CalfSMART microbiome

Sandeep Kumar

23/09/2020

This document represents statistical analyses of fecal bacterial community profile generated from calves raised on three different allowances of milk replacer: 10% of initial bodyweight (LA), 20% of initial bodyweight (HA), and ad libitum (ADLIB). Samples were collected at pre-weaning.

All data associated with analysis can be found here: ‘<https://github.com/kusandeep/Calf-Smart-microbiome-analysis-using-R>’

# Libraries needed for analysis

# Read/prepare datasets

## Read main file, define factors and make subset of data

### Bacterial Proportions  
calfsmartdata = read.table("calfsmart\_for\_R\_20052020.csv", header = T, sep = ",", stringsAsFactors = FALSE,check.names=FALSE)  
dim(calfsmartdata)

## [1] 182 150

for (k in 1:5) { calfsmartdata[,k] = as.factor(calfsmartdata[,k]) }  
  
SCFA = colnames(calfsmartdata)[6:19]; length(SCFA)

## [1] 14

Performance = colnames(calfsmartdata)[20:25]; length(Performance)

## [1] 6

Period\_diet\_all = colnames(calfsmartdata)[26:34]; length(Period\_diet\_all)

## [1] 9

Period\_diet\_week = colnames(calfsmartdata)[35:43]; length(Period\_diet\_week)

## [1] 9

Bacteria = colnames(calfsmartdata)[44:ncol(calfsmartdata)]; length(Bacteria)

## [1] 107

Period\_All\_Week = colnames(calfsmartdata)[26:43]; length(Period\_All\_Week)

## [1] 18

## Omit data/rows with missing values

This is important for SCFA data as not all samples have SCFA values

calfsmartdata\_SCFA = droplevels(calfsmartdata[complete.cases(calfsmartdata),]); dim(calfsmartdata)

## [1] 182 150

dim(calfsmartdata\_SCFA)

## [1] 81 150

### Order the facror for Diets

calfsmartdata$Diet <- factor(calfsmartdata$Diet , levels=c("LA", "HA", "ADLIB"))

## Define colours for plots

mycol1 = c("red","blue","green4","purple","darkorange", "cyan", "chartreuse")

## Choose subset data for analyses

This will later be needed for PcoA analsysis

FactorVars = c("Diet","Sampling\_date", "Antibiotic\_bef\_aft")  
  
respVars = Bacteria; respV = "Bacteria" # choose this or below  
#respVars = SCFA; respV = "SCFA"

## Susetting by sampling date

This will be needed to compare before and after antibiotic treatemant effect on baterial community

sept28 = subset(calfsmartdata, calfsmartdata$Sampling\_date =="2017-09-28")  
oct5 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-10-05")  
oct12 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-10-12")  
oct19 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-10-19")  
oct30 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-10-30")  
nov7 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-11-07")  
nov16 = subset(calfsmartdata, calfsmartdata$Sampling\_date == "2017-11-16")

### Dataset of before antibiotic treatment

calfsmartdata\_befAb = rbind(sept28, oct5, oct12, oct19)   
calfsmartdata\_befAb$Sampling\_date<-factor(calfsmartdata\_befAb$Sampling\_date)

### Dataset of after antibiotic treatment

calfsmartdata\_aftAb = rbind(oct30, nov7, nov16)  
calfsmartdata\_aftAb$Sampling\_date<-factor(calfsmartdata\_aftAb$Sampling\_date)

# Alpha diversity statistics

For alpha diversity Qiime 1.9 was used following instruction beow

### Convert otu table from txt format to biom format

biom convert -i otu\_table.txt -o otu\_table.biom –table-type “OTU table” –to-hdf5

### Perform alpha rarefaction using observed\_species, chao1, shannon at a max sampling depth of 3000, using 24 cores

alpha\_rarefaction.py -i otu\_table.biom -o alpha\_3k/ -p alpha\_param.txt -m -m otu\_mapping.txt -e 3000 -a -O 24

## Differences in diversity across diet groups

Kruskal-Wallis testing was used to test significant difference between diet groups

alpha\_div = read.table("alpha\_diversity.txt", header=T,sep="\t",stringsAsFactors = FALSE,check.names=TRUE)  
dim(alpha\_div)

## [1] 181 5

str(alpha\_div)

## 'data.frame': 181 obs. of 5 variables:  
## $ Calf.ID : int 1 2 3 4 5 6 7 9 10 11 ...  
## $ Diet : chr "ADLIB" "HA" "LA" "ADLIB" ...  
## $ Farm : chr "Farmway" "Farmway" "Farmway" "Farmway" ...  
## $ Shannon.diversity: num 4.49 4.65 4.58 4.33 5.06 ...  
## $ Chao1.richness : num 82.6 87.3 71.7 59.9 80.7 ...

for (k in 1:3) { alpha\_div[,k] = as.factor(alpha\_div[,k]) }   
  
alpha\_div$Diet <- factor(alpha\_div$Diet , levels=c("LA", "HA", "ADLIB")) ### to order factors  
  
### define colours for plots  
mycol1 = c("red","blue","green4","purple","darkorange", "cyan", "chartreuse")  
  
#### Kruskal-Wallis with pair-wise comparisions  
  
res.kruskal\_C <- alpha\_div %>% kruskal\_test(Chao1.richness ~ Diet)  
res.kruskal\_C

## # A tibble: 1 x 6  
## .y. n statistic df p method   
## \* <chr> <int> <dbl> <int> <dbl> <chr>   
## 1 Chao1.richness 181 3.73 2 0.155 Kruskal-Wallis

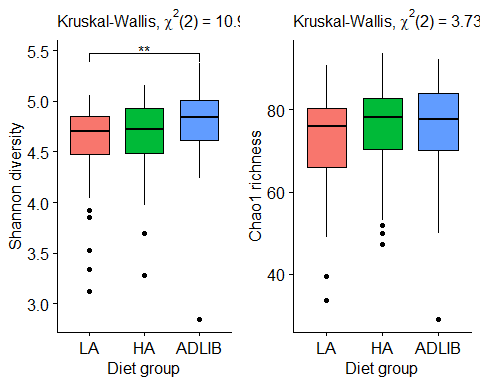
res.kruskal\_S <- alpha\_div %>% kruskal\_test(Shannon.diversity ~ Diet)  
res.kruskal\_S

## # A tibble: 1 x 6  
## .y. n statistic df p method   
## \* <chr> <int> <dbl> <int> <dbl> <chr>   
## 1 Shannon.diversity 181 11.0 2 0.0041 Kruskal-Wallis

### Plot alpha diversity for Chao1 richness and Shannon diversity

Kruskal-wallis and Wilcoxon’s testwas used for pairwise comparision between diet groups

pwc1 <- alpha\_div %>%   
 wilcox\_test(Chao1.richness ~ Diet, p.adjust.method = "bonferroni")   
  
pwc1 <- pwc1 %>% add\_xy\_position(x = "Diet")  
  
p1 <- ggboxplot(alpha\_div, x = "Diet", y = "Chao1.richness", fill = "Diet") +  
 theme(legend.position = "none") +  
 stat\_pvalue\_manual(pwc1, hide.ns = TRUE) +  
 labs(y= "Chao1 richness", x = "Diet group",  
 subtitle = get\_test\_label(res.kruskal\_C, detailed = TRUE)) # remove this line to remove heading  
  
pwc2 <- alpha\_div %>%   
 wilcox\_test(Shannon.diversity ~ Diet, p.adjust.method = "bonferroni")   
  
pwc2 <- pwc2 %>% add\_xy\_position(x = "Diet")  
  
p2 <- ggboxplot(alpha\_div, x = "Diet", y = "Shannon.diversity", fill = "Diet") +   
 theme(legend.position = "none") +  
 stat\_pvalue\_manual(pwc2, hide.ns = TRUE) +  
 labs(y= "Shannon diversity", x = "Diet group",   
 subtitle = get\_test\_label(res.kruskal\_S, detailed = TRUE)) # remove this line to remove heading  
  
grid.arrange(p2, p1, ncol=2)



# ANOSIM analysis

Use appropriate factor(eg. Diet, Farm etc.) for different group observations

anosim(calfsmartdata[,respVars], calfsmartdata$Diet)

