 | 0 | 0.681 | 0.77 |
| 24 | CASP1 | CASP1 | 9050.ENSP00000338004 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.153 | 0 | 0 | 0.617 | 0.651 |
| 25 | CASP1 | RAC2 | 9050.ENSP00000338004 | 9050.ENSP00000331318 | 0 | 0 | 0 | 0 | 0.291 | 0.053 | 0 | 0.217 | 0.249 |
| 26 | CASP1 | I6P | 9050.ENSP00000338004 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.290 | 0.071 | 0 | 0.307 | 0.309 |
| 27 | CASP1 | MW9 | 9050.ENSP00000338004 | 9050.ENSP0000033405 | 0 | 0 | 0 | 0 | 0.14 | 0 | 0 | 0.510 | 0.560 |
| 28 | CASP1 | CYP2B | 9050.ENSP00000338004 | 9050.ENSP00000331318 | 0 | 0 | 0 | 0 | 0.267 | 0 | 0 | 0.543 | 0.651 |
| 29 | CASP1 | MNP2 | 9050.ENSP00000338004 | 9006.ENSP00000337958 | 0 | 0 | 0 | 0 | 0.129 | 0 | 0 | 0.589 | 0.589 |
| 30 | CYBB | MNP2 | 9050.ENSP00000338159 | 9050.ENSP000003221970 | 0 | 0 | 0 | 0 | 0.069 | 0 | 0.9 | 0.881 | 0.957 |
| 31 | CYBB | MAD2 | 9050.ENSP00000338159 | 9050.ENSP00000324070 | 0 | 0 | 0 | 0 | 0.039 | 0.015 | 0.8 | 0.702 | 0.748 |
| 32 | CYBB | ANXA1 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.181 | 0 | 0 | 0.702 | 0.748 |
| 33 | CYBB | MNP2 | 9050.ENSP00000338159 | 9050.ENSP00000337958 | 0 | 0 | 0 | 0 | 0.178 | 0 | 0.9 | 0.887 | 0.953 |
| 34 | CYBB | MW9 | 9050.ENSP00000338159 | 9050.ENSP0000033405 | 0 | 0 | 0 | 0 | 0.207 | 0.083 | 0 | 0.720 | 0.847 |
| 35 | CYBB | MAD2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.064 | 0 | 0 | 0.458 | 0.471 |
| 36 | CYBB | UCP2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.096 | 0 | 0 | 0.378 | 0.411 |
| 37 | CYBB | LRRK2 | 9050.ENSP00000338159 | 9050.ENSP000003288910 | 0 | 0 | 0 | 0 | 0.093 | 0 | 0 | 0.591 | 0.700 |
| 38 | CYP2D6 | MNP2 | 9050.ENSP00000338159 | 9050.ENSP000003288910 | 0 | 0 | 0 | 0 | 0.067 | 0.051 | 0 | 0.494 | 0.5 |
| 39 | CYP2D6 | MAD2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.61 | 0.61 |
| 40 | CYP2D6 | UCP2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.510 | 0.518 |
| 41 | CYP2E1 | MAD2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0.062 | 0 | 0 | 0.392 | 0.406 |
| 42 | CYP2E1 | UCP2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.891 | 0.951 |
| 43 | CYP2E1 | UCP2 | 9050.ENSP00000338159 | 9050.ENSP00000323341 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.738 | 0.738 |

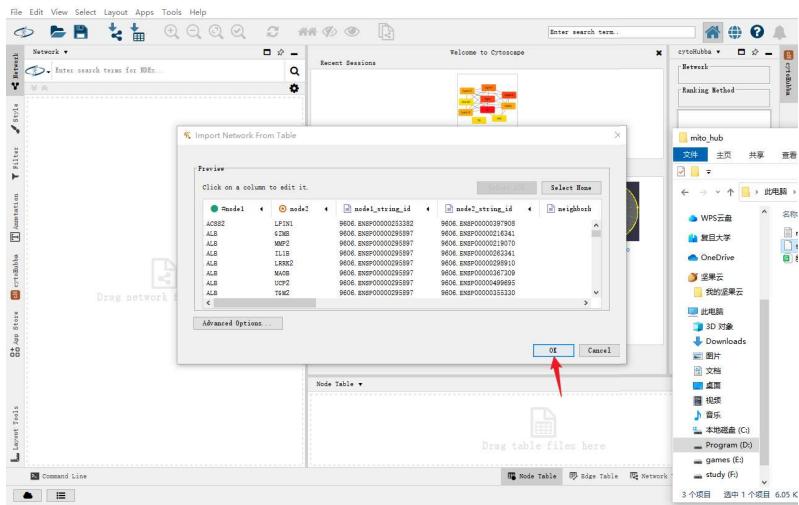
3.1.3 cytoHubba 实操

- 在没有算法的时候，我们常常会选择 edge 最多的点，接下来我们使用 cytoHubba 来完成这个操作

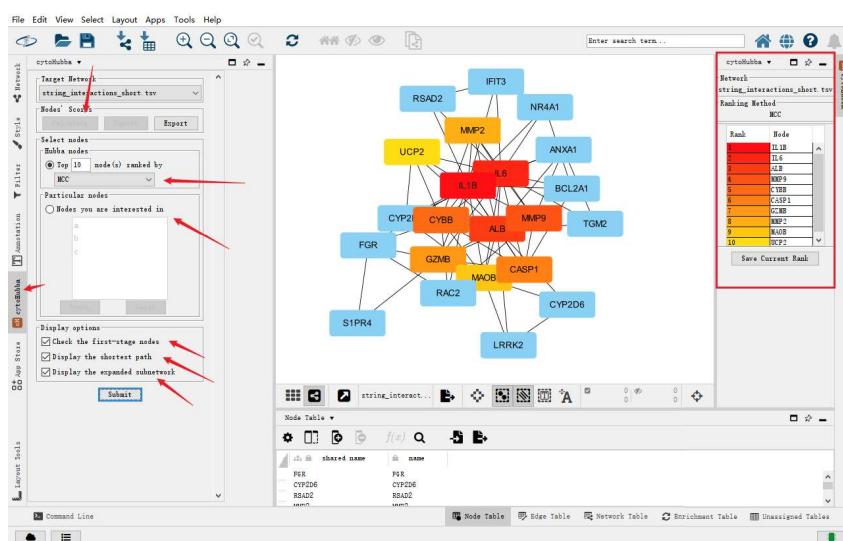
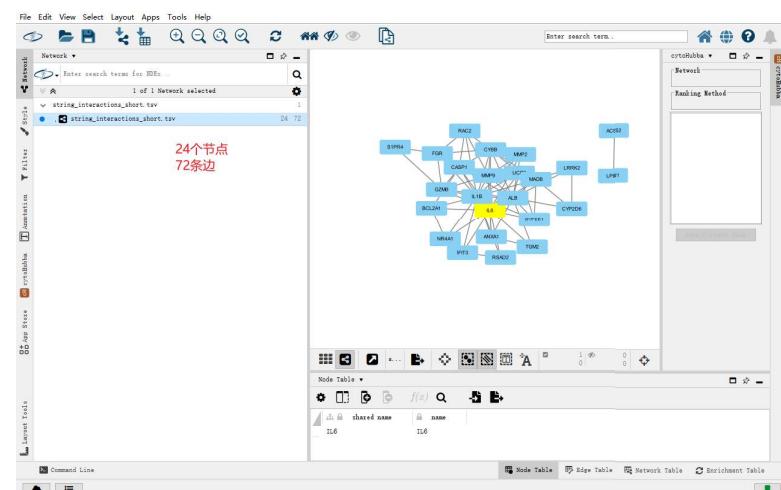
将文件拖入 cytoscape 软件，或者通过 File -> Import -> Network from File 即可导入文件



检查无误后点击 OK

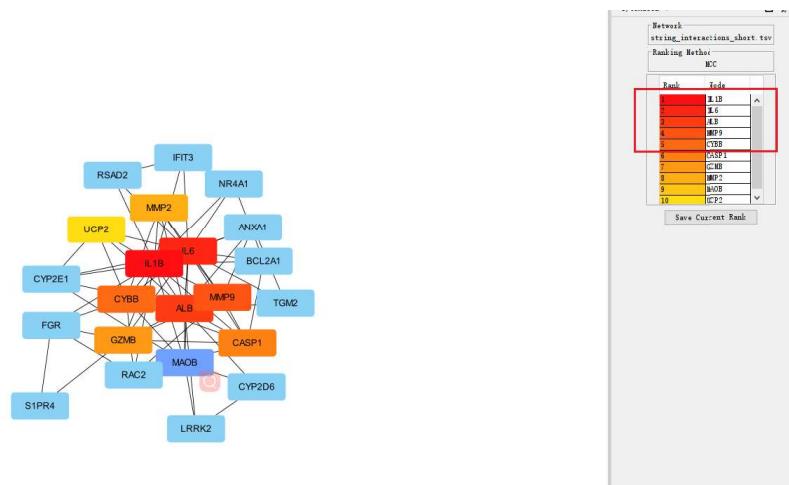


可以看到这是最初的样子

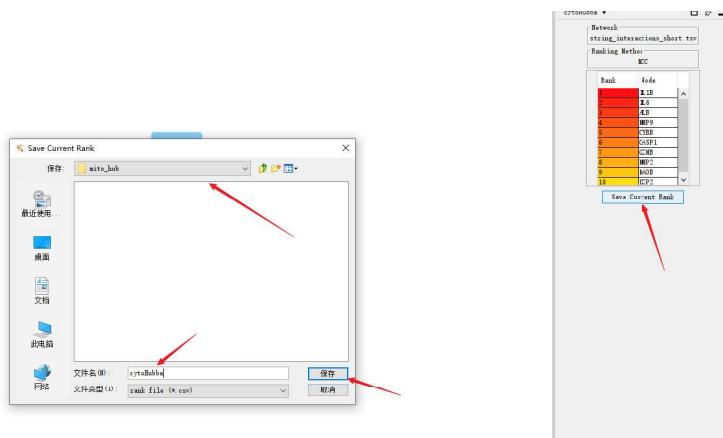


1、首先选择左边的 cytoHubba 进入工作页面 2、Target Network 选择待分析的网络 3、点击 Calculate 4、点击 Top 10，可根据自身课题需要，调整提取关键基因的数量 5、下方 MCC 可下拉选择计算方法 6、Particular nodes：选择感兴趣的基因进行分析（一般是对整个网络分析的话，此项可以忽略） 5、点击左下方 Display the shortest path：显示最短的路径；Check the first-stage nodes：显示出 hub gene 连接到的第一个节点；Display the expanded subnetwork：显示出最长的子网络 6、右边的列表就显示关键基因的排名：颜色越深，证明分数越高，越显著 7、下方可选择 Save Current Rank

同时我们会发现，得到得结果与我们最初得想法也有共通之处，但是除了计算每个节点的连线数，cytoHubba 还会对于每个节点和节点之间的相关性进行综合的打分，可以看到 IL1B, IL6, ALB, MMP9, CYBB 等几个基因在其发挥了重要的作用



随后我们将结果保存



该 csv 中包含了对应的 hub gene 和对其的打分

| A | B | C | D | E | F | G | H | I | J | K | L |
|------|--|-------|---|---|---|---|---|---|---|---|---|
| 1 | Top 10 in network string_interactions_short.tsv ranked by MCC method | | | | | | | | | | |
| Rank | Name | Score | | | | | | | | | |
| 3 | 1 IL1B | 530 | | | | | | | | | |
| 4 | 2 IL6 | 504 | | | | | | | | | |
| 5 | 3 ALB | 492 | | | | | | | | | |
| 6 | 4 MMP9 | 408 | | | | | | | | | |
| 7 | 5 CYBB | 318 | | | | | | | | | |
| 8 | 6 CASP1 | 270 | | | | | | | | | |
| 9 | 7 GZMB | 124 | | | | | | | | | |
| 10 | 8 MMP2 | 120 | | | | | | | | | |
| 11 | 9 MAOB | 54 | | | | | | | | | |
| 12 | 10 UCP2 | 48 | | | | | | | | | |

3.2 MCODE(子网络构建)

- 关于网络构建， PPI+MCODE 两者结合似乎已经成了标配。
- 定义：**MCODE (Molecular Complex Detection)** 插件是在庞大的网络中根据边和节点的关系，寻找出关键的子网络和基因，方便进行下游分析。（我理解的话，其实这方法和 **Cytohubba** 很相似，不过 Cytohubba 会提供多种算法进行选择，并会按基因的核心程度进行排序，两种方法都可以筛选出核心基因）

3.2.1 安装

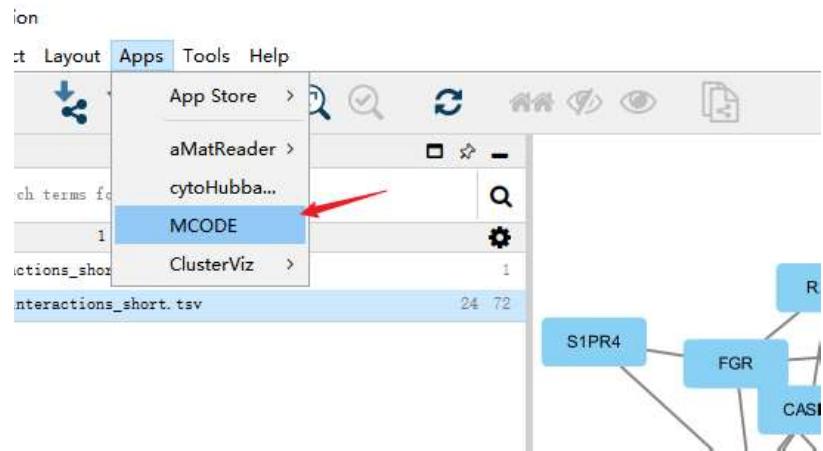
模仿 cytoHubba 安装 MCODE

The screenshot shows the CytoScape App Store interface. At the top, there's a search bar and a navigation menu. Below it, the 'MCODE' app page is displayed. The page includes the following details:

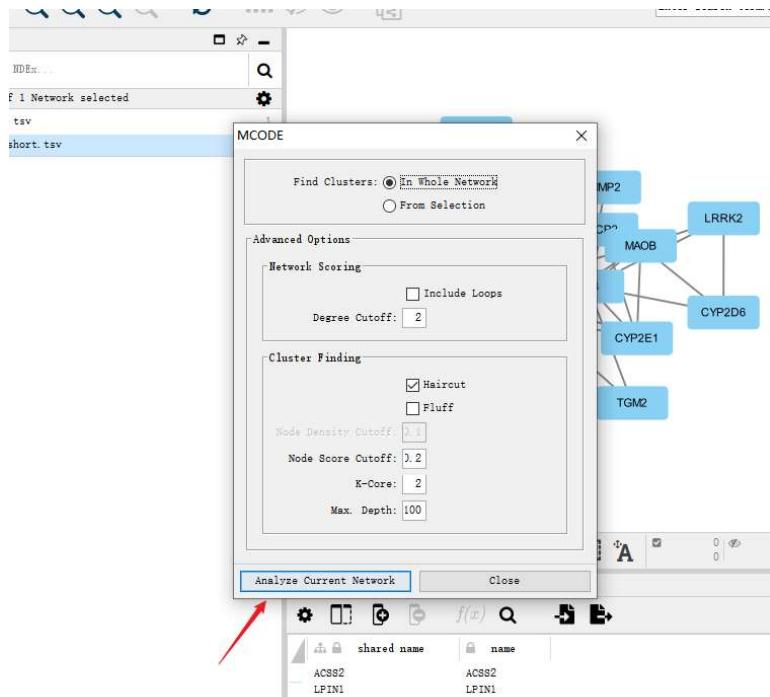
- MCODE**: The app name.
- Clusters a given network based on topology to find densely connected regions.**: A brief description.
- ★★★★★ (30) 170294 downloads | citations | discussions**: User reviews and download statistics.
- Details** and **Release History** tabs.
- Categories: automation, clustering, graph analysis**.
- CYTOSCAPE3** section: Shows the status as **Installing...**.
- Version 2.0.3**, **License Click here**, **Released 4 Jul 2023**, **Works with CytoScape 3.9**, **Download Stats Click here**, and a link to **Cytoscape v3.10.1**.
- Two small screenshots of the MCODE interface are shown below the category list.

