**不服就肝！FEAST Source Tracker：快速准确的微生物来源追溯工具 百分百畅通版~**

广东省科学院 徐锐 环境微生物方向

最近帮老板做项目想试下source tracker，刚好看到公众号推送了一篇相关攻略。不过实测以后发现“作者GitHub原版”和“文涛聊科研版”都有一些bug老是跑不通，于是自己仔细捋顺了一遍，希望能抛砖引玉，帮助小白们速入门~

# 一、感谢前人相关工作，同时大家也可对比一下~

1.【GitHub原版链接】

<https://github.com/cozygene/FEAST> （点击右侧绿色Download打包下载）

2.【文涛版链接】

<https://mp.weixin.qq.com/s?__biz=MzUzMjA4Njc1MA==&mid=2247488607&idx=1&sn=68738f1e48aa261c34a04c2adae6371c&chksm=fab9feeecdce77f841a857cbb2c2482bcc41cc82cfcc98786cb451d04c4c3af508117948b063&scene=0&xtrack=1&key=b00bc6577437f0f52d6016952ad45c699fbf61ed10b2ee8f14e1ec07d2412e7a9d12aa23b71b817c7962d08c509b8ee9567bcb3a35b41a6a9b31057d11c8276f78e7ea78fdecdda22c14a31ce6f83e23&ascene=1&uin=MTM5NDQzNzAyMA%3D%3D&devicetype=Windows+10&version=62060833&lang=zh_CN&pass_ticket=4vpn6mEW%2FkXRmBOinCIGt%2B3CaO1Ns69ZxA79iYFRaEGxMzAxM5BabqayaIjvfed5>

3.【宏基因组入门介绍】

<https://mp.weixin.qq.com/s?__biz=MzUzMjA4Njc1MA==&mid=2247488512&idx=1&sn=721697280d1901373813df23b02d153e&scene=21#wechat_redirect>

# 二、实测发现的Bugs

1.原版脚本中没有提供所依赖的各包，对小白不友好~

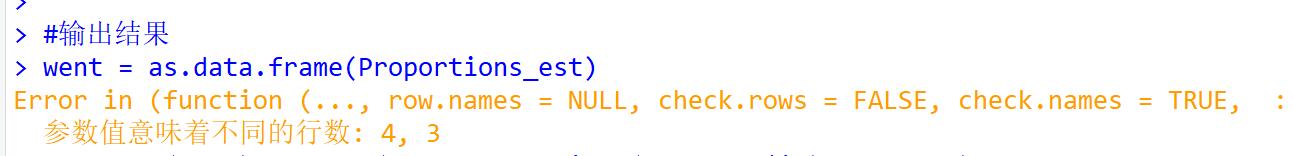
2.所需文件格式txt/csv没统一，换自己的数据时非常容易在Line68求和时报错：

