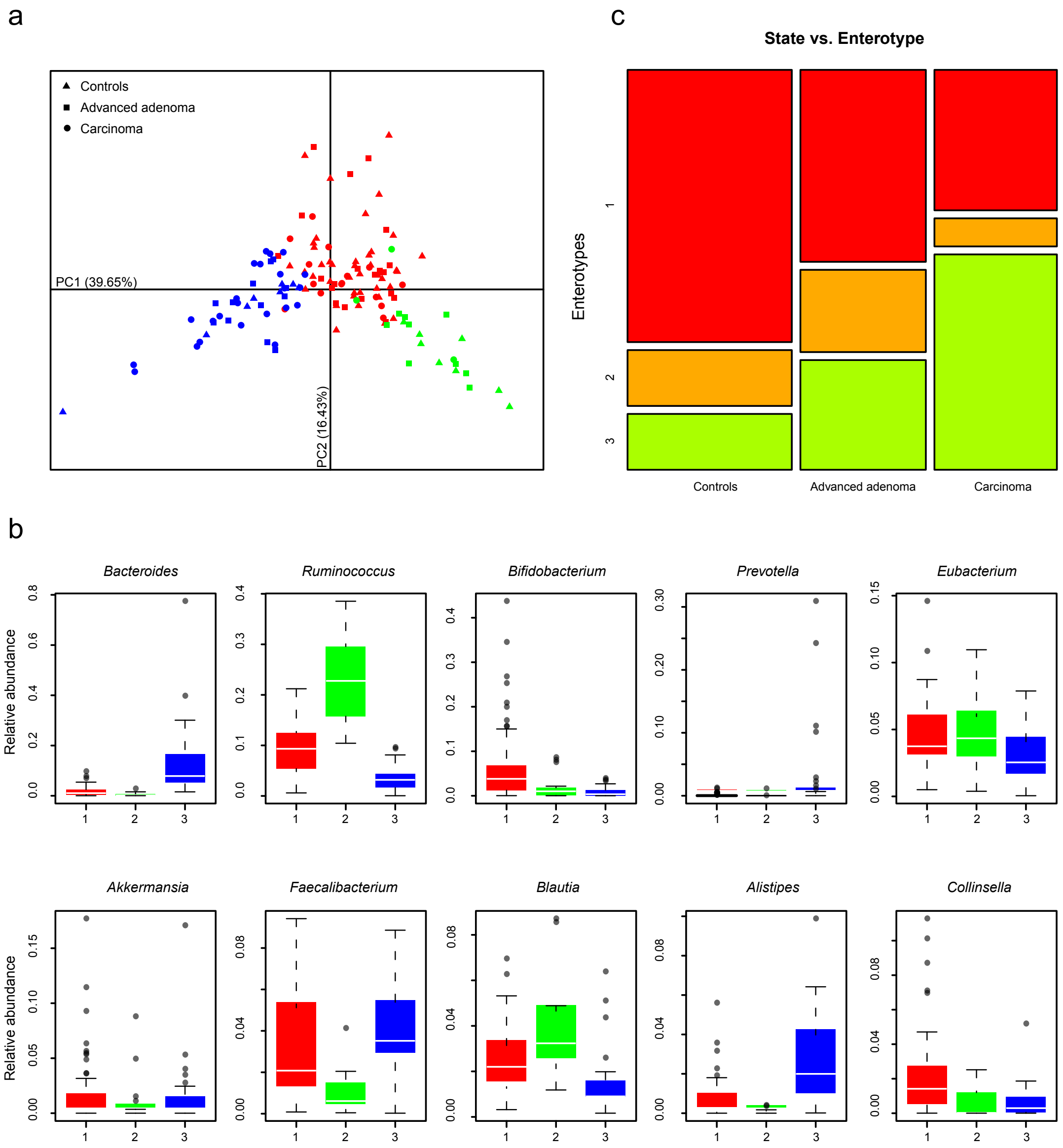
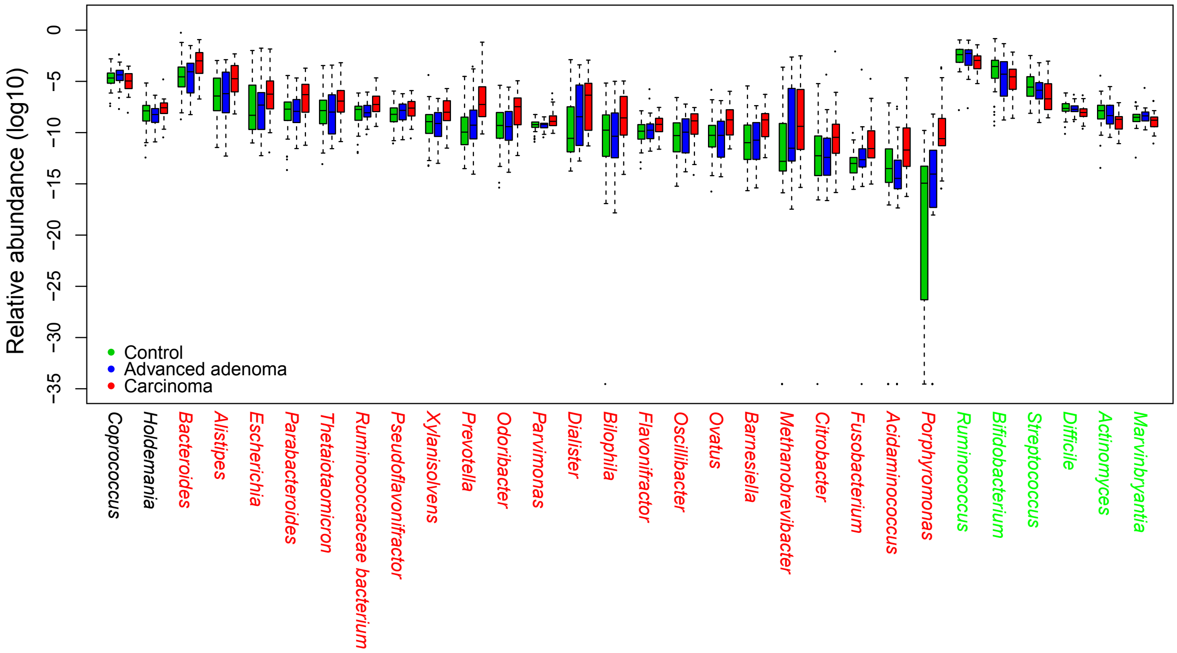
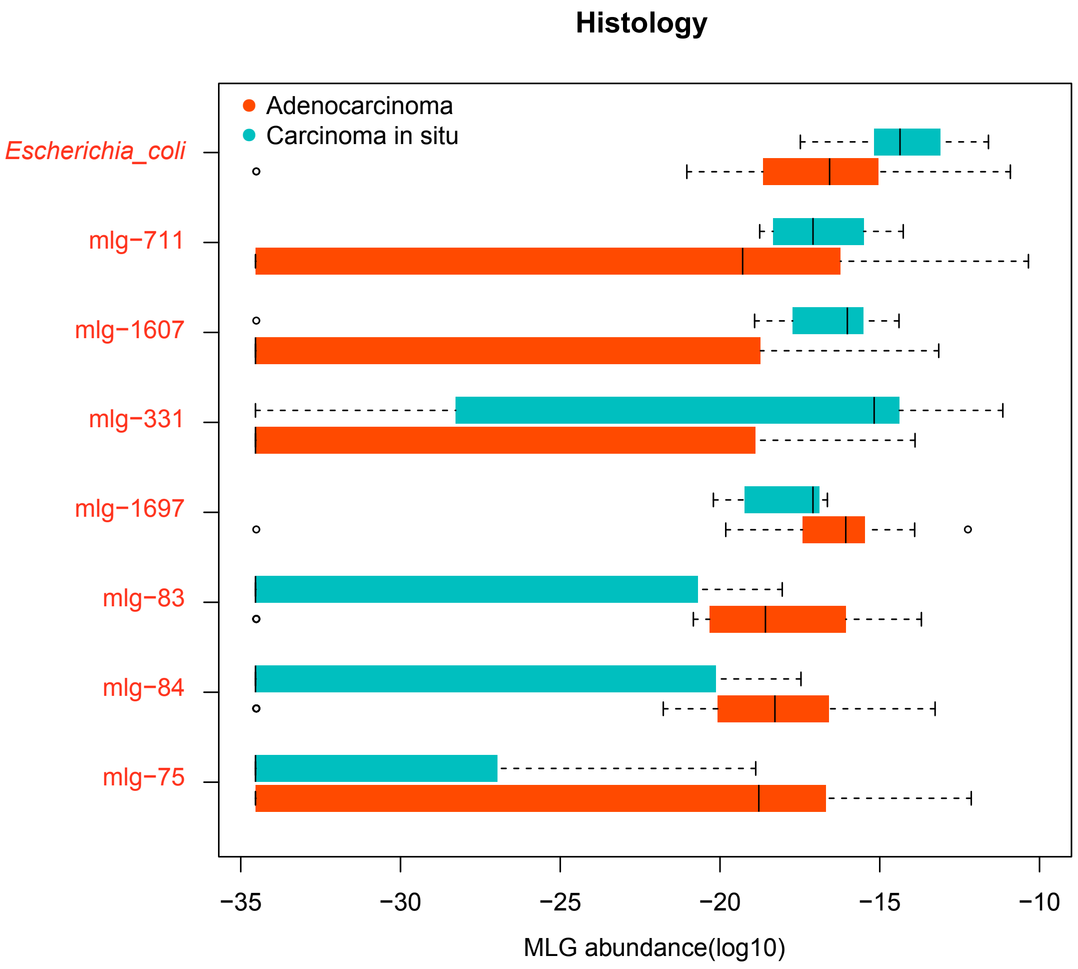
Supplementary Figures



