距离计算

非约束排序–各种距离计算

R版本与运行环境信息

```R
Date:2021-3-24
R version 4.0.3 (2020-10-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 18363)
```

```R
#工作路径设定
setwd("path")
rm(list = ls())
```

数据读入

载入相关包，读入数据

```R
library("vegan")
otu <- t(read.csv("otutable.csv",header = T,row.names = 1))
```

相似度计算

Jaccard相似性/相异度

将两个样方中**共享的数量除以两个样方中所有的物种和**所计算出的距离，其中，a代表两个样方中共享的物种数量，Jaccard距离只考虑有无，所以要对数据进行二元转化。

$$Jaccard-distabce=\frac{a}{a+b+c+...+n}$$

PS:相似度和距离的转化公式D:距离，S:相似度

$$D=1-S$$

$$D = \sqrt{1 - S}$$

$$D = \sqrt{(1 - S^2)}$$

使用 `vegdist()` 计算距离

```R
#binary是否进行二元转换
Jaccard_d <- vegdist(otu,method = "jaccard",binary = T)
```

Sørensen相似性/相异度

Jaccard相似，将共有物种数量扩大两倍，其余一样，同样要进行二元转换,**应用少**，a代表两个样方中共享的物种数量，要对数据进行二元转化

$$Sørensen - distabce = \frac{2a}{2a + b + c}$$

Simpson相似性/相异度

在两个样方的物种丰富度指数差异很大的情况下（即一个样方比另一个样方具有更多的物种），Simpson相似性指数通过从分数某一不公有物种中仅取较小的数据来消除这个问题，**与Simpson指数不一样**

$$Simpson - distabce = \frac{a}{a + min(b, c)}$$

数据转换

线性转换:对原始数据中的每个值加常数或乘以常数，通常不会改变统计检验的结果；
非线性转换，如对数转换，平方根转换等，转换后的统计检验结果与未转换的变量的统计检验结果会有不同，但是数据间的差异不会改变。

归一化

```R
#数据转换
#以理化数据为例
setwd("G:\\Desktop\\s_note\\data\\16s\\beta_diversity\\PCA")
data <- as_tibble(read.csv("LH.csv",header = T,row.names = 1))
#对数转换
data1 <- log(data+1,10)
data1 <- log2(data+1)
#ln转换
data2 <- log(data+1,exp(1))
#同乘同除取倒数
data3 <- data*2
data3 <- data/2
data3 <- 1/data
data3 <- data^0.5
```

标准化

利用数据本身计算的统计量来更改数据，常见方法有，

（1）中心化（原始数据减去均值）

（2）标准化：转换后的数据均值为0，方差为1

（3）z-scores标准化：先中心化后再除以标准差

……

vegan包中使用 **`decostand()`** 函数对数据进行标准化，其中 **`MARGIN`** 参数是选择对行或对列标准化，是一个**重要参数**** *（1：行；2列）*，R自带的 **`scale()`** 函数默认对列标准化，默认数据格式为：**行为处理，列为物种（或变量）**

```R
data <- t(data)
#行为处理，列为物种(或变量)
> head(data)
                  CK_1       CK_2     CK_3       CK_4       CK_5       CK_
6      CK_7       CK_8       T1_1       T1_2      T1_3
Temperature  41.000000   36.00000  42.0000  41.000000  54.000000  40.00000
0  58.00000  52.000000  74.000000  68.000000  74.00000
pH            6.476667    6.13000   6.1400   6.253333   6.913333   6.70333
3   6.82000   6.833333   7.573333   6.446667   7.16000
TOC         415.000000 1161.33333 428.3333 212.803333  80.973333 107.62666
7  71.83333 191.806667 152.840000 104.380000  92.60667
IC          167.333333   51.65000 112.0967 140.666667 169.806667 187.58000
0 184.60667 176.106667 165.380000 178.860000 162.60000
TC          582.333333 1213.000 00 540.6667 354.000000 250.666667 295.3333
33 256.66667 368.000000 318.000000 283.333333 255.33333
TN           81.840000   88.23333  62.4600  55.636667  56.373333  69.10666
7  59.32000  62.773333  65.520000  65.386667  75.32000

data4 <- decostand(data,method = "total")
data5 <- decostand(data,method = "hellinger")
data6 <- decostand(data,method = "max",MARGIN = 1)
data7 <- decostand(data,method = "chi.square")
data9 <- scale(t(data),center = T,scale = T)
data10[data>0] <- 1
#total：除以行或列的总和，默认MARGIN=1
#max：除以行或列的最大值，MARGIN=2
#frep:除以行或列的最大值,同时乘以非零值得个数，MARGIN=2
#normalize: 行或列的平方和为1，MARGIN=1
#range:标准化使行列的值在0，1之间，MARGIN=2
#standardize:行或列的和为1且方差为1，MARGIN=2
#pa:0-1转换
#chi.square:除以行和和列和的平方根
#hellinger:total法后在取平方根
#log:对数转换，默认自然对数
```