3.2.2 MCODE 实操

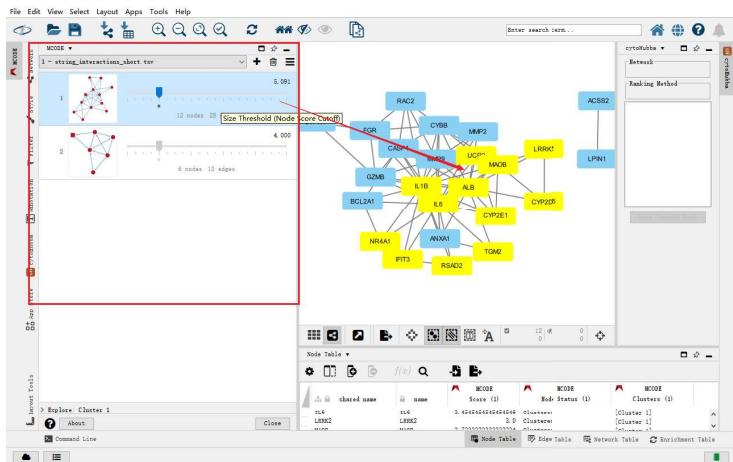
我们仍然使用之前的数据：在 Apps 中点击 MCODE，然后会在控制面板中出现 Mcode 这一面板



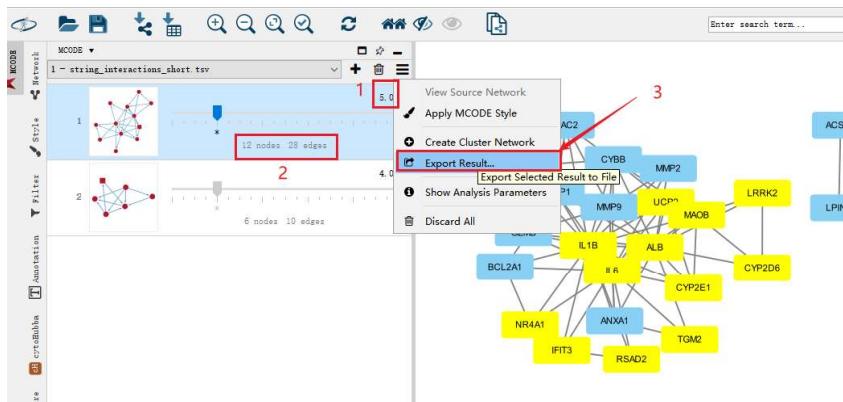
我们这里不修改默认参数，点击 Analyze Current Network 即可（当然你可以选择设置 degree 的阈值来修改子网络的筛选方式）



点击左边的子网络，就能在右面显示出来，非常方便：



- 最后，选择感兴趣的网络，点击上方三选择 Create Cluster Network
- 第 1 列的数字为网络得分，最高意味网络里的基因最关键和典型
- 第 2 个框里面是节点 nodes 和边 edges 的信息，这也是文献结果描写的重点



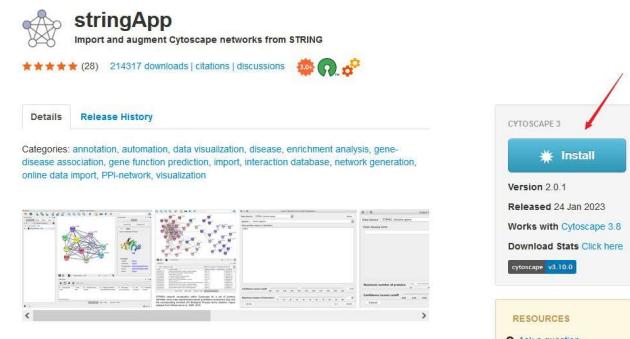
总结： - MCODE 插件一般在文章中有两种使用规律：
- 可以利用 MCODE 插件求出子网络，将分数最高的网络里面全部基因当作 hub gene
- 也有不少文章是用 Cytohubba 求 hub gene，MCODE 插件求出的结果做下游的富集分析

3.3 stringAPP (蛋白互作网络)

- 蛋白质网络一直是我们分析和可视化大量蛋白质和基因的工具。其中，最耳熟能详的是 STRING 数据库 (<https://string-db.org/>)，其适用物种有 2000 多个，能检索出编码蛋白间可能的潜在相互作用，例如物理接触、靶向调节等，最终阐述生物体中有意义的分子调节网络。
- 但有个缺点：STRING 数据库网页版只适用于构建小型网络。如要构建大型蛋白互相网络，这个网页版恐怕无法满足我们需求。
- 而这 Cytoscape 软件中的 stringApp 插件有以下好处：
 - 更适用于大型网络

- 对导入数据的可视化提供更大的灵活性
- 可与 Cytoscape 软件中其他插件联合进行数据分析
- 免费使用

3.3.1 下载

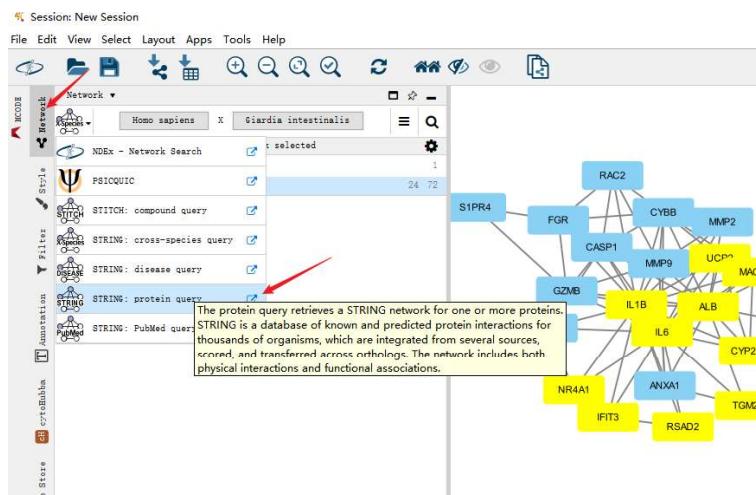


3.3.2 stringAPP 实操

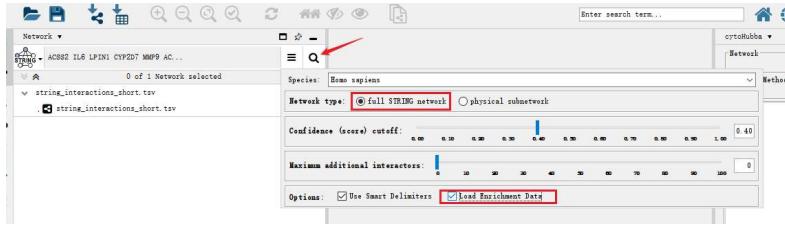
以同样的方式导入 38 个基因 - 点击图标，选择所需要的 STRING 类型，这里选择 STRING protein query

| Function | Operation | Input | Output |
|----------------------------------|---|---|----------|
| STRING: protein query | Create a network for one or multiple proteins | Protein name(s) or identifier(s) | 蛋白互作用 |
| STRING: disease query | Create a network of proteins associated with a disease | Disease name or identifier | 某疾病 |
| STRING: PubMed query | Create a network by entering a PubMed query | PubMed query | PubMed查询 |
| STITCH: protein / compound query | Create a network for one or multiple proteins and compounds | Protein/compound name(s) or identifier(s) | 化学化合物 |

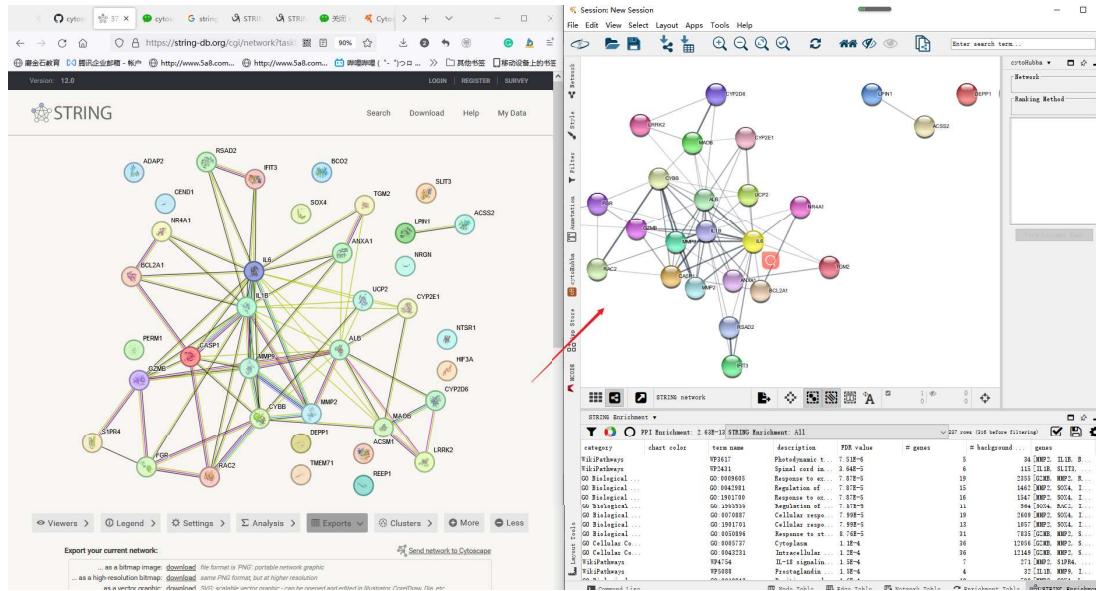
我们常用的主要就是 STRING protein query 这个选项，用于探究蛋白的互作网络



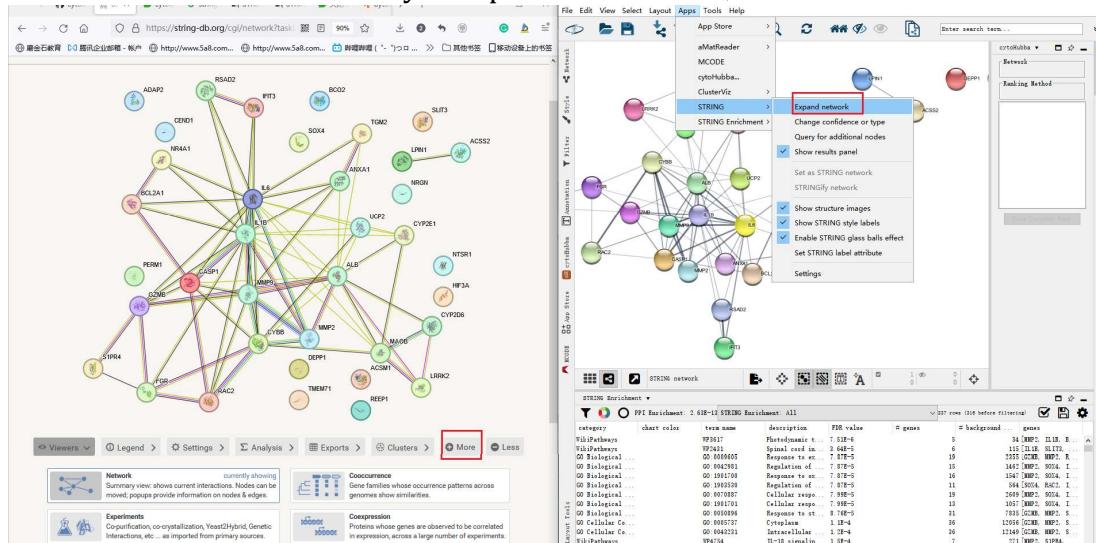
输入 38 个基因，选择好种群之后，勾选 load Enrichment Data，点击查询



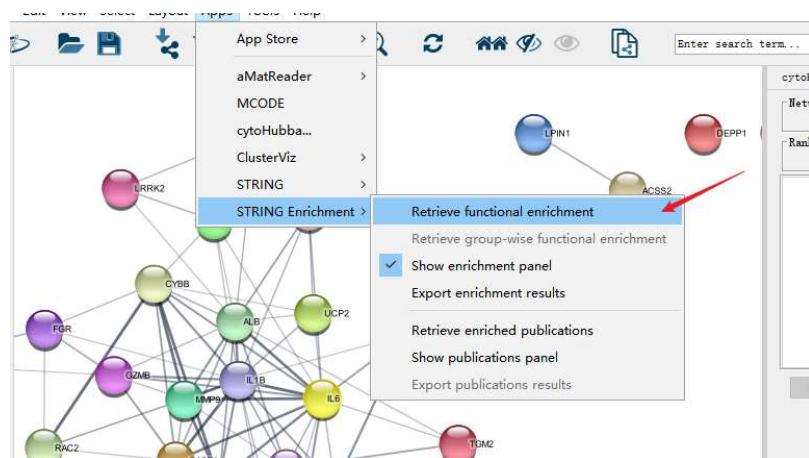
可以发现，两者是基本一致的



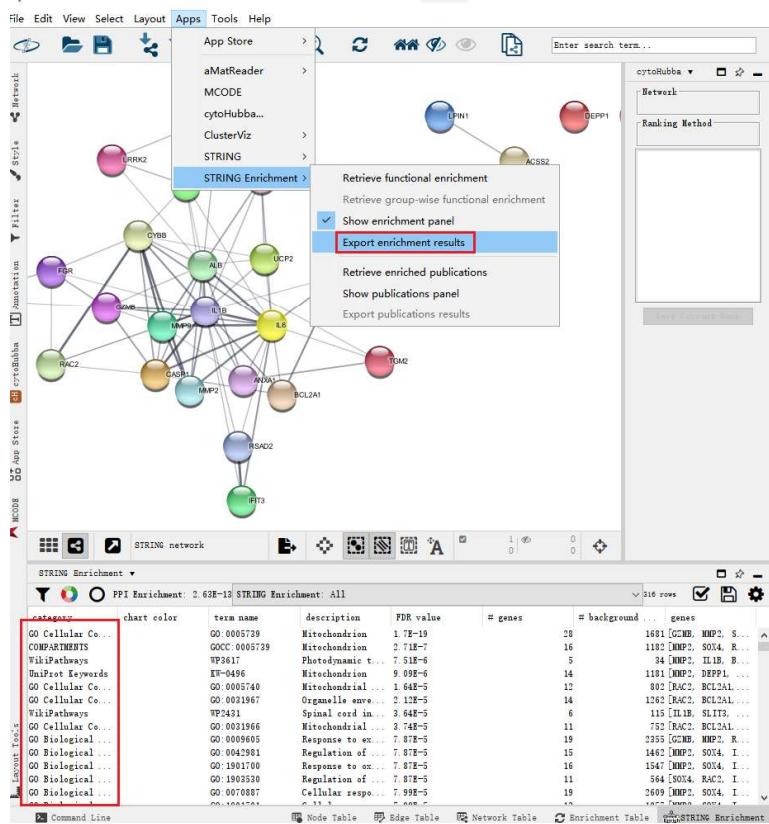
网页中的功能也可以直接再 cytoscape 上实现，包括扩展网络等



除了勾选我们的 load Enrichment Data 进行富集分析，我们还可以点击 Apps—STRING Enrichment—Retrieve function enrichment 进行富集分析