##   
## Call:  
## anosim(x = calfsmartdata[, respVars], grouping = calfsmartdata$Diet)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.1306   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata[,respVars], calfsmartdata$Farm)

##   
## Call:  
## anosim(x = calfsmartdata[, respVars], grouping = calfsmartdata$Farm)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.001523   
## Significance: 0.375   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata[,respVars], calfsmartdata$Sampling\_date)

##   
## Call:  
## anosim(x = calfsmartdata[, respVars], grouping = calfsmartdata$Sampling\_date)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.134   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata[,respVars], calfsmartdata$Antibiotic\_bef\_aft)

##   
## Call:  
## anosim(x = calfsmartdata[, respVars], grouping = calfsmartdata$Antibiotic\_bef\_aft)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.004225   
## Significance: 0.389   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_befAb[,respVars], calfsmartdata\_befAb$Diet)

##   
## Call:  
## anosim(x = calfsmartdata\_befAb[, respVars], grouping = calfsmartdata\_befAb$Diet)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.1308   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_befAb[,respVars], calfsmartdata\_befAb$Farm)

##   
## Call:  
## anosim(x = calfsmartdata\_befAb[, respVars], grouping = calfsmartdata\_befAb$Farm)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.02523   
## Significance: 0.207   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_befAb[,respVars], calfsmartdata\_befAb$Sampling\_date)

##   
## Call:  
## anosim(x = calfsmartdata\_befAb[, respVars], grouping = calfsmartdata\_befAb$Sampling\_date)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.1277   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_aftAb[,respVars], calfsmartdata\_aftAb$Diet)

##   
## Call:  
## anosim(x = calfsmartdata\_aftAb[, respVars], grouping = calfsmartdata\_aftAb$Diet)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.1966   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_aftAb[,respVars], calfsmartdata\_aftAb$Farm)

##   
## Call:  
## anosim(x = calfsmartdata\_aftAb[, respVars], grouping = calfsmartdata\_aftAb$Farm)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.01364   
## Significance: 0.342   
##   
## Permutation: free  
## Number of permutations: 999

anosim(calfsmartdata\_aftAb[,respVars], calfsmartdata\_aftAb$Sampling\_date)

##   
## Call:  
## anosim(x = calfsmartdata\_aftAb[, respVars], grouping = calfsmartdata\_aftAb$Sampling\_date)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.1938   
## Significance: 0.002   
##   
## Permutation: free  
## Number of permutations: 999

## Beta dispersion check

permutest(betadisper(vegdist(calfsmartdata[,respVars]), calfsmartdata$Diet, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 2 0.01375 0.0068731 0.5481 999 0.619  
## Residuals 179 2.24444 0.0125388

permutest(betadisper(vegdist(calfsmartdata[,respVars]), calfsmartdata$Farm, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 1 0.00241 0.0024073 0.1995 999 0.677  
## Residuals 180 2.17159 0.0120644

permutest(betadisper(vegdist(calfsmartdata[,respVars]), calfsmartdata$Sampling\_date, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)   
## Groups 6 0.14147 0.023578 1.8957 999 0.08 .  
## Residuals 175 2.17659 0.012438   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

permutest(betadisper(vegdist(calfsmartdata[,respVars]), calfsmartdata$Antibiotic\_bef\_aft, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)   
## Groups 1 0.04106 0.041056 3.6373 999 0.045 \*  
## Residuals 180 2.03174 0.011287   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

permutest(betadisper(vegdist(calfsmartdata\_befAb[,respVars]), calfsmartdata\_befAb$Diet, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 2 0.00869 0.0043471 0.3413 999 0.694  
## Residuals 111 1.41379 0.0127368

permutest(betadisper(vegdist(calfsmartdata\_befAb[,respVars]), calfsmartdata\_befAb$Farm, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 1 0.00527 0.0052721 0.4466 999 0.496  
## Residuals 112 1.32222 0.0118055

permutest(betadisper(vegdist(calfsmartdata\_befAb[,respVars]), calfsmartdata\_befAb$Sampling\_date, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 3 0.05232 0.017440 1.3673 999 0.257  
## Residuals 110 1.40308 0.012755

permutest(betadisper(vegdist(calfsmartdata\_aftAb[,respVars]), calfsmartdata\_aftAb$Diet, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 2 0.00190 0.0009517 0.0887 999 0.952  
## Residuals 65 0.69736 0.0107287

permutest(betadisper(vegdist(calfsmartdata\_aftAb[,respVars]), calfsmartdata\_aftAb$Farm, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 1 0.00062 0.0006158 0.054 999 0.837  
## Residuals 66 0.75287 0.0114071

permutest(betadisper(vegdist(calfsmartdata\_aftAb[,respVars]), calfsmartdata\_aftAb$Sampling\_date, type = "median"))

##   
## Permutation test for homogeneity of multivariate dispersions  
## Permutation: free  
## Number of permutations: 999  
##   
## Response: Distances  
## Df Sum Sq Mean Sq F N.Perm Pr(>F)  
## Groups 2 0.02222 0.011112 0.9254 999 0.459  
## Residuals 65 0.78055 0.012009

# Beta diversity

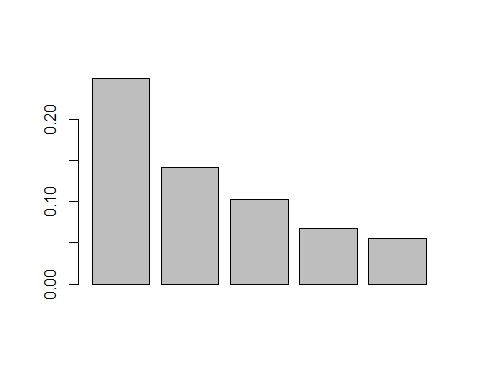
# Compute (Bray-Curtis or Euclidean) distance between reps  
if (respV == "Bacteria") { distObs <- vegdist(calfsmartdata[,respVars], method="bray") }   
as.matrix(distObs)[1:5,1:5]

## 1 2 3 4 5  
## 1 0.0000000 0.6149334 0.3517799 0.6014418 0.5935322  
## 2 0.6149334 0.0000000 0.5822125 0.5542080 0.4401264  
## 3 0.3517799 0.5822125 0.0000000 0.5858340 0.5199577  
## 4 0.6014418 0.5542080 0.5858340 0.0000000 0.3989782  
## 5 0.5935322 0.4401264 0.5199577 0.3989782 0.0000000

### PCoA   
Calf.pco <- pcoa(distObs)  
summary(Calf.pco$values) # check for negative eigenvalues

## Eigenvalues Relative\_eig Rel\_corr\_eig Broken\_stick   
## Min. :-0.201796 Min. :-0.0085863 Min. :0.000000 Min. :0.000000   
## 1st Qu.:-0.030645 1st Qu.:-0.0013039 1st Qu.:0.002841 1st Qu.:0.001562   
## Median : 0.003709 Median : 0.0001578 Median :0.003424 Median :0.003805   
## Mean : 0.129132 Mean : 0.0054945 Mean :0.005495 Mean :0.005495   
## 3rd Qu.: 0.080352 3rd Qu.: 0.0034189 3rd Qu.:0.004700 3rd Qu.:0.007625   
## Max. : 5.870810 Max. : 0.2497995 Max. :0.101164 Max. :0.032072   
## Cum\_corr\_eig Cumul\_br\_stick   
## Min. :0.1012 Min. :0.03207   
## 1st Qu.:0.5813 1st Qu.:0.60404   
## Median :0.7588 Median :0.85091   
## Mean :0.7229 Mean :0.75412   
## 3rd Qu.:0.8999 3rd Qu.:0.96783   
## Max. :1.0000 Max. :1.00000

barplot(Calf.pco$values$Relative\_eig[1:5]) # to check where is the maxim variation



pcopct = round(100\*Calf.pco$values[,3],2)  
pcokeep = length(pcopct[pcopct>10]) # keep PCs with >10%   
if (pcokeep<2) { pcokeep=2 }  
Yid = as.numeric(calfsmartdata$Diet); table(Yid)