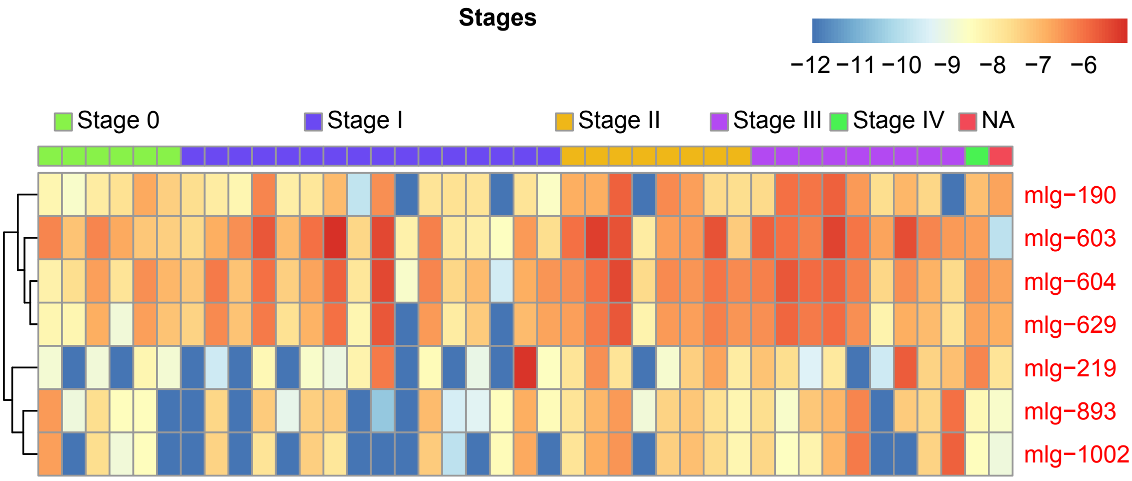
Supplementary Figure 1: Enterotypes according to the original PAM-based method.(*1*)
(a) Principle component analysis for the stool samples at the genus level. Red, community type 1; green, community type 2; blue, community type 3. **(b)** Relative abundances of the top 10 most abundant genera in the three community types. Box plot as in Fig. 1a. **(c)** Distribution of the healthy control, advanced adenoma, and carcinoma samples in the community types. P = 0.00031, Fisher test; p = 0.00028, chi-sq test.



