**BACTERIAL COMMUNITY READ NORMALIZATION**

getsetwd("C:/Users/Ester/Desktop/CNR/IntI1/for R/") #define the directory where you keep you script and your data

write("Info for article:", "info.csv") #write some information to a file that you will use for the article.

version<-R.Version() #Fist information which R version are you using?

str(version)

write(version$version.string, "info.csv", append=T) #Write it to the file.

###Prepare OTU table and taxonomy file

OTUList<-read.csv("seqtab.nochimB.csv") #reas in OTU table in csv formate for txt use read.delim

rownames(OTUList)<-OTUList[,1] #OTU names will become row names, [,1] means all the rows from the first col

OTUList<-OTUList[,-1] #remove OTU names form the dataframe

###Normalise read numbers:

readnr<- colSums(OTUList) #make the sum of each column to get the number of reads per sample

summary(readnr)

hist(readnr)

samples<-colnames(OTUList) #extract the colnames from the dataframe to get a vector with the names of the samples

sampleNr<-cbind(samples,readnr) #make a new dataframe by binding the two vectors as columns

sampleNr<-cbind(colnames(OTUList),colSums(OTUList)) #the same thing as the three rows above but coded in a more effient way.

#this is infact your first result that you will need for publication

write.csv(sampleNr, "tableS1.csv") #Write the read numers to a file which will be saved in your working direcotry

tOTUList<-as.data.frame(t(OTUList)) #Transpose the OTU table!

library("GUniFrac")

rareOTU<-Rarefy(tOTUList, depth = min(readnr)) #Rarefaction to number of reads of the sample with the least reads

rffOTU<-as.data.frame(rareOTU$otu.tab.rff) #the output is an array

summary(rowSums(rffOTU)) #if you want to check if it worked make the summary per row they should all be the same...

min<-min(readnr)

write("rarefied to nr of reads:", "info.csv", append=TRUE)

write(min, "info.csv", append=TRUE)

raOTU <- rffOTU[ ,colSums(rffOTU)!=0] #some otus don't have any reads anymore, lets remove them: != means are not equal

traOTU<-t(raOTU) #transpose dataframe

traOTU<-traOTU[order(row.names(traOTU)),] #sort by OTU number

###READ IN TAXONOMY FILE AND ADJUST OTU TABLE

taxonomy<-read.csv("taxa\_file(280F,220R).csv") #taxonomy file

row.names(taxonomy)<-taxonomy[,1] #set row names

taxonomy<-taxonomy[,-1]

STaxa<-taxonomy[order(row.names(taxonomy)),] #!!!Make sure that the OTUs in the table and in the taxonomy file are in the same order

taxaall<-STaxa[row.names(STaxa)%in%row.names(traOTU),] #We removed some OTUs now lets subset taxa to the ones that are still in the OTU table

taxaNA<-subset(taxaall, taxaall$Kingdom!="Archaea")

taxaNC<-subset(taxaNA, taxaNA$Class!="Chloroplast") #Remove OTUs from chloroplasts 16S

taxaNM<-subset(taxaNC, taxaNC$Family!="Mitochondria") #Remove OTUs from mitochondial 16S

taxa<-subset(taxaNM, taxaNM$Phylum!="NA") #Let's kick out OTUs that are not at least identified to a Phylum level

traOTU<-as.data.frame(traOTU[row.names(traOTU)%in%row.names(taxa),] ) #Keep OTUs that are in taxa

write.csv(traOTU, "traOTU.csv")

write.csv(taxa, "taxa.csv")

samples\_otus<-dim(raOTU)

write("Nr of samples and Nr of otus left after cleaning:", "info.csv", append=TRUE)

write(samples\_otus, "info.csv", append=TRUE)

**BACTERIAL COMMUNITY AND CLASS 1 INTEGRON CASSETTE ANALYSIS**

setwd("C:/Users/Admin/Desktop/CNR/IntI1/for R")

vari<-read.csv("complete\_fileR.csv")

vari$Sample <- factor(vari$Sample)

vari$Sample2<-c("LV","LV","LV","RB2","RB2","RB2", "RB3", "RB3", "RB3","LM1","LM1","LM1", "LM2","LM2","LM2", "LM3","LM3","LM3", "RB1", "RB1", "RB1")

raOTU<-read.csv("asv.csv")

row.names(raOTU)<-raOTU[,1]

raOTU<-raOTU[,-1]

library("vegan")

betabray<-vegdist(raOTU,method="bray")

plot(hclust(betabray, method="average"),hang=-1, sub='', xlab='', cex=0.5) #plot cluster analysis of betapair

adonis<-adonis(betabray~vari$Sample+vari$DATE, permutations=9999)

adonis

library("ggplot2")

library("RColorBrewer")