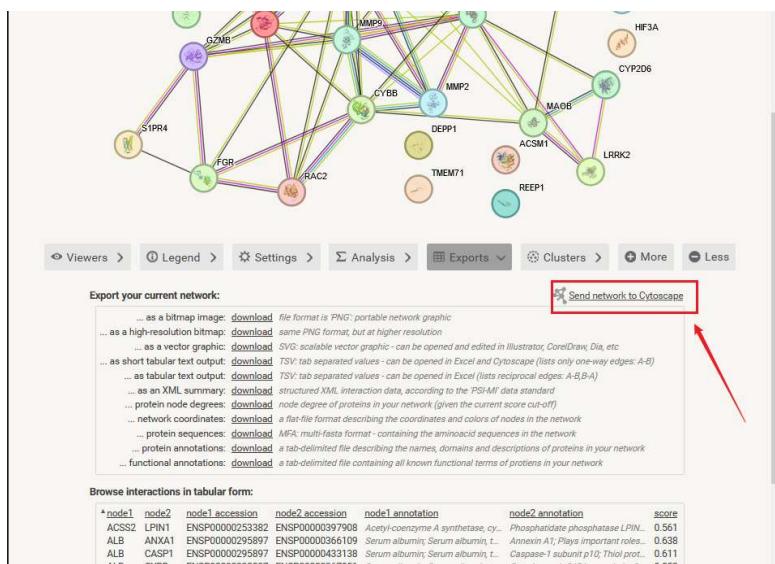


- 结果出现在网络图的下方。包含富集名称，FDR 值，基因数目及名称
- 富集的类别有六种：GO Biological Process, GO Molecular Function, GO Cellular Component, KEGG Pathways, PFAM, and InterPro protein domains.
- 点击 Apps—STRING Enrichment—Export enrichment results 进行导出保存



string 网站和 cytoscape 的互动

我们现在还可以直接点击 Send network to Cytoscape 来实现 String 网站和 Cytoscape 的互动（前提是安装了 StringAPP）



但是如今 stringApp 在文献中仍然使用不多，大家仍倾向于使用 STRING 网站构建蛋白质相互作用网络，但其实两者求出来的网络基本相同，并且如果 STRING 网站出来的网络外观不满意的话，仍需要调入 cytoscape 进行分析，这样对比使用 stringApp 就可以一步到位了。

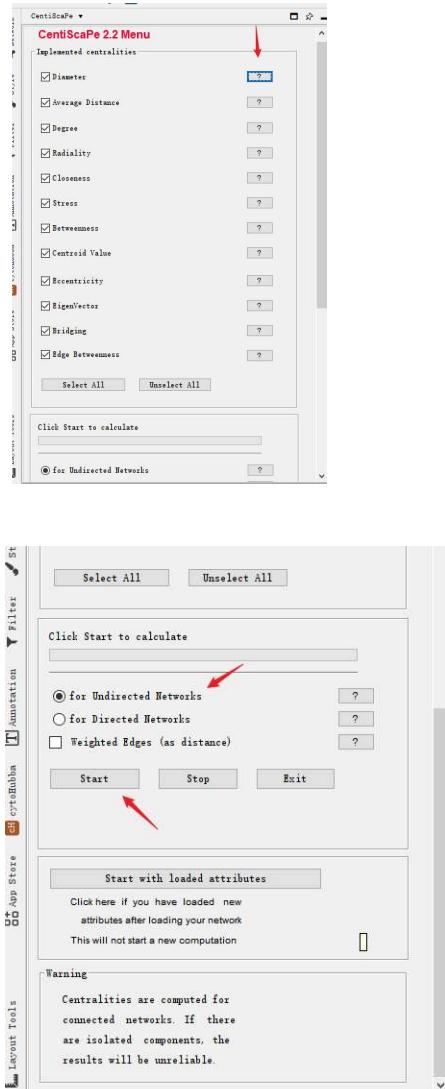
3.4 Centiscape (计算多个中心值)

Centiscape 是目前唯一能同时计算多个中心值的 Cytoscape 插件。在 Centiscape 上，通过计算出中心值或者用从实验中得到的生物参数排序，能够从网络上得到关键节点。

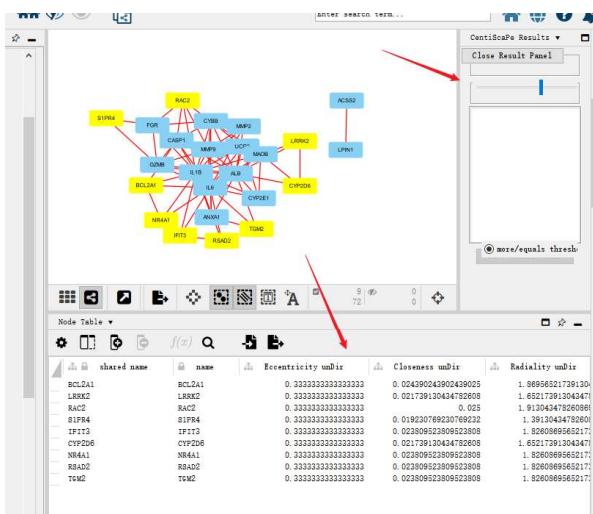
3.4.1 安装

3.4.2 实操

可以点击右边的问号来查看参数的解释



- 具体数值结果在网络下方呈现
- 网络右边还有具体调控参数



Centiscape, CytoHubba, MCODE 都可以筛选出关键基因或模块，但是使用了不同算法进行，大家可以根据筛选出来的基因数量使用其中一种或多种方法进行选择

3.5 iRegulon (转录调控)

- 基因调控网络通过调节基因的表达量和时间-空间分布特征影响生物发育，维持内稳态和疾病发生发展。因此，明确基因调控网络的拓扑学原理有助于对机制深入探讨。**基因调控网络由转录因子与其直接靶基因之间的相互作用组成。**每一种调控相互作用都代表着转录因子与靶基因附近特定 DNA 结合位点。
- 在这里，我们提出一个计算方法，称为 iRegulon，以识别目标基因的重要**调控因子**

iRegulon 插件主要使用近 10000 个 TF motifs 数据库和 1000 个 ChIP-seq 数据集或“tracks”来检测富集的 TF motifs 或 ChIPseq 峰。接下来，它将富集的 TF motifs 和“tracks”与靶点基因联系起来。iRegulon 作为一个 Cytoscape 插件，支持人类、小鼠和果蝇基因**。（可理解成 tracks 是和 motifs 差不多的数据库）

3.5.1 安装

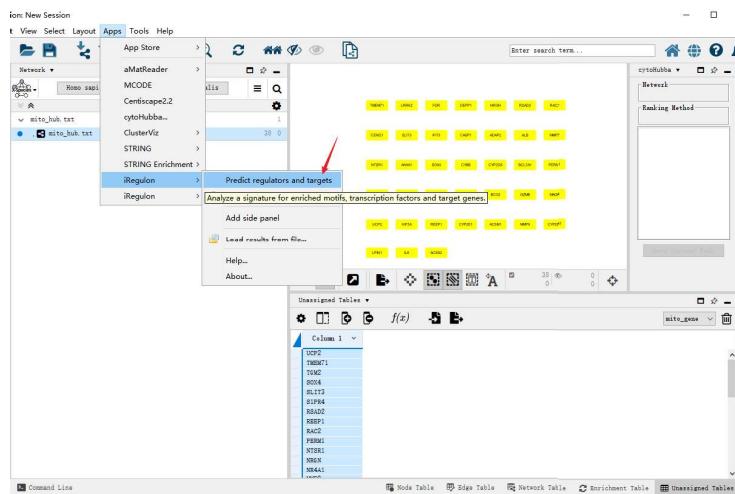
The screenshot shows the iRegulon plugin page on Bioconda. At the top, there's a logo and the text "iRegulon Detects master regulators and cis-regulatory interactions". Below that is a star rating of 4.5 stars from 12 reviews, with links to 26230 downloads, citations, and discussions. The "Details" tab is selected. Under "Categories", it lists: data integration, enrichment analysis, functional module detection, gene regulation, local data import, network generation, network inference, online data import, regulatory networks. There are four screenshots showing network visualizations. To the right, there's a section for "CYTOSCAPE 3" with an "Install" button, version information (Version 1.3), and download stats. A red arrow points to the "Install" button.

3.5.2 实操

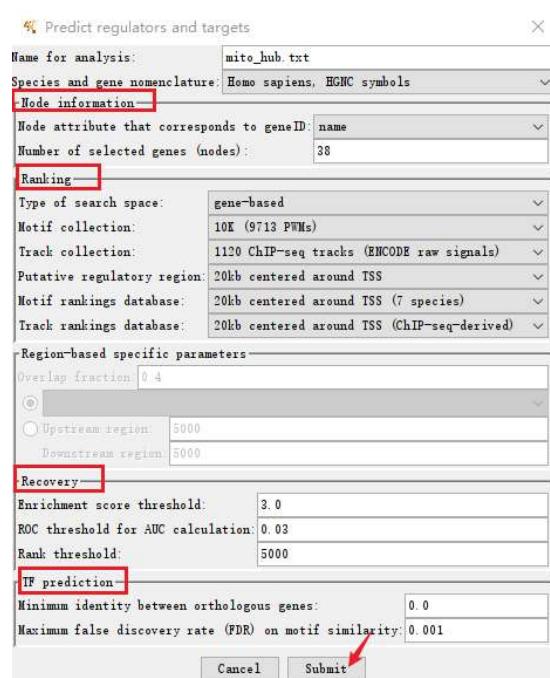
首先准备靶基因，输入之前准备的 38 个基因即可



- 先选中需要预测 TF 的靶基因（黄色为已选中）

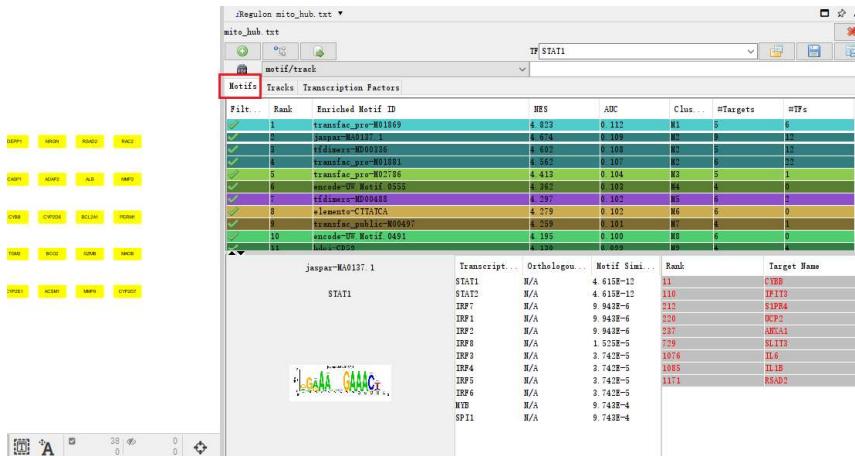


- **Node information**
 - Number of selected genes: 所预测的基因数目
- **Ranking:**
 - Type of search space: 基于基因
 - Motif collection: 10k / 6k
 - Track collection: 1120 ChIP-seq / 750
 - Putative regulatory region: 起始位点上下游端
 - Motif / Track rankings database: 排序所根据的数据库
- **Recovery:**
 - 富集分数
 - AUC 值
 - 排序阈值
- **TF prediction**
 - FDR: 发现错误率

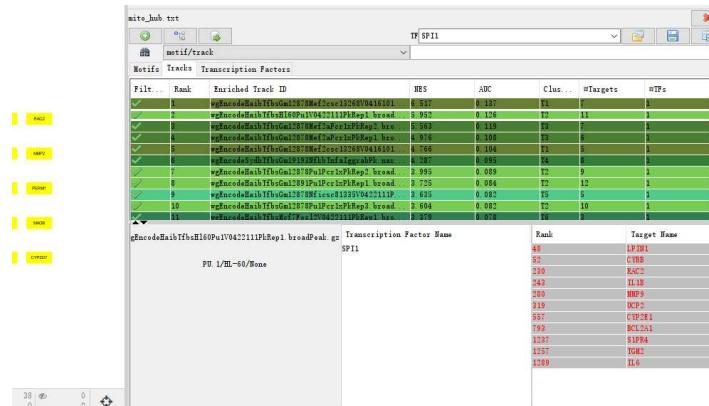
 Predict regulators and targets

| | |
|---|---|
| Name for analysis: | mito_hub.txt |
| Species and gene nomenclature: | Homo sapiens, HGNC symbols |
| Node information | |
| Node attribute that corresponds to geneID: | name |
| Number of selected genes (nodes): | 38 |
| Ranking | |
| Type of search space: | gene-based |
| Motif collection: | 10K (9713 PWNs) |
| Track collection: | 1120 ChIP-seq tracks (ENCODE raw signals) |
| Putative regulatory region: | 20kb centered around TSS |
| Motif rankings database: | 20kb centered around TSS (7 species) |
| Track rankings database: | 20kb centered around TSS (ChIP-seq-derived) |
| Region-based specific parameters | |
| Overlap fraction: | 0.4 |
| <input checked="" type="radio"/> Upstream region: | 5000 |
| <input type="radio"/> Downstream region: | 5000 |
| Recovery | |
| Enrichment score threshold: | 3.0 |
| ROC threshold for AUC calculation: | 0.03 |
| Rank threshold: | 5000 |
| TF prediction | |
| Minimum identity between orthologous genes: | 0.0 |
| Maximum false discovery rate (FDR) on motif similarity: | 0.001 |
| <input type="button" value="Cancel"/> <input style="background-color: #0070C0; color: white; font-weight: bold;" type="button" value="Submit"/> | |