距离计算

欧几里得距离

欧几里得距离没有上限，最大值取决于数据，但是其对物种的丰度更加敏感，而对物种是否存在则不敏感，**容易受到双零影响**

在处理物种数据时，会尽可能避开使用欧几里得距离这类的对称指数，或者要提前对数据进行转化如，

弦转换和**Hellinger转换**来计算弦距离、Hellinger距离、卡方距离等，降低丰度的影响
所得到的距离应该进行归一化处理

$$Dnorm = D/D_{max}$$

$$Dnorm = (D - D_{min})/(D_{max} - D_{min})$$

双零问题
根据欧几里得距离，我们得到了这样的结果：样方1和2具有比样方1和3更高的群落相似度。这显然是很难让人接受的，毕竟样方1和2没有共享任何的物种，而样方1和3共享所有物种，仅仅是丰度相差较大而已。

```R
#欧几里得距离的计算
Euclidean_distance <- vegdist(otu,method = "euclidean")
#标准化
#方法1
Euclidean_distance <- Euclidean_distance/max(Euclidean_distance)
#方法2
Euclidean_distance <- (Euclidean_distance-min(Euclidean_distance))/(max(Euclidean_distance)-min(Euclidean_distance))
```

弦距离计算

```R
#弦转换
chord_tr <- decostand(otu,method = "normalize")
#弦距离计算
chord_dis <- vegdist(chord_tr,method = "euclidean")
```

Hellinger距离

```R
#Hellinger转换
helli_tr <- decostand(otu,method = "hellinger")
#Hellinger距离计算
helli_dis <- vegdist(helli_tr,method = "euclidean")
```

Bray-curtis 距离

最常见的群落间距离的计算方法

计算方法为：物种和物种的差除以两个物种的和，有效的避免了双零的问题，可以直接通过1-D获得相似性

```R
bray <- vegdist(otu,method = "bray")
```

Unifrac距离

在Unifrac距离中，除了关注考虑了物种的存在与否及其丰度外，**还将物种之间的进化关系考虑在内，距离0更侧重于表示两个群落的进化分类完全一致。**分为加权和非加权，前者适用于群落变化较小的状态，后者则适用于群落变化较大的情况

构建发育树

首先使用 `usearch10` 对代表序列进行系统发育树的构建，得到系统发育树文件 `otus.tree` ，同样可以使用其他软件构建发育树，如 `mafft` 和 `mega` 等

```Shell
usearch -cluster_agg otu.fa -treeout otus.tree
```

读入系统发育树与数据文件

```R
setwd("G:\\Desktop\\s_note\\data\\16s\\beta_diversity\\Distance\\unifrac")
library("ape")
library("phyloseq")
#读入OTU表和分组信息
otu <- read.csv("otu_quh.csv",header = T,row.names = 1,encoding = "UTF-8")
group <- read.csv("group.csv",header = T)
names(group) <- "Tr"
#读入发育树并构建phyloseq对象,保证otu表的otu名称与发育树相对应且前者需后者包含即可
t_u10 <- read.tree("otus.tree")
phy_u10 <- phyloseq(otu_table(otu,taxa_are_rows = T),phy_tree(t_u10))
```

计算unifrac距离

使用 **`distance()`** 函数计算unifrac距离， `method` 选项指定加权或非加权unifrac距离

```R
#wunifrac:计算物种相对丰度加权后的unifrac距离
#unifrac：原始数据直接计算unifrac距离
uni_u10 <- as.matrix(distance(phy_u10,method = "wunifrac"))
```

usearch直接beta多样性分析

```Shell
#usearch上构建otus.tree, 用于Unifrac
usearch -cluster_agg otu.fa -treeout otus.tree
#生成距离矩阵：
usearch -beta_div otutab_analysis.txt -tree otus.tree
```

参考资料: 小白鱼的生统笔记