Dimension_reduction

multi-analysis Content

- PCA
- NMDS
- PCoa
- t-sne
- CA
- CCA
- RDA
- Permanova

参考文档

- 周志华《Machine Learning》
- PCA 原理
- 方法比较
- t-SNE
- CA
- CCA

demo 数据

- 瑞金糖尿病的 genus profile
- 瑞金糖尿病的表型

```
genus <- read.table("../dataSet/Ruijin.IGC_9.9M_.genus.ref.pro", header = T, row.names
                    , sep = "\t")
```

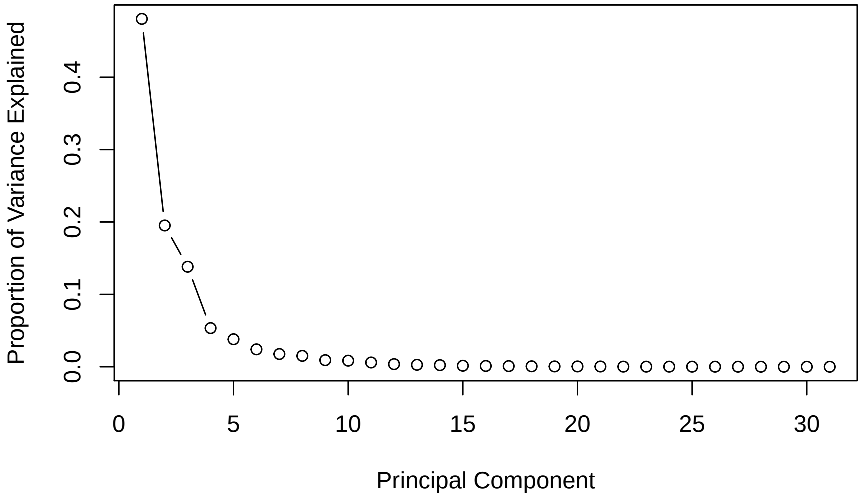
```
phe <- read.table("../dataSet/ruijin_acar_ins.txt", header = T, row.names = 1, sep = "\
source("function.R")
```

1. 数据降维后的可视化

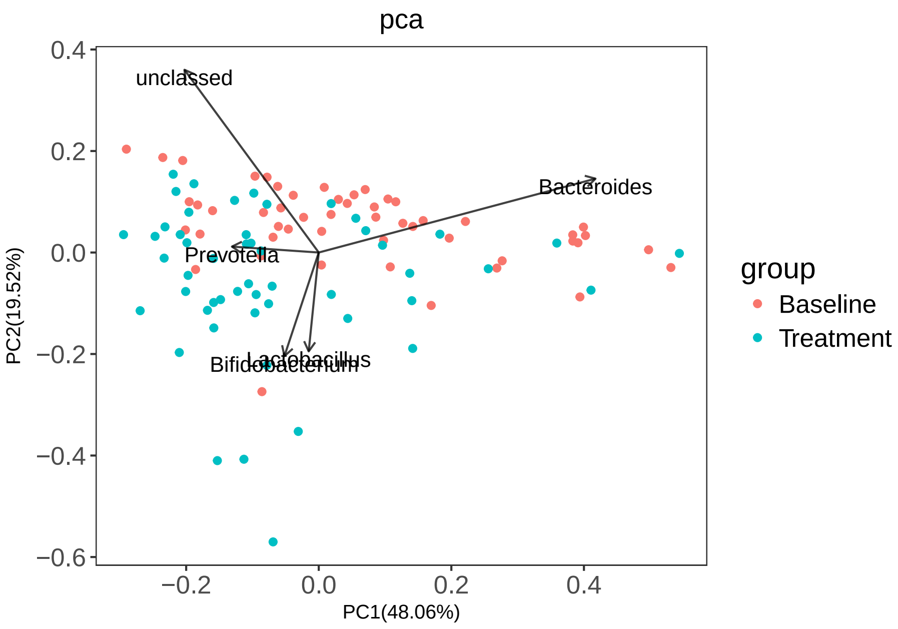
非约束性排序分析，区别于在环境因子约束下的 CCA/RDA. 主要目的是为了是实现在低维空间中样本和样本的比较。

