Analysis of the Antarctic Fur Seal Skin Microbiome

Stefanie Grosser

01/18/2019

This document provides all R codes for the analyses described in Grosser et al. “Fur seal microbiota are shaped by the social and physical environment, show mother-offspring similarities and are associated with host genetic quality”. We hope that sharing this code alongside the paper will be useful for other researchers. If you have any questions about the analyses feel free to contact me at s.grosser[at]biologie.uni-muenchen.de.

We collected skin swabs and genetic samples from 48 Antarctic fur seals (*A. gazella*) mother-offspring pairs from two breeding sites on Bird Island, South Georgia (freshwater beach and special study beach) and used 16S amplicon sequencing to characterise their bacterial communities. We hypothesise (i) that bacterial diversity should be lower at the colony with high breeding density (special study beach) due to the suppressive effects of elevated social stress on microbial communities; and (ii) that mothers and their pups should have similar microbiomes, reflecting their chemical similarity (discovered in a previous study by Stoffel et al. 2015). We additionally genotyped all of the individuals at 50 hypervariable microsatellite loci and regressed multilocus heterozygosity against microbial diversity. According to the leash model of host control, we would expect to find a negative association between genome-wide individual heterozygosity and overall bacterial diversity. For microbiome characterisation the V3-V4 region of the 16S rRNA gene was paired-end sequenced on an Illumina MiSeq instrument. The paired-end reads were merged and clustered into 97% OTUs and an OTU table was generated following the UPARSE pipeline (Usearch v.9.2.64).

Read counts

Because sequencing depth can vary between samples, we first visualise the number of read pairs sequenced per sample and the number of sequences that were successfully merged per sample. The highest read pair count is 157,204 (mother-M19), and the lowest 9,607 (pup-P39).

```
library(ggplot2)

## Read table containing information about the collected st
atistics during OTU table generation
stats.tab<-read.table("./AFSmicrobiome_SI_SequencingStatsFi
le_Rinput_DatasetS4.txt", sep="\t", header=T)

## Plot reads per sample. Total number of reads pairs is pl
otted in lightgray; number of merged read pairs is plotted
in darkgray

beach_labels <- c(FWB = "Freshwater Beach", SSB = "Special
Study Beach")
#age_labels <- c(Mother = "Mothers", Pup = "Pups")

ggplot()+
```

```
  facet_grid(.~Beach, drop=TRUE,space="free",scales="free",
labeller=labeller(Beach=beach_labels)) +
  geom_bar(data=stats.tab,aes(x=SampleID, y=TotalReads),sta
t="identity",fill="lightgrey", colour="darkgrey",width=.7)+
  geom_bar(data=stats.tab,aes(x=SampleID, y=ReadsMerged),st
at="identity",fill="darkgrey", colour="darkgrey",width=.7)+
  ylab("No. of read pairs")+
  xlab("Sample ID")+
  theme_bw()+
  guides(fill=FALSE) +
  theme(panel.grid.major = element_blank(), panel.grid.mino
r = element_blank())+
  theme(axis.text.x = element_text(angle=90, size=5), axis.
title.x = element_text(margin = margin(5, 0, 0, 0)))+
  theme(axis.title.y=element_text(margin = margin(0, 15, 0,
 0)), axis.text.y = element_text(size=10))
```

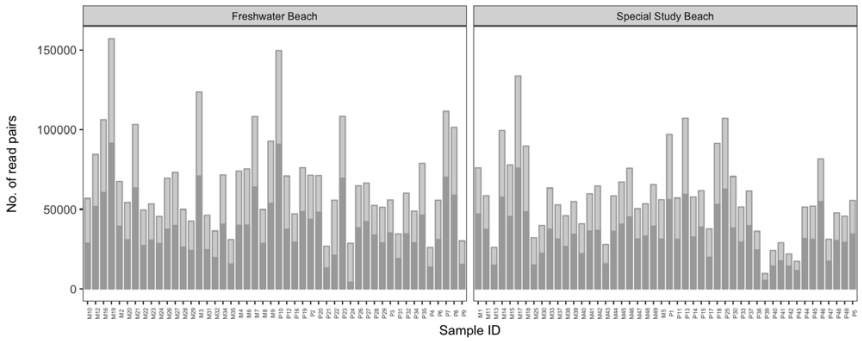


Figure 1. Number of read pairs per sample. Total number of paired-end raw reads is shown in lightgray, and the number of merged reads is shown in darkgray. M samples represent mothers, P samples represent pups. Matching numbers belong to a mother-pup pair.

Individual relatedness

Before analysing the microbiome data of the mother-pup pairs, genetic relatedness is calculated from 50 microsatellites (tested for LD and HWE). Relatedness is calculated with the package “related” following the author’s tutorial. Individual P22 is excluded from the analysis due to large amounts of missing data.

```
library(related)
library(gridExtra)
library(reshape2)
library(dplyr)

## Load genotype data (IMPORTANT NOTE: delete the header ro
w containing loci names before loading!)
msats <- readgenotypedata("./AFSmicrobiome_SI_Microsatellit
eGenotypes50_P22removed_colnames_Rinput_DatasetS5.txt")

# ## Compare the different estimators
# comp <- compareestimators(msats, 100)
# wang        0.951875  --> use Wang
# lynchli       0.950463
# lynchrd       0.933553
# quellergt 0.949013
```

```
## Simulate for 100 individuals to assess power of the anal
ysis
sim <- familysim(msats$freqs, 100)
relsim <- coancestry(sim , wang = 1)
```

```
##    user  system elapsed
##  38.027   2.153  51.701
##
## Reading output files into data.frames... Done!
```

```
relsim <- cleanuprvals(relsim$relatedness , 100)
## Extract only the column containing the wang estimates
relvalues <- as.numeric(relsim[,"wang"])
label1  <- rep("PO", 100)
label2  <- rep("Full", 100)
label3  <- rep("Half", 100)
label4  <- rep("Unrelated", 100)
labels <- c(label1 , label2 , label3 , label4)
relsimtab <- as.data.frame(cbind(relvalues,labels),stringsA
sFactors=FALSE)
relsimtab$relvalues <- as.numeric(relsimtab$relvalues)

## Calculate relatedness (wang estimator) for the fur seal
individuals
rel <- coancestry(msats$gdata, wang = 1)
```

```
##    user  system elapsed
##   0.511   0.047   0.906
##
## Reading output files into data.frames... Done!
```

```
relvals <- rel$relatedness[,c("pair.no","ind1.id","ind2.id"
,"wang")]
## write the results to a table and manually add a column d
efining status of a pair as "unrel" or "pair"
# write.table(relvals, "./relatednessWang50Msats_P22removed
.txt",sep = "\t",quote = FALSE)
relvals2 <- read.table("./AFSmicrobiome_SI_relatednessWang5
0Msats_Rinput_DatasetS6.txt",sep = "\t",header=TRUE)
```

```
## Boxplots for the simulation results
q <- ggplot(relsimtab, aes(x=labels, y=relvalues)) +
      geom_boxplot(fill="lightgrey") +
      theme(legend.position='none') +
      theme_bw() +
      theme(panel.grid.major = element_blank(),panel.grid.m
inor = element_blank()) +
      ylab("Relatedness Estimate (Wang)")+
      xlab("Relatedness Category")+
      coord_cartesian(ylim = c(-0.3, 0.7))+
      theme(axis.text.x = element_text(size=10), axis.title
.x = element_text(margin = margin(10, 0, 0, 0)))+
      theme(axis.title.y=element_text(margin = margin(0, 15
, 0, 0)), axis.text.y = element_text(size=10))

## Make a boxplot for the pairs and unrelated categories an
```

```
d mark the outlier points with the number of the comparison
.
is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - 1.5 * IQR(x) | x > quantil
e(x, 0.75) + 1.5 * IQR(x))
}
relvals2[,"pairNames"] <-  paste(relvals2[,2], relvals2[,3]
, sep="")

b <-relvals2 %>%
  group_by(pairs) %>%
  mutate(outlier = ifelse(is_outlier(wang), pairNames , as.
numeric(NA))) %>%
  ggplot(., aes(x = factor(pairs), y = wang)) +
  geom_boxplot(fill="lightgrey") +
  xlab("Relatedness Category")+
  ylab("")+
  theme_bw() +
  theme(panel.grid.major = element_blank(),panel.grid.minor
 = element_blank()) +
  coord_cartesian(ylim = c(-0.3, 0.7))+
  theme(axis.text.x = element_text(size=10), axis.title.x =
 element_text(margin = margin(10, 0, 0, 0)))+
  theme(axis.title.y=element_text(margin = margin(0, 15, 0,
 0)), axis.text.y = element_text(size=10))+
  geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.
3,size=2)

grid.arrange(q,b, ncol=2)
```