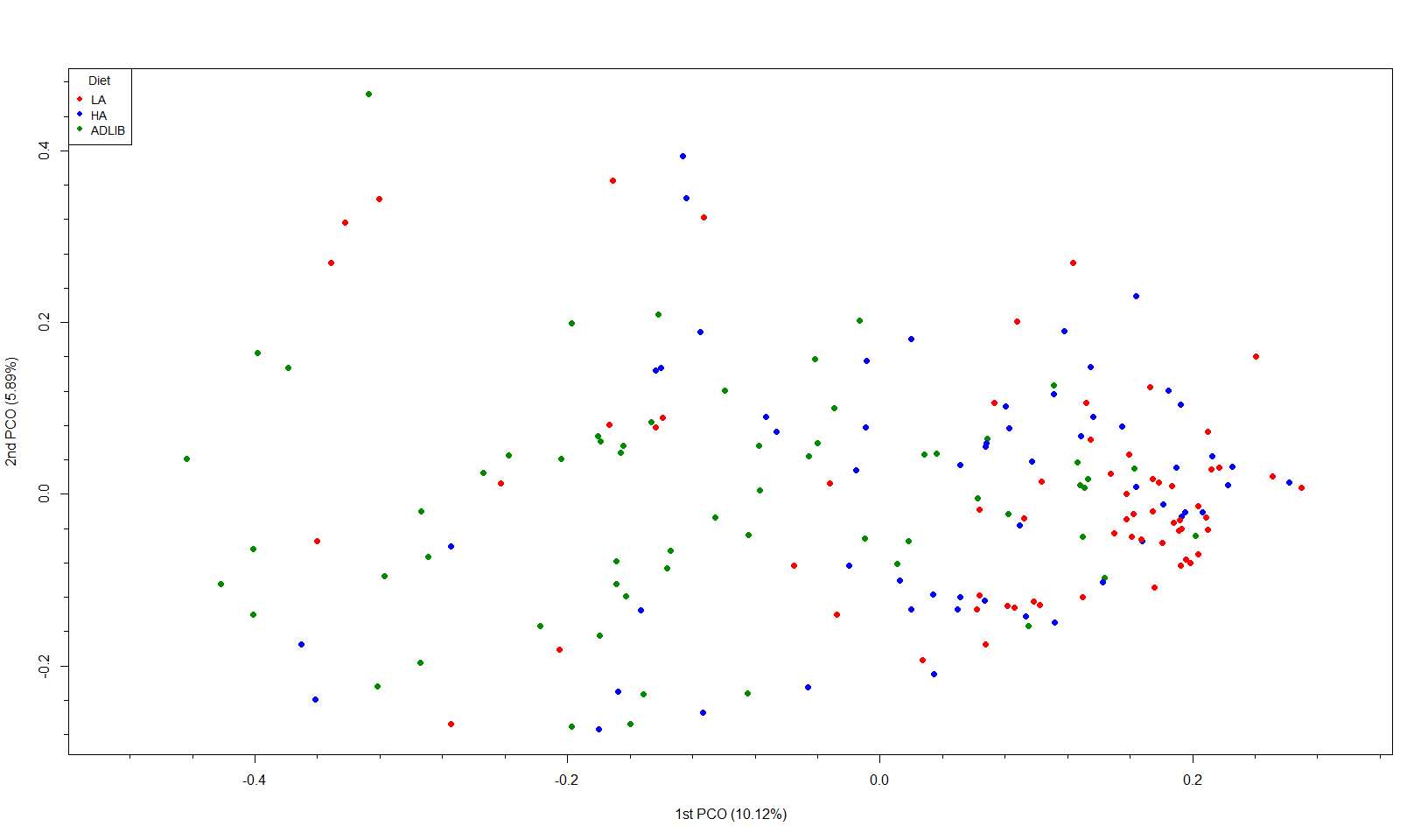
## Yid  
## 1 2 3   
## 64 57 61

PCOdata = data.frame(Calf.pco$vectors[,1:10],Yid,calfsmartdata[,FactorVars]); head(PCOdata)

## Axis.1 Axis.2 Axis.3 Axis.4 Axis.5 Axis.6  
## 1 -0.09906806 0.12108161 -0.22081704 -0.023753842 -0.08109150 -0.002326012  
## 2 -0.40118426 -0.06360284 -0.01168269 -0.001911205 0.01451923 0.198698405  
## 3 -0.19732016 0.19903155 -0.10235292 -0.035457224 -0.03826311 -0.051863985  
## 4 -0.32176421 -0.22426659 0.02605761 -0.076180124 0.03649259 0.113952300  
## 5 -0.40117426 -0.14051523 0.03351744 -0.027045993 0.01083913 0.086070742  
## 6 -0.07734871 0.05633490 -0.25272378 0.008193742 0.09007958 0.039910089  
## Axis.7 Axis.8 Axis.9 Axis.10 Yid Diet Sampling\_date  
## 1 0.002755136 -0.022721898 0.10059432 -0.01936829 3 ADLIB 2017-09-28  
## 2 -0.085433961 -0.139392804 -0.10524219 0.10150581 3 ADLIB 2017-09-28  
## 3 -0.033434229 0.062037263 -0.04796525 -0.02591353 3 ADLIB 2017-09-28  
## 4 0.021825858 -0.061133780 0.01008353 -0.09212268 3 ADLIB 2017-09-28  
## 5 -0.055721560 -0.061531628 -0.08051665 -0.04436501 3 ADLIB 2017-09-28  
## 6 0.018581256 -0.005971339 0.07025311 -0.01619699 3 ADLIB 2017-09-28  
## Antibiotic\_bef\_aft  
## 1 Pre\_AB  
## 2 Pre\_AB  
## 3 Pre\_AB  
## 4 Pre\_AB  
## 5 Pre\_AB  
## 6 Pre\_AB

## Graph using values from above

xlim1 = 1.1\*range(PCOdata[,1])  
xlab1 = paste("1st PCO (",pcopct[1],"%)",sep=""); ylab1 = paste("2nd PCO (",pcopct[2],"%)",sep="")  
plot(PCOdata[,1:2], pch=19, xlim=xlim1,  
 xlab=xlab1, ylab=ylab1, col=mycol1[calfsmartdata$Diet])  
minor.tick(nx=5,ny=5)  
legend("topleft", bty="o", pch=19, cex=0.9, pt.cex=0.9, x.intersp=1, col=mycol1, horiz=F,   
 legend=levels(calfsmartdata$Diet), title="Diet")



# Cononical correlation using spls function from mixOmics

#### For Bacteria vs periods  
X <- calfsmartdata[,Bacteria]  
dim(X)

## [1] 182 107

Y <- calfsmartdata[,Period\_All\_Week]  
dim(Y)

## [1] 182 18

####FOR SCFA Vs bacteria  
X\_SCFA <- calfsmartdata\_SCFA[,Bacteria]  
Y\_SCFA <- calfsmartdata\_SCFA[,SCFA]  
  
  
####FOR performance data Vs bacteria  
X\_perf <- calfsmartdata[,Bacteria]  
dim(X)

## [1] 182 107

Y\_perf <- calfsmartdata[,Performance]  
dim(Y)