1.1 PCA

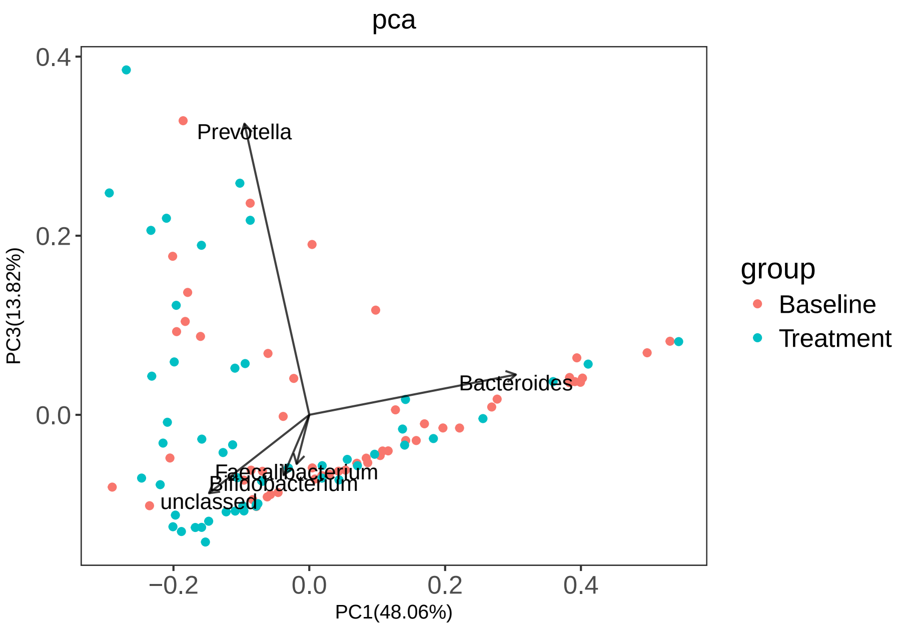
```
# 保持样本名一致
name <- intersect(rownames(phe), colnames(genus))
phe <- phe[name, ]
genus <- as.data.frame(t(genus[, name]))
genus <- genus[, colSums(genus)!=0]
genus <- genus[, core(t(genus))]
# 主成分
# 是否选择 scale
prin_comp <- prcomp(genus, scale. = F)
# 碎石图
std_dev <- prin_comp$sdev
pr_var <- std_dev^2
prop_varex <- pr_var/sum(pr_var)
plot(prop_varex, xlab = "Principal Component",
             ylab = "Proportion of Variance Explained",
             type = "b")
```



```
# figure pca
# PC1 和 PC2
mypca(genus, phe[,1,drop=F], pc1.var = 1,pc2.var = 2,top=5)
```



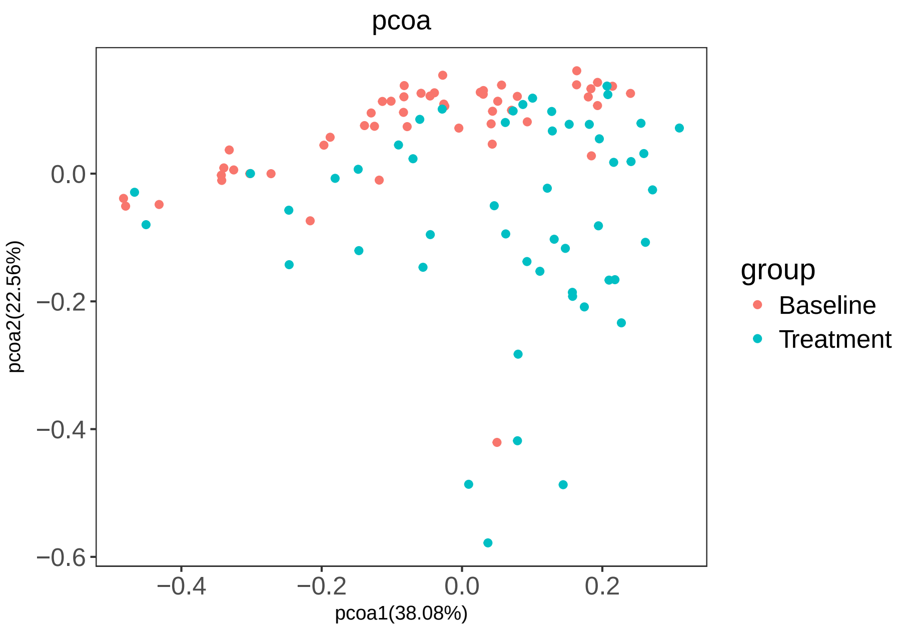
```
# PC1 和 PC3
mypca(genus, phe[,1,drop=F], pc1.var = 1,pc2.var = 3,top=5)
```