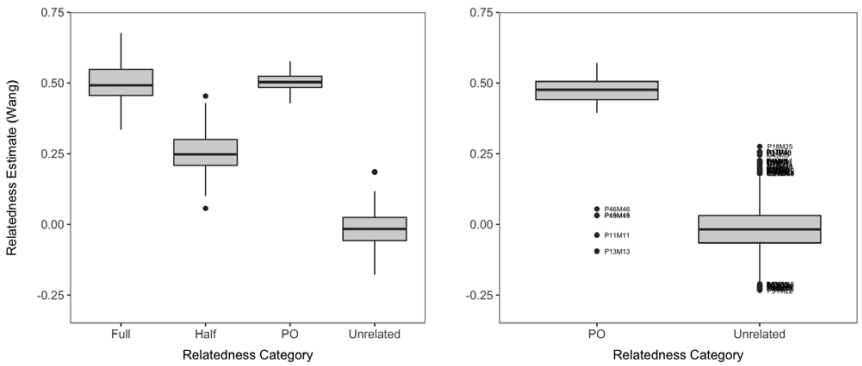


Figure 2. Pairwise relatedness estimates. The left panel shows relatedness values for simulated pairs of known relatedness. Estimates for Antarctic fur seal individuals are shown in the right panel. Estimates are divided into expected mother-pup pair (PO) and unrelated pairwise comparisons.

The wang estimator seems to be suitable to reliably distinguish parent-offspring pairs from unrelated individual pairs. The results show that five of the apparent mother-pup pairs are not related. These five pairs are all from special study beach (high-density colony). This suggests, that pairs have been wrongly identified in the field (allo-suckling occurs in this species). Pairs identified as unrelated by this analysis are: Pair49, Pair46, Pair15, Pair13, Pair11. An additional parentage analsis with the software Colony also confirms these pairs to be unrelated. Based on these results the five pairs will be treated as unrelated in analyses that require pair information.

