R统计分析-非度量多维尺度分析（NMDS）及作图方法示例

此处结合微生物群落研究中的16S扩增子分析数据，给大家分享怎样在R中进行非度量多维尺度分析（NMDS），顺便使用此处的NMDS排序结果，给大家展示怎样结合ggplot2绘制NMDS排序图。

本文示例使用生态统计分析中常用的“vegan”包，进行NMDS排序分析。

首先介绍示例数据。我们此处共有40个16S测序样本，均来自土壤。因试验需求，在土壤中添加了某化学物质，目的为探究该化学物质对土壤微生物群落的影响。这40个样本中，20个为不添加化学物质的对照组（Control组），20个为添加化学物质的处理组（Treat组）。

此处我们希望通过NMDS分析，查看该化学物质是否显著影响了土壤细菌群落的组成。

作图示例文件、R脚本等，已上传至百度盘，无提取码

https://pan.baidu.com/s/15BFnBrLXbSrqG2s5GX2i-A

## 示例文件简要

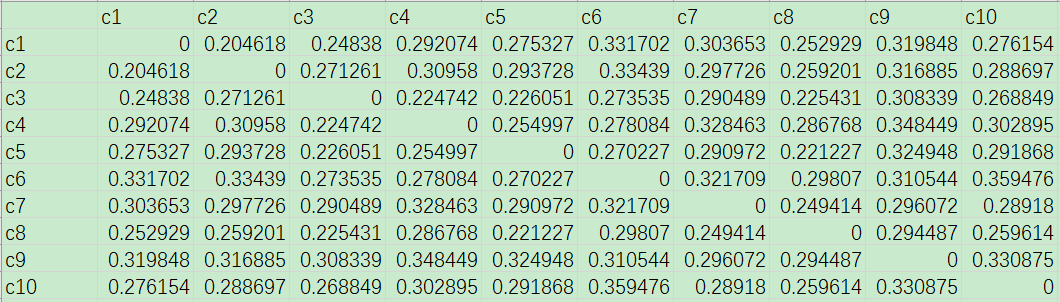
文件“otu\_table.txt”为OTU丰度表格，其内容展示如下。

每一列为一个样本，每一行为一种OTU，交叉区域为每种OTU在各样本中的丰度。

![C:\Users\lyao222lll\AppData\Roaming\Tencent\Users\504247669\QQ\WinTemp\RichOle\26FRD(AMNYA6O60N4](PP}6.png](data:image/png;base64,)

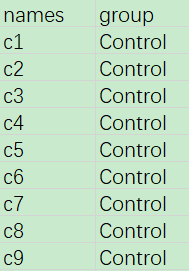
文件“bray.txt”为提前计算得到的样本距离矩阵文件（此处展示的是样本间Bray-curtis距离），其内容展示如下。

每一列为一个样本，每一行为一个样本，交叉区域为样本间的Bray-curtis距离（取值范围0-1，越接近于1表明样本间细菌群落组成差异越大）。



文件“group.txt”为样本分组信息，其内容展示如下。

第一列（names）为各样本名称；第二列（group）为各样本的分组信息，即这些样本分别属于未添加化学物质的对照组（Control组）或添加了化学物质的处理组（Treat组）。



## 基于OTU（物种）丰度表，使用vegan包进行NMDS排序分析

首先加载vegan包，并导入OTU丰度表格文件，之后使用vegan包中的metaMDS()命令进行NMDS排序。metaMDS()的详细参数可使用? metaMDS()查看。

library(vegan)

#读入 OTU 丰度表

otu <- read.delim('otu\_table.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

otu <- data.frame(t(otu))

#排序（显示 4 个排序轴）

nmds1 <- metaMDS(otu, distance = 'bray', k = 4)

读入OTU表格，赋值给新数据框“otu”，并进行转置（计算要求，每一行为一个样本，每一列为物种信息）。distance = 'bray'，基于生态统计中常用的Bray-curtis距离进行排序；k = 4，预先设定4个轴进行排序（缺省时默认为k = 2，使用2个排序轴）。虽然在本示例中设定k = 4显得多余，但为了方便在演示中加深理解，就进行了此设置。

如果预先设定的排序轴数量较少（如k = 2），那么在相同轴数的条件下，NMDS往往能够获得比PCoA更少失真的对象之间的关系。但NMDS计算需要不断迭代，因此在大样本的情况下其对计算资源的要求较高。若数据量不是很大，则很快便可以得到结果。

我们可简要查看排序结果。

#可简要查看结果

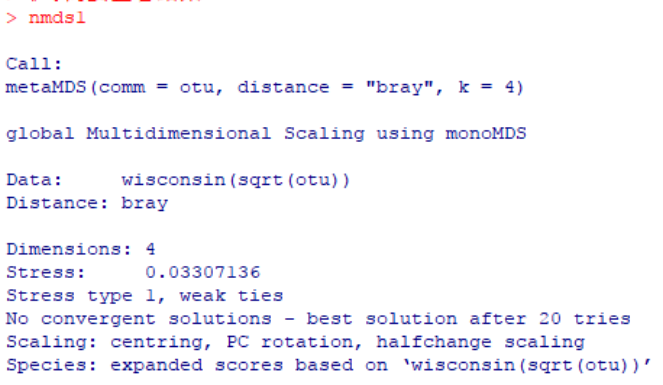
nmds1

#或

summary(nmds1)

