MetaNet Tutorial

Peng Chen

2023-02-26

Contents

Chapter 1

About

1.1 Install

MetaNet is a comprehensive network analysis package, especially in various omics.

the latest version can be found in https://github.com/Asa12138/MetaNet.

for some data manipulation, we recommend to use `dplyr`

```
remotes::install_github('Asa12138/pcutils')
remotes::install_github('Asa12138/MetaNet',dependencies=T)
library(MetaNet)
library(igraph)
#data manipulation
library(dplyr)
```

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.0.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## loaded via a namespace (and not attached):
##  [1] bookdown_0.32   digest_0.6.31   lifecycle_1.0.3 magrittr_2.0.3
##  [5] evaluate_0.19   rlang_1.0.6     stringi_1.7.8   cli_3.6.0
##  [9] rstudioapi_0.14 vctrs_0.5.1     rmarkdown_2.19  tools_4.2.2
## [13] stringr_1.5.0   glue_1.6.2      xfun_0.36       yaml_2.3.6
## [17] fastmap_1.1.0   compiler_4.2.2  htmltools_0.5.4 knitr_1.41
```

Chapter 2

Introduction

2.1 Network

In mathematics,“networks” are often referred to as “graphs”,and the mathematical field of graph research is called “graph theory”.

The basic elements in a network graph are nodes and edges. When constructing a network graph,the objects are called “nodes” (vertices or nodes),and they are usually drawn as points; the connections between nodes are called “edges” (edges). or links),and they are usually drawn as lines between points.

We Can be divide into directed and undirected networks,weighted and unweighted networks according to the edges.

2.2 Network in omics

Networks can represent various systems in the real world,and have many applications in biological research,especially in systems biology: gene expression regulatory networks,metabolic networks,ecosystem space networks,microbial co-occurrence networks,protein interaction networks,etc.

WGCNA

Co-occurrence networks

PPI

2.3 Software

- R: igraph (https://igraph.org/), network
- Python: networkx (https://pypi.org/project/networkx/)

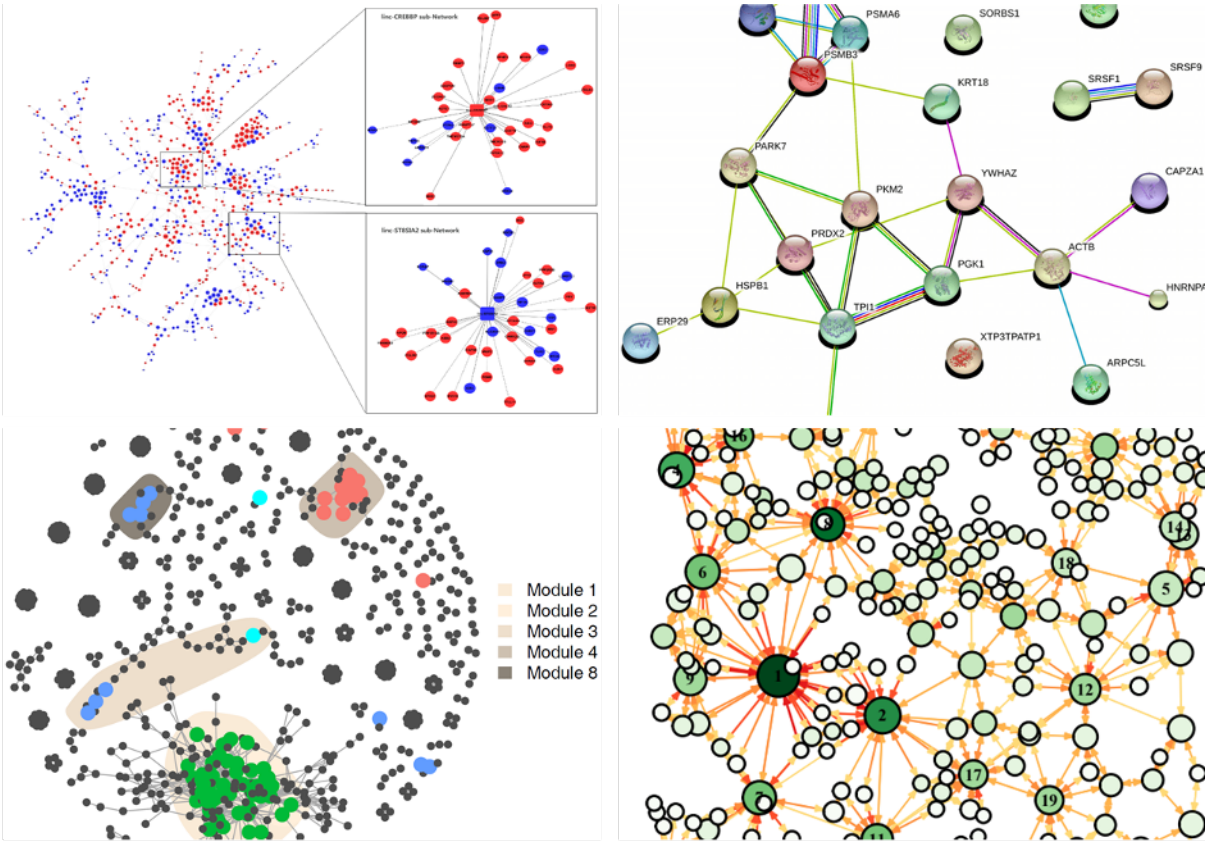


Figure 2.1: Applications of network in biology

- Pajek (http://vlado.fmf.uni-lj.si/pub/networks/pajek/)
- Cytoscape (https://cytoscape.org/)
- Gephi (https://gephi.org/)

2.4 MetaNet

MetaNet is a comprehensive network analysis package.

- Calculate correlation network quickly, accelerate lots of analysis by parallel computing.
- Support for multi-omics data, search sub-nets fluently.
- Handle bigger data, more than 10,000 nodes in each omics.
- Offer various layout method for multi-omics network and some interfaces to other software (Gephi, Cytoscape, ggplot), easy to visualize.
- Provide comprehensive topology indexes calculation including ecological network stability.

Chapter 3

Construction

3.1 Pre-processing

The `trans()` function contains many normalization methods, suitable for pre-processing of different omics, some refer to `vegan::decostand()` (Oksanen et al. [2022]).

Table 3.1: Table 3.1: Normalization methods used in omics.

Method	Description
cpm	Counts per million
minmax	linear transfer to (min, max)
acpm	Counts per million, then asinh transfer
log1	log(n+1) transformat
total	divide by total
max	divide by maximum
frequency	divide by total and multiply by the number of non-zero items, so that the average of non-zero entries is one
normalize	make margin sum of squares equal to one
range	standardize values into range 0 ... 1 (same to `minmax(0,1)`). If all values are constant, they will be transformed to 0.
rank	rank replaces abundance values by their increasing ranks leaving zeros unchanged
rank, rrank	rank replaces abundance values by their increasing ranks leaving zeros unchanged, and rrank is similar but uses relative ranks with maximum 1
pa	scale x to presence/absence scale (0/1).
standardize	scale x to zero mean and unit variance
hellinger	square root of method = "total"

Table 3.1: Table 3.1: Normalization methods used in omics.

log	logarithmic transformation as suggested by Anderson et al. (2006): logb(x)+1 for x>0, where bb is the base of the logarithm; zeros are left as zeros.
alr	Additive log ratio ("alr") transformation (Aitchison 1986) reduces data skewness and compositionality bias. $alr = [log\frac{x_1}{x_D}, ..., log\frac{x_{D-1}}{x_D}]$
clr	centered log ratio ("clr") transformation proposed by Aitchison (1986) reduces data skewness and compositionality bias. $clr = log\frac{x_r}{g(x_r)}$
rclr	robust clr ("rclr") is similar to regular clr (see above) but allows data that contains zeroes. $rclr = log\frac{x_r}{g(x_r>0)}$