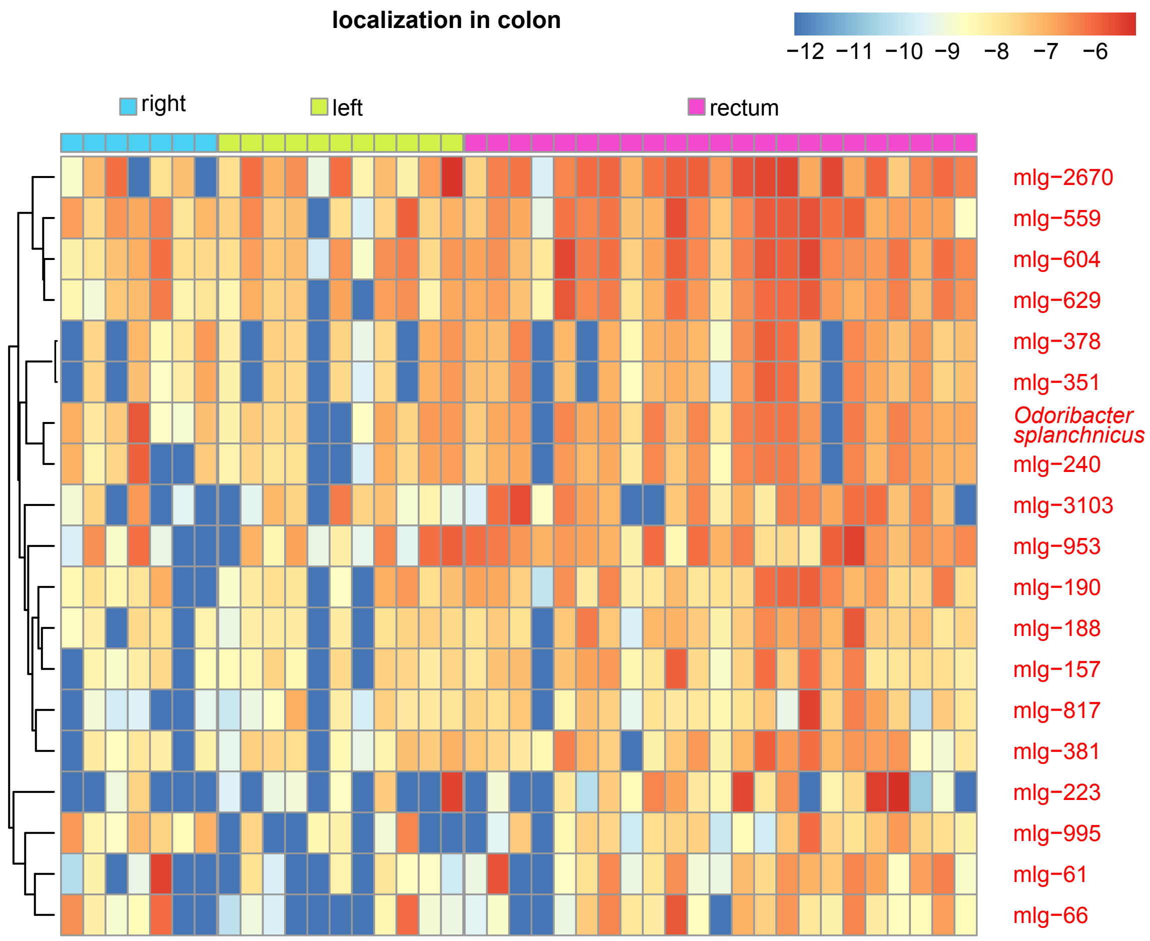
Supplementary Figure 2: **Top 30 most abundant genera differentially enriched in healthy controls, advanced adenoma or carcinoma.**
Genera with significant difference in relative abundance among the three groups were identified by KruskalWallis test ($p < 0.05$, FDR = 0.1447). Of these, the top 30 most abundant genera are shown. The name of each genus was coloured according to its direction of enrichment: green, control-enriched; red, carcinoma-enriched; black, other.



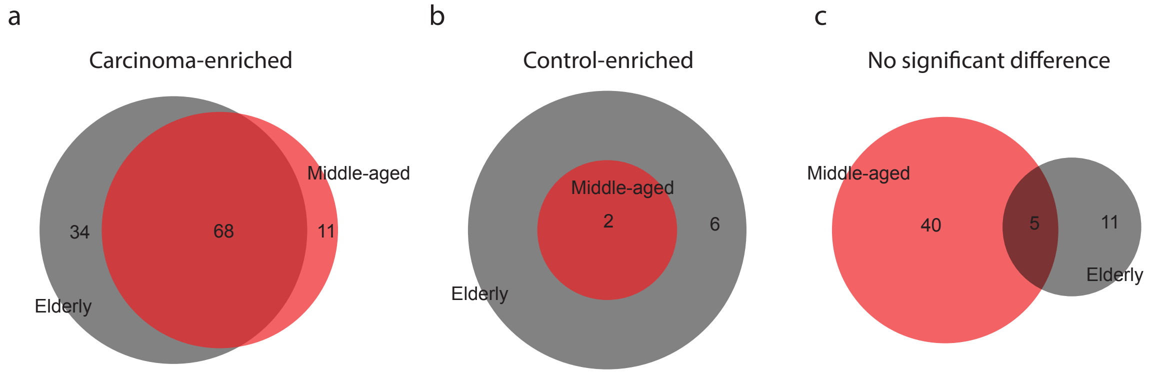
Supplementary Figure 3: **MLGs differentially enriched in faecal samples from colorectal carcinoma patients of different histology (Supplementary Datasetset 1).** MLGs (>100 genes) whose relative abundances were significantly different between carcinoma in situ and adenocarcinoma are shown ($p < 0.05$, Wilcoxon-rank sum test, FDR = 0.7216). The name of each MLG was coloured according to its direction of enrichment, i.e. red if higher in carcinomas than in controls.