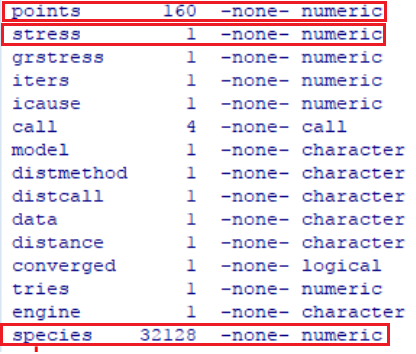
打印在屏幕中的排序结果简要，包含了运行命令、排序基于的距离类型（Distance: bray）、预设排序轴数量（Dimensions: 4）、NMDS的应力函数值（stress，此处约为0.033）等信息。

对于应力函数值，此处有必要提及。在PCA、PCoA等排序分析中，对于每个排序轴，均可计算得到其对应的解释量，一般情况下我们可以根据解释量来评估排序结果是否适用；而对于NMDS分析，它的排序轴不存在解释量一说，但可以计算得到一个总的应力函数值，因此我们需要参考应力函数值来对排序结果进行评估。在PCA、PCoA等基于坐标轴解释量的评估中，解释量越高越好；而在NMDS排序分析中，尽可能选择较低的应力函数值。一般情况下，应力函数值的值不要大于0.2。



使用summary()命令可以查看更多细节，我们可主要关注“stress”、“point”、“species”等结果。

其中，stress记录了NMDS排序分析的应力函数值，points记录了各样本的排序坐标，species记录了各物种（OTU）的排序坐标。



查看应力函数值，以及可考虑将样本排序坐标或物种（OTU）排序坐标数据提取出，转化为数据框的样式并输出保存。

#提取应力函数值（stress）

nmds1.stress <- nmds1$stress

#提取样本排序坐标

nmds1.point <- data.frame(nmds1$point)

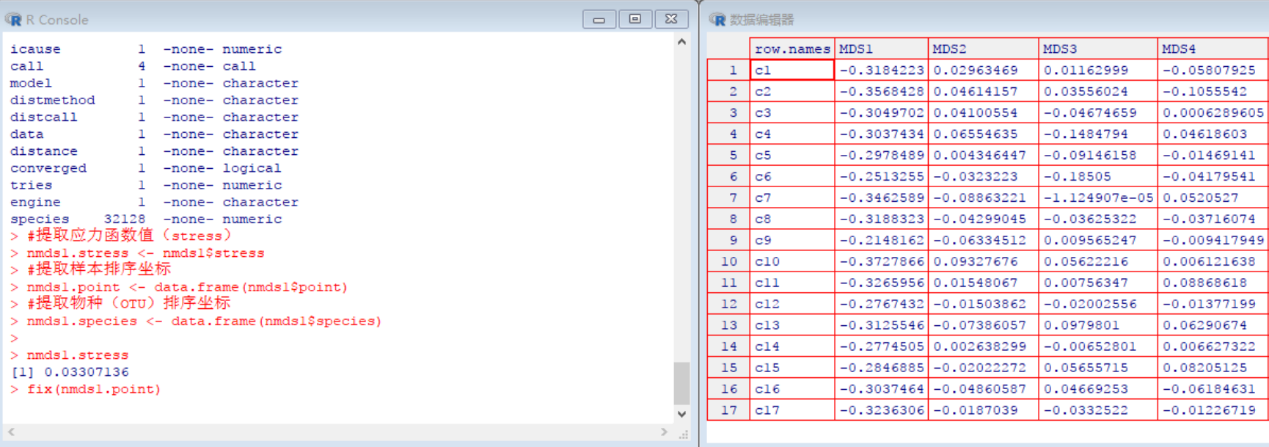
#提取物种（OTU）排序坐标

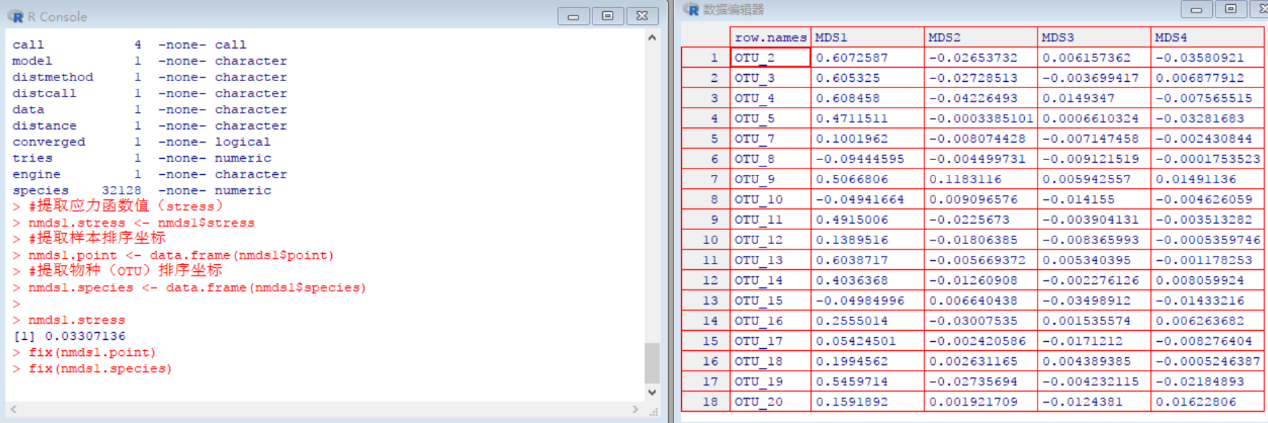
nmds1.species <- data.frame(nmds1$species)