```
library(MetaNet)
data(otutab)
#trans(otutab,method="cpm")%>%head()
trans(otutab,method="log1")%>%head(4)
```

```
##                                    NS1      NS2      NS3      NS4      NS5
## s__un_f__Thermomonosporaceae 6.996681 7.560601 6.698268 7.211557 6.970730
## s__Pelomonas_puraquae        7.582229 7.118826 7.767687 7.712891 7.973844
## s__Rhizobacter_bergeniae     6.378426 6.129050 6.791221 6.804615 7.112327
## s__Flavobacterium_terrae     5.501258 5.459586 7.501634 6.513230 7.276556
##                                    NS6      WS1      WS2      WS3      WS4
## s__un_f__Thermomonosporaceae 6.976348 7.133296 7.376508 7.193686 6.848005
## s__Pelomonas_puraquae        7.512071 6.469250 6.206576 7.115582 7.158514
## s__Rhizobacter_bergeniae     6.749931 6.405228 6.154858 6.976348 6.936343
## s__Flavobacterium_terrae     6.198479 5.765191 7.563720 7.309212 6.903747
##                                    WS5      WS6      CS1      CS2      CS3
## s__un_f__Thermomonosporaceae 7.118016 6.919684 7.746733 7.831617 7.444249
## s__Pelomonas_puraquae        6.860664 6.455199 7.174724 7.324490 6.739337
## s__Rhizobacter_bergeniae     6.741701 6.508769 6.937314 7.497207 6.910751
## s__Flavobacterium_terrae     6.359574 5.886104 6.985642 7.105786 6.626718
##                                    CS4      CS5      CS6
## s__un_f__Thermomonosporaceae 7.588830 7.266827 7.331715
## s__Pelomonas_puraquae        7.029088 7.302496 7.069023
## s__Rhizobacter_bergeniae     7.090910 7.085901 6.637258
## s__Flavobacterium_terrae     6.049733 6.940222 7.253470
```

`guolv()` and `hebing()` functions provide can help filter or aggregate the omics data.

3.2 Pairwise relationship

How to determine the pairwise relationship, because the experimental data is generally relatively rare, we mainly relying on statistical inference.

At present, there are mainly two ways, the first one is based on similarity or correlation Faust and Raes [2012]. for example: Spearman, Pearson, Bray-Curtis… based on abundance or incidence data, the similarity matrix between paired species can be calculated, and the randomized data can be used to repeatedly calculate the significance.Finally, meaningful similarities are retained in network.

The second way for networks is based on regression. Divide species into source and target, and use multiple regression to calculate the relationship between species.

Some tools use special methods to optimize the network construction, such as **SparCC**, etc.

3.2.1 Correlation

Correlation is a statistical term describing the degree to which two variables move in coordination with one-another.

Correlation calculation is the first step in all omics network analysis software, there are many method to get ρ and p-value. However, as the data size of omics grow larger and larger, many methods will become very time and computational resource consuming.

Here, we provide the `c_net_cal()` function for one single table or two tables to calculate correlation **fastly** (Figure 3.1), which will return a three elements list include ρ, p-value and p-adjust.

