GUÍA PARA ELABORACIÓN DE RED DE COOCURRENCIA

Pasos previos

 # a) Descargar Visualizador Gephi

Release Gephi 0.10.1 · gephi/gephi · GitHub

Link: https://github.com/gephi/gephi/releases/tag/v0.10.1

 # b) Descargar y copiar archivo "dataset.RData" en carpeta de trabajo (ejemplo: Coocurrencia)

El set de datos "dataset.RData" contiene 4 archivos: tabla de abundancia de ASVs, información de las muestras, taxonomía de ASVs.

 # c) Establecer directorio de trabajo

setwd("C:/Users/jpaaa/OneDrive/Escritorio/dataset")

 # d) Cargar y Explorar set de datos

load("dataset.RData")

 # e) Cargar R packages a utilizar:

install.packages("microeco")

install.packages("devtools")

install.packages("BiocManager")

library(microeco)

library(rlang)

library(ggplot2)

library(magrittr)

library(BiocManager)

library(rgexf)

Reference: Chi Liu, Yaoming Cui, Xiangzhen Li and Minjie Yao. 2021. *microeco*: an R package for data mining in microbial community ecology. FEMS Microbiology Ecology, 97(2): fiaa255. https://doi.org/10.1093/femsec/fiaa255

Link tutorial: https://chiliubio.github.io/microeco_tutorial/model-based-class.html#trans_network-class

 # f) Explorar set de datos "dataset"

Dataset

View(dataset[["otu_table"]])

View(dataset[["sample_table"]])

View(dataset[["tax_table"]])

 # g) Rango de número de secuencias

dataset$sample_sums() %>% range

##RED DE CO-OCURRENCIA ##

NetCoMi (Network Construction and Comparison for Microbiome Data) proporciona funciones estadisticas para construir, analizar y comparar redes, las cuales son adecuadas para la aplicación en datos de composición microbiana

Referencia: Peschel, S., Müller, C. L., von Mutius, E., Boulesteix, A.-L., & Depner, M. (2020). NetCoMi: network construction and comparison for microbiome data in R. Briefings in Bioinformatics. doi:10.1093/bib/bbaa290

Install NetCoMi

devtools::install_github("stefpeschel/NetCoMi",

dependencies = c("Depends", "Imports", "LinkingTo"),

repos = c("https://cloud.r-project.org/",

BiocManager::repositories()))

installNetCoMiPacks()

library(NetCoMi)

1.- Método de Correlación

t1 <- trans_network$new(dataset = dataset, cor_method = "pearson", taxa_level = "OTU", filter_thres = 0.001)

#* Metodo alternativo:

***#** SparCC (Sparse Correlations for Compositional data): técnica para inferir correlaciones a partir de datos de composición. SparCC estima las correlaciones lineales de Pearson entre los componentes transformados logarítmicamente.

**#Reference: Friedman, J., & Alm, E. J. (2012). Inferring Correlation Networks from Genomic Survey Data. PLoS Computational Biology, 8(9), e1002687. doi:10.1371/journal.pcbi.1002687

*t1 <- trans_network$new(dataset = dataset, cor_method = "sparcc", use_sparcc_method = "NetCoMi", filter_thres = 0.001)

#The correlation result list is stored in object$res_cor_p ...

return t1$res_cor_p list, containing two tables: correlation coefficient table and p value table

t1$res_cor_p

write.table(t1$res_cor_p, file='res_cor_p.txt', sep='\t', quote=FALSE)

2.- Construcción de la Red

t1$cal_network(COR_p_thres = 0.01, COR_optimization = TRUE, COR_cut = 0.5)

#The result network is stored in object$res_network ...

#return t1$res_network

t1$res_network

3.- Partición de Módulos para la Red

Calcular los módulos de red y añadir los nombres de los módulos al nodo de red.

Método utilizado para encontrar la estructura de comunidad óptima de una red.

t1$cal_module(method = "cluster_fast_greedy")

4.- Guardar Red

La siguiente operación consiste en guardar la red en formato gexf para su visualización con Gephi(https://gephi.org/). El archivo 'network.gexf' puede ser abierto directamente por Gephi.

t1$save_network(filepath = "Coocurrence_network.gexf")

5.- Calcular atributos de la Red

t1$cal_network_attr()

t1$res_network_attr

```
# get node properties

t1$get_node_table(node_roles = TRUE)

# return t1$res_node_table

t1$res_node_table

write.table(t1$res_node_table, file='res_node_table.txt', sep='\t', quote=FALSE)

# get edge properties

t1$get_edge_table()

# return t1$res_edge_table

t1$res_edge_table

write.table(t1$res_edge_table, file='res_edge_table.txt', sep='\t', quote=FALSE)
```

6.- Clasificación de Roles de los Nodos

#A continuación, vamos a trazar la clasificación de los nodos en términos de la conectividad dentro del módulo y entre módulos.

```
t1$plot_taxa_roles(use_type = 1, add_label = TRUE, add_label_text = "name")

# Graficar Roles de los Nodos con Información de los phyla

t1$plot_taxa_roles(use_type = 2)
```

7.- Correlación entre módulos y variables ambientales

#Cálculo de EigenGene de módulos. El EigenGene de un módulo, es el primer componente principal del PCA, el cual representa la variación principal de la abundancia en las especies del módulo.

```
t1$cal_eigen()

# return t1$res_eigen

t1$res_eigen

# A continuación, calcularemos la correlación de los EigenGene y las variables ambientales asociadas al set de datos.

# create trans_env object

t2 <- trans_env$new(dataset = dataset, add_data = env_data_16S)
```

```
# calculate correlations
t2$cal_cor(add_abund_table = t1$res_eigen)
# plot the correlation heatmap
t2$plot_cor()
```