#可选择将结果输出并保存在本地，例如将样本坐标输出为 csv 格式

write.csv(nmds1.point, 'nmds.sample.csv')

下图分别为样本排序坐标结果（nmds1.point）以及OTU排序坐标结果（nmds1.species），展示前4个排序轴。





我们可简要绘制NMDS排序图查看样本菌群组成差异。

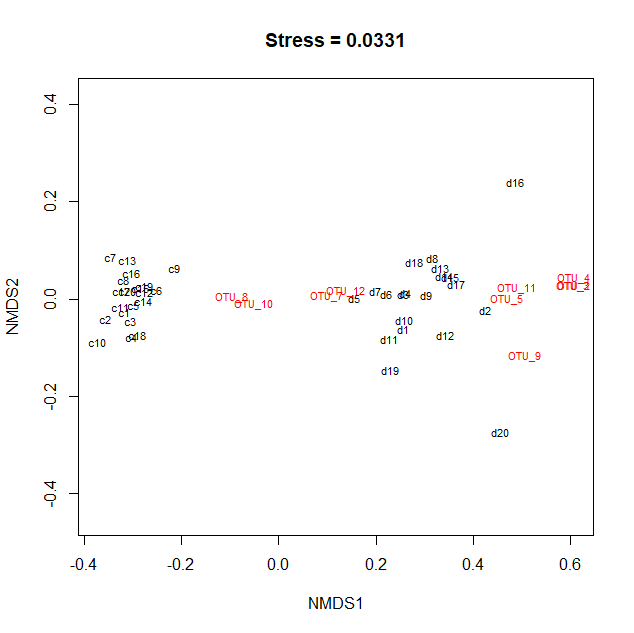
#简要绘图展示

nmds\_plot <- nmds1

nmds\_plot$species <- {nmds\_plot$species}[1:10, ]

plot(nmds\_plot, type = 't', main = paste('Stress =', round(nmds1$stress, 4)))

计算结果nmds1中，既包含了各样本的坐标，又包含了各物种（OTU）的坐标。由于OTU种类繁多（本示例中包含数千种OTU），直接绘制点图将难以观测结果，因此我们在作图时只选取其中的少部分OTU排序结果，和样本排序结果一起进行绘制。绘图时将各样本点以及OTU直接以名称展示，结果如下。



## 基于现有的距离矩阵，使用vegan包进行NMDS排序分析

导入现有的距离矩阵，之后使用vegan包中的metaMDS()命令进行NMDS排序。metaMDS()的详细参数可使用? metaMDS()查看。

#读入现有的距离矩阵

dis <- read.delim('bray.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

#排序

nmds2 <- metaMDS(as.dist(dis), k = 4)

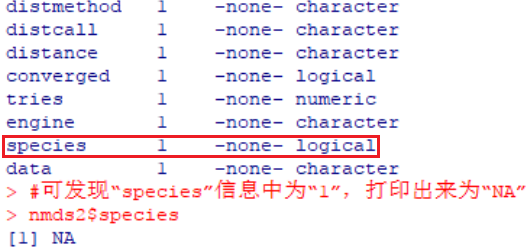
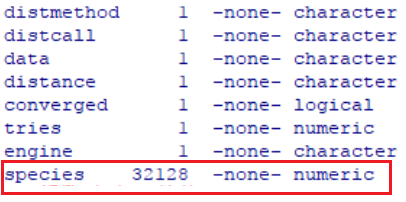
与上述使用OTU丰度表的运行命令一致，但此处放在命令中的dis为读入的距离矩阵，而非之前的OTU丰度表；k = 4，预先设定4个轴进行排序（缺省时默认为k = 2，使用2个排序轴）。

其实到这里我们可观察到一个重要的信息：在以距离矩阵作为输入文件的NMDS排序分析中，仅包含了样本距离，缺少了物种（OTU）种类及组成信息，预示着最终结果中将不包含OTU排序坐标；而上述以OTU丰度表作为输入文件的排序结果中，是包含了物种（OTU）坐标在内的。因此，两种方式的排序结果是否存在差异呢？

#可简要查看结果

summary(nmds2)

使用summary()查看结果后，可发现“species”信息中为“1”，打印出来为“NA”，即缺失了物种（OTU）排序坐标；而上述使用OTU丰度表作为输入文件的结果中，此项结果为“32128”，是包含物种（OTU）排序坐标在内的，我们此前已经顺利地提取出。

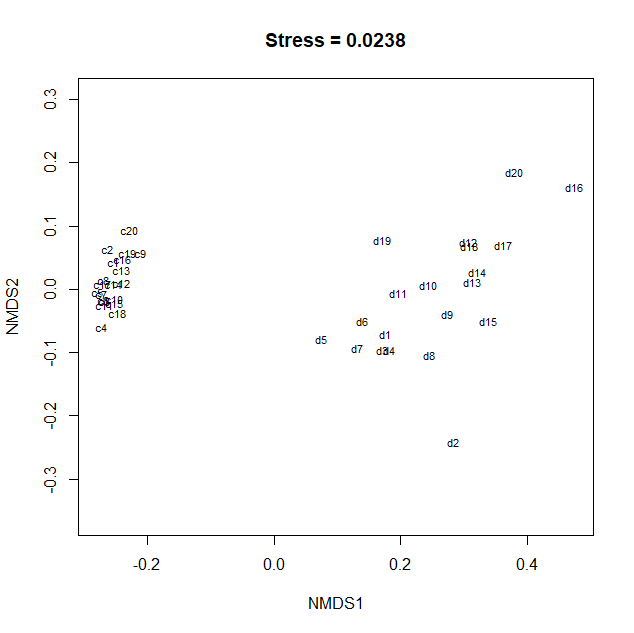
 

同样地，我们可简要绘制NMDS排序图查看样本菌群组成差异。由于本次的排序结果中不包含物种（OTU）坐标（即结果中仅包含样本坐标），因此我们可直接对结果进行绘图。

#简要绘图展示，可发现与基于 OTU 丰度表的 NMDS 排序结果不同

plot(nmds2, type = 't', main = paste('Stress =', round(nmds2$stress, 4)))

