三元图

WuYiLei

2020/6/6

Table of Contents

[简介 1](#_Toc42551846)

[实例解读 1](#_Toc42551847)

[绘图实战 1](#_Toc42551848)

## 简介

三元图是可以显示三种不同元素或化合物的混合成分的特征图，由于第3维是线性的并且仅依赖于另外两维，因此可以在2D空间中表示具有三个自由度的坐标系。

在微生物多样性分析中，普通三元图不同的点代表不同的OTUs（或分类水平），点的大小代表平均丰度。不仅如此，还可以根据OTU在各微环境中的丰度数据进行统计检验后，得出各OTU分别在哪种微环境中显著富集，并据此在图中以不同颜色的点表示，此时三元图不仅表现出OTU或者物种的分布，还包含显著性统计结果。此时的三元特异富集图，包两两比较和韦恩图比较的结果，信息高度概括，并从多方面展示，非常值得使用。

## 实例解读

[扩增子图片解读7三元图](https://blog.csdn.net/woodcorpse/article/details/74858260)

我三年前写的东西已经过时了，讲找近3年内高水平文章的矢量图进行讲解和点评。

图片必须 矢量图，或你可以重绘类似的，才能编辑修改和排版。

要求2-5个实例，每个实例相关的所有文字都要收录中、英文，方便同行参考撰写。

## 绘图实战

### 数据处理

library(tidyverse)

## -- Attaching packages ---------------- tidyverse 1.3.0 --

## √ ggplot2 3.3.0 √ purrr 0.3.3  
## √ tibble 2.1.3 √ dplyr 0.8.5  
## √ tidyr 1.0.2 √ stringr 1.4.0  
## √ readr 1.3.1 √ forcats 0.5.0

## -- Conflicts ------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# 代码要有讲解

# 设定随机数种子，确保项目可重复

set.seed(13)

# 基于种子生成测试数据OTU表  
otu <- matrix(sample(c(0:1000), 1200, replace = TRUE),   
 ncol = 12, nrow = 100,   
 dimnames =  
 list(row\_names = paste0("OTU",seq(1:100)),  
 col\_names = paste0("sample",seq(1:12))))  
  
  
  
otu <- as.data.frame(otu)  
  
# 创建分组信息数据集  
group <- data.frame(variable = paste0("sample",seq(1:12)),  
 group = rep(c("Control", "Treat", "normal"),  
 each = 4))  
  
  
# 提取OTU信息或者分类水平名字  
otu$OTU <- rownames(otu)  
# 转换宽表格为长表格  
otu <- pivot\_longer(data = otu,   
 cols = -OTU,  
 names\_to = "variable",   
 values\_to = "value")  
  
  
# 按同类项进行合并  
merge\_data <- merge(otu, group, by = "variable")  
  
otu <- subset(merge\_data, select = -variable)  
head(otu)

## OTU value group  
## 1 OTU1 711 Control  
## 2 OTU92 938 Control  
## 3 OTU82 450 Control  
## 4 OTU72 295 Control  
## 5 OTU62 573 Control  
## 6 OTU99 921 Control

# 创建三元图作图数据集  
# 需要以group, OTU为分组信息，将value分割

# 每种中的重复如何处理？不求均值？  
otu %>%  
 group\_by(group, OTU) %>%  
 mutate(index = row\_number()) %>%  
 pivot\_wider(names\_from = group,   
 values\_from = value) %>%  
 select(-index) -> otu\_tern  
  
head(otu\_tern)

## # A tibble: 6 x 4  
## # Groups: OTU [6]  
## OTU Control normal Treat  
## <chr> <int> <int> <int>  
## 1 OTU1 711 130 849  
## 2 OTU92 938 987 522  
## 3 OTU82 450 393 793  
## 4 OTU72 295 396 732  
## 5 OTU62 573 579 286  
## 6 OTU99 921 691 429