```
#single table
t(otutab) -> totu
c_net_cal(totu,method = "spearman", filename =F,p.adjust.method = NULL) -> corr
str(corr)
```

```
## List of 3
##  $ r       : num [1:464, 1:464] 0 -0.2508 0.1847 0.0114 0.2095 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rhizobacter_
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rhizobacter_
##  $ p.value : num [1:464, 1:464] 0 0.316 0.463 0.964 0.404 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rhizobacter_
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rhizobacter_
##  $ p.adjust: num [1:464, 1:464] 0 0.316 0.463 0.964 0.404 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rl
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rl
```

```
#two tables
metadata[,3:10] -> env
c_net_cal(totu,env,method = "spearman", filename =F,p.adjust.method = NULL) -> corr2
str(corr2)
```

```
## List of 3
##  $ r       : num [1:464, 1:8] 0.356 -0.5253 0.0918 -0.0114 -0.0196 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rl
##   .. ..$ : chr [1:8] "env1" "env2" "env3" "env4" ...
##  $ p.value : num [1:464, 1:8] 0.147 0.0252 0.717 0.9643 0.9384 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rl
##   .. ..$ : chr [1:8] "env1" "env2" "env3" "env4" ...
##  $ p.adjust: num [1:464, 1:8] 0.147 0.0252 0.717 0.9643 0.9384 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:464] "s__un_f__Thermomonosporaceae" "s__Pelomonas_puraquae" "s__Rl
##   .. ..$ : chr [1:8] "env1" "env2" "env3" "env4" ...
```

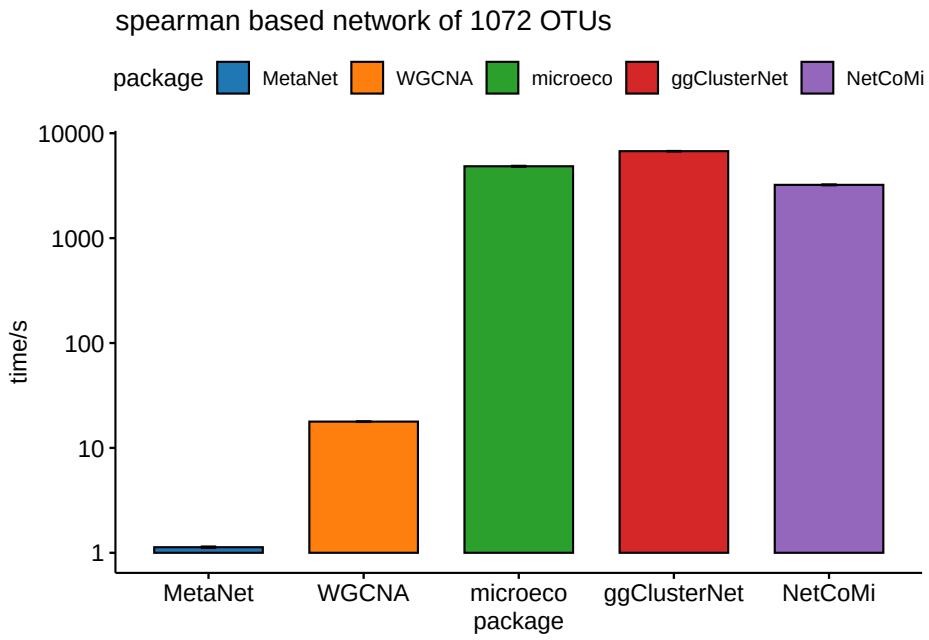


Figure 3.1: network build of packages comparison

3.2.2 Distance

you can use `par_sim()` to calculate various distance to get the pairwise similarity matrix.

3.2.3 SparCC

SparCC fits the Dirichlet distribution to the observed data, and iteratively calculates the proportion and correlation of species several times. The resulting correlation is the median of the distribution. p-values were calculated using the bootstrap method.This metric is said to be more useful with non-normal microbiome data.

$D(x_i, x_j) = var(\log(\frac{x_i}{x_j}))$

`par_sparcc()` is available for SparCC calculation.

3.2.4 Others

There are some other methods available for network construction in **NetCoMi**

3.3 Build network

3.3.1 Normally build

If you have done the `c_net_cal()`, you can get a network (igraph object) easily by `c_net_build()`. Some common attributes will be set automatically.

```
c_net_build(corr,r_thres = 0.6,p_thres = 0.05,del_single = T) -> co_net
co_net
```

```
## IGRAPH 4639c34 UNW- 462 1401 --
## + attr: n_type (g/c), name (v/c), v_group (v/c), v_class (v/c), size
## | (v/n), label (v/c), shape (v/c), color (v/c), id (e/n), from (e/c),
## | to (e/c), weight (e/n), cor (e/n), p.adjust (e/n), p.value (e/n),
## | e_type (e/c), width (e/n), v_group_from (e/c), v_group_to (e/c),
## | e_class (e/c), color (e/c), lty (e/n)
## + edges from 4639c34 (vertex names):
## [1] s__un_f__Thermomonosporaceae--s__Actinocorallia_herbida
## [2] s__un_f__Thermomonosporaceae--s__Kribbella_catacumbae
## [3] s__un_f__Thermomonosporaceae--s__Kineosporia_rhamnosa
## [4] s__un_f__Thermomonosporaceae--s__un_f__Micromonosporaceae
## + ... omitted several edges
```