NMDS<-metaMDS(raOTU, distance="bray", k=3)

plot(NMDS)

colNew2<-c("thistle3",'steelblue1',"cadetblue4", "darkslateblue",'#bcf60c','#3cb44b','#808000')

data.scores <- as.data.frame(scores(NMDS)) #Using the scores function from vegan to extract the site scores and convert to a data.frame

data.scores$site <- rownames(data.scores) # create a column of site names, from the rownames of data.scores

Site<-factor(vari$Sample2,levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3"))

cmp1<-ggplot() +

geom\_point(data=data.scores,aes(x=NMDS1,y=NMDS2,color=Site,fill=Site),size=5) + # add the point markers

scale\_fill\_manual(values=colNew2) +

scale\_color\_manual(values=colNew2)+

scale\_shape\_manual(values=c(23,21,24,22))+

coord\_equal() +

theme\_bw()

cmp1

library(cowplot)

a<-plot\_grid(cmp1)

cas<-read.csv("int\_matrixR.csv")

cas<-as.data.frame(cas)

variC<-subset(vari, vari$full\_cassetts=="Y")

rownames(cas)<-cas[,1]

cas<-cas[,-1]

cas<-cas[order(row.names(cas)),]

variC<-variC[order(variC$dev\_name),]

NMDSb<-metaMDS(cas, distance="bray", k=3)

plot(NMDSb)

data.scores <- as.data.frame(scores(NMDSb)) #Using the scores function from vegan to extract the site scores and convert to a data.frame

data.scores$site <- rownames(data.scores) # create a column of site names, from the rownames of data.scores

Site<-factor(variC$Sample2,levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3"))

cmp2<-ggplot() +

geom\_point(data=data.scores,aes(x=NMDS1,y=NMDS2,color=Site,fill=Site),size=5) + # add the point markers

scale\_fill\_manual(values=colNew2) +

scale\_color\_manual(values=colNew2)+

scale\_shape\_manual(values=c(21,24,22))+

coord\_equal() +

theme\_bw()

cmp2

b<-plot\_grid(cmp2)

plot\_grid(a, b, labels=c("A","B"), ncol=1)

plot\_grid(cmp1, cmp2, labels=c("A","B"), ncol=1)

smOTU<-subset(raOTU, rownames(raOTU)%in%rownames(cas))

smOTU<-smOTU[order(row.names(smOTU)),]

distCom<-vegdist(smOTU, method="bray")

distCas<-vegdist(cas, method="bray")

mantel(distCom, distCas)

library("ggraph")

library("reshape2")

cas<-read.csv("int\_matrixR2.csv")

rownames(cas)<-cas[,1]

cas<-cas[,-1]

mcas<-melt(as.matrix(cas))

mcas[mcas==0]<-NA

ggplot(mcas, aes(x=factor(Var1, c("LV August","LV October","LV December","RB1 October","RB2 October","RB2 December","RB3 October","RB3 December","LM1 August","LM1 OCtober","LM2 August","LM2 October","LM3 October")), y=Var2)) +

geom\_point(aes(size = value),alpha=0.4) +

theme(legend.key=element\_blank(), legend.text = element\_text(size = 10, colour ="black"),

legend.title = element\_text(size = 11),

legend.position = "right", axis.text.x = element\_text(angle = 90))+

labs(y="", x="",size = "Relative Abundance (%)")

ac<-read.csv("ARGcount.csv", as.is=F)

colNew3<-c('#bcf60c', '#3cb44b', '#808000',"cadetblue4", '#800000','#e6194b','#fffac8','#f58231','#ffe119',

"pink3", '#fabebe','#ffd8b1',"peachpuff3", '#aaffc3', 'steelblue1',"darkslateblue", '#4363d8',

"slategray4", 'grey83', "snow4","slategray2","thistle3","mediumorchid",'#e6beff')

g1<-ggplot(ac, aes(y=1, x=factor(Sample, levels=c("LV December","RB1 October","RB2 October","RB2 December",

"RB3 October","RB3 December","LM1 October","LM2 August","LM3 October")),fill=Gene)) +

geom\_bar(position="stack", stat="identity")+scale\_fill\_manual(values= colNew3)+

theme(legend.position ="left",axis.text.x = element\_text(angle = 90))+labs(x="Sample",y="Count")

g1

colS2<-c("#bcf60c","thistle3","darkslateblue")

int<-read.csv("intI1\_R.csv")

int$Sample <- factor(int$Sample , levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3"))

int$Month <-factor(int$Month , levels=c("August","October","December"))