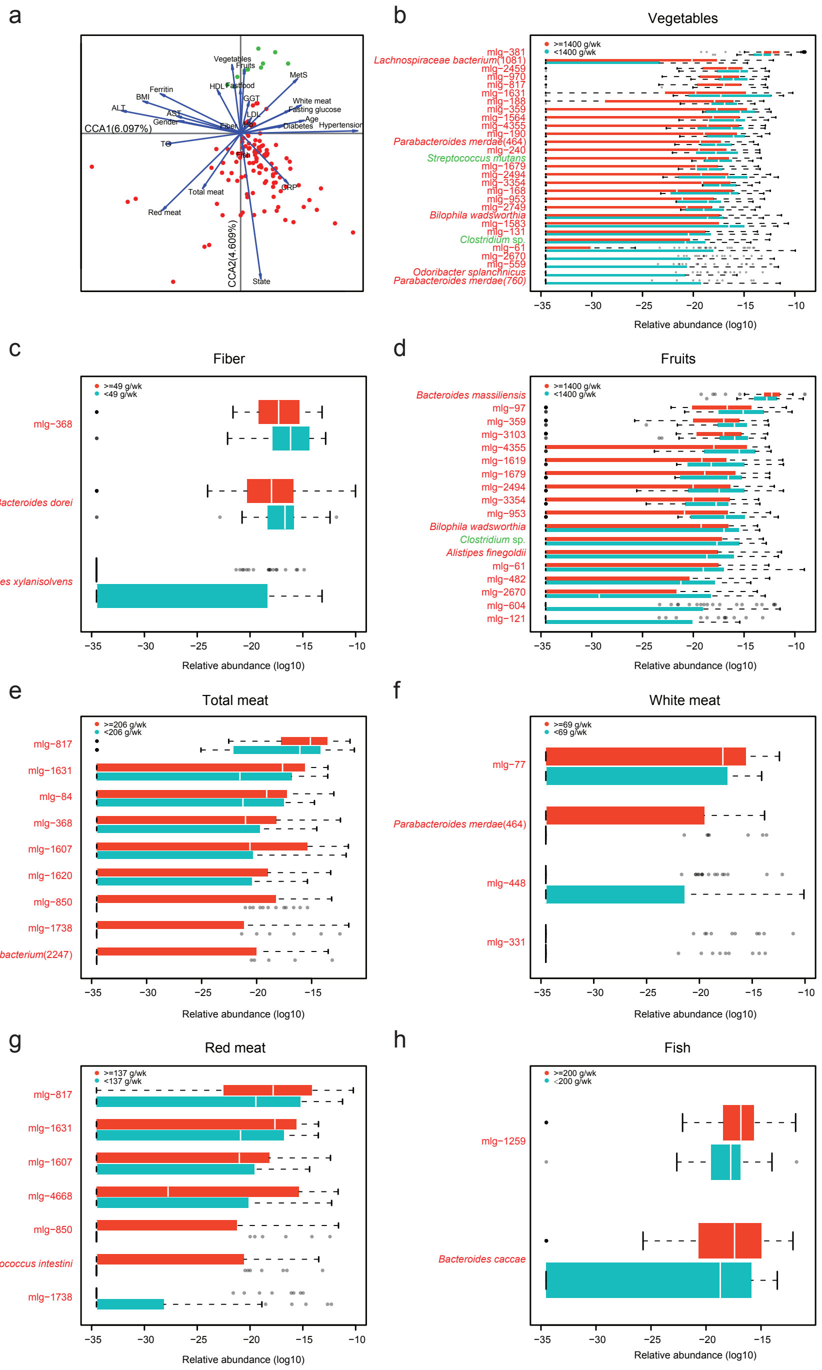
Supplementary Figure 4: **MLGs differentially enriched in faecal samples from patients of different colorectal carcinoma stages.**
MLGs (>100 genes) whose relative abundances were significantly different in different carcinoma stages are shown ($p < 0.05$, Kruskal-Wallis test, FDR = 0.531). The blocks were coloured according to relative abundance of the MLGs in each sample. The name of each MLG was coloured according to its direction of enrichment, i.e. red if higher in carcinomas than in controls.



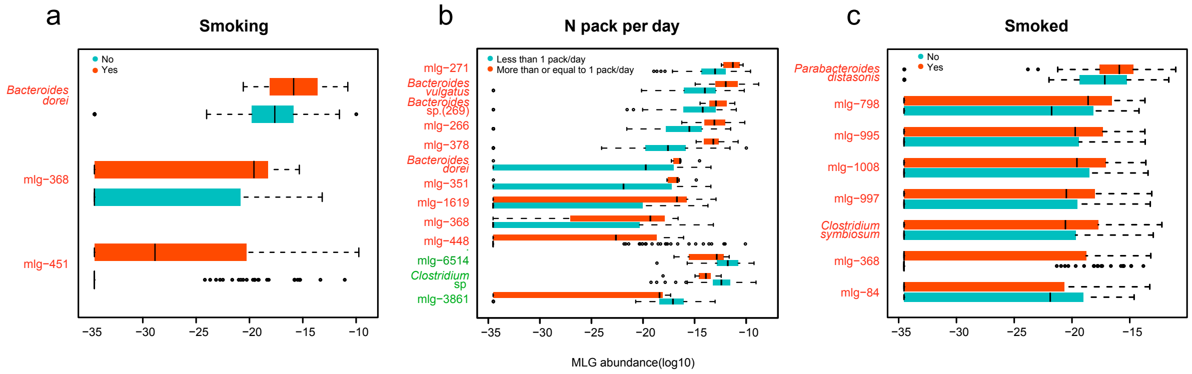
Supplementary Figure 5: **MLGs differentially enriched in faecal samples from patients with colorectal carcinomas at different locations.**
MLGs (>100 genes) whose relative abundances were significantly different in carcinomas of the right colon, left colon, or rectum are shown ($p < 0.05$, Kruskal-Wallis test, FDR = 0.318). The blocks were coloured according to relative abundance of the MLGs in each sample. The name of each MLG was coloured according to its direction of enrichment, i.e. red if higher in carcinomas than in controls.