结果如下，可发现与上述使用OTU丰度表作为输入文件的NMDS排序结果存在明显不同。



## 论以OTU（物种）丰度表/距离矩阵作为输入文件的NMDS排序分析的差异

这里，我们通过两种不同的NMDS排序过程，发现了排序结果存在差异。而这种差异在前述PCoA分析示例中是不存在的，可参考前文（<http://blog.sciencenet.cn/home.php?mod=space&uid=3406804&do=blog&id=1158563>），无论使用OTU丰度表作为输入文件还是现有的距离矩阵作为输入文件，PCoA排序的结果是完全一致的。

此处NMDS排序时所考虑的条件不同导致结果存在明显差异。一种是输入OTU丰度表，在排序时既考虑了样本间距离，也考虑了样本距离以及物种（OTU）在各样本中的分布；另一种则是直接基于样本间距离进行排序，不考虑物种（OTU）在各样本中的分布。因此在排序过程中，第一种方法中考虑的“对象”（样本+物种）比第二种方法中考虑的“对象”（仅样本）实际上要多得多，能够找到更多的潜在信息。

对于使用距离矩阵作为输入文件的排序结果，虽然不包含物种坐标，但我们也可以选择在后续将各物种（OTU）添加至NMDS排序结果中。这和上一篇博文中所讲述的PCoA排序的方法一致，可使用vegan包内置命令wascores()将OTU“被动地”加入NMDS排序图。

#可使用 wascores() 将 OTU“被动地”加入 NMDS 排序图

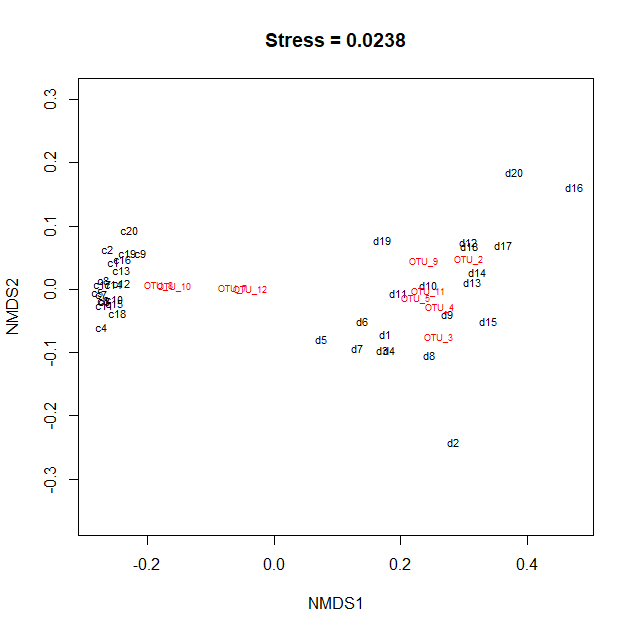
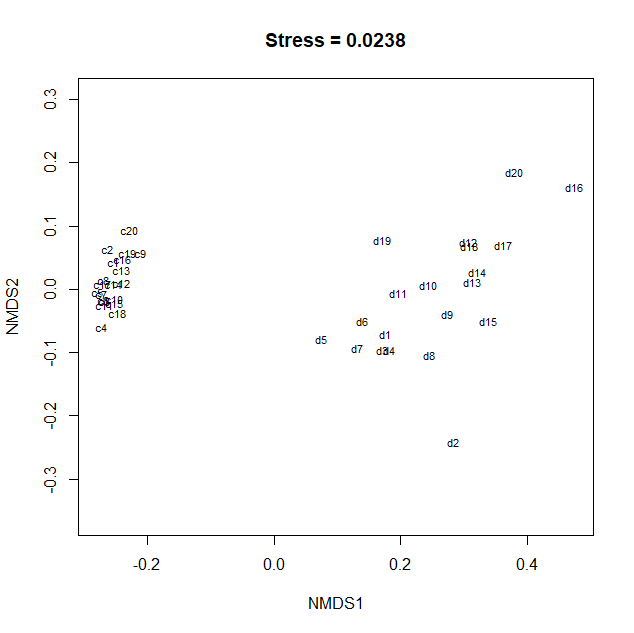
plot(nmds2, type = 't', main = paste('Stress =', round(nmds2$stress, 4)))

species <- wascores(nmds2$points, otu)

text(species[1:10, ], rownames(species)[1:10], col = 'red')