1.2 CMDS/PCOA

CMDS（Classical multidimensional scaling）主成分分析和主坐标分析的主要区别是在后者是基于距离来算的，后者的优化目标是新的坐标系下所有样本间的距离和原距离最小

```
mypcoa(genus, phe[,1,drop=F])
```



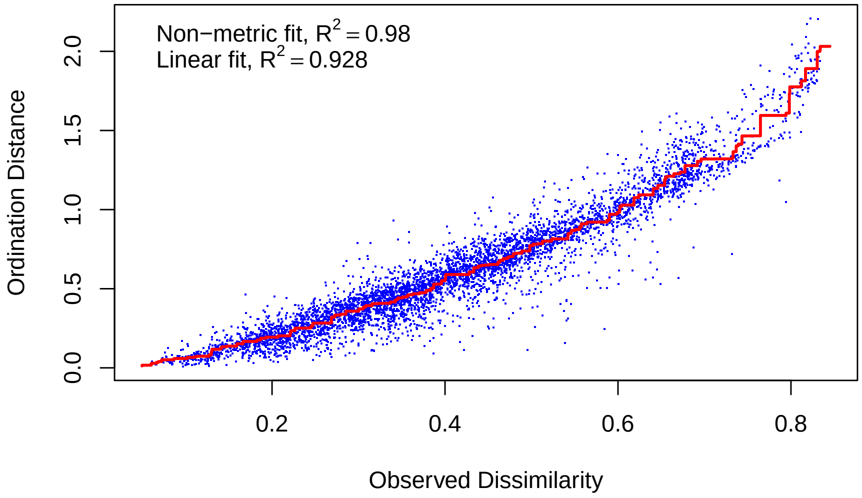
1.3 MDS/NMDS

NMDS（Non-metric multidimensional scaling）与 PCOA 不同之处在于，投影之前会对原来的距离矩阵进行一个变换，期望变换后的距离矩阵在投影后达到预期的优化目标

```
metaMDS(genus,k=2,trymax=100) -> MDSfit
```

```
## Run 0 stress 0.1420443

## Run 1 stress 0.1829243

## Run 2 stress 0.1631846

## Run 3 stress 0.1529931

## Run 4 stress 0.1482354

## Run 5 stress 0.1674978

## Run 6 stress 0.1749792

## Run 7 stress 0.1476897

## Run 8 stress 0.1886811

## Run 9 stress 0.201674

## Run 10 stress 0.1998175
## Run 11 stress 0.1420443
## ... New best solution
## ... Procrustes: rmse 0.0003198268  max resid 0.002917747
## ... Similar to previous best
## Run 12 stress 0.1461632
## Run 13 stress 0.1482352
## Run 14 stress 0.1420449
## ... Procrustes: rmse 0.0004678053  max resid 0.004279334
## ... Similar to previous best
## Run 15 stress 0.143426
## Run 16 stress 0.1681254
## Run 17 stress 0.1589389
## Run 18 stress 0.1910503
## Run 19 stress 0.1453698
## Run 20 stress 0.1424779
## ... Procrustes: rmse 0.006638223  max resid 0.06617789
## *** Solution reached
```