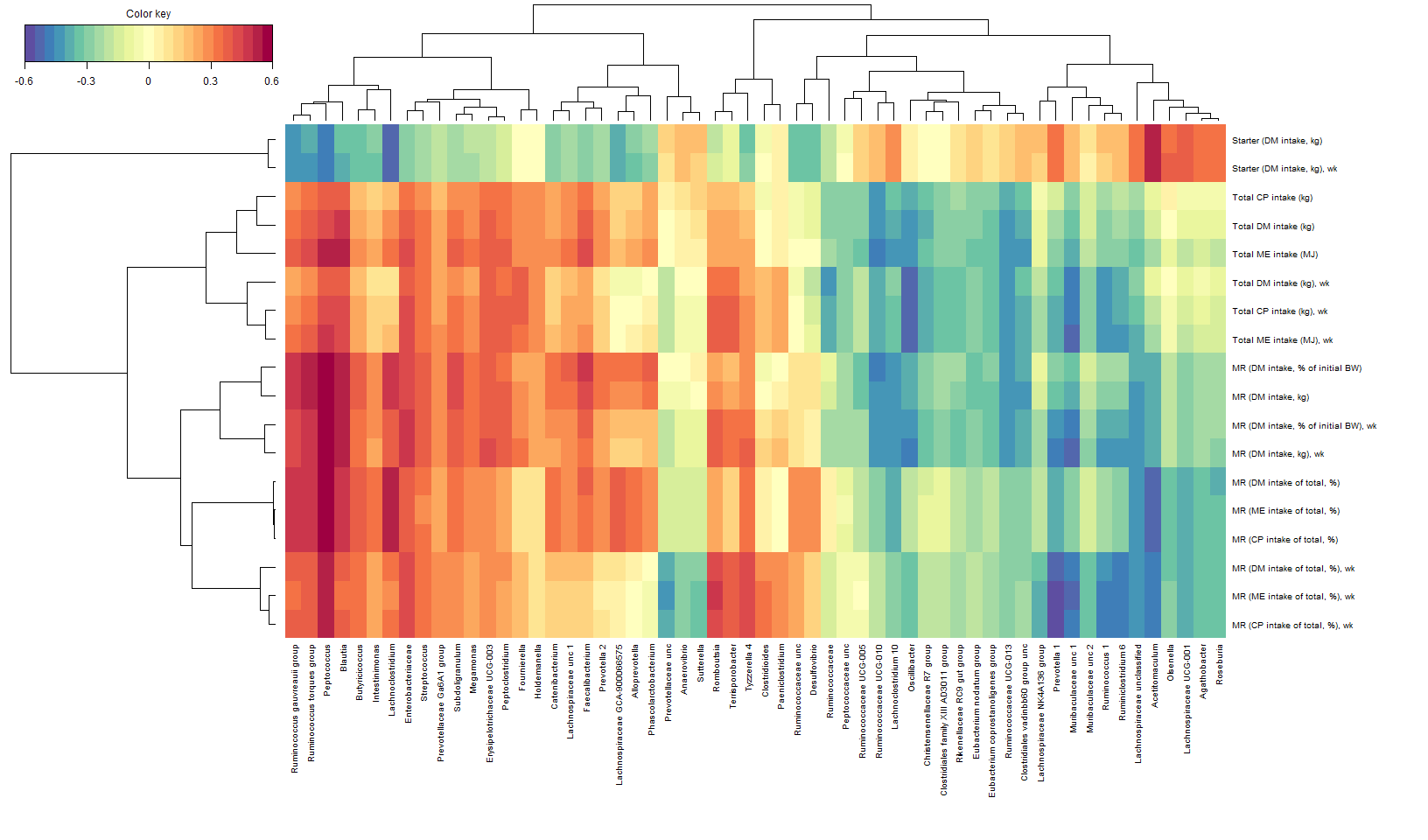
## [1] 182 18

## spls mode of cononical correlation

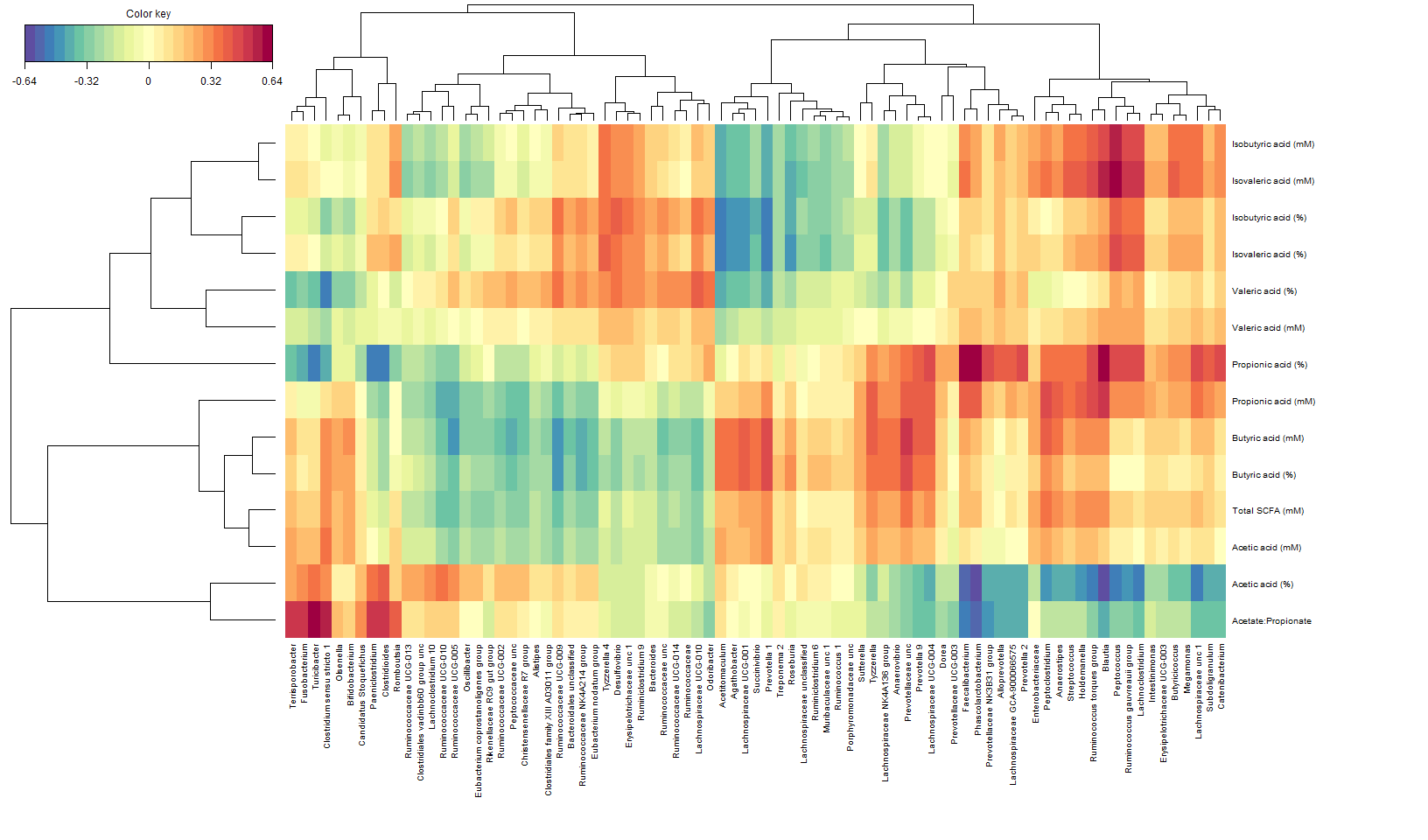
calfsmart.res <- spls(X, Y, ncomp = 3, mode="canonical")  
calfsmart\_SCFA.res <- spls(X\_SCFA, Y\_SCFA, ncomp = 3, mode="canonical")  
calfsmart\_perf.res <- spls(X\_perf, Y\_perf, ncomp = 3, mode="canonical")