因OTU种类很多，我们在作图时只展示其中的少部分OTU。

下图左，原始NMDS排序图；下图右，被动加入OTU坐标后的NMDS排序图。



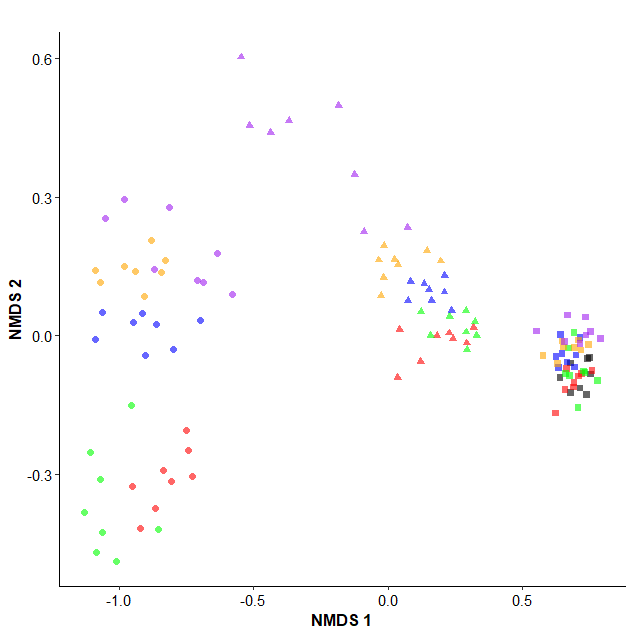
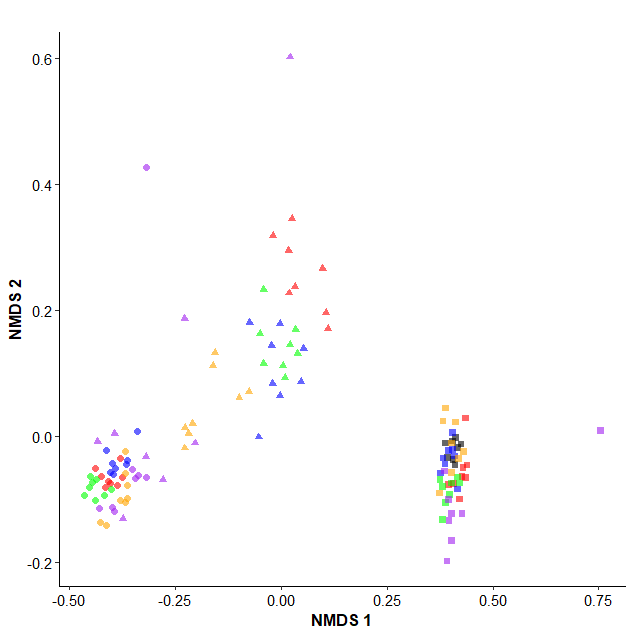
其实在这里我们可以看出，即使后续将OTU信息重新考虑在内，也仅仅是将OTU投影到已有的排序结果中（即安排OTU“回插”），并没有对原始的NMDS排序结果产生影响。因此，最终的NMDS排序结果与使用OTU丰度表作为输入文件的NMDS排序结果还是具有很大的差异。

以下内容仅为个人经验见解，仅供参考。

本人认为，使用OTU丰度表作为输入文件的NMDS排序分析能够更充分地进行信息挖掘，因为它在计算“对象位置”时考虑因素更充分，更具可靠性。

此处举个例子吧，使用我研究生期间所做课题中的真实数据说明。下图左图为使用OTU丰度表作为输入文件的NMDS排序结果，右图为使用现有的距离矩阵作为输入文件的NMDS排序结果，两者计算均基于“Bray-curtis”距离。图中不同颜色和不同形状的点，分别表示不同的样本分组。

可以看到，左图较于右图更能明显区分不同分组。

## NMDS评估

可以通过比较NMDS排序图内对象的距离与原始对象距离去评估NMDS结果（Shepard图）。若R2越大，则NMDS结果越合理。

此外，也可以直接使用排序图内对象的距离与原始距离进行线性或非线性回归的R2来评估NMDS的拟合度。可通过vegan包中的goodness()命令实现。使用气泡图在排序图中展示每个样本的拟合度，拟合度越差的点，气泡越大。

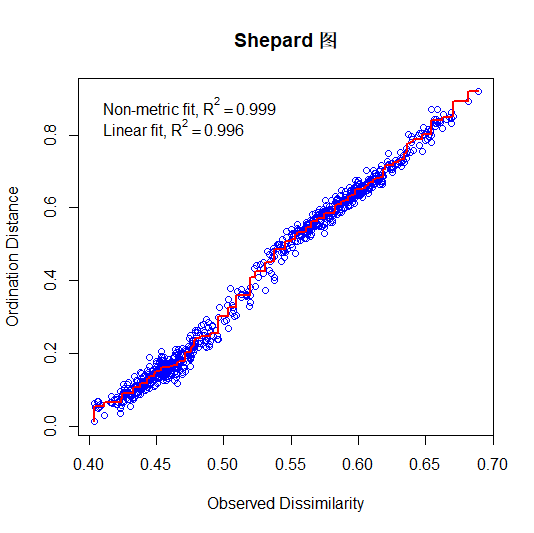
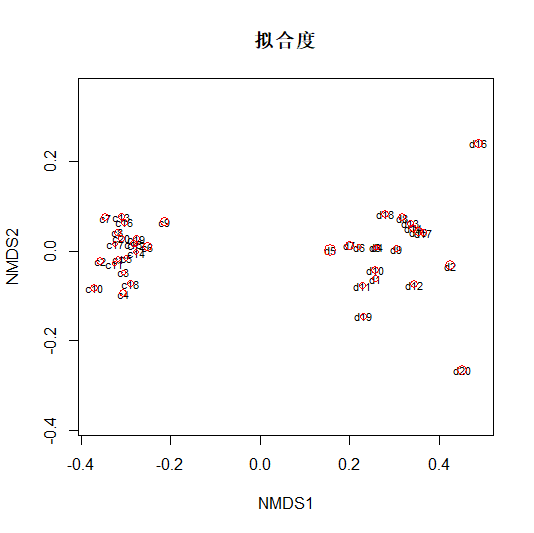
本示例使用OTU丰度表作为输入文件的NMDS排序结果进行评估，即上文中所得排序结果“nmds\_plot”。

stressplot(nmds\_plot, main = 'Shepard 图')

gof <- goodness(nmds\_plot)

plot(nmds\_plot,type = 't', main = '拟合度')

points(nmds\_plot, display = 'sites', cex = gof \* 200, col = 'red')