The *A. gazella* skin microbiome

```
## Convert the UPARSE .sintax classification table into a d
ata frame
## This script was kindly provided by Dr. Ulrich Knief

## Sintax format:
# Otu1  d:Bacteria(1.0000),p:"Proteobacteria"(1.0000),c:Gam
maproteobacteria(1.0000),o:Pseudomonadales(1.0000),f:Moraxe
llaceae(1.0000),g:Psychrobacter(1.0000),s:Psychrobacter_mar
itimus(0.3100) +   d:Bacteria,p:"Proteobacteria",c:Gammapro
teobacteria,o:Pseudomonadales,f:Moraxellaceae,g:Psychrobact
er

## Phyloseq required format
#      Domain Phylum Class Order Family Genus Species
# OTU1  "b"    "b"    "v"   "v"   "l"    "n"   "j"

path = "./"
separators <- c("d","p","c","o","f","g","s")
dat <- read.table(paste(path, "AFSmicrobiome_SI_otuRDPclass
ification_Rinput_DatasetS7.sintax", sep=""), header=FALSE,
sep="+", stringsAsFactors=FALSE)

## Get OTU IDs
OTUs <- unlist(lapply(strsplit(as.character(dat$V1),"\t",fi
xed = TRUE),"[[",1))

## Create data frame
tab <- data.frame(matrix(rep(NA,8*length(OTUs)),ncol=8))
colnames(tab) <- c("OTU","Domain","Phylum","Class","Order",
"Family","Genus","Species")

## Loop over all rows
for(i in 1:nrow(tab)) {
  out <- unlist(strsplit(as.character(dat$V2), ",", fixed =
 TRUE)[i])

  ## Find and add missing values
  Add <- which(!(separators %in% gsub("\t", "", unlist(lapp
ly(strsplit(as.character(out), ":", fixed = TRUE),"[[",1)))
))
  if(length(Add)>0) { for(j in 1:length(Add)) { out <- c(ou
t[1:(Add[j]-1)],paste(separators[Add[j]],":NA",sep=""),out[
Add[j]:length(out)]); out <- out[1:7] }}

  ## If missing values occur always on the right, this will
 work:
  ##    while(length(out) < 7) { out <- c(out,":NA") }
  out <- unlist(lapply(strsplit(as.character(out), ":", fix
ed = TRUE),"[[",2))
  tab[i, ] <- c(OTUs[i],out)
}

# write.table(tab,paste(path, "AFSmicrobiome_SI_otuRDPclass
ification_phyloseqIn_Rinput_DatasetS8.txt", sep=""), append
=FALSE, row.names=FALSE, col.names=TRUE, sep="\t", quote=FA
LSE, eol="\n")

## Manually change "_" to a space and remove "/Chloroplast"
 from the phylum column annotation "Cyanobacteria/Chloropla
```

```
st" for better readability of plots (The Phylum level annot
ation is always Cyanobacteria/Chloroplast for all Cyanobact
eria. However, none of the sequences should be chloroplast
derived in this analysis as they have been filtered after t
he OTU clustering.)
```

```
library(phyloseq)
library(ggplot2)
library(vegan)
library(dplyr)
library(scales)
library(grid)
library(reshape2)
library(gridExtra)
library(knitr)

## Import the OTU table
otu.tab <- as.matrix(read.table("./AFSmicrobiome_SI_OTUtabl
e_final_trimmed_allSamples_Rinput_DatasetS9.txt", header=T,
 sep= "\t", row.names=1, na.strings=c("","NA")))
## Import the rarefied OTU table (10,000 reads/sample)
otu_rarefied.tab <- as.matrix(read.table("./AFSmicrobiome_S
I_OTUtable_final_trimmed_raref10000_Rinput_DatasetS10.txt",
 header=T, sep= "\t", row.names=1, na.strings=c("","NA")))
## Import the taxonomy table
tax.tab <- as.matrix(read.table("./AFSmicrobiome_SI_otuRDPc
lassification_phyloseqIn_Rinput_DatasetS8.txt", header=T, s
ep= "\t", row.names=1, na.strings=c("","NA")))
## Import sample meta data
meta.tab <- read.table("./AFSmicrobiome_SI_Metadata_allSamp
les_Rinput_DatasetS11.txt", header=T, sep= "\t", row.names=
1, na.strings=c("","NA"))

## Combine all files into a phyloseq object
otu.obj <- otu_table(otu.tab, taxa_are_rows = TRUE)
tax.obj <- tax_table(tax.tab)
meta.obj <- sample_data(meta.tab)
otu_rarefied.obj <- otu_table(otu_rarefied.tab, taxa_are_ro
ws = TRUE)
## Make a phyloseq object
phylo.obj <- phyloseq(otu.obj, tax.obj, meta.obj)
phylo_rarefied.obj <- phyloseq(otu_rarefied.obj, tax.obj, m
eta.obj)

## Look at the phyloseq object
phylo.obj
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 788 taxa and 96 sampl
es ]
## sample_data() Sample Data:       [ 96 samples by 7 sampl
e variables ]
## tax_table()   Taxonomy Table:    [ 788 taxa by 7 taxonom
ic ranks ]
```

```
## Convert the OTU and taxonomy tables into a data frame
otus <- as.data.frame(otu.tab)
otus <- cbind(otus, rownames(otus))
```

```
colnames(otus)[length(otus)] <- "OTU"
tax <- as.data.frame(tax.tab)
tax <- cbind(tax, rownames(tax))
colnames(tax)[length(tax)] <- "OTU"
## Merge the tables
otu_tax <- dplyr::left_join(otus, tax, by = "OTU")
## Calculate the total number of reads for each OTU and add
 a column to the data frame
otu_tax <- cbind(otu_tax, rowSums(otu_tax[,1:96]))
colnames(otu_tax)[105] <- "TotalCount"
## Remove unwanted levels and rename NA columns to "undefin
ed"
otu_tax$Phylum <- droplevels(otu_tax$Phylum)
levels(otu_tax$Phylum) <- c(levels(otu_tax$Phylum), "undefi
ned")
otu_tax$Phylum[is.na(otu_tax$Phylum)] <- "undefined"
levels(otu_tax$Genus) <- c(levels(otu_tax$Genus), "undefine
d")
otu_tax$Genus[is.na(otu_tax$Genus)] <- "undefined"

## Make a table with phyla abundance from the unadjusted co
unts
TotalPhylaCounts <- as.data.frame(aggregate(TotalCount ~ Ph
ylum, FUN = sum, data=otu_tax))
TotalPhylaCounts <- dplyr::arrange(TotalPhylaCounts, desc(T
otalCount))
## Calculate percentages
TotalPhylaCounts <- cbind(TotalPhylaCounts,(TotalPhylaCount
s$TotalCount*100)/sum(TotalPhylaCounts$TotalCount))
colnames(TotalPhylaCounts)[3] <- "Abundance"
## Check if all reads add up to the total read count of 3,1
73,550
#sum(TotalPhylaCounts$TotalCount)
## Count the number of OTUs for each phylum
df <- data.frame(Phylum=character(),NoOTUs = integer(), str
ingsAsFactors=FALSE)
for (i in TotalPhylaCounts$Phylum) {
      len <- length(which(otu_tax$Phylum == i))
      df2 <- as.data.frame(cbind(Phylum = i, NoOTUs = len),
stringsAsFactors=FALSE)
      df <- rbind(df,df2)
}
df$NoOTUs <- as.numeric(df$NoOTUs)
## Add this information to the abundance table
TotalPhylaCounts <- dplyr::left_join(TotalPhylaCounts,df, t
ax, by = "Phylum")
colnames(TotalPhylaCounts) <- c("Phylum","Read Count", "Abu
ndance (%)", "No of. OTUs")

## Do the same for the genus level
TotalGenusCounts <- as.data.frame(aggregate(TotalCount ~ Ge
nus, FUN = sum, data=otu_tax))
TotalGenusCounts <- dplyr::arrange(TotalGenusCounts, desc(T
otalCount))
TotalGenusCounts <- cbind(TotalGenusCounts,(TotalGenusCount
s$TotalCount*100)/sum(TotalGenusCounts$TotalCount))
colnames(TotalGenusCounts)[3] <- "Abundance"
## Count the number of OTUs for each genus
df <- data.frame(Genus=character(),NoOTUs = integer(), stri
ngsAsFactors=FALSE)
```

```
for (i in TotalGenusCounts$Genus) {
      len <- length(which(otu_tax$Genus == i))
      df2 <- as.data.frame(cbind(Genus= i, NoOTUs = len),st
ringsAsFactors=FALSE)
      df <- rbind(df,df2)
}
df$NoOTUs <- as.numeric(df$NoOTUs)
## Add this information to the abundance table
TotalGenusCounts <- dplyr::left_join(TotalGenusCounts,df, t
ax, by = "Genus")
colnames(TotalGenusCounts) <- c("Genus","Read Count", "Abun
dance (%)", "No of. OTUs")
```

First, we examine the presence and abundance of bacterial phyla in the Antarctic fur seal skin microbiome.

```
library(kableExtra)

## Make a table for phyla abundance
kable(TotalPhylaCounts ,format = "html", digits = 2, row.na
mes = FALSE, caption = "Table 1. Bacterial phyla detected i
n the Antarctic fur seal skin microbiome.") %>%
  kable_styling(bootstrap_options = c("condensed","striped"
), full_width = F)
```

Table 1. Bacterial phyla detected in the Antarctic fur seal skin microbiome.