Gint<-ggplot(int, aes(x=Sample, y=int1, fill=Month, color=Month))+

geom\_point(size=5, alpha=0.5) +

scale\_fill\_manual(values=colS2)+

scale\_color\_manual(values=colS2)+

theme(legend.position = c(0.85, 0.80))+

labs(y = expression(paste(italic(intI1) ~ "gene copies/16S rRNA gene copy")),x="")

GintA<-Gint + scale\_y\_log10()

GintA

Gint2<-ggplot(int, aes(x=Sample, y=copy.cell, fill=Month, color=Month))+

geom\_point(size=5, alpha=0.5) +

scale\_fill\_manual(values=colS2)+

scale\_color\_manual(values=colS2)+

theme(legend.position = "none")+

labs(y = expression(paste(italic(intI1) ~ "gene copies/bacterial cell")),x="")

GintA2<-Gint2 + scale\_y\_log10()

GintA2

plot\_grid(GintA, GintA2, labels=c("A","B"), ncol=1)

Month<-factor(vari$DATE, c("August", "October","December"))

Gbact<-ggplot(vari, aes(x=factor(Sample2,levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3")), y=bact, group=Month, fill=Month, color=Month ))+

geom\_point(size=5, alpha=0.5)+

scale\_fill\_manual(values=colS2)+

scale\_color\_manual(values=colS2)+

labs(y=expression(paste( "bacterial cells" ~ ml^{-1} )), x="")+

scale\_y\_log10()

Gbact

Gooc<-ggplot(vari, aes(x=factor(Sample2, levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3")), y=factor(DATE , levels=c("December","October","August")), fill= occur)) +

geom\_tile(color="grey76", lwd = 1.5,

linetype = 1)+

scale\_fill\_gradient2(low="white", high="black")+

theme(legend.position = "none", axis.text.y = element\_text(size = 14), axis.text.x = element\_text(size = 14))+

labs(x="",y="")+

geom\_text(aes(label =arg\_abu, colour="white"))+

scale\_color\_manual(values="white")

Gooc

plot\_grid(Gooc, g1, labels=c("A","B"), ncol=1, hjust=0, vjust=1.5,axis="r",

rel\_heights = c(1.5,2),scale=c(0.95,1))

#taxonomy

taxa<-read.csv("taxa.csv")

traOTU<-t(raOTU)

df\_rel<-aggregate.data.frame(traOTU, by=list(taxa$Genus), FUN=sum)

rownames(df\_rel)<-df\_rel[,1]

df\_rel<-df\_rel[,-1]

write.csv(df\_rel, "Genera.csv")

dfabu<-as.matrix(subset(df\_rel, rowSums(df\_rel)>125))

tdfabu<-t(dfabu)

colNew<-c("grey50",'#3cb44b', '#808000', "thistle3", "grey10", "cadetblue4", "darkslateblue",'#f58231', '#9a6324', "tan1", "slategray4", 'grey78', '#000000', "snow4", "slategray2", '#fffac8', '#f032e6', "mediumorchid", '#800000','#e6194b', '#ffe119', "pink3", '#fabebe', '#ffd8b1',"peachpuff3", '#aaffc3', 'steelblue1', '#4363d8', '#46f0f0', '#e6beff', '#911eb4')

corTaxaint<-as.data.frame(cor(log(vari$qPCR), log(tdfabu+1)))

tcorTaxa<-t(corTaxaint)

cori<-subset(tcorTaxa, tcorTaxa>=0.75)

write.csv(cori,"cori\_new.csv")

forPlot<-subset(dfabu,rownames(dfabu)%in%rownames(cori))

tfplot<-t(forPlot)

mdf<-melt(tfplot, variable.names(c("sample","taxa","count")))

mdfint<-cbind(mdf, rep(vari$qPCR,5), rep(vari$Sample,5))

colnames(mdfint)<-c("Sample\_n","Taxa","Value","intI1","Sample")

mdfint$Sample2<-c("LV","LV","LV","RB2","RB2","RB2", "RB3", "RB3", "RB3","LM1","LM1","LM1", "LM2","LM2","LM2", "LM3","LM3","LM3", "RB1", "RB1", "RB1")

mdfint$Sample2<-factor(mdfint$Sample2,levels=c("LV", "RB1", "RB2", "RB3", "LM1","LM2", "LM3"))

ggplot(mdfint , aes(x=log(Value+1), y=log(intI1), color=Sample2)) +

geom\_point(size=4) +

facet\_wrap(~Taxa , dir="h", scales = "free") +

scale\_color\_manual(values = colNew2)+labs(x=expression(paste(log ~ "genus abundance")), y=expression(paste(log ~ italic( intI1))))+

labs(color="Site")

write.csv(corTaxaint, "cortaxaint.csv")