## 使用ggplot2包进行NMDS作图

此处给大家展示如何使用ggplot2基于已经计算得到的NMDS排序结果进行作图。

此处使用OTU丰度表作为输入文件参与排序，并且对于NMDS排序我们只展示其两个轴，因此排序时预设排序轴数量k = 2。同时我们考虑将物种（OTU）排序结果也展示出来，由于参与排序的物种（OTU）种类过多不便全部展示，因此我们在最终的绘图结果中只展示相对丰度前10的OTU。

结合试验设置说明，对于两种试验处理类型（对照组，Control；处理组，Treat），在作图结果中分别以不同颜色和不同形状的点表示出；对于主要的OTU，在图中对应的坐标位置直接展示其名称。

##ggplot2 作图（使用基于 OTU 丰度表的 NMDS 排序结果，预设 2 个排序轴）

#读入 OTU 丰度表

otu <- read.delim('otu\_table.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

otu <- data.frame(t(otu))

#读入样本分组文件

group <- read.delim('group.txt', sep = '\t', stringsAsFactors = FALSE)

#排序，预设 2 个排序轴

nmds1 <- metaMDS(otu, distance = 'bray', k = 2)

#提取样本点坐标（前两轴）

sample\_site <- nmds1.point[1:2]

sample\_site$names <- rownames(sample\_site)

names(sample\_site)[1:2] <- c('NMDS1', 'NMDS2')

#为样本点坐标添加分组信息

sample\_site <- merge(sample\_site, group, by = 'names', all.x = TRUE)

#提取相对丰度 top20 的 OTU 坐标（前两轴）

otu <- read.delim('otu\_table.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

otu$sum <- rowSums(otu)

otu <- otu[order(otu$sum, decreasing = TRUE), ]

species\_site <-{nmds1.species[rownames(otu[1:10, ]), ]}[1:2]

#整理为与 sample\_site 相同的样式，方便被 ggplot2 识别

species\_site$group <- rownames(species\_site)

names(species\_site)[1:2] <- c('NMDS1', 'NMDS2')

#使用 ggplot2 绘制 NMDS 排序图

nmds\_plot <- ggplot(sample\_site, aes(NMDS1, NMDS2, group = group)) +

geom\_point(aes(color = group, shape = group), size = 1.5, alpha = 0.8) + #可在这里修改点的透明度、大小

scale\_shape\_manual(values = c(17, 16)) + #可在这里修改点的形状

scale\_color\_manual(values = c('red', 'blue')) + #可在这里修改点的颜色

theme(panel.grid = element\_blank(), panel.background = element\_rect(color = 'black', fill = 'transparent')) + #去掉背景框

theme(legend.key = element\_rect(fill = 'transparent'), legend.title = element\_blank()) + #去掉图例标题及标签背景

labs(x = 'NMDS axis1', y = 'NMDS axis2', title = paste('Stress =', round(nmds1$stress, 4))) +

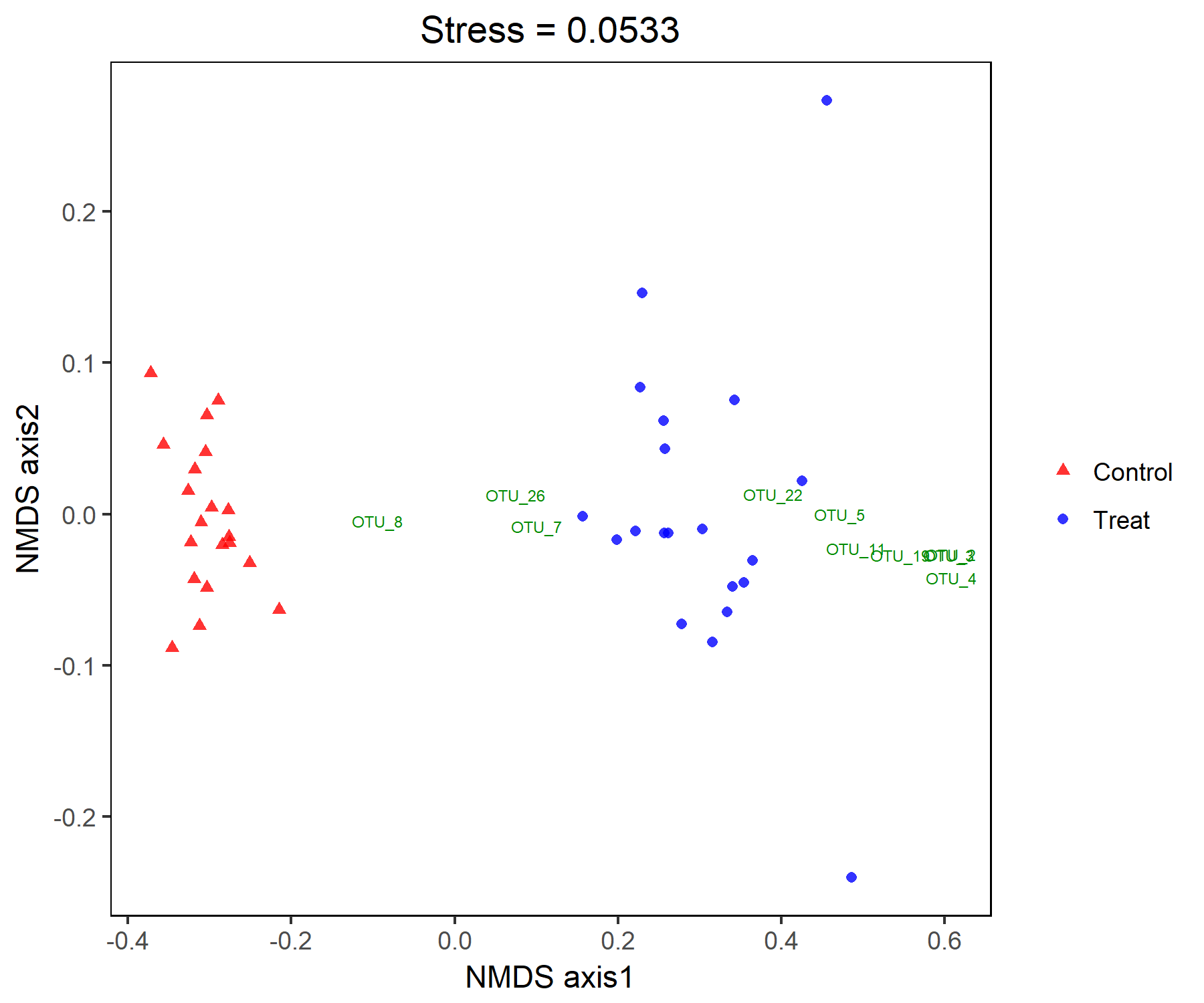
theme(plot.title = element\_text(hjust = 0.5)) + #标题居中