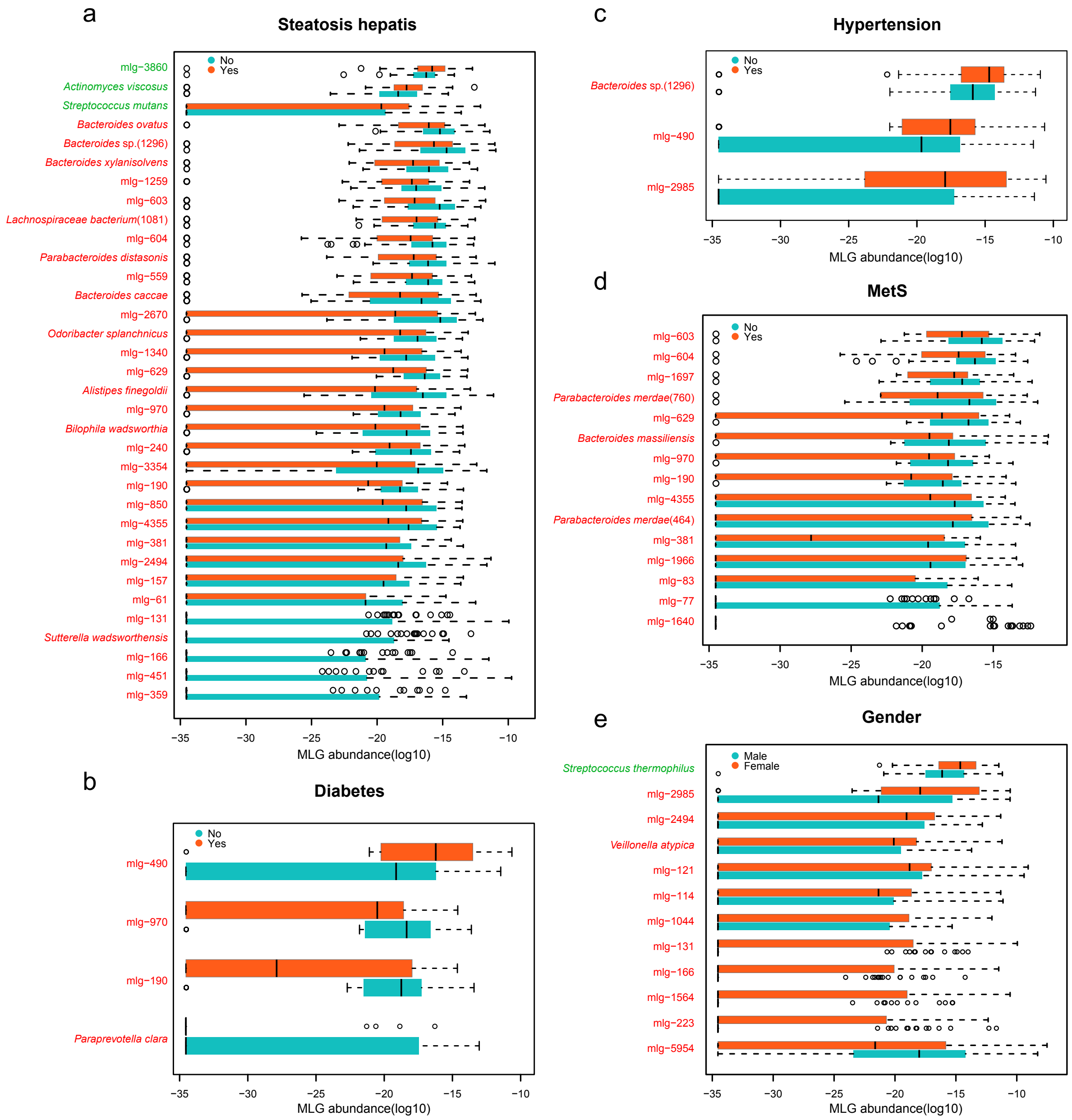
Supplementary Figure 6: **Enrichment of the 126 MLGs in elderly (>65) or middle-aged ($\geq$ 65) carcinoma versus control subjects (p <0.05).**
Wilcoxon rank sum test was performed in the age groups and the p-value was corrected according to the Benjamin-Hochberg procedure (Supplementary Dataset 5).



Supplementary Figure 7: **Contribution of diet and other factors to MLG abundance.** **(a)** Canonical correspondence analysis (CCA) for influence of the factors on the MLGs. Factors missing in >20 samples ('NA' in Supplementary Dataset 1) were not analyzed. Green points, control-enriched MLGs; red points, carcinoma-enriched MLGs. **(b-h)** MLGs differentially enriched between all samples with relatively high or low intake of vegetables **(b)**, fibre **(c)**, fruits **(d)**, total meat **(e)**, white meat **(f)**, red meat **(g)**, and fish **(h)**. P <0.05, Wilcoxon rank sum test, FDR = 0.9719 for vegetables, 0.5618 for fibre, 0.6522 for fruits, 0.3904 for total meat, 0.4900 for white meat, 0.2467 for red meat, and 0.9443 for fish. The control, adenoma, and carcinoma (n = 138) samples were divided into two groups for each dietary component according to the medium level of consumption (Supplementary Dataset 1).



Supplementary Figure 8: **MLGs influenced by smoking.**
MLGs (>100 genes) whose relative abundances were significantly different in currently smoking or not **(a)**, n packs per day **(b)** or ever smoked or not **(c)** are shown ($p < 0.05$, Wilcoxon-rank sum test, FDR = 0.601, 0.432, and 0.695, respectively). Only one person was smoking 2 or more packs a day (Supplementary Dataset 1). The name of each MLG was coloured according to its direction of enrichment, i.e. green if higher in controls than in carcinomas, red if higher in carcinomas than in controls.



Supplementary Figure 9: **MLGs influenced by other non-numerical factors.** **(a)** Steatosis hepatis. **(b)** Diabetes. **(c)** Hypertension. **(d)** Metabolic syndrome (MetS). **(e)** Gender. Only MLGs (>100 genes) whose relative abundances were significantly different between the indicated groups are shown (p <0.05, Wilcoxon-rank test). FDR = 0.182 for steatosis hepatis, 0.761 for diabetes, 0.937 for hypertension, 0.352 for MetS, and 0.506 for gender. The name of each MLG was coloured according to its direction of enrichment, i.e. green if higher in controls than in carcinomas, red if higher in carcinomas than in controls.

Supplementary Methods

Script for DMM-based community types (Figure 2)

```
#For linux system#
#step 1 : fit.rda#####
argv <- commandArgs(T)
if(length(argv)==0){stop("Rscript DMM.fit.R [input1]
input1 : Genus counts profile.
")}
################## library(DirichletMultinomial)
library(lattice)
library(xtable)
library(parallel)
library(vegan)
library(MASS)

count <- read.table(argv[1])
count <- as.matrix(count)
count <- t(count[rowSums(count)!=0,])

fit <- mclapply(1:7, dmn, count=count, verbose=TRUE)
save(fit, file="fit.rda")
#step2: plot figure#######
argv <- commandArgs(T)
if(length(argv)==0){stop("Rscript Rscript DMM.figure.R [input1] [output]
input1 : Genus counts profile.
input2 : It must be fit.rda.
")}
###library packages
library(DirichletMultinomial)
library(lattice)
library(xtable)
library(parallel)
library(vegan)
library(MASS)

count <- (as.matrix(read.table(argv[1])))
count <- count[rowSums(count)!=0,]
load(argv[2])
lplc <- sapply(fit, laplace)
##Model Fit figure.
pdf("Model.Fit.pdf")
plot(lplc, type="b", xlab="Number of Dirichlet Components", ylab="Model Fit")
dev.off()
best <- fit[[which.min(lplc)]]
coll = mixture(best)
col = apply(coll,1,which.max)
##Sample group.
write.table(col,"Sample.group.txt",quote=F,sep="\t",col.names=F)
```

```
num1 <- max(as.numeric(col))
p0 <- fitted(fit[[1]], scale=TRUE)
p2 <- fitted(best, scale=TRUE)
colnames(p2) <- paste("m", 1:num1, sep="")
diff <- rowSums(abs(p2 - as.vector(p0)))
o <- order(diff, decreasing=TRUE)
cdiff <- cumsum(diff[o]) / sum(diff)
df <- head(cbind(Mean=p0[o], p2[o,], diff=diff[o], cdiff), 10)
##Top ten most importance genus.
write.table(df,"difference_in_fit.txt",quote=F,sep="\t")

swiss.x <- as.matrix(count)
swiss.x <- sweep(swiss.x,2,apply(swiss.x,2,sum),"/")
swiss.mds <- metaMDS(t(swiss.x))
###NMDS plot.
pdf("NMDS.plot.pdf")
plot(swiss.mds$points, col = col+1,pch=19)
dev.off()
### Top five most importance genus.
num2 = 5
diff_in_fit = paste(format(sum(df[1:num2,4])*100,digit=4),"%",sep="")
write.table(diff_in_fit,"Percentage_of_difference.txt",
quote=F,sep="\t",col.names=F,row.names="diff_in_fit")
bac <- rownames(df)[1:num2]
dat <- swiss.x[pmatch(bac,rownames(swiss.x)),]
pdf(paste("Top.",num2,".importance.genus.pdf",sep=""),17,5)
par(mfcol=c(1,num2))
for(i in 1:num2){
boxplot(dat[i,]˜factor(col),col=2:8,main=rownames(dat)[i],ylab = "Relative abundance",cex.lab
= 1.5,ylim=c(0,1))
}
dev.off()
```