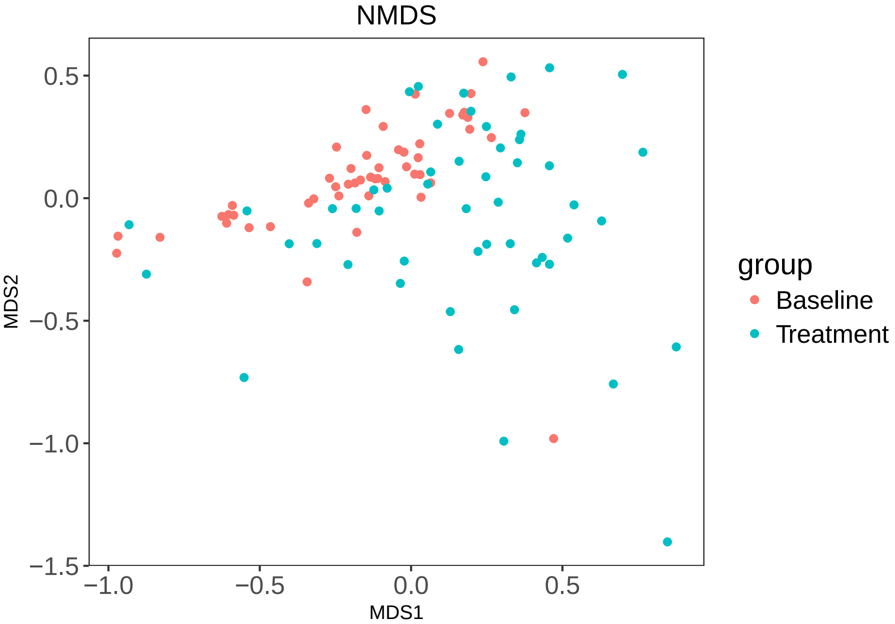
```
stressplot(MDSfit)
```



- 新生成的坐标系统下，生成的坐标间样本的距离和原始距离矩阵的距离的相关性

```
myNMDS(genus, phe[,1,drop=F])
```

```
## Run 0 stress 0.1420443
## Run 1 stress 0.1594861
## Run 2 stress 0.152034
## Run 3 stress 0.1461882
## Run 4 stress 0.1670264
## Run 5 stress 0.1842642
## Run 6 stress 0.1628785
## Run 7 stress 0.1686054
## Run 8 stress 0.1445582
## Run 9 stress 0.1424798
## ... Procrustes: rmse 0.006650992  max resid 0.06612504
## Run 10 stress 0.1806417
## Run 11 stress 0.1496178
## Run 12 stress 0.1741453
## Run 13 stress 0.1792125
## Run 14 stress 0.1739337
## Run 15 stress 0.163057
## Run 16 stress 0.1517016
## Run 17 stress 0.1740461
## Run 18 stress 0.1809296
## Run 19 stress 0.142044
## ... New best solution
## ... Procrustes: rmse 0.0001534891  max resid 0.001361534
## ... Similar to previous best
## Run 20 stress 0.1996459
## *** Solution reached
```



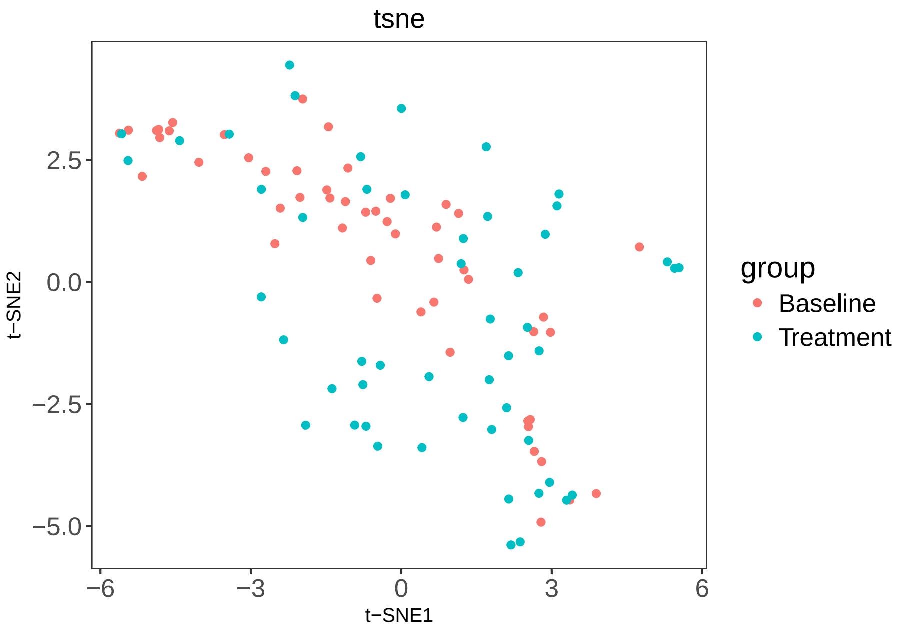
- Strees 值其实反映了 NMDS 分析结果的优劣。通常认为 stress<0.2 时，使用 NMDS 分析的结果具有一定的解释意义；当 stress<0.1 时，可认为是一个好的排序结果；当 stress<0.05 时，则表明分析结果具有极好的代表性。
- 和 PCOA、PCA 结果类似
- Strees 计算来源？