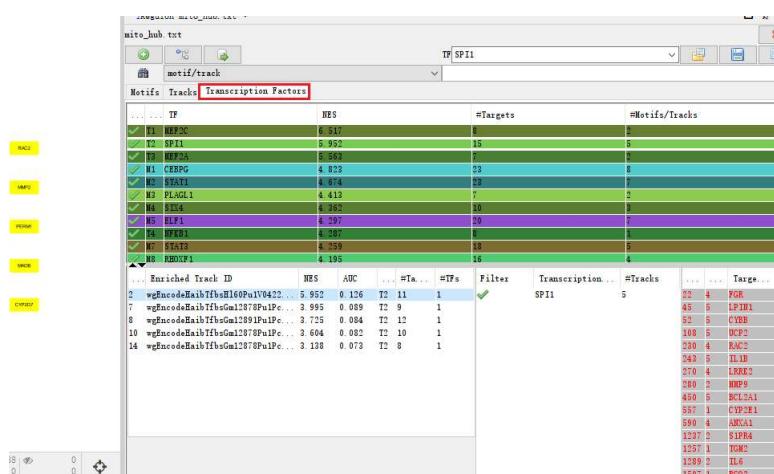
结果主要分三个部分： **Motifs, Tracks, Transcription Factors**



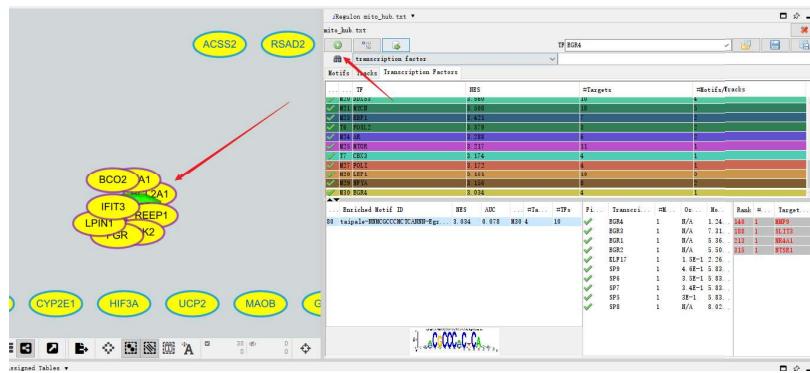
tracks: Motif 和 Tracks 应该是两个类似的求 TF 的数据库，出来的结果列名也类似，不过一般选择 Motifs 的结果



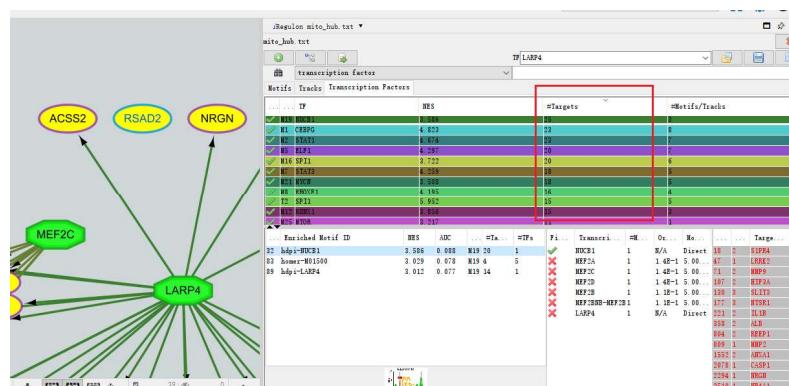
Transcription Factors



画网络图时，如果对对应靶基因最多的 TF 感兴趣，直接选中，点击上方 +，就可出现对应网络



一般文献默认参数，结果挑选对应靶基因最多的 TF 进一步研究



4. 补充

4.1 Cytoscape 可视化

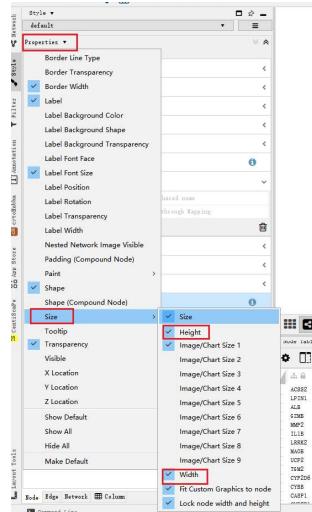
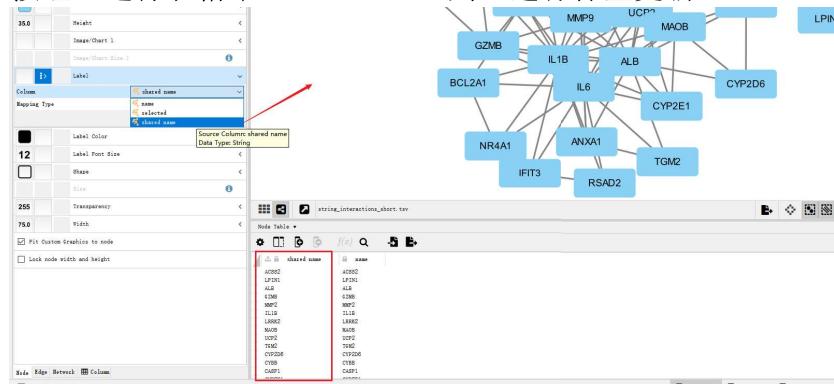
4.1.1 如何绘制蛋白互作“圈圈”网络图？

如果只是简单地将 String 得到的结果输出，对于发文章来说是不美观的，下面介绍如何绘制蛋白互作“圈圈”网络图：

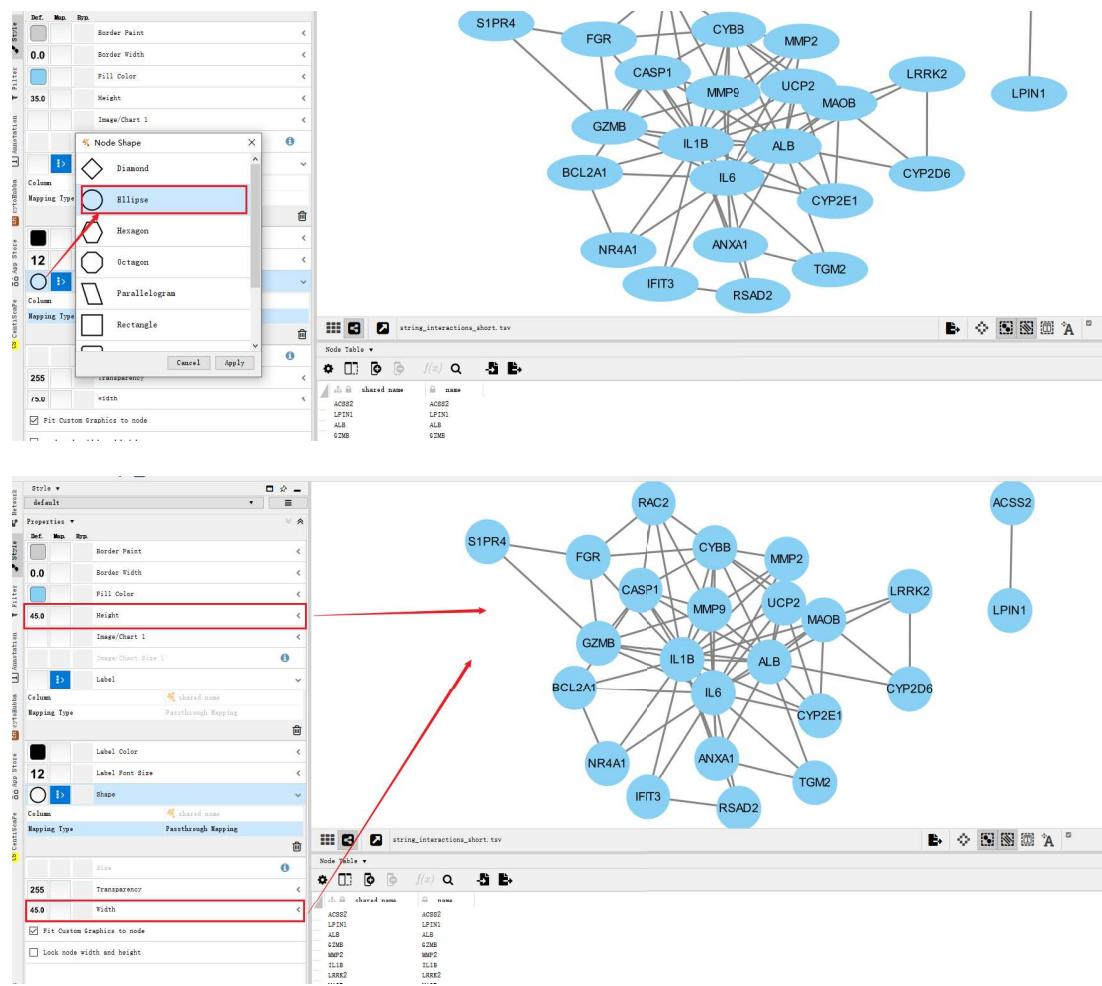
还是以之前 38 个基因得到的 PPI 网络为例：

调整网络：

- 在 Style 选项中 Node（节点）属性设置窗口中，点击 Label 属性中的 Map 按钮，选择表格中 shared name 列，进行标签更新

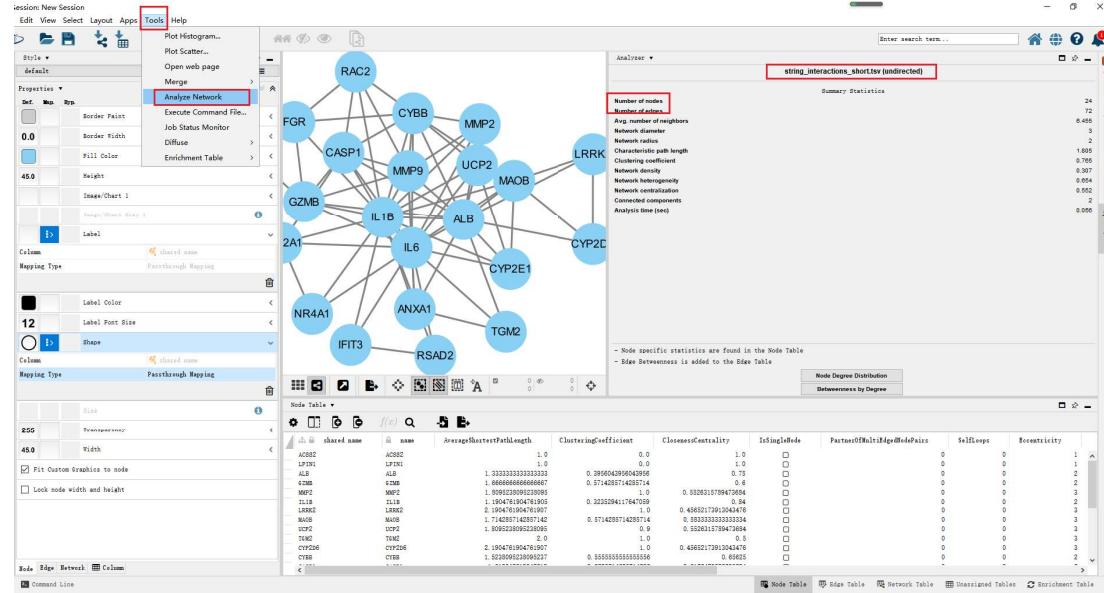


- 在 Style 选项中 Node（节点）属性设置窗口中，将节点的 Shape 改为椭圆；将椭圆的 Height、Width 数值都设置为 45，得到“正圆”节点。

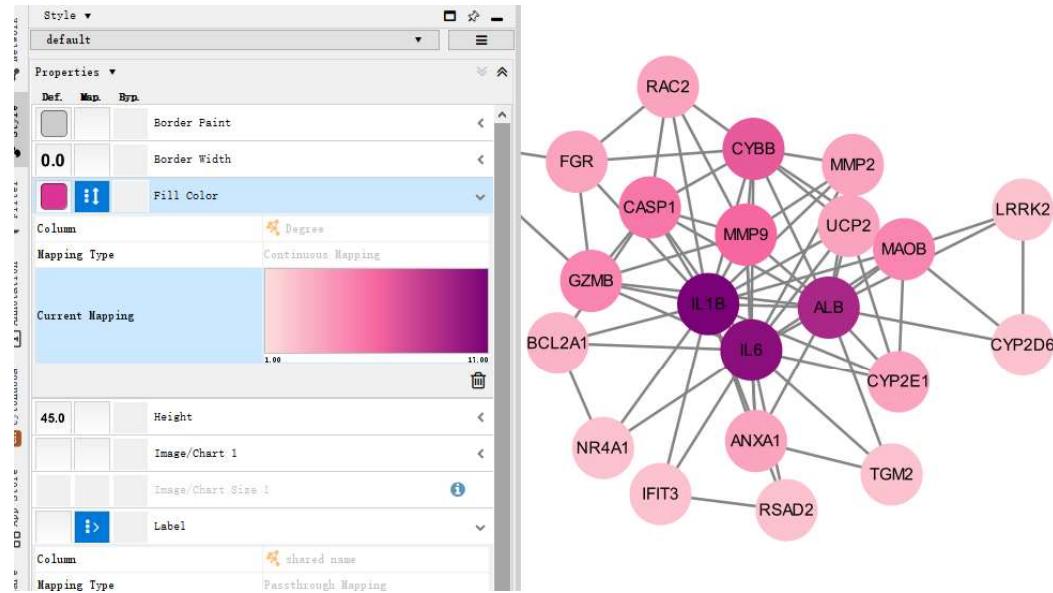


4. 通过 Tools 菜单下的 Analyze Network 命令，如下图，可分析当前网络的拓扑性质，比如 Betweenness Centrality、Closeness Centrality、Degree、