Phylum	Read Count	Abundance (%)	No of. OTUs
Proteobacteria	1231373	38.80	210
Bacteroidetes	695050	21.90	165
Firmicutes	676701	21.32	134
Actinobacteria	360848	11.37	104
Deinococcus-Thermus	32948	1.04	6
Cyanobacteria	32410	1.02	11
Verrucomicrobia	31885	1.00	35
Candidatus Saccharibacteria	29301	0.92	36
Fusobacteria	26987	0.85	9
Acidobacteria	24372	0.77	18
undefined	17517	0.55	33
Planctomycetes	3892	0.12	5
Gemmatimonadetes	2502	0.08	4
Chloroflexi	2386	0.08	5
SR1	1641	0.05	3
Tenericutes	1259	0.04	2
Armatimonadetes	1101	0.03	3
BRC1	760	0.02	2

Microgenomates	263	0.01	1
Synergistetes	183	0.01	1
Ignavibacteriae	171	0.01	1

Then we have a look at the presence and abundance of bacterial genera.

```
library(kableExtra)

## Make a table for genus abundance
kable(TotalGenusCounts, format = "html", digits = 2, row.names = FALSE, caption = "Table 2. Bacterial genera detected in the Antarctic fur seal skin microbiome") %>%
  kable_styling(bootstrap_options = c("condensed","striped"), full_width = F)
```

Table 2. Bacterial genera detected in the Antarctic fur seal skin microbiome

Genus	Read Count	Abundance (%)	No of. OTUs
undefined	870685	27.44	383
Psychrobacter	857218	27.01	8
Chryseobacterium	207721	6.55	9
Jeotgalibaca	91772	2.89	1
Streptococcus	60549	1.91	4
Gelidibacter	56676	1.79	1
Clostridium sensu stricto	56256	1.77	10
Arthrobacter	54460	1.72	2
Clostridium XI	47155	1.49	2
Jeotgalicoccus	46836	1.48	2
Tissierella	44746	1.41	4
Otariodibacter	44194	1.39	2
Flavobacterium	40695	1.28	17
Polaromonas	34744	1.09	3
Deinococcus	32948	1.04	6
Planococcus	30350	0.96	1
Saccharibacteria genera incertae sedis	24111	0.76	26
Nocardioides	22547	0.71	12
Bacteroides	22041	0.69	5
Atopostipes	20383	0.64	2

Aequorivita	18827	0.59	1
Fusobacterium	18589	0.59	5
Agrococcus	18569	0.59	1
Granulosicoccus	18143	0.57	10
Luteolibacter	15901	0.50	11
Acinetobacter	15767	0.50	3
Neisseria	14924	0.47	1
Carnobacterium	14596	0.46	1
Ilumatobacter	14174	0.45	3
Sporosarcina	13841	0.44	1
Aquihabitans	12667	0.40	5
GpIV	11124	0.35	3
Lactobacillus	10918	0.34	4
Atopobacter	10821	0.34	1
Blautia	8895	0.28	2
Erysipelothrix	8753	0.28	2
Anaerococcus	8526	0.27	2
Thermomonas	8498	0.27	1
Escherichia/Shigella	8387	0.26	1
Porphyromonas	8122	0.26	4
Marinobacter	7551	0.24	2
Hymenobacter	7429	0.23	9
Pedobacter	6764	0.21	6
Leifsonia	6290	0.20	1
Dietzia	6261	0.20	1
Arcanobacterium	6135	0.19	1
Polymorphobacter	6036	0.19	1
Staphylococcus	5918	0.19	1
Lacihabitans	5658	0.18	3
Streptobacillus	5386	0.17	2
Eubacterium	5114	0.16	1
Arenibacter	4919	0.15	2
Pricia	4886	0.15	2
Dokdonella	4857	0.15	2
Clostridium XlVb	4760	0.15	2

Corynebacterium	4692	0.15	4
Coxiella	4671	0.15	1
Ornithinicoccus	4509	0.14	1
Helcococcus	4018	0.13	2
Lachnospiracea incertae sedis	3844	0.12	1
Spirosoma	3830	0.12	4
Methylobacterium	3806	0.12	1
Rhodoferax	3731	0.12	1
Capnocytophaga	3315	0.10	1
Brachybacterium	3238	0.10	1
Psychromonas	3238	0.10	1
Bisgaardia	2676	0.08	1
Rhodococcus	2555	0.08	2
Gemmatimonas	2502	0.08	4
Moraxella	2467	0.08	1
Pseudomonas	2461	0.08	2
Brumimicrobium	2448	0.08	2
Globicatella	2399	0.08	1
Ferruginibacter	2396	0.08	1
Sphingorhabdus	2393	0.08	1
Trichococcus	2345	0.07	1
Sphingomonas	2220	0.07	3
Rhodanobacter	2110	0.07	2
Collinsella	2107	0.07	1
Nakamurella	2009	0.06	1
Tomitella	1991	0.06	1
Finegoldia	1923	0.06	1
Butyricicoccus	1919	0.06	1
Lysobacter	1853	0.06	1
Gp6	1823	0.06	2
Blastocatella	1797	0.06	1
Terrimonas	1736	0.05	2
Dyadobacter	1714	0.05	2
Enterococcus	1699	0.05	2
Spartobacteria genera incertae	1625	0.05	5

sedis			
Leadbetterella	1623	0.05	2
Peptoniphilus	1607	0.05	3
Algoriphagus	1483	0.05	1
Roseomonas	1461	0.05	1
Acetoanaerobium	1429	0.05	1
Prosthecobacter	1354	0.04	3
Mycoplasma	1259	0.04	2
Macrococcus	1218	0.04	1
Parvimonas	1204	0.04	1
Terrisporobacter	1201	0.04	1
Labilithrix	1189	0.04	1
Methylotenera	1185	0.04	1
Hydrogenophaga	1175	0.04	1
Coenonia	1152	0.04	1
Sulfitobacter	1136	0.04	1
Gp16	1135	0.04	2
Aeromicrobium	1122	0.04	1
Devosia	1114	0.04	2
Simplicispira	1101	0.03	1
Peptostreptococcus	1072	0.03	1
Jannaschia	1053	0.03	2
Rheinheimera	1029	0.03	1
Rubritalea	950	0.03	3
Catellicoccus	947	0.03	1
Loktanella	923	0.03	2
Desulfobulbus	921	0.03	2
Pseudoalteromonas	901	0.03	1
Armatimonas/Armatimonadetes gp1	889	0.03	2
Alloprevotella	841	0.03	1
Demequina	835	0.03	1
Rhodobacter	835	0.03	1
Nitrosospira	830	0.03	2
Alkanindiges	829	0.03	1

Campylobacter	788	0.02	2
Lewinella	787	0.02	1
Propionivibrio	786	0.02	1
Arenimonas	766	0.02	2
Alistipes	760	0.02	2
Flavimarina	756	0.02	1
Fusibacter	708	0.02	1
Ezakiella	695	0.02	2
Anaerovorax	691	0.02	2
Hahella	675	0.02	1
Facklamia	610	0.02	2
Mesorhizobium	604	0.02	1
Tannerella	580	0.02	1
Sutterella	573	0.02	2
Pyrinomonas	554	0.02	1
Rhizobacter	535	0.02	1
Anaerobiospirillum	530	0.02	2
Aquabacterium	501	0.02	1
Oceanisphaera	495	0.02	1
Thiobacillus	460	0.01	1
Bradymonas	459	0.01	1
Allofustis	450	0.01	1
Paludibacter	447	0.01	2
Proteocatella	437	0.01	1
Faecalibacterium	431	0.01	1
Marmoricola	419	0.01	1
Maribacter	402	0.01	1
Desulfonispora	385	0.01	1
Geobacter	382	0.01	1
Peredibacter	380	0.01	2
Sphingobium	374	0.01	1
Lactococcus	373	0.01	1
Phascolarctobacterium	369	0.01	1
Methylobacter	353	0.01	1
Planktotalea	352	0.01	1