1.4 t-SNE

t-SNE 是基于 SNE 随机邻域嵌入这种方法发展的它相对前几种的一个不同之处在于，将局部的优化考虑到你最终的损失函数中。

```
mytsne(genus, phe[,1,drop=F])
```

```
## Read the 102 x 102 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
##  - point 0 of 102
## Done in 0.02 seconds (sparsity = 0.966359)!
## Learning embedding...
## Iteration 50: error is 50.483038 (50 iterations in 0.03 seconds)
## Iteration 100: error is 49.966507 (50 iterations in 0.03 seconds)
## Iteration 150: error is 50.140095 (50 iterations in 0.02 seconds)
## Iteration 200: error is 50.252556 (50 iterations in 0.02 seconds)
## Iteration 250: error is 49.175161 (50 iterations in 0.02 seconds)
## Iteration 300: error is 1.354231 (50 iterations in 0.02 seconds)
## Iteration 350: error is 0.584602 (50 iterations in 0.01 seconds)
## Iteration 400: error is 0.271197 (50 iterations in 0.02 seconds)
## Iteration 450: error is 0.257606 (50 iterations in 0.02 seconds)
## Iteration 500: error is 0.256107 (50 iterations in 0.02 seconds)
## Fitting performed in 0.20 seconds.
```



更进一步的研究： * 基于降维后的数据的模型构建

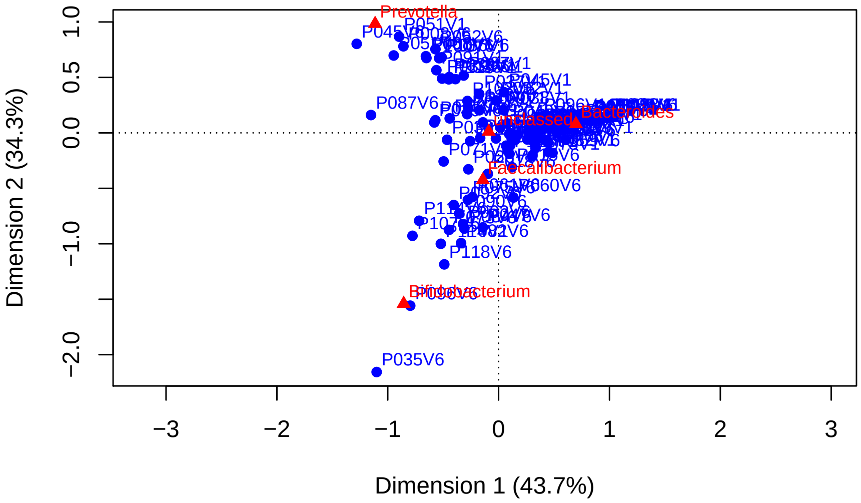
2. 两组数据关联的可视化

约束性排序分析，用于分析环境因子（表型数据）对样本菌群结构的影响。基线胆汁酸数据和 genus 的关联

2.1 CA

CA：对应分析分为简单对应分析 (两个变量间) 和多重对应分析 (多个变量间)，思想是对一个数据的行和列分别做因子分析，期望在同一坐标体系下将行和列的信息反应到二维图中。简单对应分析也可以认为是卡方检验的可视化 (需多个维度)。

```r
genus_top5 <- genus[,c(1:5)]
plot(ca(genus_top5))
```



- 该图反应了所有样本和 top5 的菌的关系，从图中可以看出，大部分的样本和 B 集中在一块（可以认为是大部分趋向 B 肠型），很大一部分治疗后的样本和 Bifi、F 集中在一块，说明治疗后的样本有 Bifi 和 F 升高