geom\_text(aes(label = group), data = species\_site, color = 'green4', size = 2) #添加物种排序（top10 OTU，展示为标签）

ggsave('NMDS.png', nmds\_plot, width = 6, height = 5)

ggplot2各作图参数不再详细说明，最终结果如下。

排序结果中，对照组（Control）和处理组（Treat）土壤中的细菌群落组成具有显著不同，在排序图中明显区分；经过添加某种化学物质处理后，相关的细菌类群发生显著的富集并占据群落的主要地位，因此对于相对丰度top10的OTU来讲，它们在处理组（Treat）土壤中的相对丰度显著增大。



## 寻求帮助（已解决）

之前一直没能找到解决方案的问题（可能好多同学也有这种疑问）：

就是以上内容均在R中进行，所使用的距离均为“Bray-curtis”距离，这个距离可直接通过现有的命令计算得到。在微生物扩增子分析中，(un)Weighted UniFrac距离（以下简称为UniFrac）也被广泛使用，而这个距离无法使用vegan包计算得到（通常来自UniFrac 软件计算所得）。因此，这里无法将metaMDS()命令写为“metaMDS(otu, distance = 'unifrac')”之类的形式，似乎只能将现有的样本间UniFrac距离文件导入，然后基于样本距离直接排序，或者在后续被动地添加物种投影等，这样排序时就会“丢失”大量的物种信息。如上所说，有时候考虑物种信息在内，结果或许会更佳。

由于本人确实也没理解明白UniFrac距离的计算公式，自然也无法做到自编计算命令。所以在这里求教各位大神们，在R中是否存在某个R包（或者是某大神写出来了计算命令），可以计算UniFrac距离呢？这样就可以方便对vegan包中的计算命令进行修改，解决这种困惑了。

这里感谢贺江舟老师在评论处给出了解决方案，感激不尽（实在太给力了，博文写出来也就1天时间就看到评论出有提到的解决方法了）。以下内容为追加内容。

然后我尝试了GUniFra包，使用它来计算UniFrac距离，同时修改了vegan包中3个命令，metaMDS()、metaMDSdist()、vegdist()，让它们可以引入UniFrac距离的计算。修改后的命令可见网盘附件“UniFrac/unifrac\_nmds.r”（3个修改后的命令分别命名为metaMDS2()、metaMDSdist2()、vegdist2()）。

然后这里加载GUniFra包，使用修改后的排序命令，使用OTU丰度表以及进化树文件作为原始输入文件，基于加权UniFrac距离（Weighted UniFrac）进行的NMDS排序分析。同时作为比较，还进行了直接以加权UniFrac距离（Weighted UniFrac）矩阵作为输入文件的排序。

注：以下命令的进行，需要使用ape包中的read.tree()读入进化树文件；GUniFra包中的GUniFrac()计算UniFrac距离，对于GUniFra包，可点击“<https://cran.r-project.org/web/packages/GUniFrac/GUniFrac.pdf>”查看说明；所使用的进化树必须为有根树，无根树无法用于计算。

##基于 Weighted UniFrac 距离的 NMDS 排序（测试）

#加载函数

source('unifrac\_nmds.r')

#OTU 丰度表

otu <- read.delim('test\_otu.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

otu <- data.frame(t(otu))

#进化树（使用 ape 包中的 read.tree() 读取，此处进化树必须为有根树）

otu\_tree <- read.tree('test\_tree.tre')

#距离矩阵（weighted unifrac 距离矩阵，使用 GUniFrac 包中的命令计算）

unifrac <- GUniFrac(otu, otu\_tree)

unifrac <- unifrac$unifracs

dis <- as.dist(unifrac[, , 'd\_1']) # Weighted UniFrac

#分组文件

group <- read.delim('test\_group.txt', sep = '\t', stringsAsFactors = FALSE)