Script for cross validation of the random forest classifiers (Figure 5)

```
#For windows system#
##ramdomforest.crossvalidation.r##
##Begin##
rfcv1 <-
function (trainx, trainy, cv.fold = 5, scale = "log", step = 0.5,
mtry = function(p) max(1, floor(sqrt(p))), recursive = FALSE,
...)
{
classRF <- is.factor(trainy)
n <- nrow(trainx)
p <- ncol(trainx)
if (scale == "log") {
k <- floor(log(p, base = 1/step))
n.var <- round(p * step^(0:(k - 1)))
same <- diff(n.var) == 0
if (any(same))
n.var <- n.var[-which(same)]
if (!1 %in% n.var)
n.var <- c(n.var, 1)
}
else {
n.var <- seq(from = p, to = 1, by = step)
}
k <- length(n.var)
cv.pred <- vector(k, mode = "list")
for (i in 1:k) cv.pred[[i]] <- rep(0,length(trainy))
if (classRF) {
f <- trainy
}
else {
f <- factor(rep(1:5, length = length(trainy))[order(order(trainy))])
}
nlvl <- table(f)
idx <- numeric(n)
for (i in 1:length(nlvl)) {
idx[which(f == levels(f)[i])] <- sample(rep(1:cv.fold,
length = nlvl[i]))
}
res=list()
for (i in 1:cv.fold) {
all.rf <- randomForest(trainx[idx != i, , drop = FALSE],
trainy[idx != i],importance = TRUE)
aa = predict(all.rf,trainx[idx == i, , drop = FALSE],type="prob")
cv.pred[[1]][idx == i] <- as.numeric(aa[,2])
impvar <- (1:p)[order(all.rf$importance[, 3], decreasing = TRUE)]
res[[i]]=impvar
for (j in 2:k) {
```

```
imp.idx <- impvar[1:n.var[j]]
sub.rf <- randomForest(trainx[idx != i, imp.idx,
drop = FALSE], trainy[idx != i]
)
bb <- predict(sub.rf,trainx[idx ==i,imp.idx, drop = FALSE],type="prob")
cv.pred[[j]][idx == i] <- as.numeric(bb[,2])
if (recursive) {
impvar <- (1:length(imp.idx))[order(sub.rf$importance[,
3], decreasing = TRUE)]
}
NULL
}
NULL
}
if (classRF) {
error.cv <- sapply(cv.pred, function(x) mean(factor(ifelse(x>0.5,1,0))!=trainy))
}
else {
error.cv <- sapply(cv.pred, function(x) mean((trainy -
x)^2))
}
names(error.cv) <- names(cv.pred) <- n.var
list(n.var = n.var, error.cv = error.cv, predicted = cv.pred,res=res)
}
##End##

library(randomForest)
set.seed(999)
par(mfrow = c(1,5))
dat1 <- read.table("MLG.more100gene.profile", head=T,sep="\t",row.names=1)
conf<-read.table("clinical_data.txt",head=T,row.names=1,sep="\t")

cN.prof <- colnames(dat1)
cN.prof <- sub("X","",cN.prof)

rN.conf <- rownames(conf)

gid <- intersect(cN.prof ,rN.conf)
dat1 <- dat1[,pmatch(gid,cN.prof)]
conf <- conf[pmatch(gid,rN.conf),]
dat2<-dat1[,conf$state=="controls" | conf$state=="advanced adenoma"]
conf2<-conf[conf$state=="controls" | conf$state=="advanced adenoma",]
conf2$state=as.factor(as.character(conf2$state))

outcome=conf2$state
outcome<-sub("controls","0",outcome)
outcome<-sub("advanced adenoma","1",outcome)
outcome<-as.factor(outcome)
```

```
dat<-dat2
X <- as.data.frame(t(dat))
X$outcome = outcome

######5*10_crossvalidation####
set.seed(999)
source("ramdomforest.crossvalidation.r")
result <- replicate(5, rfcv1(X[,-ncol(X)], X$outcome, cv.fold=10,step=0.9), simplify=FALSE)
error.cv <- sapply(result, "[[", "error.cv")
matplot(result[[1]]$n.var, cbind(rowMeans(error.cv), error.cv), type="l",
lwd=c(2, rep(1, ncol(error.cv))), col=1, lty=1, log="x",
xlab="Number of variables", ylab="CV Error")
abline(v=10,col="pink",lwd=2)
error.cv.cbm<-cbind(rowMeans(error.cv), error.cv)
cutoff<-min (error.cv.cbm[,1])+sd(error.cv.cbm[,1])
error.cv.cbm[error.cv.cbm[,1]<cutoff,]

#####pick 10 marker by corossvalidation#######
k=1
b <- matrix(0,ncol=126,nrow=50)
for(i in 1:5){
for(j in 1:10){
b[k,]<-result[[i]]$res[[j]]
k=k+1
}
}
mlg.list<-b[,1:10]
list<-c()
k=1
for(i in 1:10){
for(j in 1:50){
list[k]<-mlg.list[j,i]
k=k+1
}
}
mlg.sort<-as.matrix(table(list))
mlg.sort<-mlg.sort[rev(order(mlg.sort[,1])),]
pick<- as.numeric(names(head(mlg.sort,10)))

tmp=X[,-ncol(X)]
mlg.pick<-colnames(tmp)[pick]
write.table(mlg.pick,"cross_validation_pick_10_in_con2aa.txt",
sep="\t",quote=F)
###train.set
train1 <- X[,c(pick,127)]
set.seed(999)
train1 <-data.frame(train1)
train1.rf <- randomForest(outcome ˜ ., data =train1,
importance = TRUE)
```

```
train1.pre <- predict(train1.rf,type="prob")
p.train<-train1.pre[,2]

boxplot(p.train˜outcome,col=c(3,4),main="Probability of Adenoma")
write.table(p.train,"con2aa.cross_validation.10makr.predict.in.train.txt",
sep="\t",quote=F)

########ROC in train######
library(pROC)
roc(outcome,p.train)

####################
roc1 <- roc(outcome,
p.train,
percent=TRUE,
partial.auc.correct=TRUE,
ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
plot=F, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
)
######################
roc1 <- roc(outcome, p.train,
ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
plot=TRUE, percent=roc1$percent,col=2)

sens.ci <- ci.se(roc1, specificities=seq(0, 100, 5))
plot(sens.ci, type="shape", col=rgb(0,1,0,alpha=0.2))
plot(sens.ci, type="bars")
plot(roc1,col=2,add=T)
legend("bottomright",c(paste("AUC=",round(roc1$ci[2],2),"%"),
paste("95% CI:",round(roc1$ci[1],2),"%-",round(roc1$ci[3],2),"%")))
outcome.train<-outcome

##########test.set###########
dat3<-dat1[,conf$state=="carcinoma"]
dat4<-read.table("9.ecoli.rm.samples.MLG.more100gene.profile")
dat5<-read.table("9.newsamples.MLG.more100gene.profile")
dat5<-cbind(dat3,dat4,dat5)
dat5<-data.frame(t(dat5))
set.seed(999)
predict(train1.rf, dat5)
set.seed(999)
test<-predict(train1.rf, dat5,type="prob")
conf3<-conf[conf$state=="carcinoma",1:2]
conf4<-read.table("9.ecoli.rm.sample.info.txt",sep="\t")
conf5<-read.table("9.newsample.info.txt",sep="\t",head=T,row.names=1)
conf5<-rbind(conf3,conf4,conf5)

rN.test <- rownames(test)
rN.test <- sub("X","",rN.test)
rN.conf <- rownames(conf5)
gid <- intersect(rN.test ,rN.conf)
test <- test[pmatch(gid,rN.test),]
conf5 <- conf5[pmatch(gid,rN.conf),]

write.table(test[,2],"con2aa.cross_validation.10makr.predict.in.test.txt",
sep="\t",quote=F)
#########plot######

train_type<-conf5[,1]
col1=c()
col1[train_type=="controls"]="green"
col1[train_type=="carcinoma"]="red"
col1[train_type=="advanced adenoma"]="blue"

plot(rank(test[,2]),test[,2],col=col1,pch=16,
xlab="",ylab="Probability of Carcinoma",main="Testset")
abline(h=0.5)

########test.ROC##########
outcome=conf5$state
outcome<-sub("controls","0",outcome)
outcome<-sub("carcinoma","1",outcome)
outcome<-sub("advanced adenoma","1",outcome)

library(pROC)
roc(outcome,test[,2])
###################
roc1 <- roc(outcome,
test[,2],
percent=TRUE,
partial.auc.correct=TRUE,
ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
plot=F, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
)
#####################
roc1 <- roc(outcome, test[,2],
ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
plot=TRUE, percent=roc1$percent,col=2)

sens.ci <- ci.se(roc1, specificities=seq(0, 100, 5))
plot(sens.ci, type="shape", col=rgb(0,1,0,alpha=0.2))
plot(sens.ci, type="bars")
plot(roc1,col=2,add=T)
legend("bottomright",c(paste("AUC=",round(roc1$ci[2],2),"%"),
paste("95% CI:",round(roc1$ci[1],2),"%-",round(roc1$ci[3],2),"%")))
outcome.train<-outcome
```