2.2 CCA/RDA

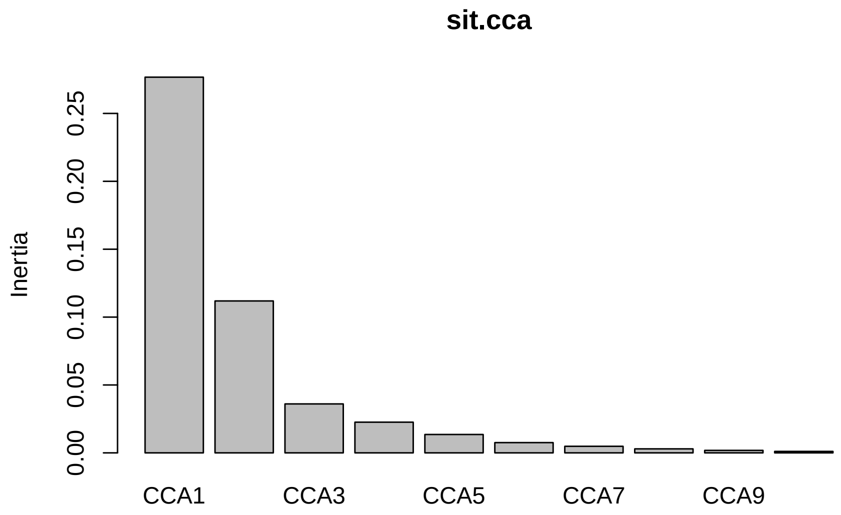
基于 CA CCA（canonical correlation analysis）：典型相关分析 CCA 的优化目标是在两个数据集分别降维后，的相关系数最大；这里的降维是线性降维，同时相关性也是指的是线性相关；进一步的优化方法是 kCCA

```
phe[,20:34] -> bileacid
genus.cle <- genus[1:51,]
env.cle <- bileacid[rownames(genus.cle),]
# rm the P038V1 P081V1
rm_index <- pmatch(c("P038V1", "P081V1"), rownames(genus.cle))
genus.cle2 <- genus.cle[-rm_index,]
genus.cle2.core <- genus.cle2[,core(t(genus.cle2))]
env.cle2 <- env.cle[-rm_index, ]
sit.cca <- cca(genus.cle2.core, env.cle2)
# CCA result
sit.cca
```

```
## Call: cca(X = genus.cle2.core, Y = env.cle2)
##
##               Inertia Proportion Rank
## Total          1.0781     1.0000
## Constrained    0.4811     0.4463   15
## Unconstrained  0.5969     0.5537   23
## Inertia is mean squared contingency coefficient
##
## Eigenvalues for constrained axes:
##    CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7    CCA8    CCA9
## 0.27671 0.11188 0.03606 0.02261 0.01354 0.00755 0.00484 0.00294 0.00184
##   CCA10   CCA11   CCA12   CCA13   CCA14   CCA15
## 0.00108 0.00071 0.00058 0.00039 0.00023 0.00017
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.18858 0.13173 0.08804 0.06572 0.04186 0.02777 0.01157 0.00910
## (Showed only 8 of all 23 unconstrained eigenvalues)
```

- summary(sit.cca) 0.4917 表明 X（genus）解释了总体变异的百分比，表示了 CCA 的 power.

```r
screeplot(sit.cca)
```



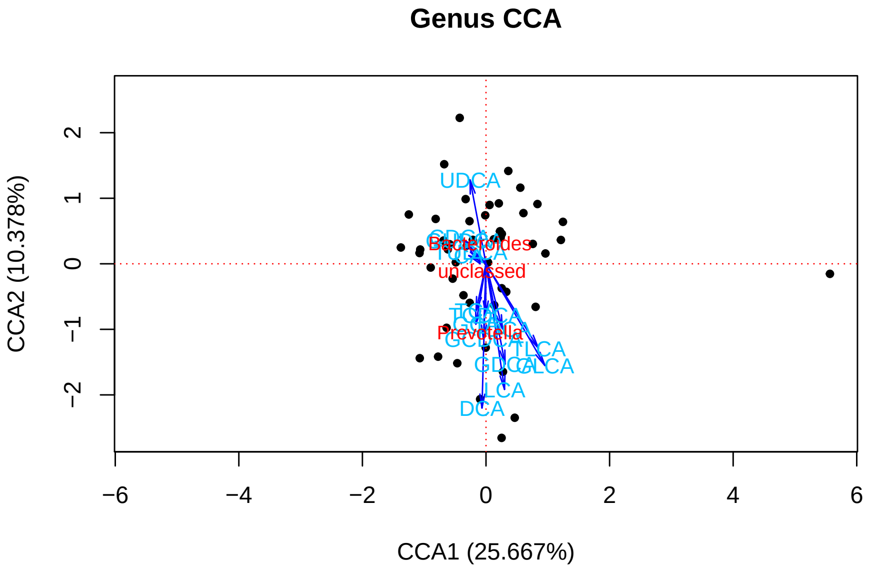
- 碎石图反应了 Constrained 在每个典型坐标的解释度

```r
anova.cca(sit.cca)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(X = genus.cle2.core, Y = env.cle2)
##          Df ChiSquare      F Pr(>F)
## Model    15   0.48113 1.7732  0.141
## Residual 33   0.59694
```