```
plot(co_net)
```

```
c_net_build(corr2) -> co_net2
plot(co_net2)
```

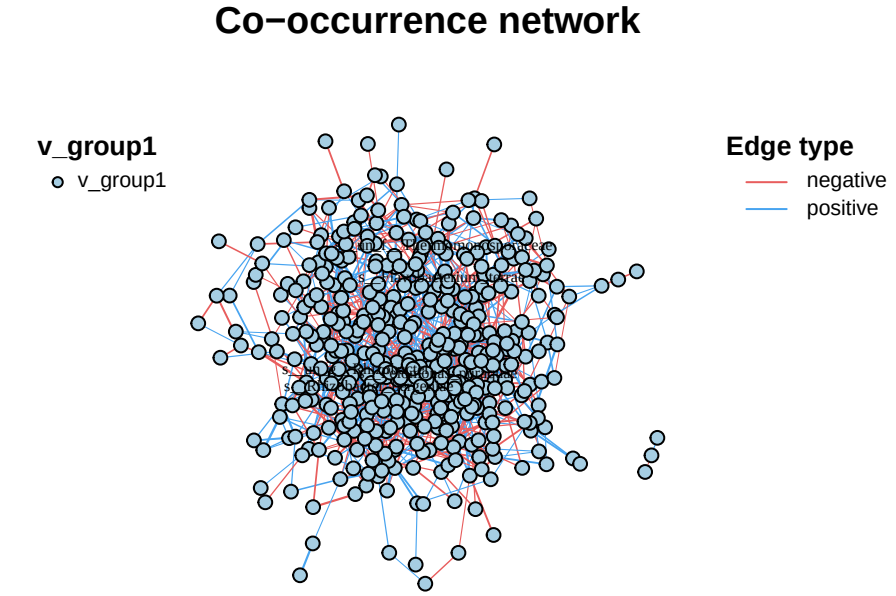
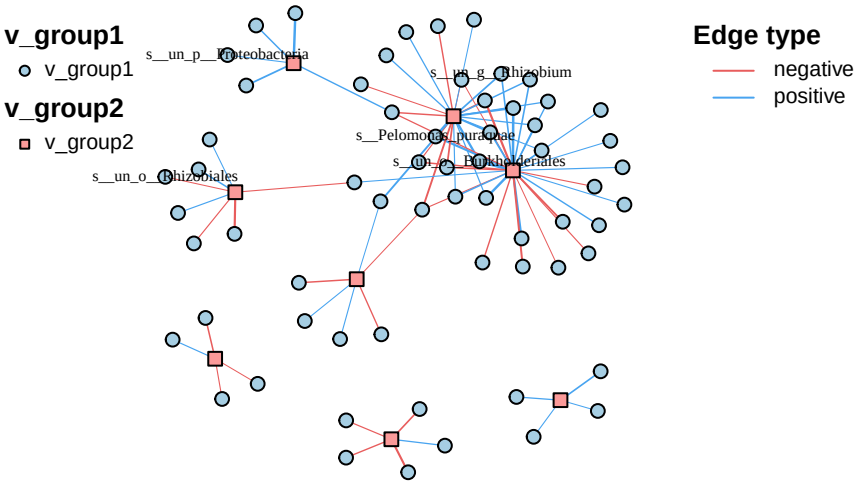


Figure 3.2: Simple co-occurrence network

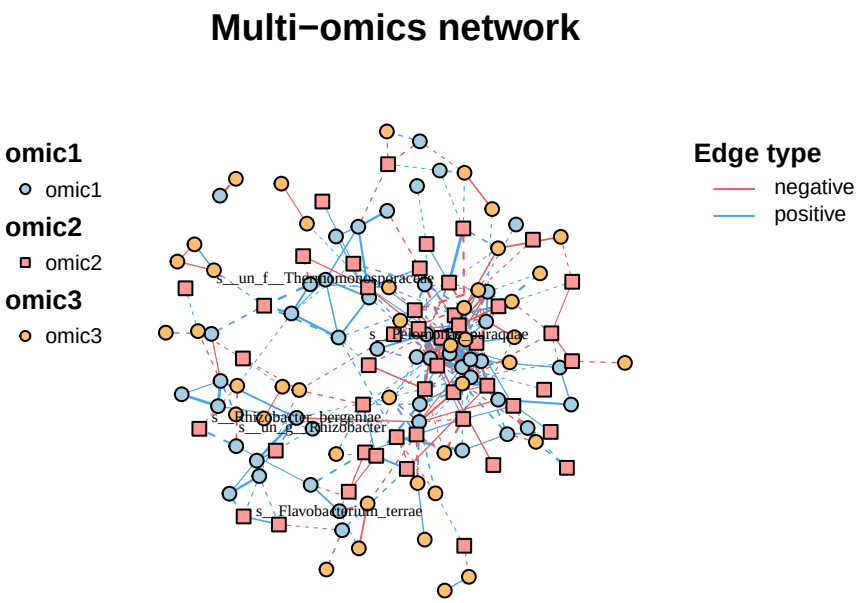


Multi-tables When you have more than two tables for correlation network analysis, you can choose the `multi_net_build()` to calculate and build network. For subsequent multi-omics analysis, see Chapter 8.

```
data("multi_test")
#microbiome
dim(micro)
## [1] 18 50
#metabolome
dim(metab)
## [1] 18 50
#transcriptome
dim(transc)
## [1] 18 50

multi_net_build(micro,metab,transc,mode = "full",method = "spearman",filename = F)->multi1

plot(multi1)
```



Edgelist If you already get the pairwise relationship of data from other approaches (database), you can form it into a edgelist and use `c_net_from_edgelist` to build network. It is useful for following analysis.