Massilia	346	0.01	1
Nosocomiicoccus	335	0.01	1
Pseudoxanthomonas	332	0.01	1
Rhizobium	329	0.01	1
Undibacterium	329	0.01	1
Deefgea	327	0.01	1
Peptostreptococcaceae incertae sedis	322	0.01	1
Aerococcus	321	0.01	1
Oleispira	317	0.01	1
Gpl	312	0.01	1
Desulfobacterium	306	0.01	1
SR1 genera incertae sedis	299	0.01	1
Cocleimonas	297	0.01	1
Roseiarcus	287	0.01	1
Taibaiella	285	0.01	1
Subdivision3 genera incertae sedis	283	0.01	1
Slackia	247	0.01	1
Romboutsia	244	0.01	1
Vagococcus	243	0.01	1
Dialister	241	0.01	1
Leptotrichia	241	0.01	1
Gp1	237	0.01	1
Terrimicrobium	229	0.01	1
Weissella	216	0.01	1
Mycobacterium	211	0.01	1
Acidiphilium	206	0.01	1
Cryomorpha	204	0.01	1
Polaribacter	203	0.01	1
Terriglobus	195	0.01	1
Arcicella	190	0.01	1
Catonella	190	0.01	1
Iamia	180	0.01	1
Gp17	178	0.01	1
Saccharofermentans	176	0.01	1

Brevundimonas	172	0.01	1
Arcobacter	169	0.01	1
Actinomyces	164	0.01	1
Oscillibacter	164	0.01	1

The core microbiome

Next, we examine the core microbiome. We can define the core microbiome of a group of hosts as the OTUs that are present in a certain percentage of the sampled individuals. Here we want to know which OTUs are present in all of the sampled individuals and in 90% of the sampled individuals.

```
## Get the core microbiome (i.e. all OTUs that are present
in 100% of the samples)
core.tab <- otu_tax[which((apply(otu_tax[,1:96], 1, functio
n(row) all(row !=0 )))=="TRUE"),]
#dim(core.tab) # 29 OTUs are present in all samples
## Make a table for the taxonomic information of the core m
icrobiome
core.tax <- core.tab[,c(97:105)]
## Calculate the percentages
core.tax <- cbind(core.tax,(core.tax$TotalCount*100)/317355
0)
core_print.tax <- core.tax[,c("OTU","Phylum","Family","Genu
s","(core.tax$TotalCount * 100)/3173550")]
colnames(core_print.tax)[5] <- "Abundance (%)"
## Export as tab delimited table
# write.table(core_print.tax, "./AFSCoreMicrobiomeOTUs.txt"
, sep = "\t", quote = FALSE)

## Get the core microbiome present in 90% of the samples
## 90% of samples is 86.4, i.e, OTUs have to be present in
87 or more samples.
core90.tab <- otu_tax[apply(otu_tax[,1:96] != 0, 1, sum) >=
 87, ]
#dim(core90.tab) # 123 OTUs are present in 90% of the sampl
es

## Make a table with the additional OTUs not already presen
t in the 100% core microbiome table
core90.tax <- core90.tab[,c(97:105)]
## Calculate the percentages
core90.tax <- cbind(core90.tax,(core90.tax$TotalCount*100)/
3173550)
## Remove the OTUs that are already present in the 100% cor
e microbiome table
core90_reduced.tax <- core90.tax[-which(core90.tax$OTU %in%
 core.tax$OTU),]
core90_reduced_print.tax <- core90_reduced.tax[,c("OTU","Ph
ylum","Family","Genus","(core90.tax$TotalCount * 100)/31735
50")]
colnames(core90_reduced_print.tax)[5] <- "Abundance (%)"
## Export as tab delimited table
# write.table(core90_reduced_print.tax, "./AFSCoreMicrobiom
```

```
e90OTUs.txt", sep = "\t", quote = FALSE)
```

```
kable(core_print.tax, format = "html", digits = 2, row.names = FALSE, caption = "**Table 3.** Antarctic fur seal skin core microbiome (OTUs present in all sampled individuals)")
 %>%
  kable_styling(bootstrap_options = c("condensed","striped"), full_width = F)
```

Table 3. Antarctic fur seal skin core microbiome (OTUs present in all sampled individuals)

OTU	Phylum	Family	Genus	Abundance (%)
Otu1	Proteobacteria	Moraxellaceae	Psychrobacter	17.28
Otu3	Bacteroidetes	Flavobacteriaceae	Chryseobacterium	5.50
Otu22	Proteobacteria	Moraxellaceae	Psychrobacter	3.79
Otu4	Firmicutes	Carnobacteriaceae	Jeotgalibaca	2.89
Otu2253	Proteobacteria	Moraxellaceae	Psychrobacter	2.83
Otu6	Actinobacteria	Intrasporangiaceae	undefined	2.47
Otu13	Proteobacteria	Moraxellaceae	Psychrobacter	2.12
Otu29	Firmicutes	Streptococcaceae	Streptococcus	1.60
Otu16	Actinobacteria	Propionibacteriaceae	undefined	1.29
Otu15	Firmicutes	Clostridiales Incertae Sedis XI	Tissierella	0.99
Otu31	Actinobacteria	Micrococcaceae	Arthrobacter	0.88
Otu11	Actinobacteria	Micrococcaceae	Arthrobacter	0.84
Otu14	Firmicutes	Clostridiaceae 1	Clostridium sensu stricto	0.83
Otu5	Firmicutes	Peptostreptococcaceae	Clostridium XI	0.80
Otu26	Deinococcus-Thermus	Deinococcaceae	Deinococcus	0.67
Otu18	Bacteroidetes	Flavobacteriaceae	Flavobacterium	0.60
Otu19	Firmicutes	Clostridiaceae 1	Clostridium sensu stricto	0.59
Otu78	Firmicutes	Carnobacteriaceae	Atopostipes	0.59
Otu17	Bacteroidetes	Flavobacteriaceae	undefined	0.59
Otu36	Actinobacteria	Microbacteriaceae	Agrococcus	0.59
Otu401	Proteobacteria	Moraxellaceae	Psychrobacter	0.57
Otu1771	Bacteroidetes	Flavobacteriaceae	Chryseobacterium	0.50
Otu25	Firmicutes	Carnobacteriaceae	Carnobacterium	0.46