# 用于定义图中点的大小，对 3 个样本的平均值取log   
otu\_tern$size <- log((apply(otu\_tern[2:4], 1, mean)))

### ggtern可视化

ggtern Nicholas Hamilton是开发的，用于创建三元图的ggplot2的扩展包，详细参数和用法见[官方说明文档](http://www.ggtern.com/)。

# installation  
# install.packages('ggtern')  
  
library(ggtern)

## --  
## Remember to cite, run citation(package = 'ggtern') for further info.  
## --

##   
## Attaching package: 'ggtern'

## The following objects are masked from 'package:ggplot2':  
##   
## aes, annotate, ggplot, ggplot\_build, ggplot\_gtable, ggplotGrob,  
## ggsave, layer\_data, theme\_bw, theme\_classic, theme\_dark,  
## theme\_gray, theme\_light, theme\_linedraw, theme\_minimal, theme\_void

p <- ggtern(data = otu\_tern,   
 aes(x = Control, y = Treat, z = normal)) +   
 geom\_mask() + # 可将超出边界的点正常显示出来  
 geom\_point(aes(color = OTU, size = size),   
 alpha = 0.8, show.legend = TRUE) +  
 scale\_size(range = c(0, 6)) +  
 # 去掉颜色legend  
 guides(colour = "none") +  
 theme\_bw() +  
 theme(axis.text = element\_blank(),   
 axis.ticks = element\_blank())  
p

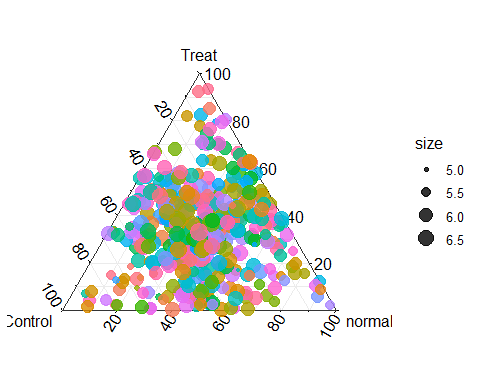


图. 图注，你想表达什么，为什么要用这个图。

### edgeR 显著性检验

# installation

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

if (!requireNamespace("edgeR", quietly = TRUE))

BiocManager::install("edgeR")  
  
library(edgeR)  
set.seed(13)  
otu <- matrix(sample(c(0:1000), 1200, replace = TRUE),   
 ncol = 12, nrow = 100,   
 dimnames =  
 list(row\_names = paste0("OTU",seq(1:100)),  
 col\_names = paste0("sample",seq(1:12))))  
  
  
group <- data.frame(variable = paste0("sample",seq(1:12)),  
 group = rep(c("Control", "Treat", "normal"),  
 each = 4))  
  
dge\_list <- DGEList(counts = otu, group = group$group)  
  
# 数据过滤  
# Remove the lower abundance/(cpm, rpkm)  
keep <- rowSums(dge\_list$counts) >= 0  
# 统计保留和排除的数量

table(keep)

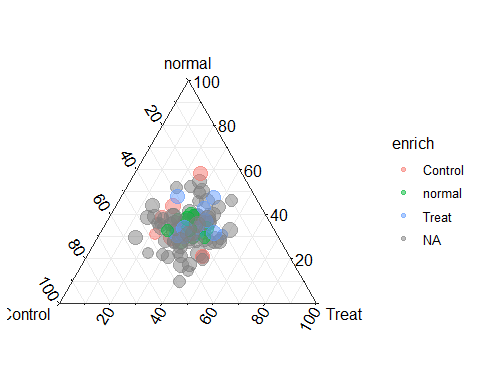
dge\_keep <- dge\_list[keep, ,keep.lib.sizes = FALSE]  
  
# scale the raw library sizes dgelist  
  
dge <- calcNormFactors(dge\_keep)  
  
# fit the GLM  
design.mat <- model.matrix(~ 0 + dge$samples$group)  
d2 <- estimateGLMCommonDisp(dge, design.mat)  
d2 <- estimateGLMTagwiseDisp(d2, design.mat)  
  