```
load("../MetaNet/data/edgelist.rda")
dnet=c_net_from_edgelist(arc_count,direct = T)
plot(dnet)
```

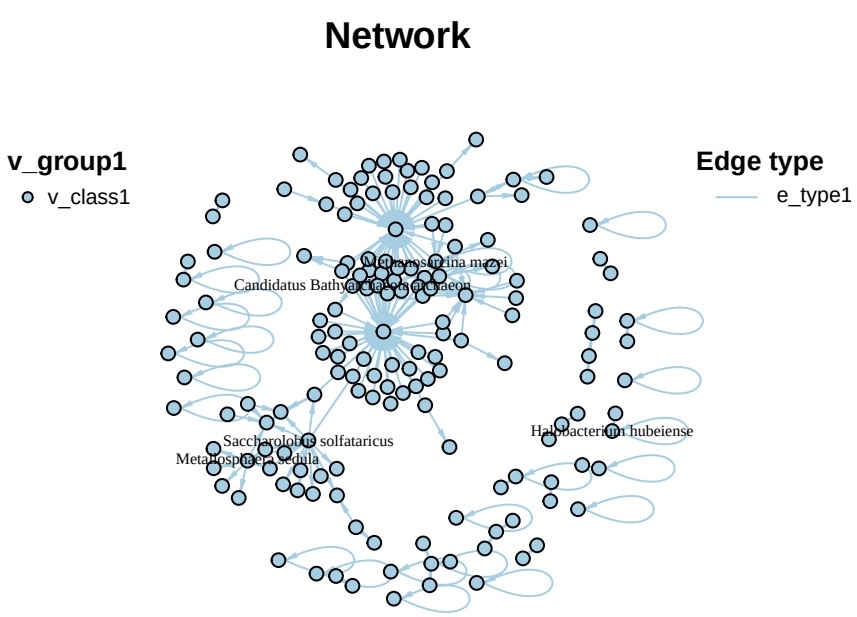


Figure 3.3: Simple directed network

3.4 RMT optimize

The correlation-based relevance network method is most commonly used largely due to its simple calculation procedure and noise tolerance. However, most studies involving relevance network analysis use arbitrary thresholds (usually, we use r>0.6, p<0.05), and thus the constructed networks are subjective rather than objective.

This problem has been solved by a random matrix theory (RMT)-based approach (Figure 3.4), which is able to automatically identify a threshold for cellular network construction from microarray data (Deng et al. [2012]).

use `RMT_threshold()` , we can find the best r_threshold to make the network with smallest noise.

the bigger log_LE, less log_LW, less log_see, bigger p_ks_test indicate the better r_threshold for a meaningful network construction.You can change the threshold range to calculate more finely.

```
RMT_threshold(corr$r,min_threshold = 0.5,max_threshold = 0.9, step = 0.02)->rmt_res
## ========================Calculating1:  threshold =0.5=========================
## ========================Calculating2:  threshold =0.52========================
## ========================Calculating3:  threshold =0.54========================
## ========================Calculating4:  threshold =0.56========================
## ========================Calculating5:  threshold =0.58========================
```

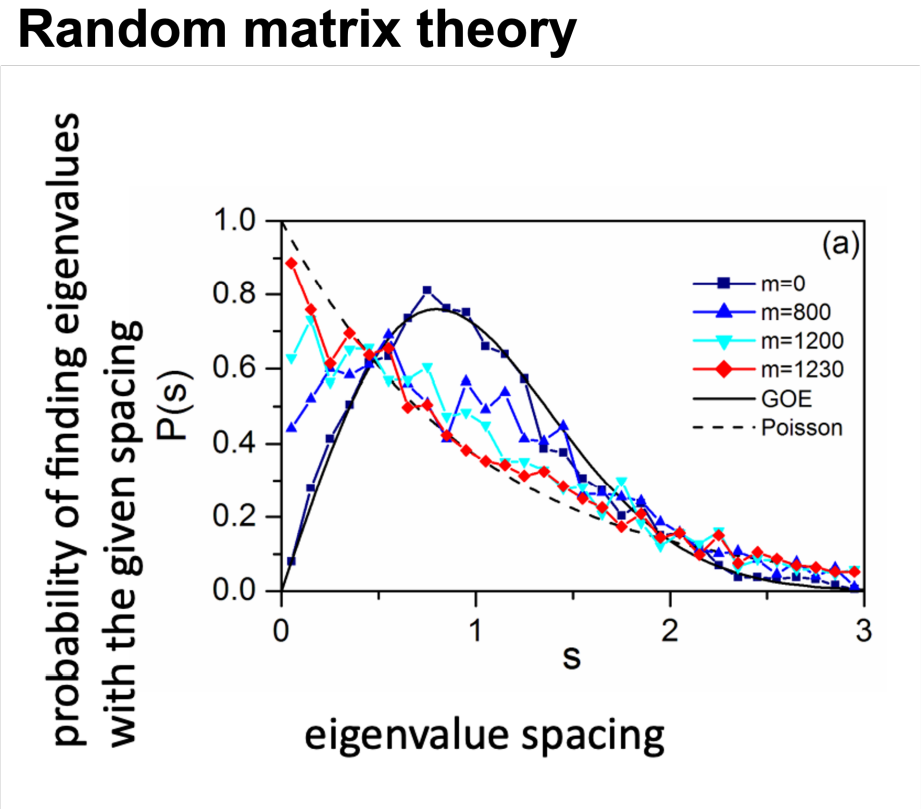


Figure 3.4: random matrix theory (RMT)-based approach