（# COVERAGE = min(rowSums(otus[c(train.ix, test.ix),]))），很可能是面前表头读取错误导致。

3.原版测试数据data frame的行列个数有1处不对（其余都是4，有1个是3），导致后续将结果as.data.frame时报错。



图。原数据的bug



图。将原数据的计算结果as.data.frame的bug

4.文涛版提供了很好的结果可视化脚本，但可惜没有提供测试数据。

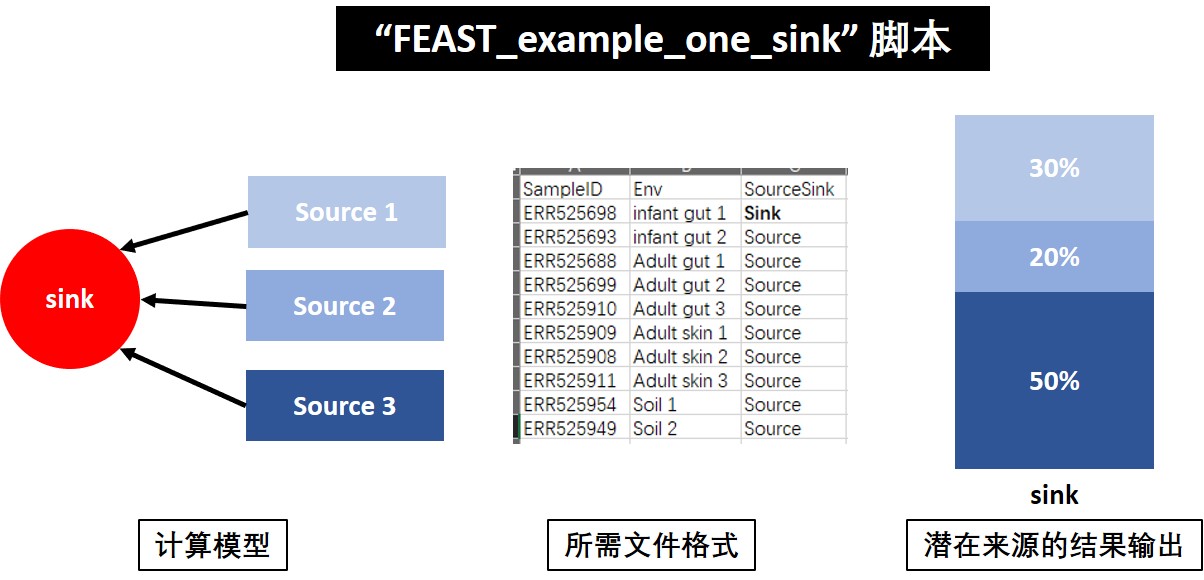
# 三、修改使用说明

1.将本修订脚本文件夹打包复制到工作目录下，其中已含有原版可供参考

2.菌群潜在来源定义为“Source”，待计算目标定义为“Sink”。

3.FEAST提供了2组脚本（计算1个sink时选“FEAST\_example\_one\_sink.R”；计算多个sink时选“FEAST\_example\_Multiple\_sinks.R”），2组脚本都依赖主程序“FEAST\_scr/src.R”

4.其中多个sink的脚本需要设置参数“different\_sources\_flag”（每个sink的source不同时=1，相同时=0），具体选择流程图如下：



图。1个sink时的选择方式

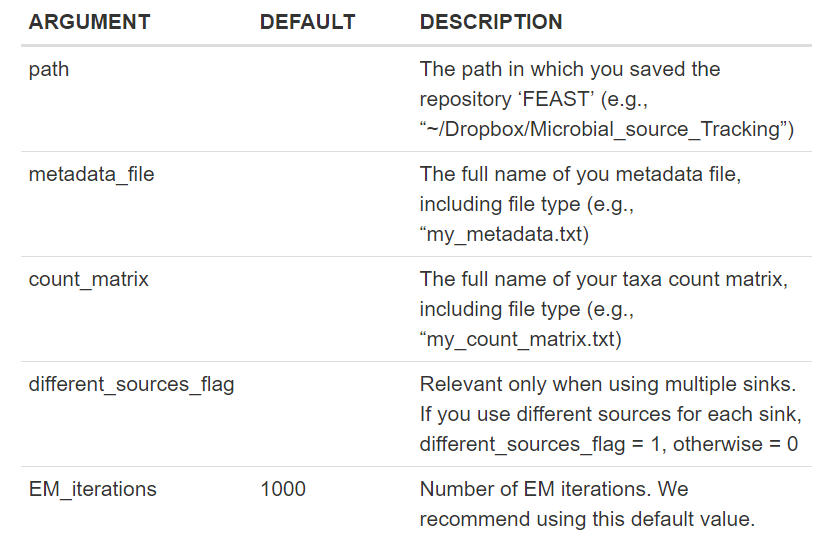


图。多个sink时的选择方式

5.修改后的脚本统一使用csv数据格式

# 四、脚本注释与说明

1.几个重要的Argument



2.所需文件：meta\_data（分组文件，要求见上流程图）和otu\_data（样品的otu丰度表），在脚本中改好对应的文件名即可。

3.以最复杂的Multiple\_sinks脚本为例，解析如下：

#加载必要的包

library("vegan")

library("dplyr")

library("doParallel")

library("foreach")

library("mgcv")

library("reshape2")

library("ggplot2")

library("cowplot")

library("Rcpp")

library("RcppArmadillo")

#准备

rm(list = ls())

gc()

#所有需要设置的变量，共3个

#Set the arguments of your data设置计算数据，最好都统一为csv格式

metadata\_file = "all\_meta.csv" #分组信息

count\_matrix = "all\_otu.csv" #otu表

EM\_iterations = 1000 #default value=1000

##if you use different sources for each sink, different\_sources\_flag = 1, otherwise = 0

different\_sources\_flag = 0

# Load main code加载主程序

print("Change directory path")

dir\_path = paste("C:/ ...your path/ FEAST/") #修改成FEAST文件夹所在目录

setwd(paste0(dir\_path, "FEAST\_src"))

source("src.R")

# Load sample metadata加载数据，仍旧统一为csv格式

setwd(paste0(dir\_path, "Data\_files"))

metadata <- read.csv(metadata\_file, header=T, sep = ",", row.names = 1)

# Load OTU table

otus <- read.csv(count\_matrix, header=T, sep = ",", row.names = 1)

otus <- t(as.matrix(otus))

#计算过程，不用管

# Extract only those samples in common between the two tables

common.sample.ids <- intersect(rownames(metadata), rownames(otus))

otus <- otus[common.sample.ids,]

metadata <- metadata[common.sample.ids,]

# Double-check that the mapping file and otu table

# had overlapping samples

if(length(common.sample.ids) <= 1) {

message <- paste(sprintf('Error: there are %d sample ids in common '),

'between the metadata file and data table')

stop(message)