## Generate CIM values and plot

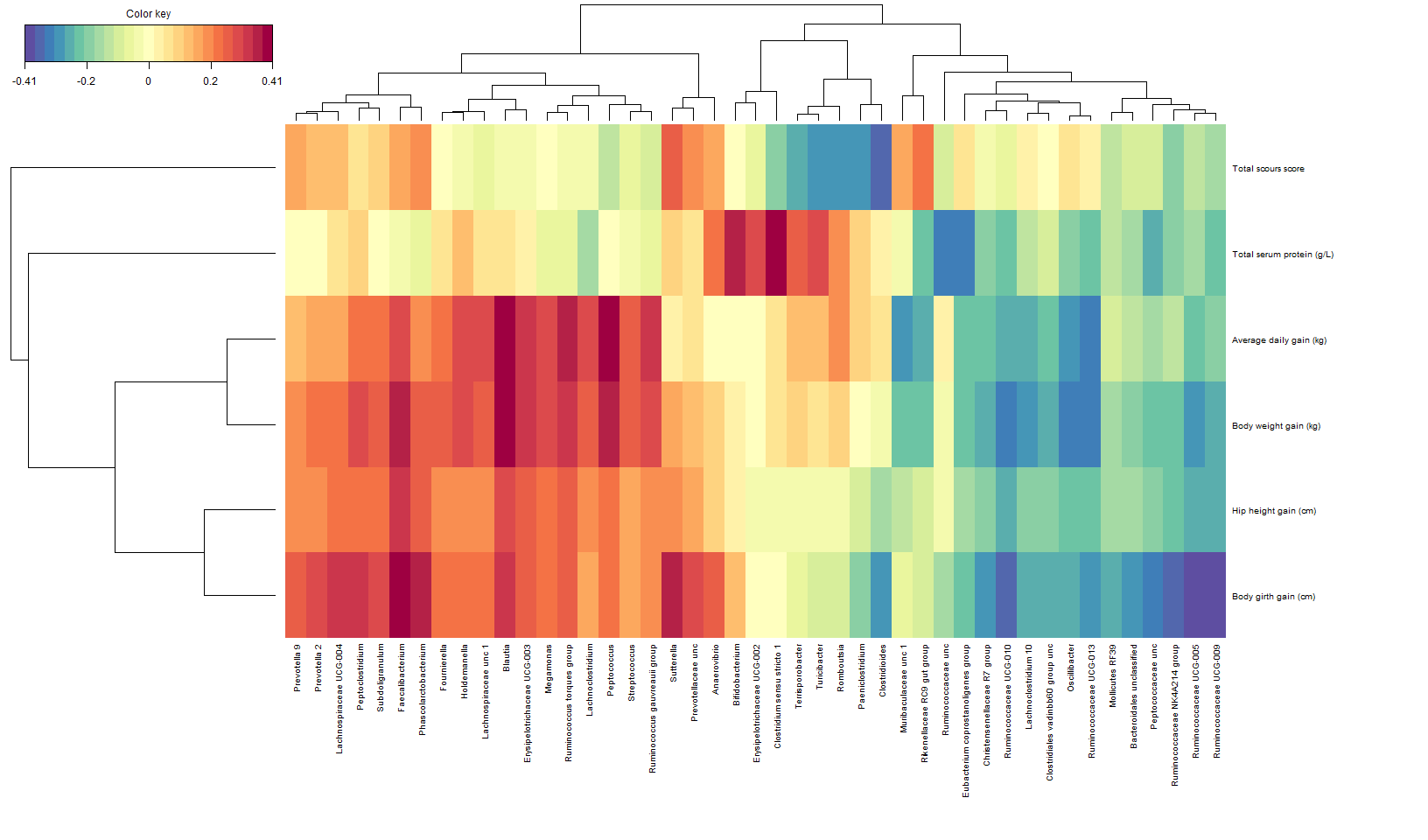
cim\_periods <- cim(calfsmart.res, margins = c(16, 16), threshold = 0.30, transpose = T,  
 row.cex = 0.8, col.cex = 0.8, keysize=c(1,0.7))



cim\_SCFA <- cim(calfsmart\_SCFA.res, margins = c(16, 16),threshold = 0.25, transpose = T,  
 row.cex = 0.8, col.cex = 0.8, keysize=c(1,0.7))



cim\_Performance <- cim(calfsmart\_perf.res, margins = c(16, 16), threshold = 0.25, transpose = T,  
 row.cex = 0.8, col.cex = 0.8, keysize=c(1,0.7))



Print/save CIM values for from above and pest abundance data by different diet groups for heatmap generation.

#write.table(cim\_periods$mat,file="bacteria\_vs\_periods.txt",sep="\t") #Period\_All\_Week  
#write.table(cim\_SCFA$mat,file="bacteria\_vs\_scfa.txt",sep="\t") #SCFA  
#write.table(cim\_Performance$mat,file="bacteria\_vs\_performance.txt",sep="\t") #Performance`

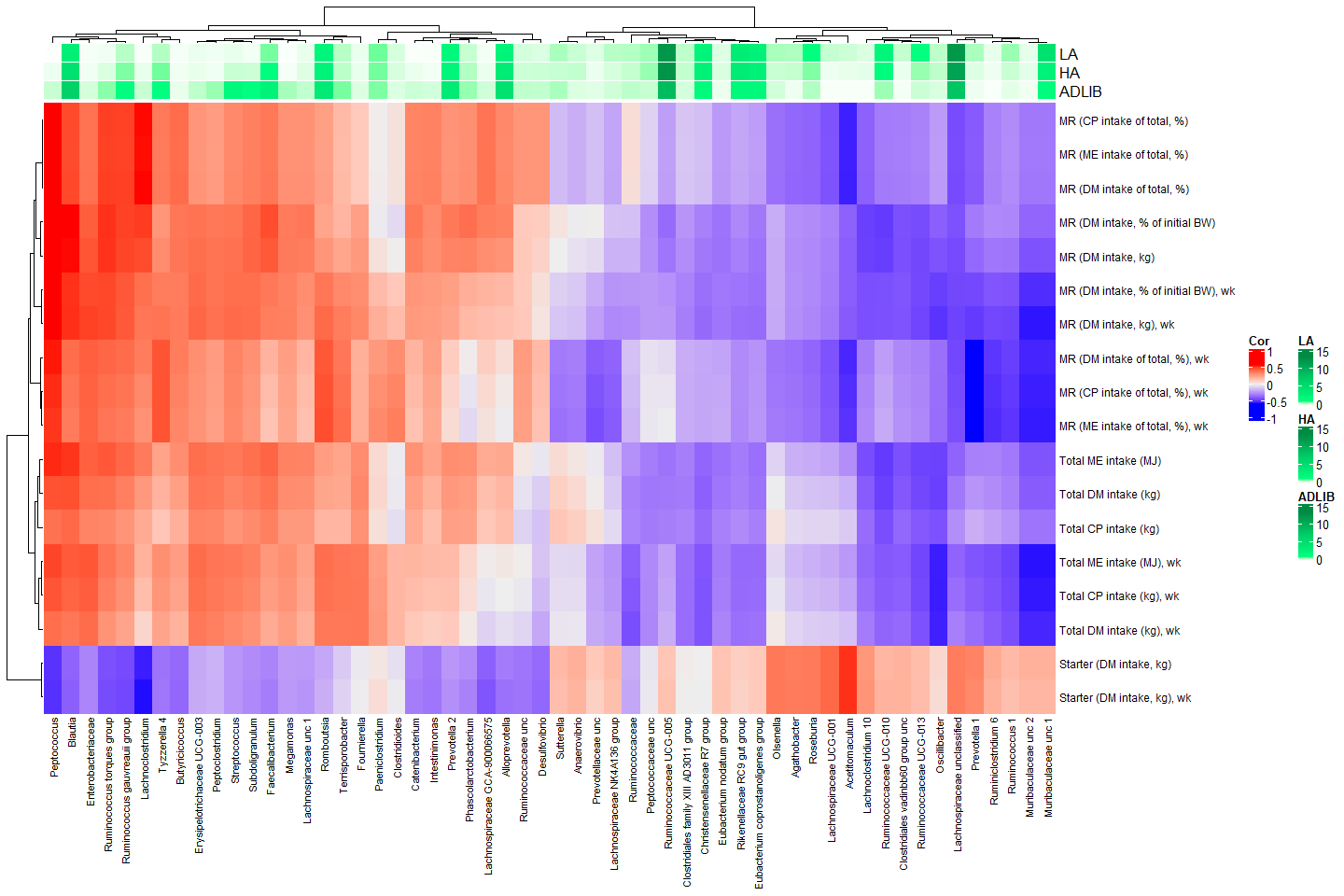
Modified files have been provided for convenience