```
## ========================Calculating6:  threshold =0.6=========================
## ========================Calculating7:  threshold =0.62========================
## ========================Calculating8:  threshold =0.64========================
## ========================Calculating9:  threshold =0.66========================
## ========================Calculating10:  threshold =0.68=======================
## ========================Calculating11:  threshold =0.7========================
## ========================Calculating12:  threshold =0.72=======================
## ========================Calculating13:  threshold =0.74=======================
## ========================Calculating14:  threshold =0.76=======================
## ========================Calculating15:  threshold =0.78=======================
## ========================Calculating16:  threshold =0.8========================
## ========================Calculating17:  threshold =0.82=======================
## ========================Calculating18:  threshold =0.84=======================
## ========================Calculating19:  threshold =0.86=======================
## The Intermediate files saved in ./RMT_temp/ .
plot(rmt_res)
```

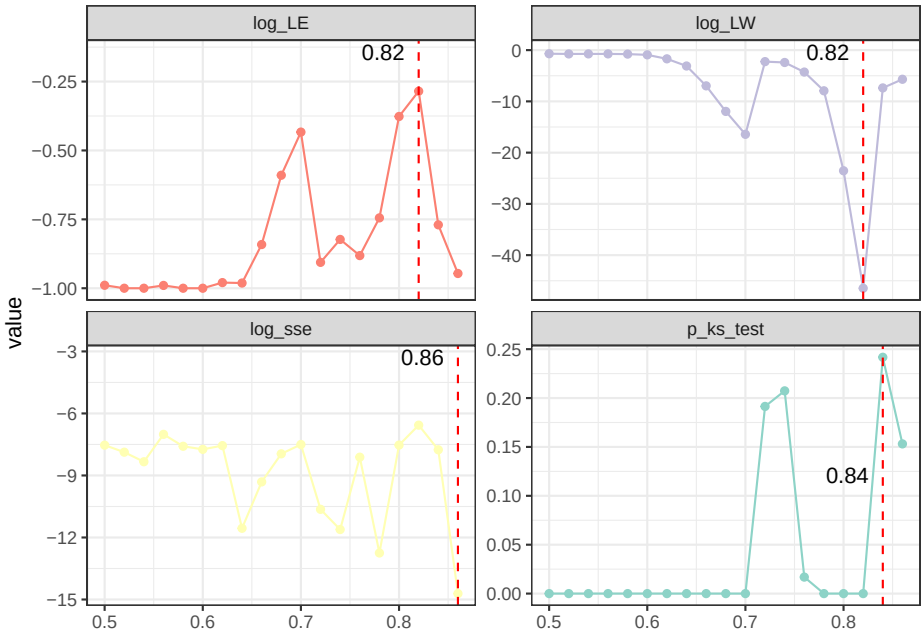


Figure 3.5: RMT_threshold result from 0.5 to 0.9

```
## [1] "recommend r_threshold:  0.835"
## [1] 0.835
```

You can set the `gif=T` in `RMT_threshold` and get a gif file to observe the distribution of eigenvalue spacing for different r-thresholds.

