The Manual of GRaMM and GRaMMnonlinfit Functions (Matlab)

1. Introduction

GRaMM is designed specifically for inter-correlation identification between metabolome and microbiome. It can detect both linear and nonlinear correlations, both function and nonfunction relationships. The main workflows contain: (1) data preprocessing, (2) identifying linear or nonlinear correlation, (3) correlation analysis, (4) the p value adjustment.

The preprocessing of metabolic variables includes log transformation and normalization by sample total intensity. Microbial variables will be normalized by sample total abundance if they have not been, and then will be transformed using the centered log-ratio (CLR) transformation. Step 2 is to identify roughly the correlation type. Since the linear relationship is simple and is favorable in most cases, LR will be applied to the pair and the p and r values will be obtained. Based on the threshold values of p and/or r, this pair will be handled by different arms. Step 3, for a linear correlation (e.g. LR p<0.05), results of LR will be outputted directly when there are no confounders. Or, multiple linear regression (mLR) will be applied, taking the metabolic variables as y and the microbial variables and confounders as x. The adjusted p and r values will be outputted. For a nonlinear correlation (e.g. LR p>0.05), MIC will be conducted and the p and r values will be outputted directly when there are no confounders. When there are confounders, the variables will be processed by the Metabolic Confounding Effect Elimination (MCEE) before the application of MIC. Besides, GRaMM will conduct curve fitting using 5 typical models (Table. S4) for a nonlinear correlated pair and output the best one. Finally, step 4, the p value will be adjusted using Benjamini-Hochberg to control the multiple testing false discovery rate (FDR).

GRaMMnonlinfit is used to fit several special curves for nonlinear correlation pairs. It concludes five models: (1): y = b(1).* x.^2 + b(2).* x + b(3); (2): y = b(1) .* log(x) + b(2); (3): y = exp(b(1)+b(2) .* x); (4): y = 1./(b(1) + b(2).*exp(-x)); (5): y = b(1).*x.^b(2)+b(3). As for each pairs, five curves will be tried and tested, and the final output will be the one with the smallest sum of squares due to error (SSE).

First, you can use the core function, GRaMM, to explore possible relationship between metabolites and microbiota. And then, some interested nonlinear correlation pairs can be fitted using GRaMMnonlinfit to intuitively show their relationship.

2. Precautions before use

A core function is mine() for correlation analysis. It was provided from minepy package on github. Before using GRaMM, this package should be downloaded and added to the default path of Matlab.

3. To use GRaMM() function

function [result,GRaMM] = GRaMM(A,B,C,Anorm,Bnorm,alpha,linR)

Input:

A is a matrix of metabolome data (p*n). The raw(p) is the variates, column(n) is samples.

B is a matrix of microbiome data (v*n). The raw(v) is the variates, column(n) is samples.

C is optional for the covariates matrix , u*n. The raw(u) is the covariates,column(n) is samples.

Anorm = 'yes',or 'no'(defualt). If choose "yes", the metabolome data will be benormalized by samples."no" means the metabolome data will not be normalized by samples.

Bnorm = 'yes',or 'no'(defualt). If choose "yes", the microbiome data will be benormalized by samples."no" means the microbiome data will not be normalized by samples.

Alpha (defualt = 0.05), the threshold of significant difference for linear correlation.

linR (defualt = 0.1), the linear correlation threshold to determine whether or not it is linear.

"> linR" means it tends to a strong linear correlation;

" <linR " means it it's more llikely to be nonlinear correlation.

Output:

(1): "results" includes 4 columns, correlation strength(r),p value,FDR value and the correlation types("0"=linear correlation, "1"=nonlinear correlation).

(2): "GRaMM", as a structure, includes 4 matrixes(v*p) for r, p, FDR and correlation types respectively.

4. To use GRaMMnonlinfit() function.

function []=GRaMMnonlinfit(dataA,dataB)

Input:

dataA: The Variable matrix1. The raw is the variates, column is samples, such as the preprocessing metabolome data matrix.

dataB: The Variable matrix2. The raw is the variates, column is samples, such as the preprocessing microbiome data matrix.

Output:

The picture of best fitting curve for each pair.