#Heatmaps using CIM values and abundance; figure 4, 5 and 6

CIM values form above were used to generate heatmaps with bacterial abundance

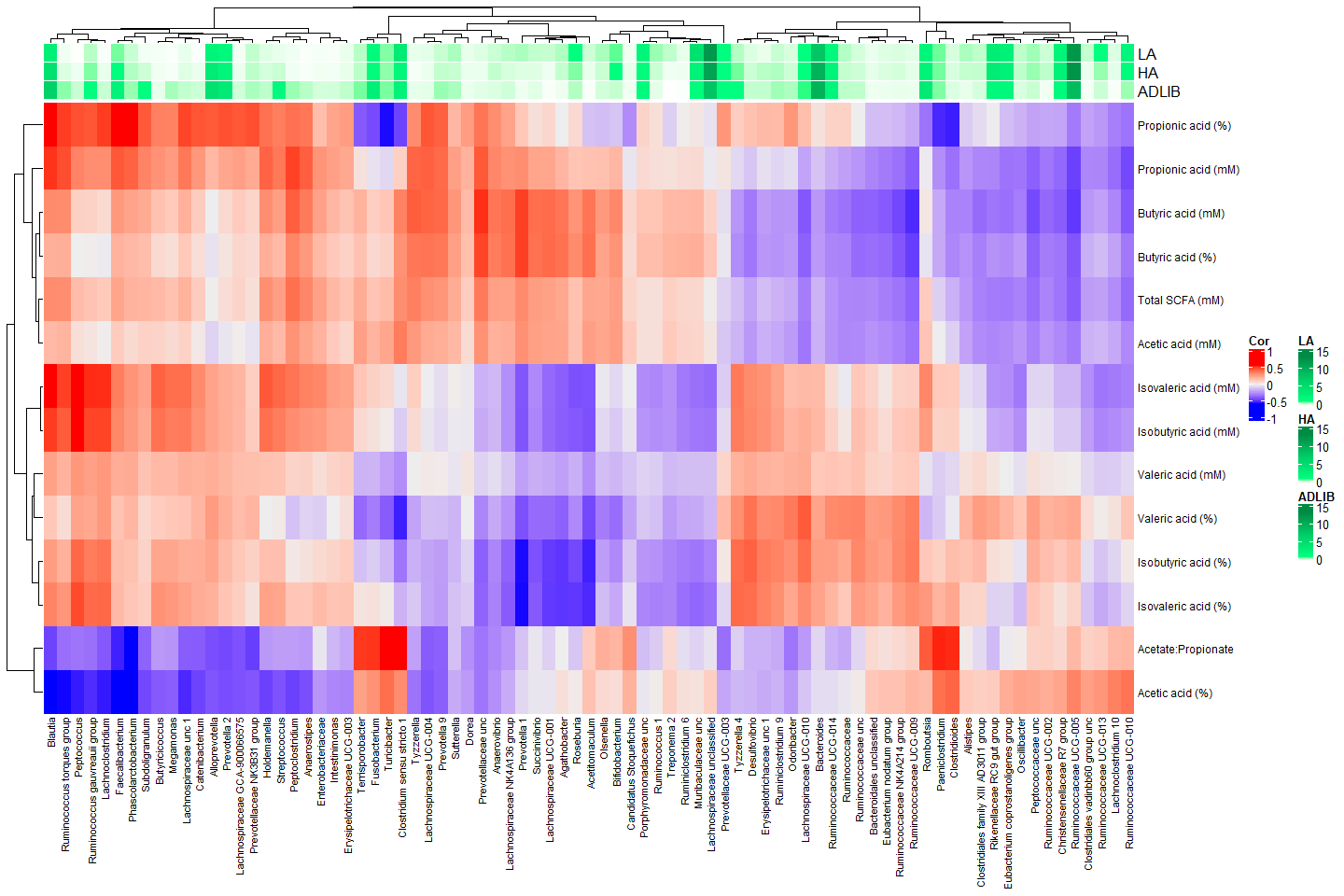
## Heatmap for bacteria vs periods; figure 4

heatmap\_periods.data <- read.table("bacteria\_vs\_periods.txt",header=T,sep="\t",stringsAsFactors = FALSE,check.names=FALSE)  
#using check.names=FALSE can keep the column name as it but then can't use $ function to detect the column  
rownames(heatmap\_periods.data) <- heatmap\_periods.data[,1] # keep the taxon name as it is. No space replaced by .  
  
#define colour  
col\_fun = colorRamp2(c(0, 1,3,7, 13), c("white", "springgreen","springgreen2","springgreen3","springgreen4"))  
  
####Print heatmap  
#pdf("heatmap\_fig4.pdf", height=6, width=12, pointsize=6)  
column\_ha = HeatmapAnnotation(LA = heatmap\_periods.data[,20], HA = heatmap\_periods.data[,21], ADLIB = heatmap\_periods.data[,22],  
 col = list(LA = col\_fun, HA = col\_fun, ADLIB = col\_fun), show\_legend = T, show\_annotation\_name = T)  
Heatmap(t(as.matrix(heatmap\_periods.data[,2:19])), name = "Cor", top\_annotation = column\_ha, column\_names\_gp = gpar(fontsize = 8),  
 row\_names\_gp = gpar(fontsize = 9), show\_heatmap\_legend = T)



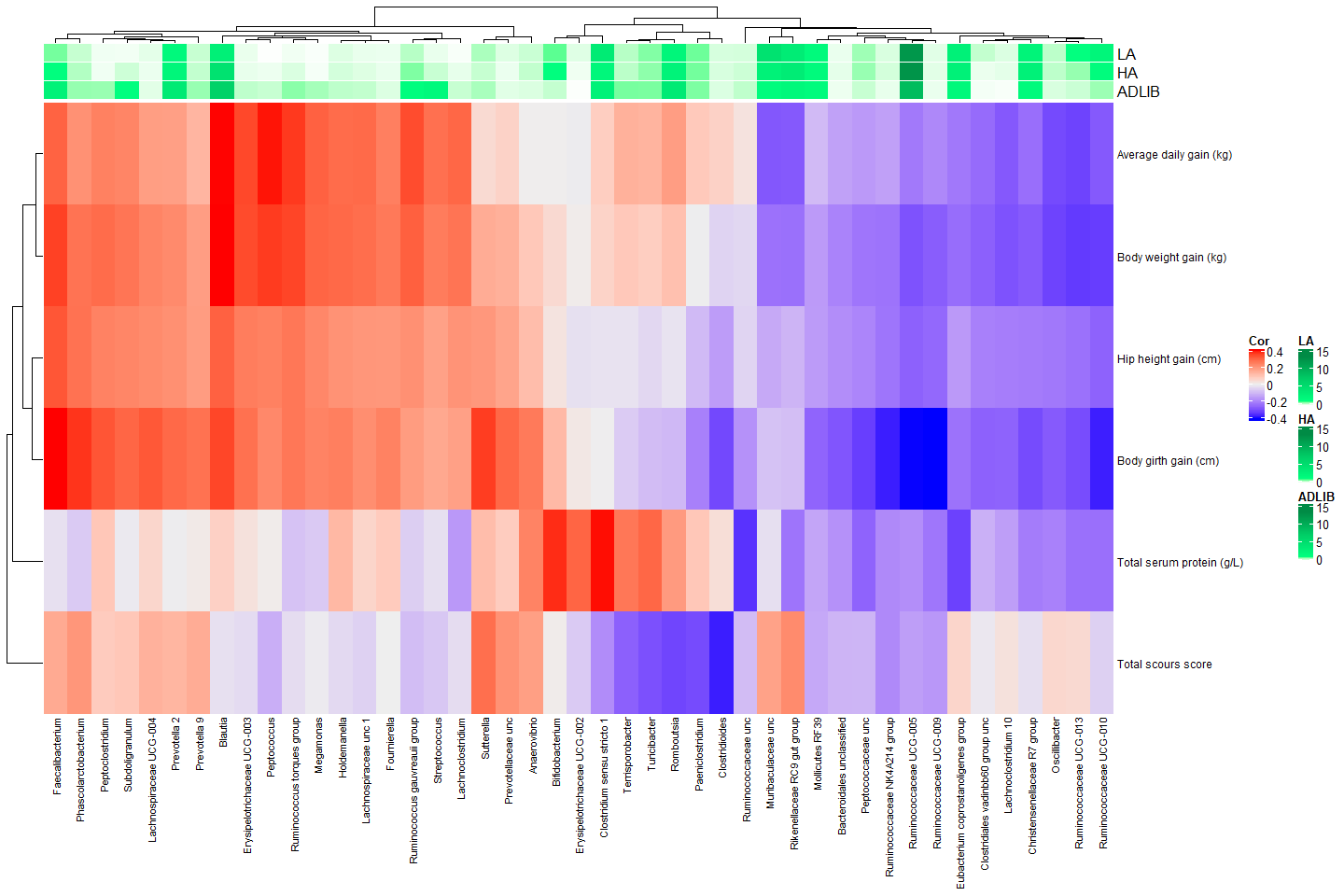
## Heatmap for bacteria vs SCFA; figure 5

heatmap\_scfa.data <- read.table("bacteria\_vs\_scfa.txt",header=T,sep="\t",stringsAsFactors = FALSE,check.names=FALSE)  
#using check.names=FALSE can keep the column name as it but then can't use $ function to detect the column  
rownames(heatmap\_scfa.data) <- heatmap\_scfa.data[,1] # keep the taxon name as it is. No space replaced by .  
  