Otu32	Firmicutes	Planococcaceae	Sporosarcina	0.44
Otu43	Actinobacteria	Acidimicrobiaceae	Ilumatobacter	0.37
Otu82	Bacteroidetes	Flavobacteriaceae	undefined	0.34
Otu72	Proteobacteria	Rhodobacteraceae	undefined	0.21
Otu145	Actinobacteria	Microbacteriaceae	Leifsonia	0.20
Otu488	Actinobacteria	Nocardioidaceae	Nocardioides	0.11

```
kable(core90_reduced_print.tax, format = "html", digits = 2
, row.names = FALSE, caption = "**Table 4.** Antarctic fur
seal skin extended core microbiome (OTUs present in 90% of
the sampled individuals)") %>%
  kable_styling(bootstrap_options = c("condensed","striped"
), full_width = F)
```

Table 4. Antarctic fur seal skin extended core microbiome (OTUs present in 90% of the sampled individuals)

OTU	Phylum	Family	Genus	Abundance (%)
Otu9	Bacteroidetes	Flavobacteriaceae	Gelidibacter	1.79
Otu7	Bacteroidetes	Flavobacteriaceae	undefined	1.74
Otu12	Firmicutes	Planococcaceae	undefined	1.31
Otu24	Bacteroidetes	Flavobacteriaceae	undefined	1.04
Otu83	Proteobacteria	Comamonadaceae	Polaromonas	0.88
Otu90	Proteobacteria	Pasteurellaceae	Otariodibacter	0.86
Otu60	Proteobacteria	Comamonadaceae	undefined	0.73
Otu8	Firmicutes	Peptostreptococcaceae	Clostridium XI	0.68
Otu995	Proteobacteria	Pasteurellaceae	Otariodibacter	0.54
Otu93	Proteobacteria	Rhodobacteraceae	undefined	0.49
Otu51	Proteobacteria	Neisseriaceae	Neisseria	0.47
Otu38	Firmicutes	Clostridiaceae 1	undefined	0.46
Otu35	Proteobacteria	Moraxellaceae	Acinetobacter	0.44
Otu54	Bacteroidetes	Bacteroidaceae	Bacteroides	0.40
Otu129	Bacteroidetes	Chitinophagaceae	undefined	0.40
Otu80	Firmicutes	Carnobacteriaceae	Atopobacter	0.34
Otu21	Cyanobacteria	Family IV	GpIV	0.32
Otu28	Fusobacteria	Fusobacteriaceae	Fusobacterium	0.29
Otu141	Proteobacteria	Xanthomonadaceae	Thermomonas	0.27
Otu57	Proteobacteria	Enterobacteriaceae	Escherichia/Shigella	0.26

Otu84	Actinobacteria	Intrasporangiaceae	undefined	0.26
Otu50	Actinobacteria	NA	undefined	0.26
Otu59	Firmicutes	Clostridiales Incertae Sedis XI	Anaerococcus	0.26
Otu47	Firmicutes	Clostridiales Incertae Sedis XI	Tissierella	0.25
Otu96	Actinobacteria	Iamiaceae	Aquihabitans	0.25
Otu45	Firmicutes	Ruminococcaceae	undefined	0.24
Otu52	Bacteroidetes	Flavobacteriaceae	Flavobacterium	0.24
Otu53	Firmicutes	Lachnospiraceae	Blautia	0.24
Otu102	Proteobacteria	Burkholderiaceae	undefined	0.24
Otu68	Actinobacteria	Nocardioidaceae	Nocardioides	0.23
Otu101	Bacteroidetes	Flavobacteriaceae	Chryseobacterium	0.23
Otu46	Actinobacteria	NA	undefined	0.21
Otu76	Deinococcus-Thermus	Deinococcaceae	Deinococcus	0.20
Otu86	Actinobacteria	Dietziaceae	Dietzia	0.20
Otu39	Firmicutes	Lachnospiraceae	undefined	0.20
Otu88	Actinobacteria	Actinomycetaceae	Arcanobacterium	0.19
Otu157	Firmicutes	Planococcaceae	undefined	0.19
Otu74	Firmicutes	Erysipelotrichaceae	Erysipelothrix	0.19
Otu2175	Proteobacteria	Comamonadaceae	Polaromonas	0.19
Otu73	Bacteroidetes	Chitinophagaceae	undefined	0.18
Otu339	Proteobacteria	Comamonadaceae	undefined	0.17
Otu49	Firmicutes	Lachnospiraceae	undefined	0.17
Otu228	Bacteroidetes	NA	undefined	0.17
Otu85	Bacteroidetes	Flavobacteriaceae	Flavobacterium	0.16
Otu55	Firmicutes	Eubacteriaceae	Eubacterium	0.16
Otu213	Verrucomicrobia	Verrucomicrobiaceae	Luteolibacter	0.16
Otu146	Bacteroidetes	Bacteroidaceae	Bacteroides	0.16
Otu69	Actinobacteria	Intrasporangiaceae	Ornithinicoccus	0.14
Otu189	Firmicutes	Clostridiaceae 1	Clostridium sensu stricto	0.14
Otu122	Actinobacteria	Micrococcaceae	undefined	0.14
Otu214	Proteobacteria	Moraxellaceae	Psychrobacter	0.13
Otu103	Proteobacteria	Xanthomonadaceae	Dokdonella	0.13

Otu58	Proteobacteria	Methylobacteriaceae	Methylobacterium	0.12
Otu1883	Proteobacteria	Comamonadaceae	Rhodoferax	0.12
Otu209	Actinobacteria	Dermacoccaceae	undefined	0.12
Otu130	Firmicutes	Ruminococcaceae	undefined	0.12
Otu105	Firmicutes	Clostridiales Incertae Sedis XI	Helcococcus	0.11
Otu89	Actinobacteria	NA	undefined	0.11
Otu65	Candidatus Saccharibacteria	NA	Saccharibacteria genera incertae sedis	0.11
Otu115	Firmicutes	Streptococcaceae	Streptococcus	0.11
Otu167	Bacteroidetes	Cytophagaceae	Hymenobacter	0.11
Otu120	Candidatus Saccharibacteria	NA	Saccharibacteria genera incertae sedis	0.11
Otu143	Actinobacteria	Iamiaceae	Aquihabitans	0.10
Otu135	Firmicutes	Erysipelotrichaceae	Erysipelothrix	0.09
Otu75	Fusobacteria	Fusobacteriaceae	undefined	0.09
Otu201	Firmicutes	NA	undefined	0.09
Otu95	Firmicutes	Clostridiaceae 1	Clostridium sensu stricto	0.08
Otu177	Candidatus Saccharibacteria	NA	Saccharibacteria genera incertae sedis	0.08
Otu174	Bacteroidetes	Chitinophagaceae	Ferruginibacter	0.08
Otu108	Actinobacteria	Nocardiaceae	Rhodococcus	0.08
Otu156	Proteobacteria	Xanthomonadaceae	undefined	0.08
Otu110	Proteobacteria	Rhodobacteraceae	undefined	0.07
Otu133	Proteobacteria	NA	undefined	0.07
Otu236	Actinobacteria	Acidimicrobiaceae	Ilumatobacter	0.07
Otu136	Actinobacteria	Coriobacteriaceae	Collinsella	0.07
Otu629	Proteobacteria	Comamonadaceae	undefined	0.06
Otu234	Actinobacteria	Nakamurellaceae	Nakamurella	0.06
Otu207	Actinobacteria	Corynebacterineae incertae sedis	Tomitella	0.06
Otu148	Firmicutes	Lachnospiraceae	Clostridium XlVb	0.06
Otu121	Firmicutes	Ruminococcaceae	Butyricicoccus	0.06
Otu222	Proteobacteria	Xanthomonadaceae	Lysobacter	0.06