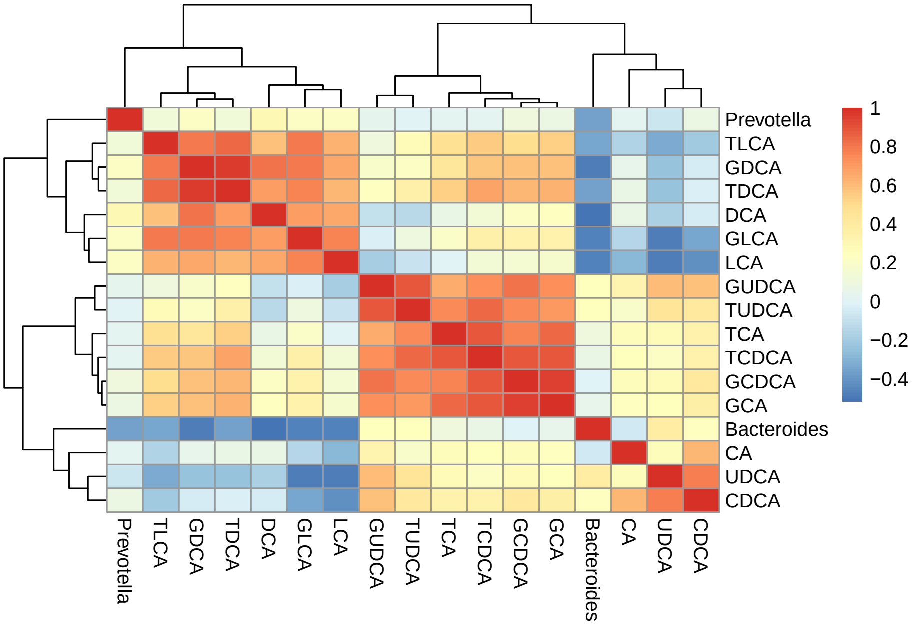
- 当前 CCA model 是否有意义，The analysis is based on the differences in residual deviance in permutations of nested models.

```
genus.s <- scores(sit.cca, display = "sp")
env.s <- scores(sit.cca, display = "bp")
sample.s <- sit.cca$CCA$u[,1:2]
summary <- summary(sit.cca)

xlab <- paste0("CCA1"," (",summary$cont$importance[2,1]*100, "%",")")
ylab <- paste0("CCA2"," (",summary$cont$importance[2,2]*100, "%",")")


plot(sample.s, pch = 20, main = "Genus CCA",xlim=c(-max(abs(c(env.s[,1],genus.s[,1], sa
     ylim=c(-max(abs(c(env.s[,2],genus.s[,2], sample.s[,2]))),max(abs(c(env.s[,2],genus
     xlab = xlab, ylab = ylab)
abline(h = 0, col = 2, lty = 3)
abline(v = 0, col = 2, lty = 3)
s <- 3
arrows(0, 0, env.s[, 1] * s, env.s[, 2] * s, col = 4, angle = 10, length = 0.1)
text(env.s[, 1] * s, env.s[, 2] * s, rownames(env.s), cex = 0.9, col = "deepskyblue")
enter.index <- c("Bacteroides", "unclassed", "Prevotella")
enter.index  <- pmatch(enter.index, rownames(genus.s))
text(genus.s[enter.index, 1], genus.s[enter.index, 2], rownames(genus.s)[enter.index],
    cex=0.8, col="red")
```



- 怎么理解这个结果?

```
library(pheatmap)
env.cle3 <- env.cle2
env.cle3$Bacteroides <- genus.cle2.core[,2]
env.cle3$Prevotella <- genus.cle2.core[,3]
pheatmap(cor(env.cle3, method="s"))
```