#define colour  
col\_fun = colorRamp2(c(0, 1,3,7, 13), c("white", "springgreen","springgreen2","springgreen3","springgreen4"))  
  
####Print heatmap  
#pdf("heatmap\_fig5.pdf", height=6, width=12, pointsize=6)  
column\_ha = HeatmapAnnotation(LA = heatmap\_scfa.data[,16], HA = heatmap\_scfa.data[,17], ADLIB = heatmap\_scfa.data[,18],  
 col = list(LA = col\_fun, HA = col\_fun, ADLIB = col\_fun), show\_legend = T, show\_annotation\_name = T)  
Heatmap(t(as.matrix(heatmap\_scfa.data[,2:15])), name = "Cor", top\_annotation = column\_ha, column\_names\_gp = gpar(fontsize = 8), row\_names\_gp = gpar(fontsize = 9), show\_heatmap\_legend = T)



## Heatmap for bacteria vs performance; figure 6

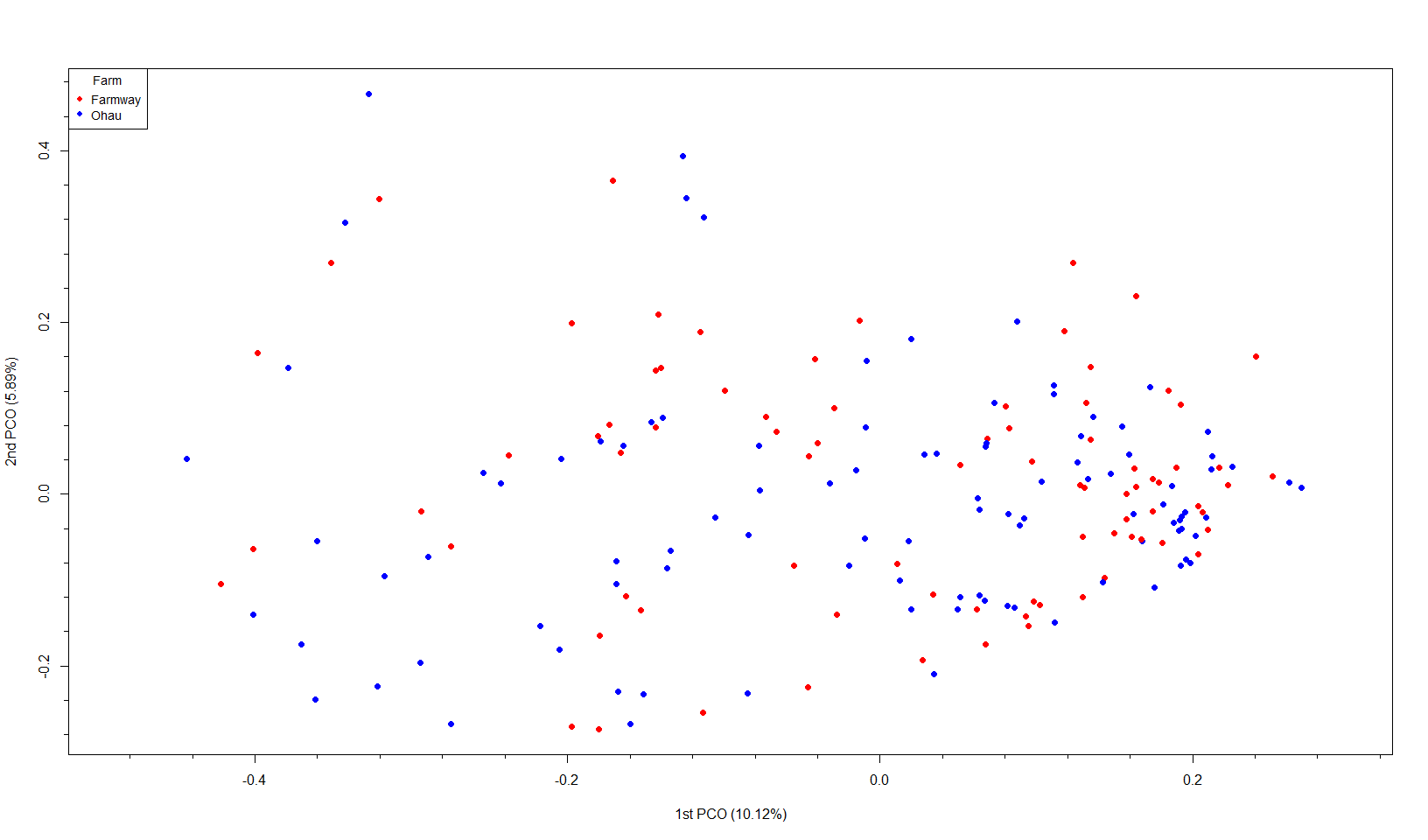
heatmap\_performance.data <- read.table("bacteria\_vs\_performance.txt",header=T,sep="\t",stringsAsFactors = FALSE,check.names=FALSE)  
#using check.names=FALSE can keep the column name as it but then can't use $ function to detect the column  
rownames(heatmap\_performance.data) <- heatmap\_performance.data[,1] # keep the taxon name as it is. No space replaced by .  
  
#define colour  
col\_fun = colorRamp2(c(0, 1,3,7, 13), c("white", "springgreen","springgreen2","springgreen3","springgreen4"))  
  
####Print heatmap  
#pdf("heatmap\_fig6.pdf", height=6, width=12, pointsize=6)  
column\_ha = HeatmapAnnotation(LA = heatmap\_performance.data[,8], HA = heatmap\_performance.data[,9], ADLIB = heatmap\_performance.data[,10],  
 col = list(LA = col\_fun, HA = col\_fun, ADLIB = col\_fun), show\_legend = T, show\_annotation\_name = T)  
Heatmap(t(as.matrix(heatmap\_performance.data[,2:7])), name = "Cor", top\_annotation = column\_ha, column\_names\_gp = gpar(fontsize = 8),  
 row\_names\_gp = gpar(fontsize = 9), show\_heatmap\_legend = T)