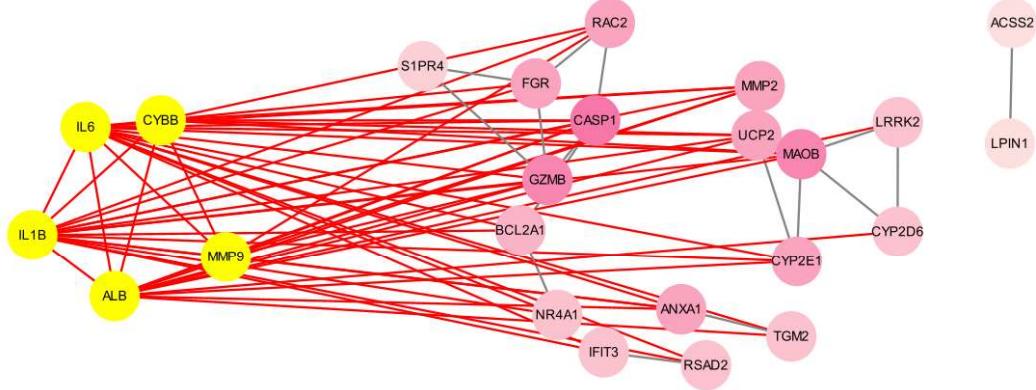
Neighborhood Connectivity、Edge Betweenness 等。



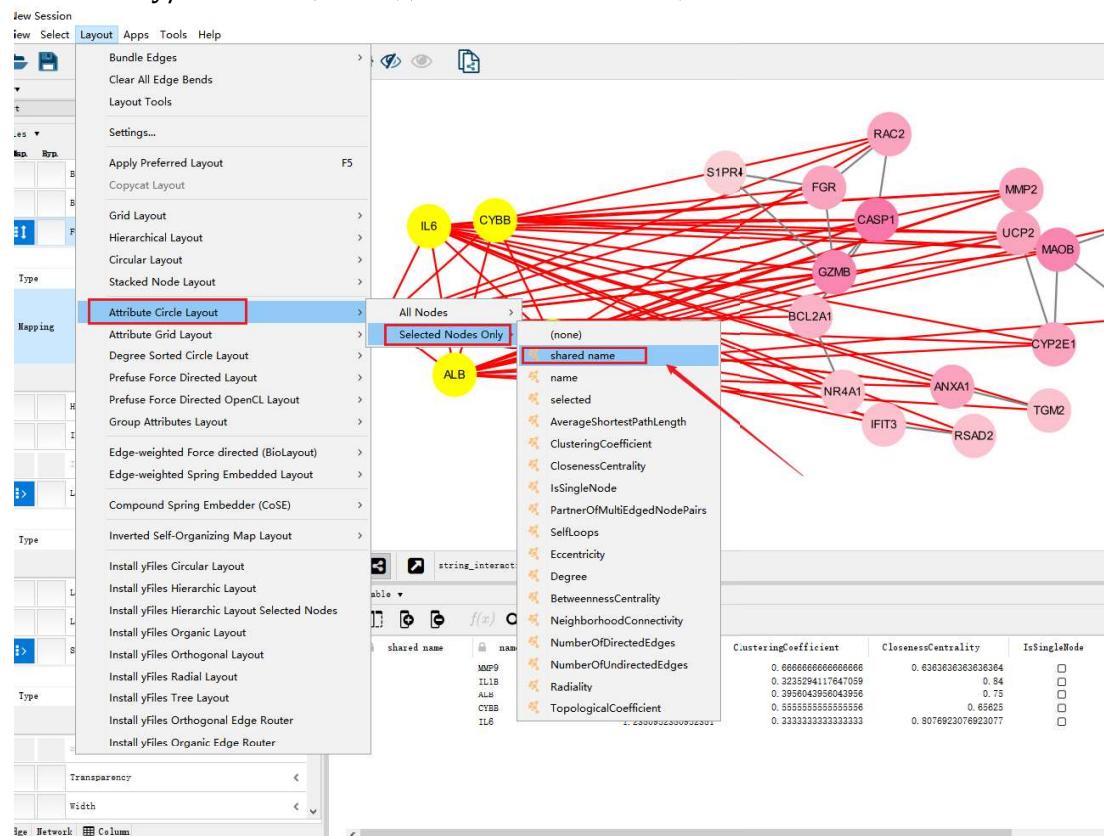
5. 点击 Fill Color 属性中的 Map 按钮，将点的颜色和计算得到网络性质（如这里的 Degree 值）建立映射关系。点击 Current Mapping 中的渐变色条，可以自定义渐变色



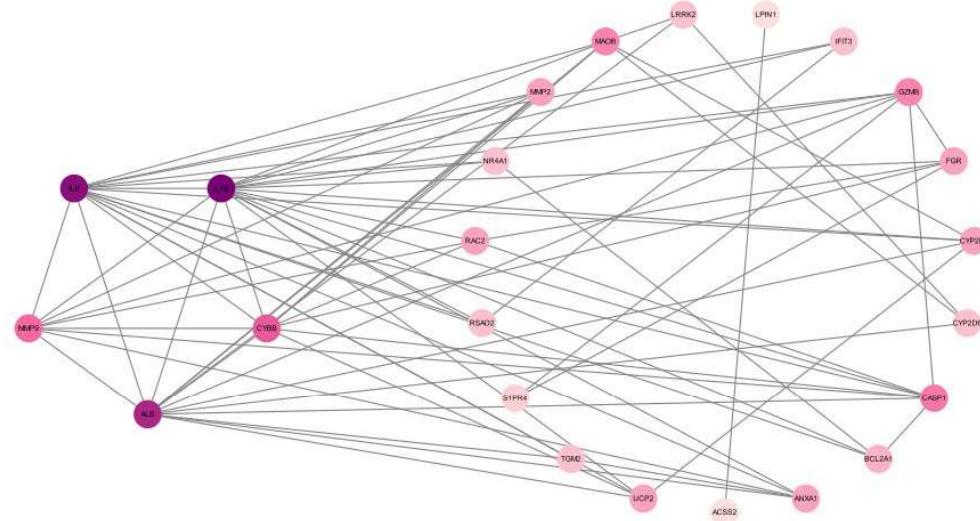
6. 按中 Ctrl 按钮，逐一选中 Degree 值较高的节点，可用鼠标将这些节点拖动到一边



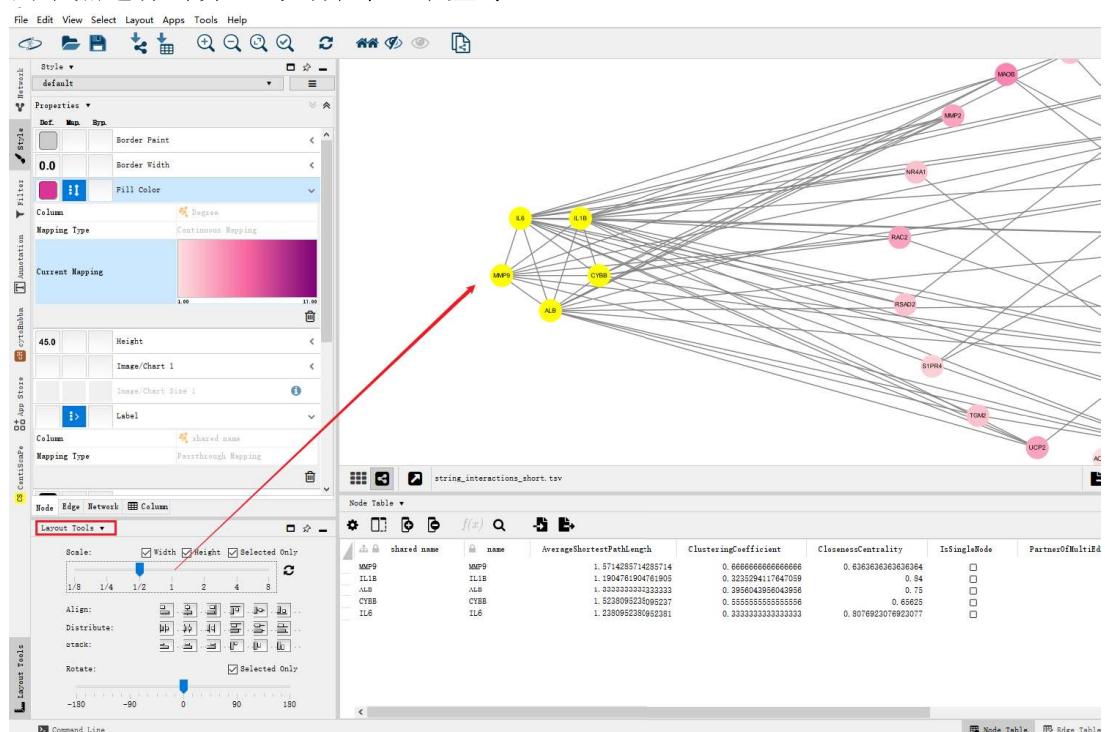
7. 选中目标节点，通过 Layout 菜单下的 Attribute Circle Layout/Selected Nodes Only/Label 命令，可将选中的节点排成环状网络，如下图



8. 按住 Ctrl 按钮框选剩余节点，同样的方法，可以将其余节点调整成环形。

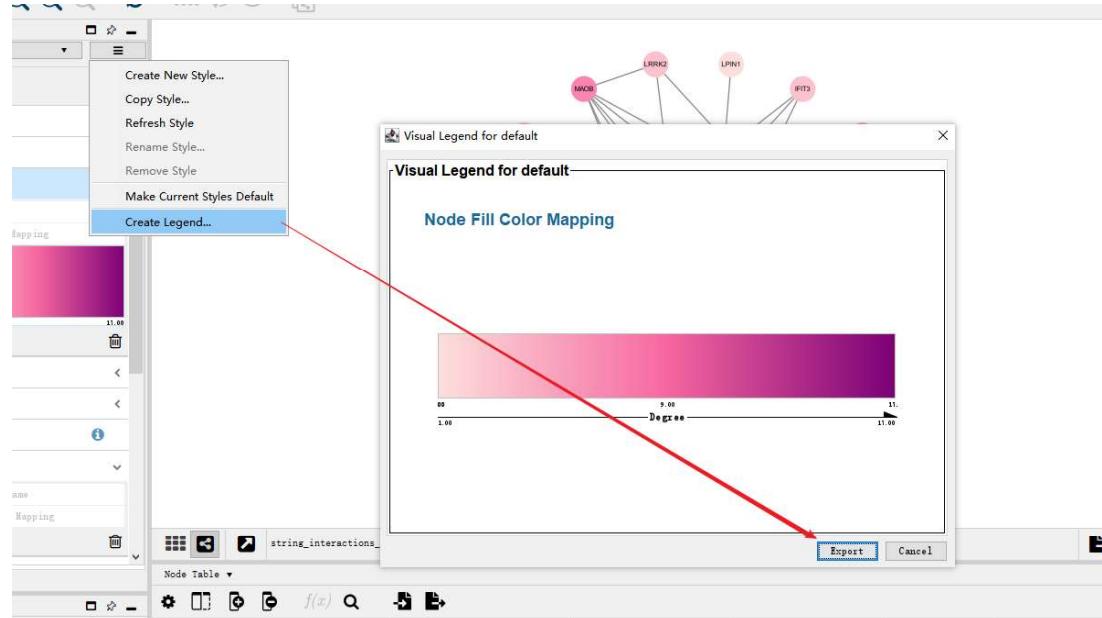


9. 同时，直接通过 Cytoscape 软件 layout-Layout Tools，可修改选中节点的布局（Layout），比如对网络图进行缩放（节点大小不变）、旋转以及对选中的节点进行对齐、均匀分布、堆叠等。

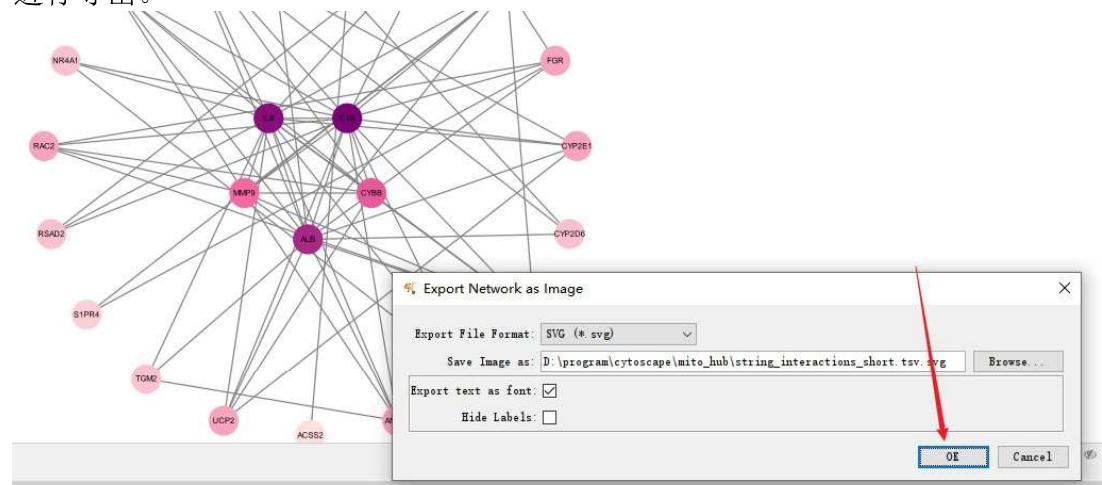


10. 按中 Ctrl 按钮框选“小环”中的节点，可将“小环”中的节点拖到“大环”中，点击 default 选项右侧的“三道杠”按钮，点击 Create Legend 选项可直接创建图例。当然，图例是以单独图片进行保存的，之后可用 Ai(Adobe illustrator) 将图

例拼到网络图上。



11. 调整满意后，通过 File/Export/Network to Image 命令，可将当前的网络以图片的形式（除了 JPG、PNG 格式的位图，也支持 PDF、SVG 格式的矢量图）进行导出。



4.2 基于 python 的网络图绘制

4.2.1 基于 omicverse

我们主要基于 omicverse 体验 Python 版 WGCNA 分析和蛋白质相互作用 PPI 分析教程，代码已上传 [omicverse_rna #####](#) 导入环境

```
import omicverse as ov
import pandas as pd
ov.utils.ov_plot_set()
```

加权基因共表达网络分析(WGCNA)