2.3 RDA(Redundancy analysis)：冗余分析

```
sit.rda <- rda(genus.cle2.core, env.cle2)
sit.rda
```

```
## Call: rda(X = genus.cle2.core, Y = env.cle2)
##
##                Inertia Proportion Rank
## Total          0.06555    1.00000
## Constrained    0.02496    0.38080   15
## Unconstrained 0.04059    0.61920   24
## Inertia is variance
##
## Eigenvalues for constrained axes:
##     RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7     RDA8
## 0.017301 0.003751 0.002835 0.000436 0.000363 0.000125 0.000071 0.000044
##     RDA9    RDA10    RDA11    RDA12    RDA13    RDA14    RDA15
## 0.000009 0.000007 0.000005 0.000005 0.000004 0.000002 0.000001
```

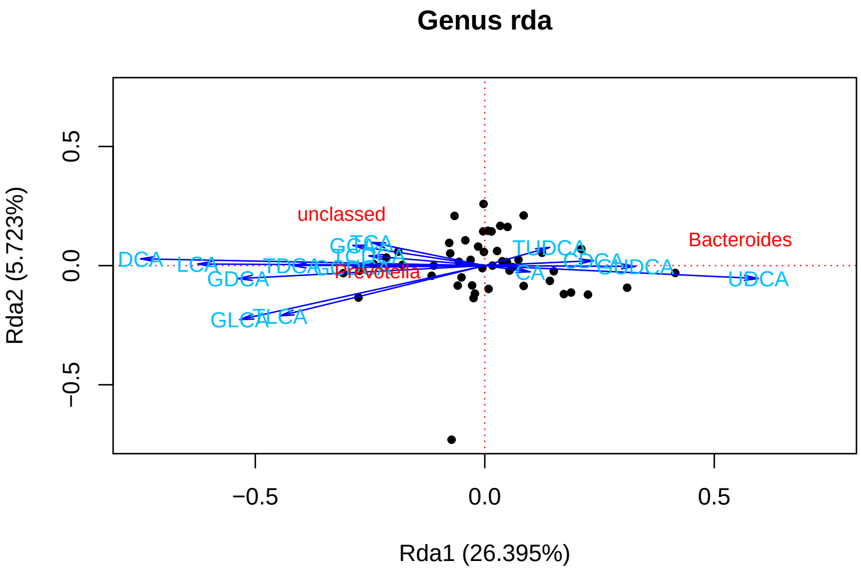
```
##
## Eigenvalues for unconstrained axes:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.027509 0.007385 0.002735 0.001175 0.000838 0.000342 0.000215 0.000124
## (Showed only 8 of all 24 unconstrained eigenvalues)
```

```
anova.cca(sit.rda)
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = genus.cle2.core, Y = env.cle2)
##          Df Variance     F Pr(>F)
## Model    15 0.024960 1.353   0.15
## Residual 33 0.040586
```

```
genus.s <- scores(sit.rda, display = "sp")
env.s <- scores(sit.rda, display = "bp")
sample.s <- sit.rda$CCA$u[,1:2]
summary <- summary(sit.rda)

xlab <- paste0("Rda1"," (",summary$cont$importance[2,1]*100, "%",")")
ylab <- paste0("Rda2"," (",summary$cont$importance[2,2]*100, "%",")")



plot(sample.s, pch = 20, main = "Genus rda",xlim=c(-max(abs(c(env.s[,1],genus.s[,1], sa
     ylim=c(-max(abs(c(env.s[,2],genus.s[,2], sample.s[,2]))),max(abs(c(env.s[,2],genus
     xlab = xlab, ylab = ylab)
abline(h = 0, col = 2, lty = 3)
abline(v = 0, col = 2, lty = 3)
s <- 1
arrows(0, 0, env.s[, 1] * s, env.s[, 2] * s, col = 4, angle = 10, length = 0.1)
text(env.s[, 1] * s, env.s[, 2] * s, rownames(env.s), cex = 0.9, col = "deepskyblue")
```

```
enter.index <- c("Bacteroides", "unclassed", "Prevotella")
enter.index  <- pmatch(enter.index, rownames(genus.s))
text(genus.s[enter.index, 1], genus.s[enter.index, 2], rownames(genus.s)[enter.index],
     cex=0.8, col="red")
```


Genus rda
Rda1 (26.395%)
Rda2 (5.723%)
unclassed
Bacteroides
Prevotella
DCA
LCA
GDCA
TDCA
GCA
TCA
TCDCA
GLCA
TLCA
TUDCA
CDCA
GUDCA
UDCA
CA