#Supplementary figures

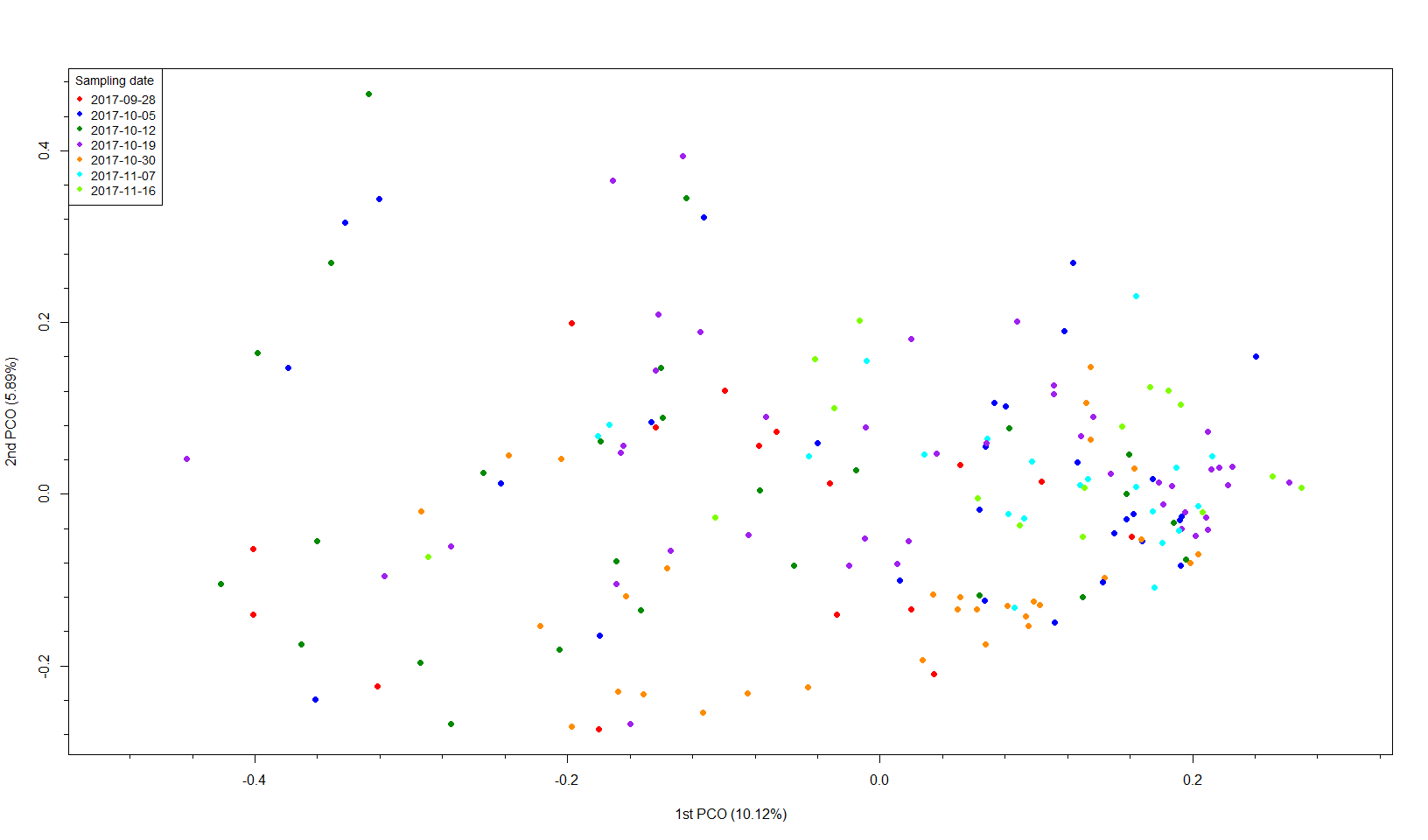
## Figure S1- PCoA by Farm

xlim1 = 1.1\*range(PCOdata[,1])  
xlab1 = paste("1st PCO (",pcopct[1],"%)",sep=""); ylab1 = paste("2nd PCO (",pcopct[2],"%)",sep="")  
plot(PCOdata[,1:2], pch=19, xlim=xlim1,  
 xlab=xlab1, ylab=ylab1, col=mycol1[calfsmartdata$Farm])  
minor.tick(nx=5,ny=5)  
legend("topleft", bty="o", pch=19, cex=0.9, pt.cex=0.9, x.intersp=1, col=mycol1, horiz=F,   
 legend=levels(calfsmartdata$Farm), title="Farm")



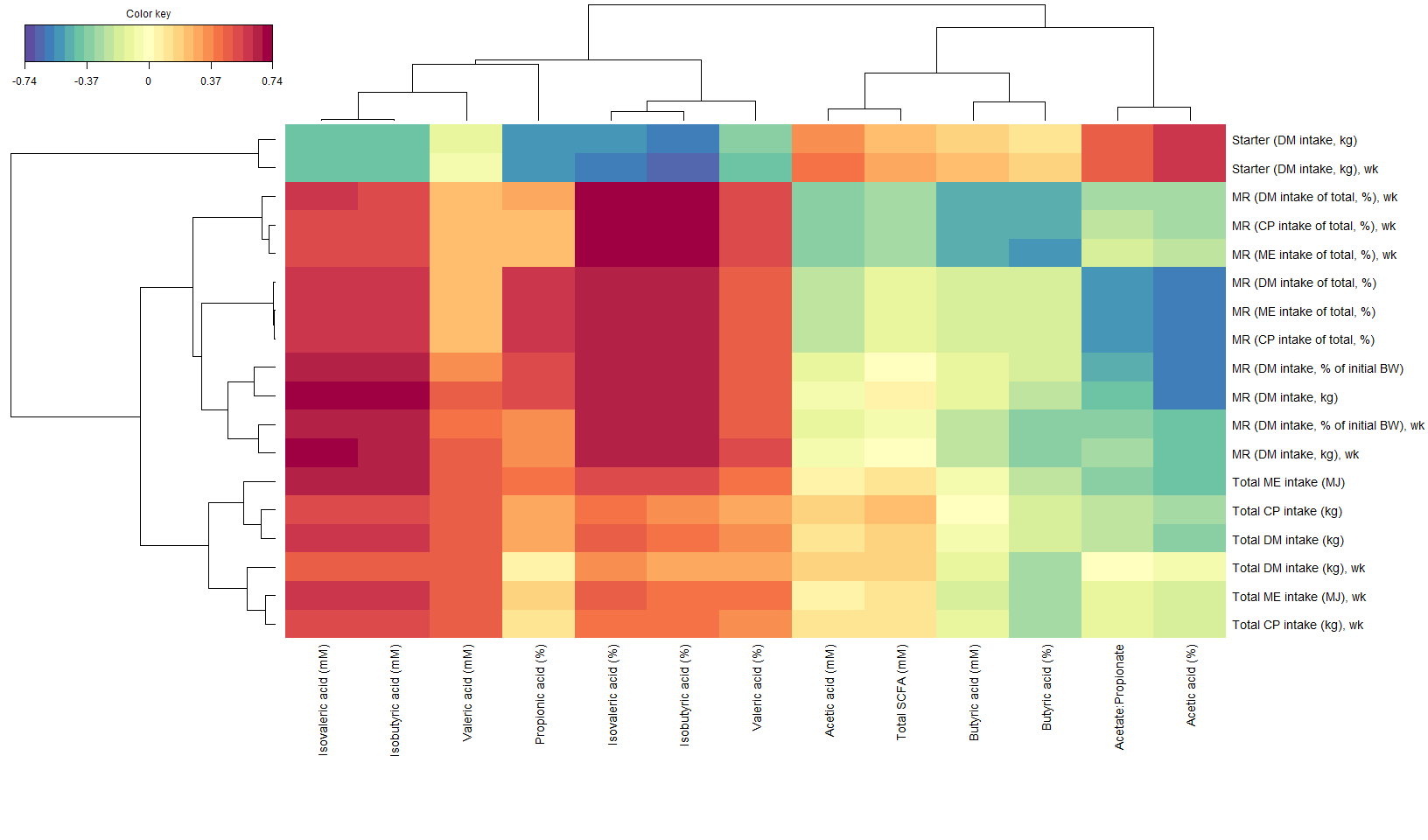
## Figure S1- PCoA by Sampling date

xlim1 = 1.1\*range(PCOdata[,1])  
xlab1 = paste("1st PCO (",pcopct[1],"%)",sep=""); ylab1 = paste("2nd PCO (",pcopct[2],"%)",sep="")  
plot(PCOdata[,1:2], pch=19, xlim=xlim1,  
 xlab=xlab1, ylab=ylab1, col=mycol1[calfsmartdata$Sampling\_date])  
minor.tick(nx=5,ny=5)  
legend("topleft", bty="o", pch=19, cex=0.9, pt.cex=0.9, x.intersp=1, col=mycol1, horiz=F,   
 legend=levels(calfsmartdata$Sampling\_date), title="Sampling date")



## Cononical correlation between SCFA profiles and nutritional intakes

X\_SCFA\_int <- calfsmartdata\_SCFA[,SCFA]  
Y\_SCFA\_int <- calfsmartdata\_SCFA[,Period\_All\_Week]  
  
calfsmart\_SCFA\_int.res <- spls(X\_SCFA\_int, Y\_SCFA\_int, ncomp = 3, mode="canonical")  
#pdf("Figure\_S3.pdf", height=6, width=12, pointsize=6)  
cim\_SCFA\_int <- cim(calfsmart\_SCFA\_int.res, margins = c(16, 16), threshold = 0.30, transpose = T,  
 row.cex = 1.2, col.cex = 1.2, keysize=c(1,0.7))



## Cononical correlation between SCFA profiles and calf performance

X\_SCFA\_perf <- calfsmartdata\_SCFA[,SCFA]  
Y\_SCFA\_perf <- calfsmartdata\_SCFA[,Performance]  
  
calfsmart\_SCFA\_perf.res <- spls(X\_SCFA\_perf, Y\_SCFA\_perf, ncomp = 3, mode="canonical")  
#pdf("Figure\_S4.pdf", height=6, width=12, pointsize=6)  
cim\_SCFA\_perf <- cim(calfsmart\_SCFA\_perf.res, margins = c(16, 16), threshold = 0.30, transpose = T,  
 row.cex = 1.2, col.cex = 1.2, keysize=c(1,0.7))