(WGCNA)是一种系统生物学方法，用于表征不同样品之间的基因关联模式，可用于鉴定高度协同的基因集，并基于基因集的内生性和基因集与表型之间的关联来鉴定候选生物标志物基因或治疗靶点。目前引用量已超过15,000。但Python中完成WGCNA分析相关的包仍是空白。

```
import pandas as pd
# 如果出现加载报错，请直接手动下载文件
data=ov.utils.read_csv(filepath_or_buffer='https://github.com/pigudog/cytoscape/blob/main/omicver_rna/LiverFemale3600.csv',
                      index_col=0)
data.head()
```

| substanceBXH | gene_symbol | LocusLinkID | ProteomeID | cytogeneticLoc | CHROMOSOME | StartPosition | EndPosition | F2_2 | F2_3 | F2_14 | ... | F2_324 | F2_325 |
|--------------|---------------|-------------|------------|----------------|------------|---------------|-------------|----------|---------|-----------|-----|-----------|---------|
| MMT00000044 | I700007N18Rik | 69339 | 286025 | 0 | 16 | 50911260 | 50912491 | -0.01810 | 0.0642 | 0.000064 | ... | 0.047700 | -0.0488 |
| MMT00000046 | Mast2 | 17776 | 157466 | 0 | 4 | 115215318 | 115372404 | -0.07730 | -0.0297 | 0.112000 | ... | -0.049200 | -0.0350 |
| MMT00000051 | Ankrd32 | 105377 | 321939 | 0 | 13 | 74940309 | 74982847 | -0.02260 | 0.0617 | -0.129000 | ... | 0.000612 | 0.1210 |
| MMT00000076 | 0 | 383154 | 0 | 0 | 16 | 49345114 | 49477048 | -0.00924 | -0.1450 | 0.028700 | ... | -0.270000 | 0.0803 |
| MMT00000080 | Ldb2 | 16826 | 157383 | 0 | 5 | 43546124 | 43613704 | -0.04870 | 0.0582 | -0.048300 | ... | 0.113000 | -0.0859 |

```
type(data)
```

```
col_name = "gene_symbol"
# 将基因列去重并作为行名
data = data.drop_duplicates(subset='gene_symbol').set_index('gene_symbol')
data.head()
```

| gene_symbol | LocusLinkID | ProteomeID | cytogeneticLoc | CHROMOSOME | StartPosition | EndPosition | F2_2 | F2_3 | F2_14 | F2_15 | ... | F2_324 | F2_325 |
|---------------|-------------|------------|----------------|------------|---------------|-------------|----------|---------|-----------|---------|-----|-----------|--------|
| I700007N18Rik | 69339 | 286025 | 0 | 16 | 50911260 | 50912491 | -0.01810 | 0.0642 | 0.000064 | -0.0580 | ... | 0.047700 | |
| Mast2 | 17776 | 157466 | 0 | 4 | 115215318 | 115372404 | -0.07730 | -0.0297 | 0.112000 | -0.0589 | ... | -0.049200 | |
| Ankrd32 | 105377 | 321939 | 0 | 13 | 74940309 | 74982847 | -0.02260 | 0.0617 | -0.129000 | 0.0871 | ... | 0.000612 | |
| 0 | 383154 | 0 | 0 | 16 | 49345114 | 49477048 | -0.00924 | -0.1450 | 0.028700 | -0.0439 | ... | -0.270000 | |
| Ldb2 | 16826 | 157383 | 0 | 5 | 43546124 | 43613704 | -0.04870 | 0.0582 | -0.048300 | -0.0371 | ... | 0.113000 | |

```
# 去掉多余的index相关的列
data = data.iloc[:,6:]
data.head()
```

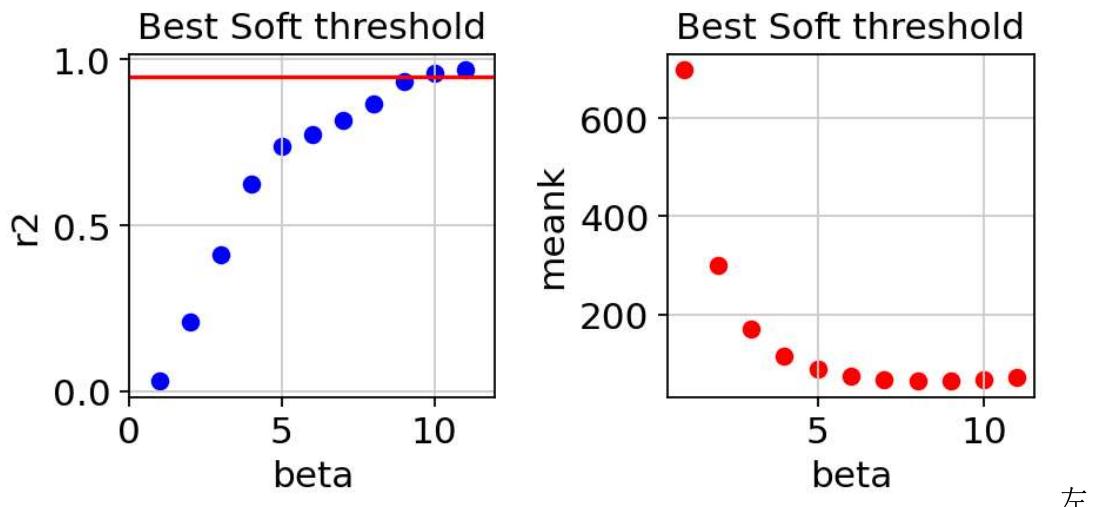
| gene_symbol | F2_2 | F2_3 | F2_14 | F2_15 | F2_19 | F2_20 | F2_23 | F2_24 | F2_26 | F2_37 | ... | F2_324 | F2_325 | F2_326 | F2_327 | F2_328 | F2_329 | F2_330 | F2_332 | F2_355 | F2_357 |
|---------------|----------|---------|-----------|---------|----------|-----------|----------|----------|---------|----------|-----|-----------|---------|---------|---------|----------|--------|---------|---------|---------|-----------|
| I700007N18Rik | -0.01810 | 0.0642 | 0.000064 | -0.0580 | 0.04830 | -0.151974 | -0.00129 | -0.23600 | -0.0307 | -0.02610 | ... | 0.047700 | -0.0488 | 0.0168 | -0.0309 | 0.02740 | -0.031 | 0.0660 | -0.0199 | -0.0146 | 0.065000 |
| Mast2 | -0.07730 | -0.0297 | 0.112000 | -0.0589 | 0.04430 | -0.093800 | 0.09340 | 0.02690 | -0.1330 | 0.07570 | ... | -0.049200 | -0.0350 | -0.0738 | -0.1730 | -0.07380 | -0.201 | -0.0820 | -0.0939 | 0.0192 | -0.049900 |
| Ankrd32 | -0.02260 | 0.0617 | -0.129000 | 0.0871 | -0.11500 | -0.065026 | 0.00249 | -0.10200 | 0.1420 | -0.10200 | ... | 0.000612 | 0.1210 | 0.0996 | 0.1090 | 0.02730 | 0.120 | -0.0629 | -0.0395 | 0.1090 | 0.000253 |
| 0 | -0.00924 | -0.1450 | 0.028700 | -0.0439 | 0.00425 | -0.236100 | -0.06900 | 0.01440 | 0.0363 | -0.01820 | ... | -0.270000 | 0.0803 | 0.0424 | 0.1610 | 0.05120 | 0.241 | 0.3890 | 0.0251 | -0.0348 | 0.114000 |
| Ldb2 | -0.04870 | 0.0582 | -0.048300 | -0.0371 | 0.02510 | 0.085043 | 0.04450 | 0.00167 | -0.0680 | 0.00567 | ... | 0.113000 | -0.0859 | -0.1340 | 0.0639 | 0.00731 | 0.124 | -0.0212 | 0.0870 | 0.0512 | 0.024300 |

```

# WGCNA 的第一步 analysis_meta_correlation 是计算基因间的相关性矩阵，这里我们采用皮尔森系数的计算方法，来完成基因间的直接相关性矩阵计算。
gene_wgcna=ov.bulk.pyWGCNA(data,save_path='result')
# 会输出一个基因与基因的相关性矩阵
gene_wgcna.calculate_correlation_direct(method='pearson',save=True)
...correlation coefficient matrix is being calculated
...direction correlation have been saved

# 在 pyWGCNA 模块中，我们需要将直接相关矩阵转换为间接相关矩阵来计算软阈值，软阈值可以帮助我们将原来的相关网络转换为无尺度网络
gene_wgcna.calculate_correlation_indirect(save=False)
gene_wgcna.calculate_soft_threshold(save=False)

```



左边的垂直坐标是无尺度网络的评估指标 r^2 。 r^2 越接近 1，网络就越接近无尺度网络，通常需要 $r^2 > 0.8$ 或 0.9。右侧垂直坐标为平均连通度，随 β 值的增加而减小。将这两个图结合起来，通常选择 r^2 首次达到 0.8 或 0.9 或更高时的 β 值。利用 β 值，我们可以根据方程将相关矩阵转换成邻接矩阵。

```

# 然后我们构造拓扑重叠矩阵
gene_wgcna.calculate_corr_matrix()

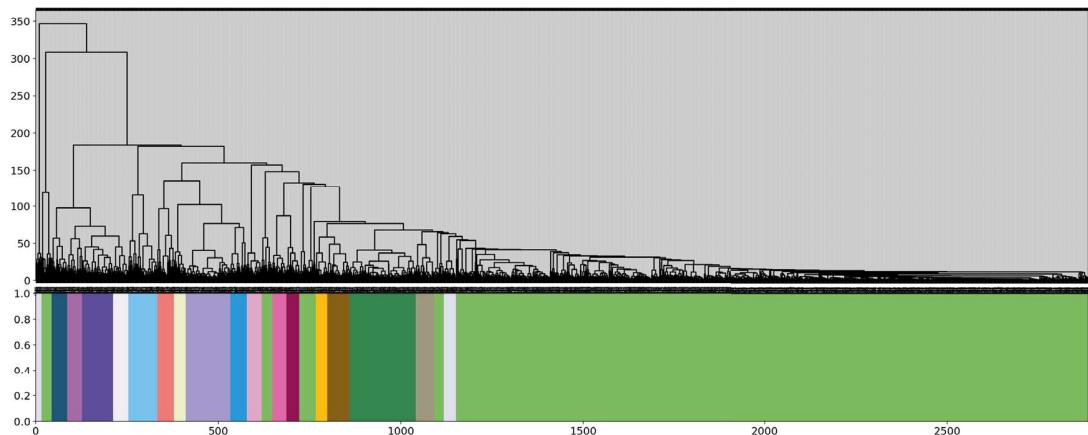
# 在获得基因间的拓扑重叠矩阵后，我们使用动态剪切树的方式来寻找基因间的模块。在这里，我们使用 WGCNA 作者发表的 DynamicTree 的算法来实现此功能。
gene_wgcna.calculate_distance()
gene_wgcna.calculate_geneTree()
gene_wgcna.calculate_dynamicMods()
module=gene_wgcna.calculate_gene_module()

```

```

...distance have being calculated
...geneTree have being calculated
...dynamicMods have being calculated
..cutHeight not given, setting it to 343.31256903199113 ==> 99% of t
he (truncated) height range in dendro.
..done.
...total: 18

```



png

```

# 在这里，我们成功地计算了每个基因的模块，共有18个模块，它们的颜色都显示在图
中。我们使用.head()查看每个基因所属的模块。
module.head()

```

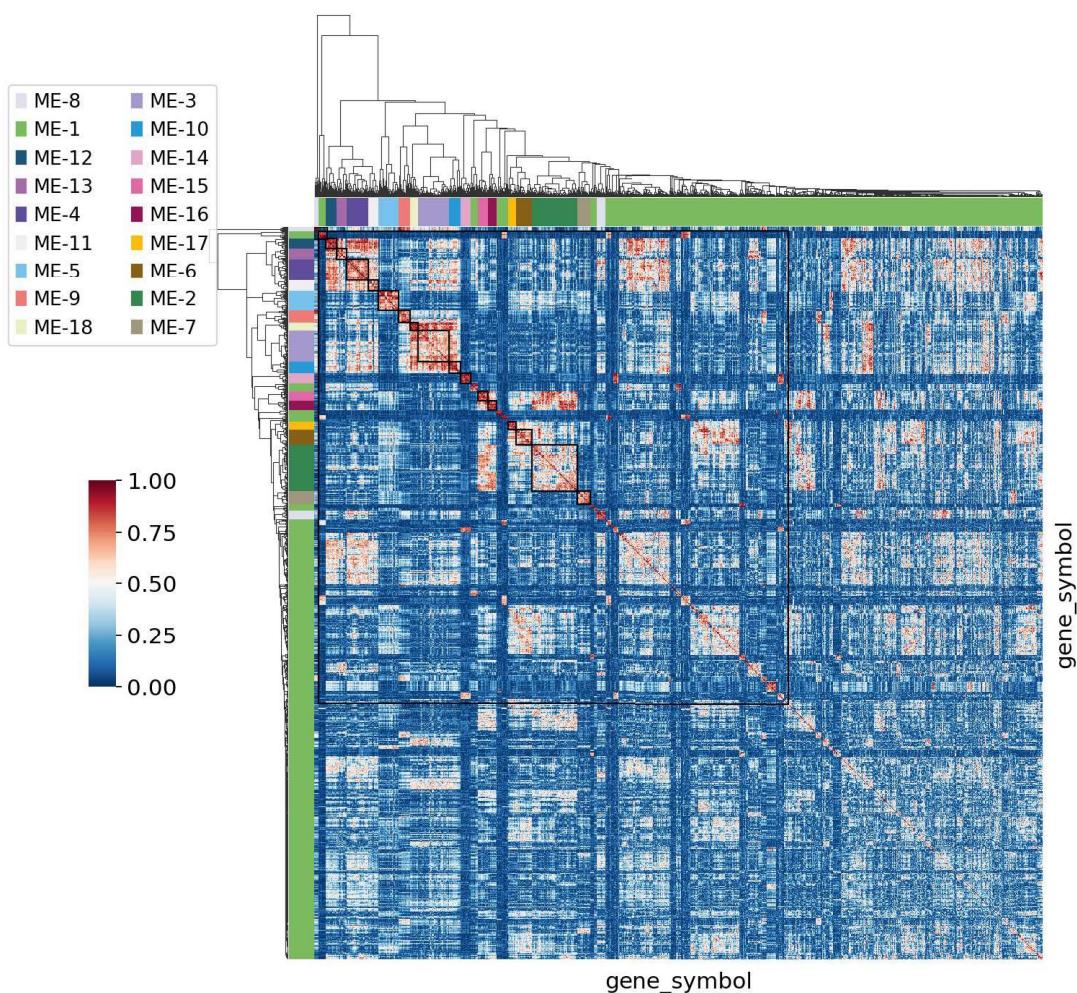
| | ivl | module | name | color |
|---|------|--------|-------|---------|
| 0 | 585 | 8 | Il1rn | #E0DFED |
| 1 | 1741 | 8 | Moap1 | #E0DFED |
| 2 | 2389 | 8 | Bicd2 | #E0DFED |
| 3 | 2061 | 8 | Ilf2 | #E0DFED |
| 4 | 538 | 8 | Bing4 | #E0DFED |

```

# 我们还可以使用.plot_matrix()来可视化拓扑重叠矩阵与模块之间的关系。
gene_wgcnna.plot_matrix()