}

if(different\_sources\_flag == 0){

metadata$id[metadata$SourceSink == 'Source'] = NA

metadata$id[metadata$SourceSink == 'Sink'] = c(1:length(which(metadata$SourceSink == 'Sink')))

}

envs <- metadata$Env

Ids <- na.omit(unique(metadata$id))

Proportions\_est <- list()

for(it in 1:length(Ids)){

# Extract the source environments and source/sink indices

if(different\_sources\_flag == 1){

train.ix <- which(metadata$SourceSink=='Source' & metadata$id == Ids[it])

test.ix <- which(metadata$SourceSink=='Sink' & metadata$id == Ids[it])

}

else{

train.ix <- which(metadata$SourceSink=='Source')

test.ix <- which(metadata$SourceSink=='Sink' & metadata$id == Ids[it])

}

num\_sources <- length(train.ix)

COVERAGE = min(rowSums(otus[c(train.ix, test.ix),])) #Can be adjusted by the user

str(COVERAGE)

# Define sources and sinks

sources <- as.matrix(rarefy(otus[train.ix,], COVERAGE))

sinks <- as.matrix(rarefy(t(as.matrix(otus[test.ix,])), COVERAGE))

print(paste("Number of OTUs in the sink sample = ",length(which(sinks > 0))))

print(paste("Seq depth in the sources and sink samples = ",COVERAGE))

print(paste("The sink is:", envs[test.ix]))

# Estimate source proportions for each sink

FEAST\_output<-FEAST(source=sources, sinks = t(sinks), env = envs[train.ix], em\_itr = EM\_iterations, COVERAGE = COVERAGE)

Proportions\_est[[it]] <- FEAST\_output$data\_prop[,1]

names(Proportions\_est[[it]]) <- c(as.character(envs[train.ix]), "unknown")

if(length(Proportions\_est[[it]]) < num\_sources +1){

tmp = Proportions\_est[[it]]

Proportions\_est[[it]][num\_sources] = NA

Proportions\_est[[it]][num\_sources+1] = tmp[num\_sources]

}

print("Source mixing proportions")

print(Proportions\_est[[it]])

}

print(Proportions\_est)#原版仅可得到这个数据，可视化程度较差

#可视化过程，参考文涛脚本并修正若干bug

#输出计算结果

FEAST\_output = as.data.frame(Proportions\_est)

colnames(FEAST\_output) = paste("repeat\_",Ids,sep = "") #取Ids作为平行代号

head(FEAST\_output)

filename = paste(dir\_path,"Result/FEAST.csv",sep = "")

write.csv(FEAST\_output,filename,quote = F)

#简单出图(每个repeat一张)

library(RColorBrewer)

library(dplyr)

library(graphics)

head(FEAST\_output)

plotname = paste(dir\_path,"Result/FEAST\_repeat.pdf",sep = "")

pdf(file = plotname,width = 12,height = 12)

par(mfrow=c((length(unique(metadata$SampleType))%/%2 +2 ),2), mar=c(1,1,1,1))

# layouts = as.character(unique(metadata$SampleType))

for (i in 1:length(colnames(FEAST\_output))) {

labs <- paste0(row.names(FEAST\_output)," \n(", round(FEAST\_output[,i]/sum(FEAST\_output[,i])\*100,2), "%)")

pie(FEAST\_output[,i],labels=labs, init.angle=90,col = brewer.pal(nrow(FEAST\_output), "Paired"),#最多可用12种颜色梯度

border="black",main =colnames(FEAST\_output)[i] )

}

dev.off()

#简单出图（所有repeat求平均后出1张图）

head(FEAST\_output)

asx = as.data.frame(rowMeans(FEAST\_output))

asx = as.matrix(asx)

asx\_norm = t(t(asx)/colSums(asx)) #\* 100 # normalization to total 100

head(asx\_norm)

plotname = paste(dir\_path,"Result/FEAST\_mean.pdf",sep = "")

pdf(file = plotname,width = 6,height = 6)

labs <- paste0(row.names(asx\_norm)," \n(", round(asx\_norm[,1]/sum(asx\_norm[,1])\*100,2), "%)")

pie(asx\_norm[,1],labels=labs, init.angle=90,col = brewer.pal(nrow(FEAST\_output), "Paired"),#最多12个颜色梯度

border="black",main = "mean of source tracker")

dev.off()

# 五、实测体会

1.关于速度：听说FEAST比SourceTracker快300倍，实测20个样×2000个otu的数据表计算仅需10分钟（i5, 4G），确实快！

2.关于准确性：采用A、B两组性质类似的数据计算（每组均含20个平行，A=source，B=sink，详见data\_files/compare\_otu和compare\_meta），理论上source结果应接近100%，但实测约为70%~80%。是否准确就见仁见智了~

图。准确性对比