Script for PERMANOVA of influencing factors on microbial gene profile (Supplementary Dataset 6)

```
#For linux system#
argv <- commandArgs(T)
    if(length(argv) == 0){stop("Rscript permanova_single_adjust.R [input1] [input2] [prefix]
input1: Distance Matrix
input2: Phenotype
prefix: eg: Bray Euclid")
}

adonis1<-function (formula, data = NULL, permutations = 999, method = "bray",
strata = NULL, contr.unordered = "contr.sum", contr.ordered = "contr.poly")
{

TOL <- 1e-07
Terms <- terms(formula, data = data)
lhs <- formula[[2]]
lhs <- eval(lhs, data, parent.frame())
formula[[2]] <- NULL
rhs.frame <- model.frame(formula, data, drop.unused.levels = TRUE)
op.c <- options()$contrasts
options(contrasts = c(contr.unordered, contr.ordered))
rhs <- model.matrix(formula, rhs.frame)
options(contrasts = op.c)
grps <- attr(rhs, "assign")
qrhs <- qr(rhs)
rhs <- rhs[, qrhs$pivot, drop = FALSE]
rhs <- rhs[, 1:qrhs$rank, drop = FALSE]
grps <- grps[qrhs$pivot][1:qrhs$rank]
u.grps <- unique(grps)
nterms <- length(u.grps) - 1
H.s <- lapply(2:length(u.grps), function(j) {
Xj <- rhs[, grps %in% u.grps[1:j]]
qrX <- qr(Xj, tol = TOL)
Q <- qr.Q(qrX)
tcrossprod(Q[, 1:qrX$rank])
})
if (inherits(lhs, "dist")) {
if (any(lhs <-TOL))
stop("dissimilarities must be non-negative")
dmat <- as.matrix(lhs^2)
}else {
dist.lhs <- as.matrix(vegdist(lhs, method = method))
dmat <- dist.lhs^2
write.table(dist.lhs,"sample.dist.table",sep="\t",quote=F)
#########output the distance matrix#############
}
n <- nrow(dmat)
```

```
I <- diag(n)
ones <- matrix(1, nrow = n)
A <- -(dmat)/2
G <- -0.5 * dmat %*% (I - ones %*% t(ones)/n)
SS.Exp.comb <- sapply(H.s, function(hat) sum(G * t(hat)))
SS.Exp.each <- c(SS.Exp.comb - c(0, SS.Exp.comb[-nterms]))
H.snterm <- H.s[[nterms]]
if (length(H.s) > 1)
for (i in length(H.s):2) H.s[[i]] <- H.s[[i]] - H.s[[i -
1]]
SS.Res <- sum(G * t(I - H.snterm))
df.Exp <- sapply(u.grps[-1], function(i) sum(grps == i))
df.Res <- n - qrhs$rank
if (inherits(lhs, "dist")) {
beta.sites <- qr.coef(qrhs, as.matrix(lhs))
beta.spp <- NULL
}else {
beta.sites <- qr.coef(qrhs, dist.lhs)
beta.spp <- qr.coef(qrhs, as.matrix(lhs))
}
colnames(beta.spp) <- colnames(lhs)
colnames(beta.sites) <- rownames(lhs)
F.Mod <- (SS.Exp.each/df.Exp)/(SS.Res/df.Res)
f.test <- function(tH, G, df.Exp, df.Res, tIH.snterm) {
(sum(G * tH)/df.Exp)/(sum(G * tIH.snterm)/df.Res)
}
SS.perms <- function(H, G, I) {
c(SS.Exp.p = sum(G * t(H)), S.Res.p = sum(G * t(I - H)))
}
if (missing(strata))
strata <- NULL

permuted.index<-function (n, strata)
{
if (missing(strata) || is.null(strata))
out <- sample(n, n)
else {
out <- 1:n
inds <- names(table(strata))
for (is in inds) {
gr <- out[strata == is]
if (length(gr) > 1)
out[gr] <- sample(gr, length(gr))
}
}
out
}

p <- sapply(1:permutations, function(x) permuted.index(n,
strata = strata))
tH.s <- sapply(H.s, t)
tIH.snterm <- t(I - H.snterm)
f.perms <- sapply(1:nterms, function(i) {
sapply(1:permutations, function(j) {
f.test(H.s[[i]], G[p[, j], p[, j]], df.Exp[i], df.Res,
tIH.snterm)
})
})
f.perms <- round(f.perms, 12)
F.Mod <- round(F.Mod, 12)
SumsOfSqs = c(SS.Exp.each, SS.Res, sum(SS.Exp.each) + SS.Res)
tab <- data.frame(Df = c(df.Exp, df.Res, n - 1), SumsOfSqs = SumsOfSqs,
MeanSqs = c(SS.Exp.each/df.Exp, SS.Res/df.Res, NA), F.Model = c(F.Mod,
NA, NA), R2 = SumsOfSqs/SumsOfSqs[length(SumsOfSqs)],
P = c((rowSums(t(f.perms) >= F.Mod) + 1)/(permutations +
1), NA, NA))
rownames(tab) <- c(attr(attr(rhs.frame, "terms"), "term.labels")[u.grps],
"Residuals", "Total")
colnames(tab)[ncol(tab)] <- "Pr(>F)"
class(tab) <- c("anova", class(tab))
tab
}

library(vegan)
X <- read.table(argv[2],sep="\t",head=T,row.names=1)
X <- data.frame(X)

sampledist <- read.table(argv[1])
dim(X)
lab <- colnames(X)
prefix <- as.vector(argv[3])
colname = c("phenotype","Df","SumsOfSqs","MeanSqs","F.Model","R2","Pr(>F)")
write.table(t(colname),paste("perm_",prefix,"_","single.txt",sep=""),
quote=F,sep="\t",append=T,col.names=F,row.names=F)
for(i in 1:ncol(X)){
set.seed(0)
m=which(!is.na(X[,i]))
sampledist_tem=as.dist(sampledist[m,m])
if(length( levels(as.factor(X[m,i])))==1){next}
if(length( levels(as.factor(X[m,i])))<=5){X[,i]=as.factor(X[,i])}
run=paste("X_tem=data.frame(",lab[i],"=X[m,i])",sep="")
eval(parse(text = run))
run <- paste("adonis1(","sampledist_tem˜",lab[i],",data=X_tem,","permutations=9999)",sep="")
tab = eval(parse(text = run))
tabb <- tab[1,]
names(tabb) = NULL
write.table(tabb,paste("perm_",prefix,"_","single.txt",sep=""),
quote=F,sep="\t",append=T)}
```