```

```
<seaborn.matrix.ClusterGrid at 0x7fdef18ff250>
```



png

```
# 有时候我们对一个基因或一个通路的模块感兴趣，我们需要提取基因的子模块进行分析
# 和定位。例如，我们选择了两个模块 3, 18 作为分析的子模块
gene_wgcna.get_sub_module([3,18]).shape
(154, 4)

# 我们发现模块 3, 18 共有 154 个基因。接下来，我们使用之前构建的无尺度网络，将阈值设置为 0.95，为模块 3 构建一个基因相关网络图
sub_G= gene_wgcna.get_sub_network([3,18],correlation_threshold=0.95)
sub_G

<networkx.classes.graph.Graph at 0x7fdef1a4d790>

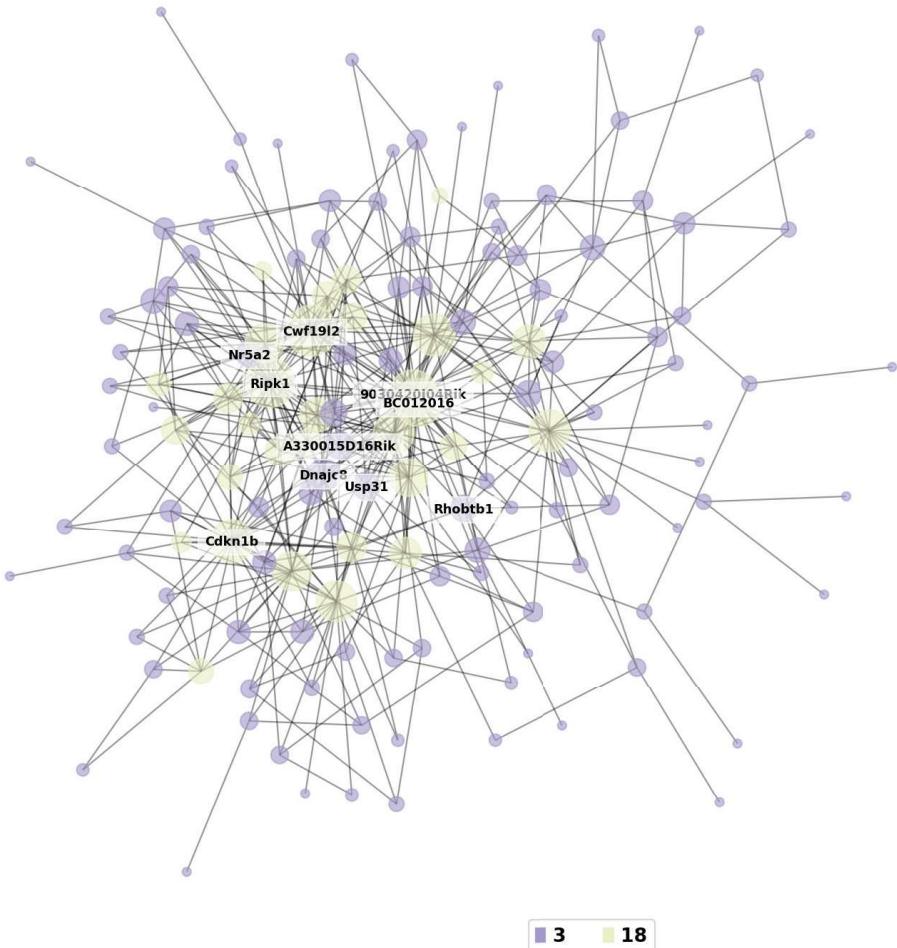
import logging
logging.getLogger('matplotlib.font_manager').setLevel(level=logging.CRITICAL)
# pyWGCNA 提供了一个简单的可视化函数 plot_sub_network 来可视化我们感兴趣的
# 基因相关性网络。
```

```

gene_wgcna.plot_sub_network([3,18],pos_type='kamada_kawai',pos_scale=5,
pos_dim=2,
figsize=(12,12),node_size=30,label_fontsize=8,
label_bbox={"ec": "white", "fc": "white", "alpha": 0.6})

(<Figure size 960x960 with 1 Axes>, <AxesSubplot: >)

```



png

蛋白质互作网络分析

我们接下来介绍蛋白质互作网络的分析，STRING 是一个已知和预测的蛋白质-蛋白质相互作用的数据库。这种相互作用包括直接的(物理的)和间接的(功能的)关联；它们来源于计算预测，来源于生物体之间的知识转移，以及来自其他(主要的)数据库的相互作用。

在这里，我们制作了一个教程，使用 `omicverse` 来构建蛋白质-蛋白质相互作用网络。

使用 `omicverse` 完成蛋白质相互作用网络分析需要三个数据：蛋白列表,蛋白类别字典和蛋白颜色字典，颜色字典是绘图时的每个蛋白的颜色，一般与类别字典相同。在这里我们随机生成一个颜色字典和类别字典。

```
gene_list=['FAA4', 'POX1', 'FAT1', 'FAS2', 'FAS1', 'FAA1', 'OLE1', 'YJU3', 'TGL3', 'INA1', 'TGL5']
gene_type_dict=dict(zip(gene_list,['Type1']*5+['Type2']*6))
gene_color_dict=dict(zip(gene_list,['#F7828A']*5+['#9CCCA4']*6))
gene_type_dict

{'FAA4': 'Type1',
 'POX1': 'Type1',
 'FAT1': 'Type1',
 'FAS2': 'Type1',
 'FAS1': 'Type1',
 'FAA1': 'Type2',
 'OLE1': 'Type2',
 'YJU3': 'Type2',
 'TGL3': 'Type2',
 'INA1': 'Type2',
 'TGL5': 'Type2'}
```

`omicverse` 提供了一个十分简单的`APIov.bulk.string_interaction`, 只需要输入蛋白列表即可完成相互作用关系的预测。

```
G_res=ov.bulk.string_interaction(gene_list,4932)
G_res.head()
```

| | stringId_A | stringId_B | preferredName_A | preferredName_B | ncbiTaxonId | score | nscore | fscore | pscore | ascore | escore | dscore | tscore |
|---|--------------|--------------|-----------------|-----------------|-------------|-------|--------|--------|--------|--------|--------|--------|--------|
| 0 | 4932.YBR041W | 4932.YKL182W | FAT1 | FAS1 | 4932 | 0.576 | 0 | 0 | 0 | 0.121 | 0 | 0 | 0.538 |
| 1 | 4932.YBR041W | 4932.YKL182W | FAT1 | FAS1 | 4932 | 0.576 | 0 | 0 | 0 | 0.121 | 0 | 0 | 0.538 |
| 2 | 4932.YBR041W | 4932.YOR081C | FAT1 | TGL5 | 4932 | 0.601 | 0 | 0 | 0 | 0 | 0 | 0 | 0.601 |
| 3 | 4932.YBR041W | 4932.YOR081C | FAT1 | TGL5 | 4932 | 0.601 | 0 | 0 | 0 | 0 | 0 | 0 | 0.601 |
| 4 | 4932.YBR041W | 4932.YPL231W | FAT1 | FAS2 | 4932 | 0.63 | 0 | 0 | 0 | 0.113 | 0 | 0 | 0.63 |

| Field | Description |
|-----------------|---------------------------------|
| stringId_A | STRING identifier (protein A) |
| stringId_B | STRING identifier (protein B) |
| preferredName_A | common protein name (protein A) |
| preferredName_B | common protein name (protein B) |
| ncbiTaxonId | NCBI taxon identifier |
| score | combined score |
| nscore | gene neighborhood score |
| fscore | gene fusion score |
| pscore | phylogenetic profile score |
| ascore | coexpression score |
| escore | experimental score |
| dscore | database score |
| tscore | textmining score |

```

# 当然, omicverse 还有非常漂亮的可视化函数, 来可视化蛋白质相互作用网络, 在这里, 我们需要使用 pyPPI 类来完成上述分析
#4932 代表酵母, 人类一般是 9606, 小鼠一般是 10090
ppi=ov.bulk.pyPPI(gene=gene_list,
                    gene_type_dict=gene_type_dict,
                    gene_color_dict=gene_color_dict,
                    species=4932)

# 然后我们连接到 string-db 来计算蛋白质-蛋白质的相互作用

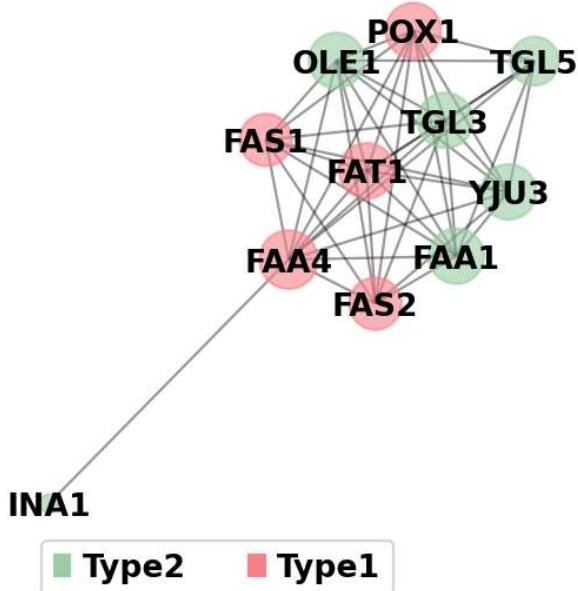
ppi.interaction_analysis()

<networkx.classes.graph.Graph at 0x7fde41ac8220>

# 我们提供了 .plot_network() 来可视化蛋白质相互作用网络, 更多的参数您可以使用 help(ov.utils.plot_network) 来寻找。
ppi.plot_network()

(<Figure size 320x320 with 1 Axes>, <AxesSubplot: >)

```



png

4.3 基于 R 的网络图绘制

STRING 数据库是一个基于公共数据库和文献信息的蛋白质相互作用网络数据库, 使用 STRING 数据库, 可以便于我们进行 PPI 网络分析。对于数据库产生的结果, 我们可以进行可视化。通常使用 cytoscape 软件, 此外, 我们也可使用 R 语言进行互作网络的可视化。我们现在使用 R 包 *igraph* 和 *ggraph* 对 PPI 网络进行可视化。

数据和代码以上传 https://github.com/pigudog/cytoscape/tree/main/igraph_r

安装并加载 R 包

```
if (!require("igraph"))
  BiocManager::install("igraph")
if (!require("ggnewscale"))
  install.packages("ggnewscale")
if (!require("tidyverse"))
  install.packages("tidyverse")
if (!require("ggraph"))
  install.packages("ggraph")
library(igraph)
library(ggraph)
library(tidyverse)
library(ggnewscale)
```

需要准备两个文件 - String 网站的输出文件 -

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 |
|----|-------|-------|--------------|--------------|----|----|-------|-------|-------|-------|-----|-------|-------|
| 1 | BUB2 | CDC15 | 4932.YMR055C | 4932.YAR019C | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.995 | 0.0 | 0.931 | 0.999 |
| 2 | BUB2 | CDC14 | 4932.YMR055C | 4932.YFR028C | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.994 | 0.0 | 0.910 | 0.999 |
| 3 | BUB2 | DBF2 | 4932.YMR055C | 4932.YGR092W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.162 | 0.0 | 0.972 | 0.976 |
| 4 | BUB2 | SPO12 | 4932.YMR055C | 4932.YHR152W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.699 | 0.999 |
| 5 | BUB2 | MO81 | 4932.YMR055C | 4932.YIL106W | 0 | 0 | 0.000 | 0.000 | 0.046 | 0.336 | 0.0 | 0.870 | 0.911 |
| 6 | BUB2 | NET1 | 4932.YMR055C | 4932.YIL076W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.799 | 0.799 |
| 7 | BUB2 | SIC1 | 4932.YMR055C | 4932.YLR079W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.164 | 0.0 | 0.605 | 0.655 |
| 8 | BUB2 | TEM1 | 4932.YMR055C | 4932.YML064C | 0 | 0 | 0.261 | 0.000 | 0.051 | 0.995 | 0.9 | 0.965 | 0.999 |
| 9 | BUB2 | SLK19 | 4932.YMR055C | 4932.YOR195W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.632 | 0.632 |
| 10 | BUB2 | CLB2 | 4932.YMR055C | 4932.YPR119W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.225 | 0.0 | 0.774 | 0.817 |
| 11 | CDC14 | CDC15 | 4932.YFR028C | 4932.YAR019C | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.994 | 0.0 | 0.987 | 0.999 |
| 12 | CDC14 | MO81 | 4932.YFR028C | 4932.YIL106W | 0 | 0 | 0.000 | 0.000 | 0.097 | 0.477 | 0.9 | 0.920 | 0.995 |
| 13 | CDC14 | DBF2 | 4932.YFR028C | 4932.YGR092W | 0 | 0 | 0.000 | 0.000 | 0.111 | 0.859 | 0.9 | 0.950 | 0.999 |
| 14 | CDC14 | NET1 | 4932.YFR028C | 4932.YIL076W | 0 | 0 | 0.000 | 0.000 | 0.055 | 0.925 | 0.9 | 0.999 | 0.999 |
| 15 | CDC14 | SPO12 | 4932.YFR028C | 4932.YHR152W | 0 | 0 | 0.000 | 0.000 | 0.052 | 0.994 | 0.0 | 0.905 | 0.999 |
| 16 | CDC14 | SLK19 | 4932.YFR028C | 4932.YOR195W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.994 | 0.0 | 0.900 | 0.999 |
| 17 | CDC14 | SIC1 | 4932.YFR028C | 4932.YLR079W | 0 | 0 | 0.000 | 0.000 | 0.000 | 0.999 | 0.9 | 0.903 | 0.999 |
| 18 | CDC14 | TEM1 | 4932.YFR028C | 4932.YML064C | 0 | 0 | 0.000 | 0.000 | 0.140 | 0.994 | 0.0 | 0.980 | 0.999 |
| 19 | CDC14 | CLB2 | 4932.YFR028C | 4932.YPR119W | 0 | 0 | 0.000 | 0.000 | 0.148 | 0.994 | 0.0 | 0.917 | 0.999 |