Otu1017	Proteobacteria	Xanthomonadaceae	Rhodanobacter	0.05
Otu691	Bacteroidetes	Flavobacteriaceae	Chryseobacterium	0.05
Otu211	Actinobacteria	Nocardiaceae	undefined	0.05
Otu270	Bacteroidetes	Cyclobacteriaceae	Algoriphagus	0.05
Otu1469	Bacteroidetes	Flavobacteriaceae	Chryseobacterium	0.04
Otu1703	Firmicutes	Lachnospiraceae	Blautia	0.04
Otu2423	Bacteroidetes	Chitinophagaceae	undefined	0.04
Otu365	Actinobacteria	Nocardioidaceae	Aeromicrobium	0.04
Otu332	Actinobacteria	NA	undefined	0.03
Otu458	Actinobacteria	Microbacteriaceae	undefined	0.03
Otu308	Actinobacteria	Demequinaceae	Demequina	0.03
Otu429	Actinobacteria	NA	undefined	0.03
Otu612	Actinobacteria	Microbacteriaceae	undefined	0.02

Bacterial abundance

Above we have examined the overall abundance of the different bacterial phyla on Antarctic fur seal skin (Table 1). We now want to visualise the abundance of bacterial phyla for each individual separately. We only display phyla with more than 1% abundance in each sample.

```
library(phyloseq)
library(scales)
library(reshape2)
library(dplyr)

## Plot phyla relative abundance for each individual (Phyla
 with more than 1% abundance).
## Non-normalised data (plot looks the same when rarefied O
TU table is used)
afs_phylum <- phylo.obj %>%
  tax_glom(taxrank = "Phylum") %>%                      # ag
glomerate at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Tr
ansform to rel. abundance
  psmelt() %>%                                          # Me
lt to long format
  filter(Abundance > 0.01) %>%                          # Fi
lter out low abundance taxa
  arrange(desc(Abundance))                              # So
rt data frame alphabetically by phylum

## Define phylum colours
phylum_colors <- c("#673770", "#5F7FC7", "orange","#DA5724"
, "#508578", "#CD9BCD", "#AD6F3B", "#CBD588","#D14285", "#6
52926" , "#C84248", "#8569D5")
## Define plot labels
beach_labels <- c(Freshwater = "FWB", SSB = "SSB")
```

```
age_labels <- c(M = "Mothers", P = "Pups")


ggplot(afs_phylum, aes(x = PlotLabel, y = Abundance, fill =
 Phylum)) +
      facet_grid(Beach~Age, drop=TRUE,space="free",scales="
free",labeller=labeller(Age=age_labels,Beach=beach_labels))
 +
      geom_bar(stat = "identity") +
      theme_bw() +
      scale_fill_manual(values = phylum_colors) +
      theme(axis.text.x = element_blank(),axis.ticks = elem
ent_blank(),axis.title.x = element_blank(), axis.title.y =
element_text(size=14), axis.text.y = element_text(size=12))
+
      guides(fill = guide_legend(keywidth = 1, keyheight =
1)) +
      theme(legend.text = element_text(size = 11),legend.ti
tle = element_text(face="bold")) +
      ylab("Relative abundance (phyla > 1%) \n") +
      theme(panel.grid.major = element_blank(),panel.grid.m
inor = element_blank()) +
      theme(panel.spacing.x = unit(0, "lines"))+
      theme(strip.text.x = element_text(size=12), strip.tex
t.y = element_text(size=12))
```

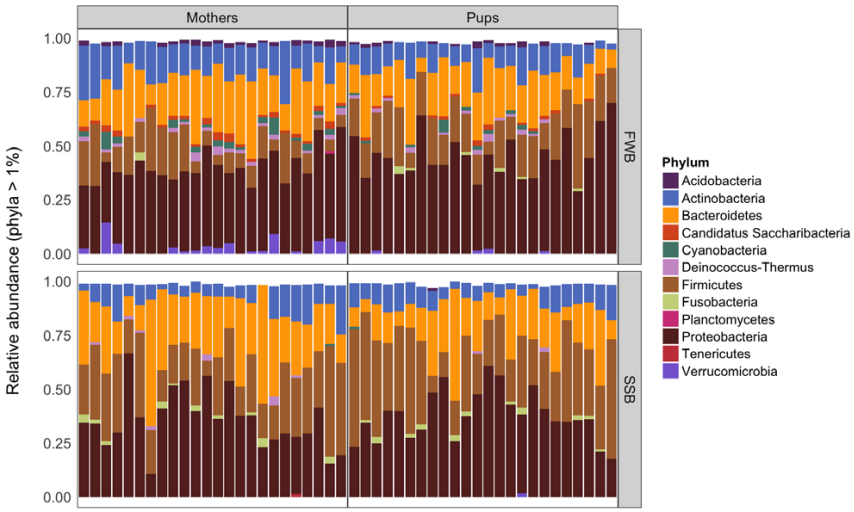


Figure 3. Relative abundance of bacterial phyla present in each sample based on non-normalised counts. For each sample only phyla with an abundance > 1% are shown (not all columns add up to 1.0).

```
## Same plot as above but plotting the rarefied data.

library(phyloseq)
library(scales)
library(reshape2)

## Plot phyla relative abundance for each individual (Phyla
 with more than 1% abundance)
## Rarefied data
afs_phylum_rarefied <- phylo_rarefied.obj %>%
  tax_glom(taxrank = "Phylum") %>%                      # ag
glomerate at phylum level
```

```
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Tr
ansform to rel. abundance
  psmelt() %>%                                        # Me
lt to long format
  filter(Abundance > 0.01) %>%                        # Fi
lter out low abundance taxa
  arrange(desc(Abundance))
     # Sort data frame alphabetically by phylum

phylum_colors <- c("#673770", "#5F7FC7", "orange","#DA5724"
, "#508578", "#CD9BCD", "#AD6F3B", "#CBD588","#D14285", "#6
52926" , "#C84248", "#8569D5")
beach_labels <- c(Freshwater = "Freshwater Beach", SSB = "S
pecial Study Beach")
age_labels <- c(M = "Mothers", P = "Pups")

ggplot(afs_phylum_rarefied, aes(x = PlotLabel, y = Abundanc
e, fill = Phylum)) +
      facet_grid(Beach~Age, drop=TRUE,space="free",scales="
free",labeller=labeller(Age=age_labels,Beach=beach_labels))
 +
      geom_bar(stat = "identity") +
      theme_bw()+
      scale_fill_manual(values = phylum_colors) +
      theme(axis.text.x = element_blank(),axis.ticks = elem
ent_blank(),axis.title.x = element_blank(), axis.title.y =
element_text(size=12), axis.text.y = element_text(size=10))
+
      guides(fill = guide_legend(keywidth = 1, keyheight =
1)) +
      theme(legend.text = element_text(size = 10),legend.ti
tle = element_text(face="bold"))+
      ylab("Relative abundance (phyla > 1%) \n") +
      theme(panel.grid.major = element_blank(),panel.grid.m
inor = element_blank())+
      theme(panel.spacing.x = unit(0, "lines"))+
      theme(strip.text.x = element_text(size=11), strip.tex
t.y = element_text(size=11))
```

Based on the abundance plot we might assume that bacterial diversity is higher at the low density colony freshwater beach. This will be properly tested below.