```
r if (knitr::is_html_output()) ' ![(\#fig:unnamed-chunk-13)the
```

```
distribution of eigenvalue spacing from 0.5 to 0.9](images/rmt_nnsd.gif)
'
```

Chapter 4

Manipulation

4.1 Attributes

get_v get_e get_n

4.2 Annotation

anno_vertex anno_edge

summ2col

4.3 Filter (Sub-net)

c_net_filter

4.4 Export

c_net_save

Chapter 5

Visualization

5.1 Basic plot

5.2 Layout

5.3 Other styles

5.3.1 ggplot

5.3.2 Gephi

5.3.3 Cytoscape

Chapter 6

Topology

6.1 Complex network

The microbial co-occurrence network we study is a complex network, which generally has the following characteristics, scale-free, small-world attributes, modularity and hierarchy.

Table 6.1: Common characteristic

Terminology	Explanation
Scale-free	Scale-free It is a most notable characteristic in complex systems. It was used to desibe the finding that most nodes in a network have few neighbors while few nodes have large amount of neighbors. In most cases, the connectivity distribution asymptotically follows a power law. It can be expressed in , where $P(k) \sim k^{-y}$ P(k) is the number of nodes with k degrees, k is connectivity/degrees and is a constant.
Small-world	Small-world It is a terminology in network analyses to depict the average distance between nodes in a network is short, usually logarithmically with the total number of nodes. It means the network nodes are always closely related with each other.

Table 6.1: Common characteristic

Modularity	The modularity of a graph with respect to some division (or vertex types) measures how good the division is, or how separated are the different vertex types from each other. It defined as $Q = \frac{1}{2m}\sum_{i,j}(A_{ij} - \gamma\frac{k_i k_j}{2m})\delta(c_i, c_j)$ here mm is the number of edges, A_{ij} is the element of the A adjacency matrix in row i and column j, k_i is the degree of i, k_j is the degree of j, c_i is the type (or component) of i, c_j that of j, the sum goes over all i and j pairs of vertices, and $\delta(x, y)$ is 1 if x=y and 0 otherwise. The resolution parameter γ allows weighting the random null model, which might be useful when finding partitions with a high modularity.The original definition of modularity is retrieved when setting γ to 1.
Hierarchy	Hierarchy It was used to depict the networks which could be arranged into a hierarchy of groups representing in a tree structure. Several studies demonstrated that metabolic networks are usually accompanied by a hierarchical modularity.

`fit_power()` is used to prove the scale-free. `smallworldness()` can calculate the smallworld index.

6.2 Topology indexes

Indexes	Formula	Note	Description
Part I: network indexes for individual nodes			
Connectivity/ Degree (centrality)	$k_i = \sum_{j \neq i} a_{ij}$	a_{ij} is the connection strength between nodes i and j. when a_{ij}=1, ki is the unweighted degree	It is also called node degree. It is the most commonly used concept for describing the topological property of a node in a network.

Indexes	Formula	Note	Description
Betweenness centrality	$B_i = \sum_{j,k} \frac{\sigma(i,j,k)}{\sigma(j,k)}$	(, ,) is the number of shortest paths between nodes j and k that pass through node i. (,) is the total number of shortest paths between j and k.	It is used to describe the ratio of paths that pass through the ith node. High Betweenness node can serve as a broker similar to stress centrality.
Closeness centrality	$Ci = 1/\sum_{i \neq j} d_{ij}$	The closeness centrality of a vertex is defined as the inverse of the sum of distances to all the other vertices in the graph. dij is the shortest distances from node i to j.	Closeness centrality measures how many steps is required to access every other vertex from a given vertex.
Eigenvector centrality	$EC_i = \frac{1}{\lambda} \sum_{j \in M(i)} EC_j$	M( ) is the set of nodes that are connected to the ith node and is a constant eigenvalue.	It is used to describe the degree of a central node that it is connected to other central nodes.
Clustering coefficient	$CCo_i = \frac{2l_i}{k_i'(k_i'-1)}$	li is the number of links between neighbors of node i and k i ' is the number of neighbors of node i.	It describe how well a node is connected with its neighbors. If it is fully connected to its neighbors, the clustering coefficient is 1. A value close to 0 means that there are hardly any connections with its neighbors. It was used to describe hierarchical properties of networks.
Eccentricity	$Ey = \max_{j \neq i}(d_{ij})$	dij is the shortest distance from node i to node j	The eccentricity of a vertex is its shortest path distance from the farthest other node in the graph.
Page.rank	$PR_i = \sum_{j \in B_i} \frac{PR_j}{l_j}$	i is the node whose pr value needs to be calculated, and Bi is the set of all nodes pointing to node i. PRj is the pr value of node j and lj is the number of links between neighbors of node j.	Calculates the Google PageRank for the specified vertices. PageRank, or webpage ranking, also known as webpage level, is an indicator to measure the importance of webpages.

Indexes	Formula	Note	Description
Kleinberg's hub and authority centrality	$HC = \lambda_{AA^T}$ $AC = \lambda_{A^TA}$	The hub scores of the vertices are defined as the principal eigenvector of AAT, the authority scores of the vertices are defined as the principal eigenvector of ATA. where A is the adjacency matrix of the graph.	A node is an authority if it is linked to by hubs; it is a hub if it links to authorities.
Part II: The overall network topological indexes			
Average connectivity/ degree	$\overline{k} = \frac{\sum_{i=1}^{n} k_i}{n}$	k i is degree of node i and n is the number of nodes.	Higher avgK means a more complex network.
Average path length/ Average geodesic distance	$L = \frac{\sum_{i \neq j} d_{ij}}{n(n-1)}$	dij is the shortest path between node i and j.	A smaller GD means all the nodes in the network are closer.
global efficiency/ Geodesic efficiency	$E_g = \frac{\sum_{i \neq j} 1/d_{ij}}{n(n-1)}$	all parameters shown above.	It is the opposite of GD. A higher E means that the nodes are closer.

Indexes	Formula	Note	Description
Centralization of degree	$CD = \sum_{i=1}^{n}(\max(k) - k_i)$	max(k) is the maximal value of all connectivity values and k i represents the connectivity of ith node. Finally this value is normalized by the theoretical maximum centralization score.	It is close to 1 for a network with star topology and in contrast close to 0 for a network where each node has the same connectivity.
Centralization of betweenness	$CB = \sum_{i=1}^{n}(\max(B) - B_i)$	max(B) is the maximal value of all betweenness values and B i represents the betweenness of ith node. Finally this value is normalized by the theoretical maximum centralization score.	It is close to 0 for a network where each node has the same betweenness, and the bigger the more difference among all betweenness values.