#NMDS 排序（基于 OTU 丰度表和进化树文件，使用 weighted unifrac 距离）

nmds\_un1 <- metaMDS2(otu, distance = 'dw', tree = otu\_tree)

#NMDS 排序（直接基于 weighted unifrac 距离矩阵）

nmds\_un2 <- metaMDS(as.dist(dis))

#分别作图展示

sample\_site1 <- data.frame(nmds\_un1$point); sample\_site2 <- data.frame(nmds\_un2$point)

sample\_site1$names <- rownames(sample\_site1); sample\_site2$names <- rownames(sample\_site2)

names(sample\_site1)[1:2] <- c('NMDS1', 'NMDS2'); names(sample\_site2)[1:2] <- c('NMDS1', 'NMDS2')

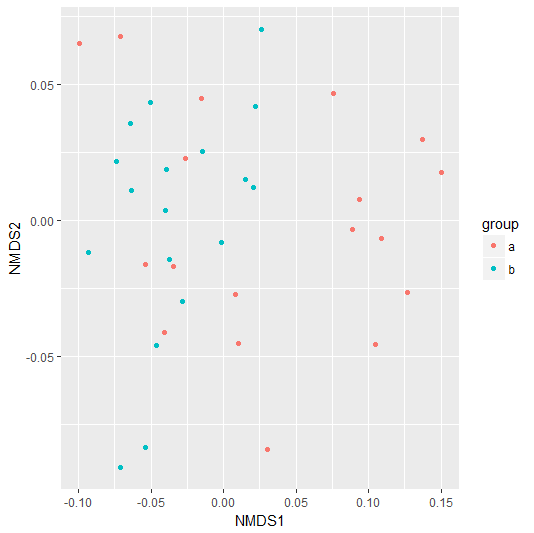
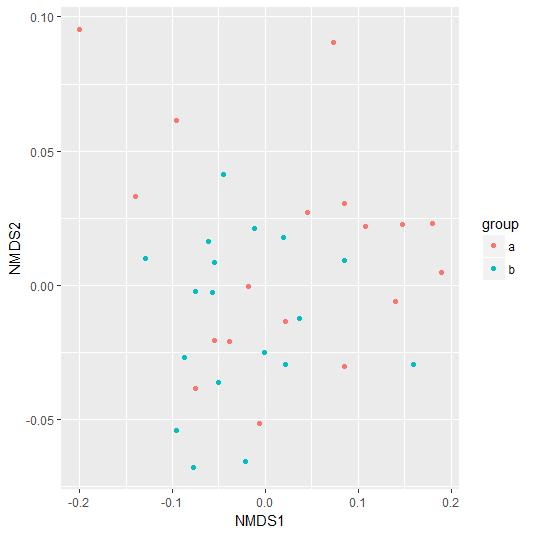
sample\_site1 <- merge(sample\_site1, group, by = 'names', all.x = TRUE); sample\_site2 <- merge(sample\_site2, group, by = 'names', all.x = TRUE)

ggplot(sample\_site1, aes(NMDS1, NMDS2, color = group)) + geom\_point()

ggplot(sample\_site2, aes(NMDS1, NMDS2, color = group)) + geom\_point()

中间过程省略，最终结果分别见下图。左图为使用OTU丰度表和进化树文件作为输入文件的NMDS排序结果，右图为使用现有的距离类型（Weighted UniFrac）直接进行NMDS排序的结果。

虽然不很明显，但个人感觉还是左图较于右图的分组更明显一些。这里我的数据不是很好（因为测试时没有有根树，还得现构建；考虑到之前的数据集OTU代表序列太多，序列对齐会很慢，就随便现找了个OTU代表序列少的小数据集，构建有根树），大家可尝试换一下自己的数据试一下。

备注：

使用GUniFra包计算时，进化树一定要为有根树，无根树不可以。一般我们送测16S样本时，测序公司给的进化树文件都是无根树，需要重新构建。对于16S测序结果中的进化树文件，可以基于OTU代表序列使用qiime计算获得。在qiime中，构建进化树的命令示例“make\_phylogeny.py -i sample\_aligned.fasta -o sample.tre -r midpoint”，这儿加上“-r midpoint”就可以获得有根树了（一般公司给做进化树的时候，是没有这个 -r参数的，最后就是无根树），这儿的“sample\_aligned.fasta”为对齐后的OTU序列，在qiime中可使用“align\_seqs.py”脚本实现。但构建进化树之前的序列对齐步骤会比较慢（运行时间根据OTU代表序列总量决定，特别是样本多时，特别还是自己在windows下安装的qiime虚拟机时，而序列对齐结果一般的测序公司也是不提供的，这儿有点坑……），虽然总体用时也不是特别长，实在不行的话可以要求测序公司做售后给更改一下。

此外还有一点，就是GUniFra包计算的UniFrac距离，这里可能存在另一个疑问。GUniFra的Unweighted UniFrac和使用qiime得到的Unweighted UniFrac是一致的，但GUniFra的Weighted UniFrac和使用qiime得到的Weighted UniFrac是有差异的。猜测原因应该是两个程序对Weighted UniFrac进行标准化的算法不一样（我看到计算公式存在差异）。

## 参考文献

DanielBorcard, FranoisGillet, PierreLegendre, et al. 数量生态学:R语言的应用（赖江山 译）. 高等教育出版社, 2014.