# glmFit()将值拟合到模型中  
fit <- glmFit(d2, design.mat)  
  
# 查看分组情况，主要关注各分组的排序位置  
fit$design  
  
  
# 检测第一分组相交于其他两个分组的显著性  
# null hypothesis: the coefficients between two groups are equal to zero.，解释1，-1，0代表什么  
lrt\_Control\_normal <- glmLRT(fit, contrast = c(1, -1, 0))  
  
lrt\_Control\_Treat <- glmLRT(fit, contrast = c(1, 0, -1))  
  
# 确认命名与实际比较组是否有差异  
test <- topTags(lrt\_Control\_Treat , n = nrow(dge\_list$counts))  
  
  
# 默认值判别是否显著富集：-1表示下调，1表示富集，0表示无差异  
de\_Control\_normal <- decideTestsDGE(lrt\_Control\_normal, adjust.method = "fdr",   
 p.value = 0.5)  
de\_Control\_Treat <- decideTestsDGE(lrt\_Control\_Treat, adjust.method = "fdr",   
 p.value = 0.5)  
  
# 查看?decideTestsDGE选用合适的阈值  
  
# 选出Control组中相对normal和Treat均显著富集的OTUs  
rich\_Control <- rownames(otu)[de\_Control\_normal == 1 & de\_Control\_Treat == 1]  
  
enrich\_Control <- data.frame(rownames = rich\_Control,   
 enrich = rep("Control", length(rich\_Control)))  
  
# 分别对三个分组进行显著性分析并合并最终数据  
  
# enrich\_index <- rbind(enrich\_Control, enrich\_normal, enrich\_Treat)

### 根据富集OTUs索引，回到otu表中标记相应数据

# 示例富集索引  
enrich\_index <- data.frame(OTU = paste0("OTU", sample(seq(1:100),30, replace = FALSE)),  
 enrich = sample(c("Control", "normal", "Treat" ),30, replace = TRUE) )  
  
# 生成示例数据  
set.seed(13)  
otu <- matrix(sample(c(0:1000), 1200, replace = TRUE),   
 ncol = 12, nrow = 100,   
 dimnames =  
 list(row\_names = paste0("OTU",seq(1:100)),  
 col\_names = paste0("sample",seq(1:12))))  
  
data <- as.data.frame(otu)  
data[is.na(data)] <- "unknow"  
data$OTU <- rownames(data)  
  
otu <- pivot\_longer(data = data,   
 cols = -OTU,  
 names\_to = "variable",   
 values\_to = "value")  
  
# 合并数据  
otu <- merge(otu, group, by = "variable")  
otu <- select(otu, select = - "variable")  
  
head(otu)

## OTU value group  
## 1 OTU1 711 Control  
## 2 OTU92 938 Control  
## 3 OTU82 450 Control  
## 4 OTU72 295 Control  
## 5 OTU62 573 Control  
## 6 OTU99 921 Control

otu %>%  
 group\_by(group, OTU) %>%  
 mutate(index = row\_number()) %>%  
 pivot\_wider(names\_from = group,   
 values\_from = value) %>%  
 select(-index) -> otu\_tern  
  
# 合并相同的OTUs  
na.omit(otu\_tern) %>% group\_by(OTU) %>%   
 summarise\_all(sum)-> data\_all  
  
# 根据富集索引，合并富集信息  
plot\_data <- merge(data\_all, enrich\_index, by = "OTU", all = TRUE)  
  
# 点大小，这里取 3 个样本的平均值的 0.5 次方  
plot\_data$size <- (apply(plot\_data[, c(2:4)], 1,   
 mean))^0.5  
  
  
ggtern(data = plot\_data,   
 aes(x = Control, y = normal, z = Treat)) +   
 geom\_mask() + # 可将超出边界的点正常显示出来  
 geom\_point(aes(size = size, color = enrich),  
 alpha = 0.5) +   
 guides(size = "none") +  
 theme\_bw() +  
 theme(axis.text = element\_blank(),   
 axis.ticks = element\_blank())



```