Script for CCA (Figure 6a)

```
#For windows system#
library(vegan)
   prof<-read.table("MLG.more100gene.profile")
prof<-t(prof)
dca<-decorana(prof)
summary(dca) ####cca or rda#####
plot(dca)

conf<-read.table("clinical_data.txt",sep="\t",head=T,row.names=1)
rN.prof<-rownames(prof)
rN.conf<-rownames(conf)
rN.prof<-sub("X","",rN.prof)
gid<-pmatch(rN.conf,rN.prof)
rownames(prof)<-rN.prof
###remove NA######
NA.n=apply(conf,2,function(x){sum(is.na(x))})
en<-conf[,-c(4,5,6,8,12,14,16,25,26,27,36,37,38,39)]
en.f<-en[!is.na(en$redmeat)&!is.na(en$TG)&!is.na(en$ferritin )&!is.na(conf$fastingglucose),]
prof.f<-prof[!is.na(en$redmeat)&!is.na(en$TG)&!is.na(en$ferritin )&!is.na(conf$fastingglucose),]

en.f$state<-sub("advanced adenoma","1",en.f$state)
en.f$state<-sub("carcinoma","2",en.f$state)
en.f$state<-sub("controls","0",en.f$state)
en.f$state<-as.numeric(en.f$state)
en.cca <-cca(prof.f,en.f)
anova(en.cca)
plot(en.cca)

#####plot fig####
enrich<-read.table("con2cc.MLG.more100gene.profile.wilcox.enrichment.test")
pmatch(rownames(en.cca$CCA$v[,1:2]),enrich$V1)
col<-enrich$V5
col<-sub("1","2",col)
col<-sub("0","3",col)

plot(en.cca$CCA$v[,1:2],col=col,pch=20)
abline(h=0,col = "lightgray", lty = 3)
abline(v=0,col = "lightgray", lty = 3)

for(i in 1:25){
arrows(0,0,en.cca$CCA$biplot[i,1]*4.5,en.cca$CCA$biplot[i,2]*4.5,col=4,angle=10,length=0.1)
}
text(en.cca$CCA$biplot[,1]*4.5,en.cca$CCA$biplot[,2]*4.5,rownames(en.cca$CCA$biplot),cex=0.9)
```

Supplementary References

1. Arumugam, M. et al. Enterotypes of the human gut microbiome. Nature 473, 174-80 (2011).