Centralization of closeness	$CC = \sum_{i=1}^{n}(\max(C) - C_i)$	max(C) is the maximal value of all closeness values and Ci represents the closeness of ith node. Finally this value is normalized by the theoretical maximum centralization score.	It is close to 0 for a network where each node has the same closeness, and the bigger the more difference among all closeness values.
Centralization of eigenvector centrality	$CE = \sum_{i=1}^{n}(\max(EC) - EC_i)$	max(EC) is the maximal value of all eigenvector centrality values and EC i represents the eigenvector centrality of ith node. Finally this value is normalized by the theoretical maximum centralization score.	It is close to 0 for a network where each node has the same eigenvector centrality, and the bigger the more difference among all eigenvector centrality values.
Density	$D = \frac{2l}{n(n-1)}$	l is the sum of total links.	The density of a graph is the ratio of the number of edges and the number of possible edges. It is closely related to the average connectivity.

Indexes	Formula	Note	Description
Average clustering coefficient	$\overline{CCo} = \frac{\sum_{i=1}^{n} CCo_i}{n}$	oi is the clustering coefficient of node i.	It is used to measure the extent of module structure present in a network.
Transitivity	$Tr = \frac{\sum_{i=1}^{n} 2l_i}{\sum_{i=1}^{n}(k_i')(k_i'-1)}$	li is the number of links between neighbors of node i and k i ' is the number of neighbors of node i.	Sometimes it is also called the entire clustering coefficient. It has been shown to be a key structural property in social networks.
Natural connectivity	$NC = \ln\left(\frac{1}{N}\sum_{i=1}^{N} e^{\lambda_i}\right)$	Where N is nodes number of the network, represents the eigenvalue of the network adjacency matrix.	

Chapter 7

Stability

It is very important to compare networks stability based on different groups. So MetaNet collects lots of methods to reflect the stability and complexity, these algorithms are coded using Parallel Computing which can be much faster.

7.1 Robustness tests

Robustness tests of networks were done with natural connectivity as it can reflect the stability of networks (WU Jun [2010]). Specifically, natural connectivity was calculated after removing the nodes (remove five nodes from a network at one time until 70% of nodes disappear), and the downtrend level of natural connectivity indicated the connectivity performance of the network after being damaged to a certain extent.

7.2 Community stability

Community stability can be characterized by various indexes, such as **robustness**, **vulnerability** and **cohesion**. Networks with higher robustness and lower vulnerability tend to be more stable (Yuan et al. [2021]). Also, community stability is commonly associated with negative interactions, and high percentage of negative correlations within communities is essential for maintaining a stable ecological system. ### Robustness The robustness was regarded as when 50% of nodes were randomly removed and results were based on 100 repetitions of the simulation.

7.2.1 Vulnerability

To evaluate the speed of disturbance spreading within a network, the global efficiency was regarded as the average of the efficiency over all pairs of nodes,

which was calculated by the number of edges in the shortest path between paired nodes. The vulnerability, which reflected the relative contribution of each node to the globe efficiency, was represented by the maximal vulnerability of nodes in the network.

7.2.2 Cohesion

Cohesion was calculated to quantify the connectivity of microbial communities in each group. Cohesion contains both positive and negative cohesion values, which indicate that associations between taxa attributed to positive and negative species interactions as well as similarities and differences in niches of microbial taxa11. Briefly, pairwise Pearson correlation matrix across taxa was calculated based on abundance-weighted matrix. After "taxa shuffle" null module-correcting with 200 simulations, average positive and negative correlations was calculated to get a connectedness matrix. Finally, positive and negative cohesions were calculated for each sample respectively by multiplying the abundance-weighted matrix and connectedness matrix. The absolute value of negative: positive cohesion is an important index for community stability. $cohesion = \sum_{i=1}^{m} \text{abundance}_i \times connectness_i$

Chapter 8

multi-omics

8.1 Case study

In-depth multi-omic profiling was performed on each sample including plasma proteomics (targeted and untargeted), metabolomics (untargeted), lipidomics (semi-targeted), and gene expression (transcriptomics) from peripheral blood mononuclear cells (PBMCs).(Contrepois et al. [2020])

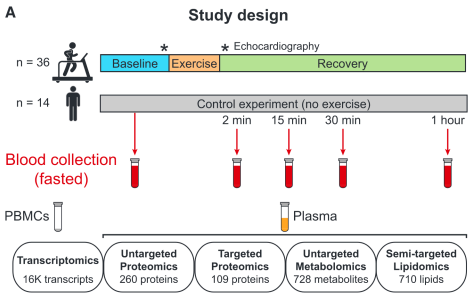


Figure 8.1: Overview of the study design including an acute bout of exercise

Bibliography

Kévin Contrepois, Si Wu, Kegan J. Moneghetti, Daniel Hornburg, Sara Ahadi, Ming-Shian Tsai, Ahmed A. Metwally, Eric Wei, Brittany Lee-McMullen, Jeniffer V. Quijada, Songjie Chen, Jeffrey W. Christle, Mathew Ellenberger, Brunilda Balliu, Shalina Taylor, Matthew G. Durrant, David A. Knowles, Hani Choudhry, Melanie Ashland, Amir Bahmani, Brooke Enslen, Myriam Amsallem, Yukari Kobayashi, Monika Avina, Dalia Perelman, Sophia Miryam Schüssler-Fiorenza Rose, Wenyu Zhou, Euan A. Ashley, Stephen B. Montgomery, Hassan Chaib, Francois Haddad, and Michael P. Snyder. Molecular Choreography of Acute Exercise. *Cell*, 181(5):1112–1130.e16, May 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.04.043.

Ye Deng, Y. Jiang, Yunfeng Yang, Zhili He, Feng Luo, and Jizhong Zhou. Molecular ecological network analyses. *BMC bioinformatics*, 2012. doi: 10.1186/1471-2105-13-113.

Karoline Faust and J. Raes. Microbial interactions: From networks to models. *Nature Reviews Microbiology*, 2012. doi: 10.1038/nrmicro2832.

Jari Oksanen, Gavin L. Simpson, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R.B. O'Hara, Peter Solymos, M. Henry H. Stevens, Eduard Szoecs, Helene Wagner, Matt Barbour, Michael Bedward, Ben Bolker, Daniel Borcard, Gustavo Carvalho, Michael Chirico, Miquel De Caceres, Sebastien Durand, Heloisa Beatriz Antoniazi Evangelista, Rich FitzJohn, Michael Friendly, Brendan Furneaux, Geoffrey Hannigan, Mark O. Hill, Leo Lahti, Dan McGlinn, Marie-Helene Ouellette, Eduardo Ribeiro Cunha, Tyler Smith, Adrian Stier, Cajo J.F. Ter Braak, and James Weedon. *vegan: Community Ecology Package*, 2022. URL https://github.com/vegandevs/vegan. R package version 2.6-4.

Mauricio Barahona WU Jun. Natural Connectivity of Complex Networks. *Chinese Physics Letters*, 27(7):78902–078902, June 2010. ISSN 0256-307X. doi: 10.1088/0256-307X/27/7/078902.

Mengting Maggie Yuan, Xue Guo, Linwei Wu, Ya Zhang, Naijia Xiao, Daliang Ning, Zhou Shi, Xishu Zhou, Liyou Wu, Yunfeng Yang, James M. Tiedje, and Jizhong Zhou. Climate warming enhances microbial network complexity and

stability. *Nature Climate Change*, 11(4):343–348, April 2021. ISSN 1758-6798. doi: 10.1038/s41558-021-00989-9.