差异分析结果文件

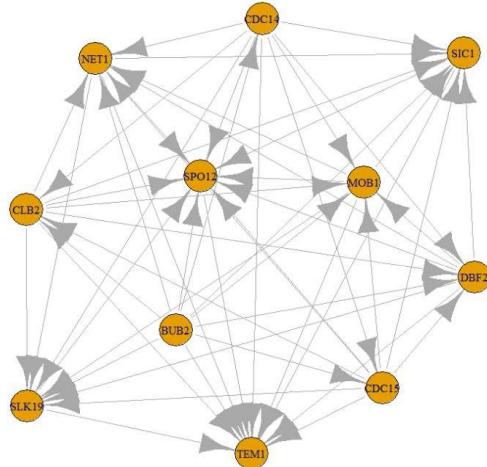
| | nodename | DEG | logFC | Sample |
|----|----------|------|-----------|--------|
| 1 | BUB2 | up | 16.465718 | A |
| 2 | CDC14 | up | 11.472131 | A |
| 3 | CDC15 | up | 11.255639 | B |
| 4 | CLB2 | up | 5.554841 | D |
| 5 | DBF2 | up | 4.592605 | E |
| 6 | MOB1 | down | -5.728319 | A |
| 7 | NET1 | down | -4.946238 | C |
| 8 | SIC1 | down | -4.615053 | E |
| 9 | SLK19 | down | -3.833520 | B |
| 10 | SPO12 | down | -4.685742 | D |
| 11 | TEM1 | down | -3.105211 | D |

```
#保留 string_interactions_short.tsv 的第 1, 2, 10 列, 分别是节点的起点、终点,
和试验验证的可信度
nodelink<-ppidif[,c(1,2,10)] %>%
  rename_with(~c("from","to","experimentally_determined_interaction"),c
(1,2,3))
```

```
#对可信度分组, 大于 0.5 为已验证, 否则为非验证
nodelink$experimentally_group <- ifelse(nodelink$experimentally_determined_interaction>0.5, "validation", "no validation")
```

| from | to | experimentally_determined_interaction | experimentally_group |
|------|-------|---------------------------------------|----------------------|
| 1 | BUB2 | CDC15 | 0.995 validation |
| 2 | BUB2 | CDC14 | 0.994 validation |
| 3 | BUB2 | DBF2 | 0.162 no validation |
| 4 | BUB2 | SPO12 | 0.000 no validation |
| 5 | BUB2 | MOB1 | 0.336 no validation |
| 6 | BUB2 | NET1 | 0.000 no validation |
| 7 | BUB2 | SIC1 | 0.164 no validation |
| 8 | BUB2 | TEM1 | 0.995 validation |
| 9 | BUB2 | SLK19 | 0.000 no validation |
| 10 | BUB2 | CLB2 | 0.225 no validation |
| 11 | CDC14 | CDC15 | 0.994 validation |
| 12 | CDC14 | MOB1 | 0.477 no validation |
| 13 | CDC14 | DBF2 | 0.859 validation |
| 14 | CDC14 | NET1 | 0.925 validation |
| 15 | CDC14 | SPO12 | 0.994 validation |

```
# 创建igraph 图形
mygraph<-graph_from_data_frame(nodelink,vertices = nodes)
plot(mygraph)
```



但是这上面这个图实在太丑了,稍微修改一下,但是还是好丑

```
ov = c('#A499CC',
      '#5E4D9A',
      '#EF7B77',
      '#A56BA7',
      '#E0A7C8',
      '#E069A6',
      '#941456',
      '#01A0A7',
      '#75C8CC',
      '#279AD7',
      '#1F577B',
      '#78C2ED',
      '#F0D7BC',
      '#FCBC10',
      '#EAEFC5',
      '#D5B26C',
      '#D5DA48',
      '#B6B812',
      '#9DC3C3',
      '#A89C92',
      '#FEE00C',
      '#FEF2A1',
      '#7CBB5F',
      '#368650',
      '#866017',
      '#9F987F',
      '#E0DFED',
      '#F0EEF0')

ggraph(mygraph, layout = "linear", circular=T) +
  geom_edge_bend(aes(edge_width=experimentally_determined_interaction, e
dge colour=experimentally group),
```

```

strength = 0.02, #strength 参数后接数值, 在 0-1 之间, 越大,
代表曲线的弯曲程度越大
alpha=0.6)+  

scale_edge_width_continuous(range = c(0.5,1.2))+ #设置连线的粗细范围  

scale_edge_color_manual(values = c("skyblue","#fe817d"))+ #设置连线的
颜色  

geom_node_point(aes(colour=Sample,size=logFC))+ #添加第一层散点, 映射样
本信息和 LogFC  

scale_size_continuous(range =c(3,15))+ #设置散点大小范围  

scale_colour_manual(values = ov[1:10])+ #设置散点颜色  

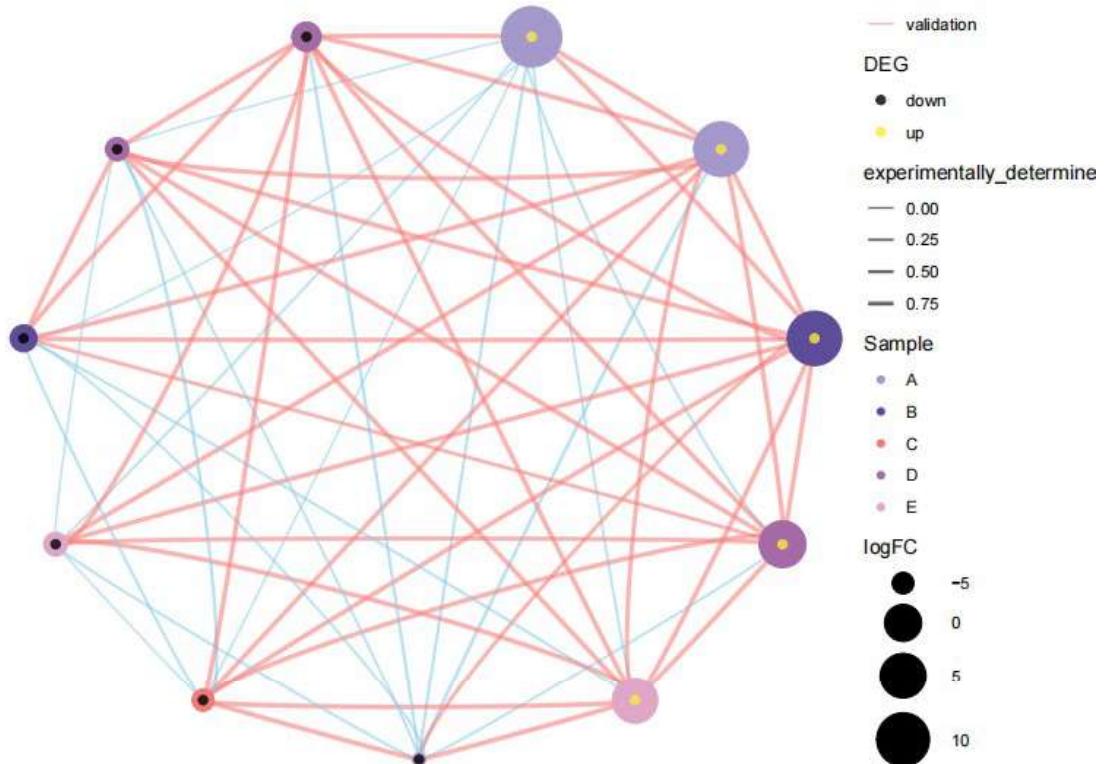
new_scale_colour()#添加新的scale  

geom_node_point(aes(colour=DEG),alpha=0.8,size=2)+ #添加第二层散点, 映
射差异表达基因信息  

scale_colour_manual(values = c("black","#FFEB3B"))+ #设置散点颜色  

theme_void()

```



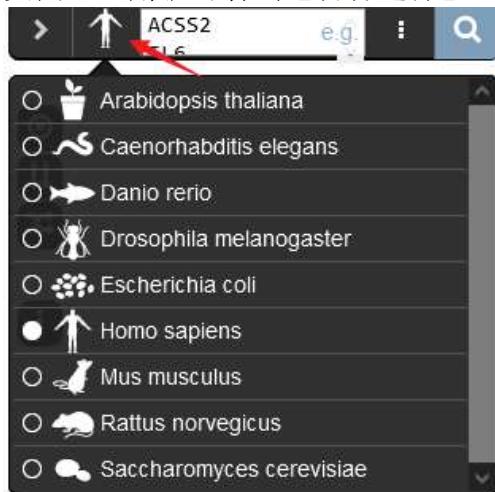
4.4 GENEMANIA 网站

- GeneMANIA 不仅可以支持网络互作，其可视化也是相当不错的，直接用 GeneMANIA 做 PPI 网络，非常美观，一般无需任何加工
- GeneMANIA (<http://genemania.org/>) 是个可生成关于基因功能的假设，分析基因列表和根据功能分析基因优先级的数据库。给出一个查询基因列表，其根据丰富的基因组学和蛋白质组学发现功能相似的基因，并根据预测值对其实现加权。

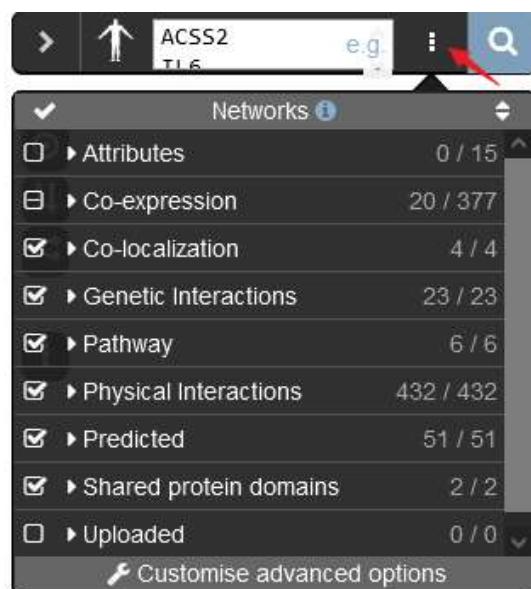
- 另一个用途是基因功能预测。只要有一个查询基因，GeneMANIA 可根据相互作用，找到共同功能基因。
- 支持物种：Arabidopsis thaliana, Caenorhabditis elegans, Danio rerio, Drosophila melanogaster, Escherichia coli, Homo sapiens, Mus musculus, Rattus norvegicus and Saccharomyces cerevisiae

4.4.1 实操

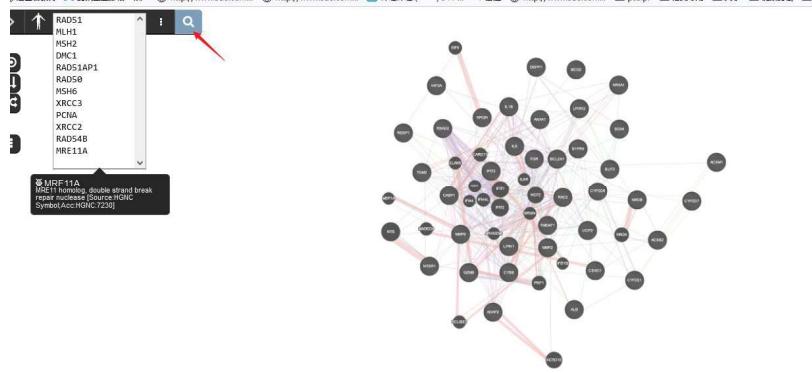
页面左上角就可看到【物种选择】，图标很直观。默认就直接是个人，代表人类



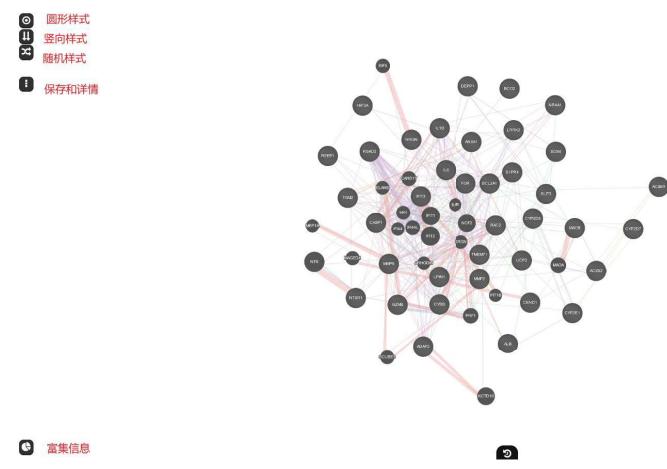
点击右侧，可选择分析的网络数据集，系统会自带默认的数据集，一般包括包括蛋白质-蛋白质、蛋白质-DNA 和遗传相互作用、途径、反应、基因和蛋白质表达数据、蛋白质结构域和表型筛选情况



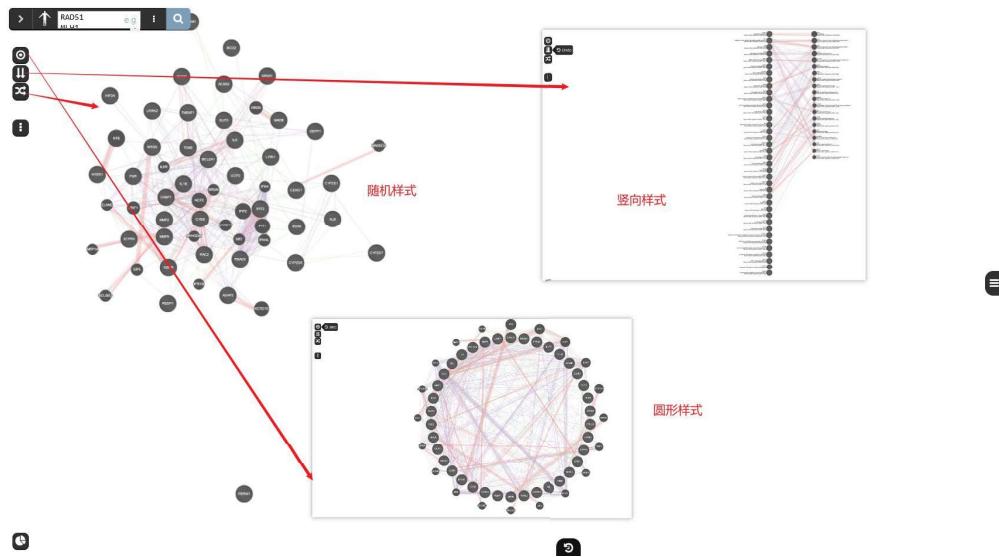
输入 38 个基因后点击 search



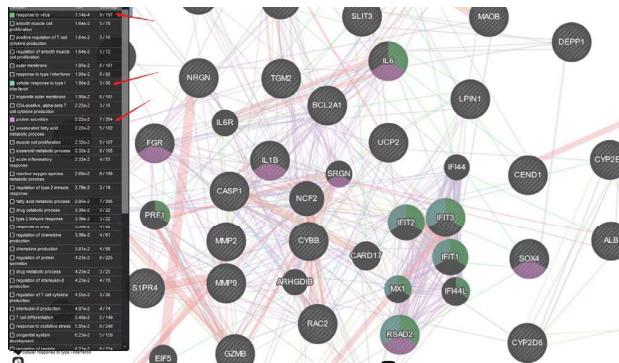
整体页面



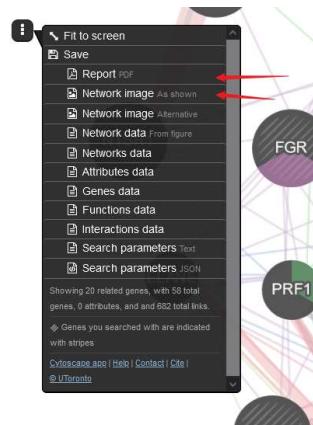
样式调整



选择你想查看的通路



导出保存，推荐前两种图片保存方式



比如 pdf 格式的 report:

GeneMANIA report

Created on : 19 November 2023 11:26:11
Last database update : 13 August 2021 00:00:00
Application version : 3.6.0