######################

int\_mod<-aov(asin(sqrt(intI1))~Sample+Month, data=int)

performance::check\_model(int\_mod)

summary(int\_mod)

out<-capture.output(summary(int\_mod))

write(out,"stats.txt",append=T)

ph<-TukeyHSD(int\_mod)

out1<-capture.output(ph$Sample)

write(out1,"stats.txt",append=T)

out2<-capture.output(ph$Month)

write(out2,"stats.txt",append=T)

cell\_mod<-aov(asin(sqrt(copy.cell))~Sample+Month, data=int)

check\_model(cell\_mod)

summary(cell\_mod)

out<-capture.output(summary(cell\_mod))

write(out,"stats.txt",append=T)

ph<-TukeyHSD(cell\_mod)

out1<-capture.output(ph$Sample)

write(out1,"stats.txt",append=T)

out2<-capture.output(ph$Month)

write(out2,"stats.txt",append=T)

tdf<-as.data.frame(t(df\_rel))

tdf$Escherichia\_Shigella<-tdf[,256]

psm\_mod<-lm(log(Pseudomonas+1)~Sample+Month,data=tdf)

check\_model(psm\_mod)

Anova(psm\_mod)

out<-capture.output(Anova(psm\_mod))

write(out,"stats.txt",append=T)

ph1<-emmeans(psm\_mod, pairwise ~ Sample)

out1<-capture.output(ph1)

write(out1,"stats.txt",append=T)

ph2<-emmeans(psm\_mod, pairwise ~ Month)

out2<-capture.output(ph2)

write(out2,"stats.txt",append=T)

plot(Pseudomonas~Sample, data=tdf)

plot(Pseudomonas~Month, data=tdf)

arm\_mod<-lm(log(Aeromonas+1)~Sample+Month,data=tdf)

check\_model(arm\_mod)

Anova(arm\_mod)

out<-capture.output(Anova(arm\_mod))

write(out,"stats.txt",append=T)

ph1<-emmeans(arm\_mod, pairwise ~ Sample)

out1<-capture.output(ph1)

write(out1,"stats.txt",append=T)

ph2<-emmeans(arm\_mod, pairwise ~ Month)

out2<-capture.output(ph2)

write(out2,"stats.txt",append=T)

plot(Aeromonas~Sample, data=tdf)

plot(Aeromonas~Month, data=tdf)

prv\_mod<-lm(log(Prevotella\_9+1)~Sample+Month,data=tdf)

check\_model(prv\_mod)

Anova(prv\_mod)

out<-capture.output(Anova(prv\_mod))

write(out,"stats.txt",append=T)

ph1<-emmeans(prv\_mod, pairwise ~ Sample)

out1<-capture.output(ph1)

write(out1,"stats.txt",append=T)

plot(Prevotella\_9~Sample, data=tdf)

plot(Prevotella\_9~Month, data=tdf)

ess\_mod<-lm(log(Escherichia\_Shigella+1)~Sample+Month,data=tdf)

check\_model(ess\_mod)

Anova(ess\_mod)

out<-capture.output(Anova(ess\_mod))

write(out,"stats.txt",append=T)

ph1<-emmeans(ess\_mod, pairwise ~ Sample)

out1<-capture.output(ph1)

write(out1,"stats.txt",append=T)

plot(Escherichia\_Shigella~Sample, data=tdf)

plot(Escherichia\_Shigella~Month, data=tdf)

pld\_mod<-lm(log(Paludibacter+1)~Sample+Month,data=tdf)

check\_model(pld\_mod)

Anova(pld\_mod)

out<-capture.output(Anova(pld\_mod))

write(out,"stats.txt",append=T)

ph1<-emmeans(pld\_mod, pairwise ~ Sample)

out1<-capture.output(ph1)

write(out1,"stats.txt",append=T)

ph2<-emmeans(pld\_mod, pairwise ~ Month)

out2<-capture.output(ph2)

write(out2,"stats.txt",append=T)

plot(Paludibacter~Sample, data=tdf)

plot(Paludibacter~Month, data=tdf)