Alpha diversity

Alpha diversity for each sample was calculated in USEARCH. We calculated the Jost index which is based on a family of metrics called Hill numbers of parameter q. q determines how abundance is weighted. These indices are transformed into the effective number of species. We use q=1, which is equivalent to Shannon entropy and balances differently abundant OTUs. There is an argument about whether to rarefy OTU tables to even number of reads per sample. To test, if uneven read depth affected the calculation of diversity measures we calculated alpha diversity as described above for the non-normalised OTU table, and the OTU table rarefied to 10,000 reads per sample using QIIME (samples with <10,000 reads (P24, P39) were removed from the latter).

```
## Read the table with the alpha-diversity estimates calcul
ated in USEARCH
alpha_div <- read.table("./AFSmicrobiome_SI_alphaDiversity_
Rinput_DatasetS12.txt", header=T, sep= "\t", row.names=1, n
a.strings=c("","NA"))

## Test how well the estimates for the non-normalised and r
arefied OTU tables are correlated
cor.test(alpha_div$jost1_all, alpha_div$jost1_raref)
```

```
##
##  Pearson's product-moment correlation
##
## data:  alpha_div$jost1_all and alpha_div$jost1_raref
## t = 280.56, df = 92, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to
 0
## 95 percent confidence interval:
##  0.9991195 0.9996128
## sample estimates:
##       cor
## 0.9994161
```

```
ggplot(alpha_div, aes(x = jost1_all, y=jost1_raref)) +
  geom_point()+
  stat_smooth(method="lm", se=FALSE)+
  theme_bw()+
  ylab("Effective no. of species (rarefied data)")+
  xlab("Effective no. of species (non-normalised data)")+
  theme(panel.grid.major = element_blank(), panel.grid.mino
r = element_blank())+
  annotate("text", x=60, y=100, label= paste("r==0.99"), pa
rse=TRUE, size=6)
```

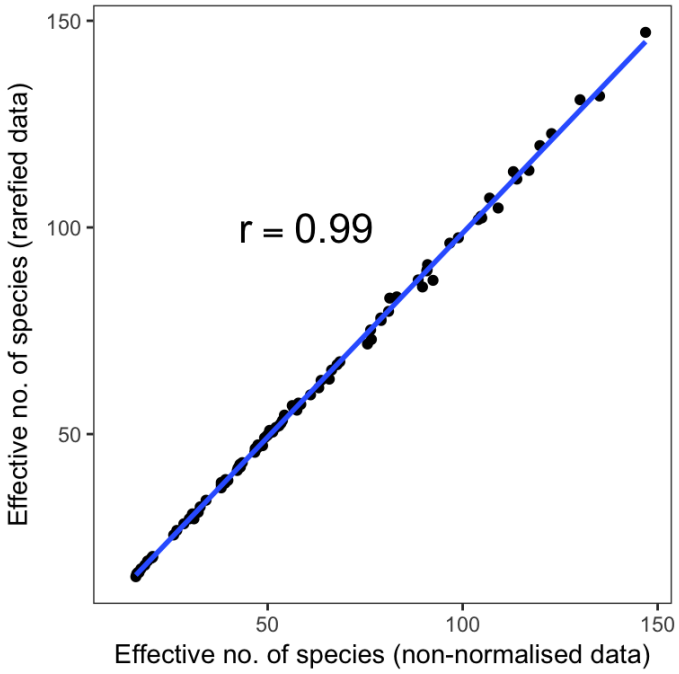


Figure 4. Comparison of alpha diversity estimates calculated from the non-normalised and rarefied OTU tables.

In the next step, we test if there is a difference in alpha-diversity between beaches and the two age groups. We first have to establish if alpha diversity has a gaussian distribution

```
library(gridExtra)
library(ggplot2)
library(dplyr)
library(lme4)

## Check distribution of alpha diversity estimates.
## Effective number of species values can be treaded as a c
ontinuous variable (not a count table anymore)
h <- ggplot(alpha_div, aes(jost1_all)) +
        geom_histogram(binwidth=17,fill="lightgrey", colour
="gray26") +
        theme_bw()+
        ylab("Count")+
        xlab("alpha diversity (Jost 1)")+
        theme(panel.grid.major = element_blank(), panel.gri
d.minor = element_blank())
## -> not normal

##  Square root transform data
th <- ggplot(alpha_div, aes(sqrt(jost1_all))) +
        geom_histogram(binwidth=1, fill="lightgrey", colour
="gray26") +
        theme_bw()+
        ylab("Count")+
        xlab("alpha-diversity (Jost 1)")+
        theme(panel.grid.major = element_blank(), panel.gri
d.minor = element_blank())
## -> square root transformation achieves normality

grid.arrange(h, th, ncol=2)
```

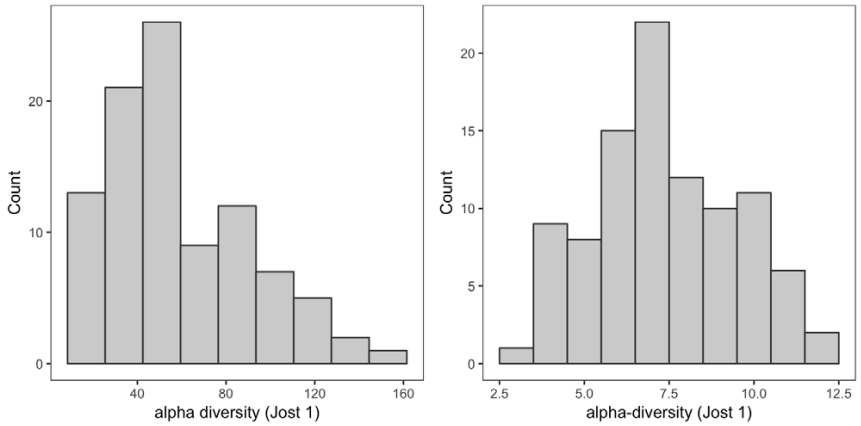


Figure 5. Histgram of untransformed alpha-diversity estimates (left panel) and square-root transformed alpha-diversity estimates (right panel).

Now we can perform linear mixed models (LMM) and likelihood ratio test to test for differences in alpha diversity between the groups. We make use of LMMs to include pair ID as a random effect as we can not consider the two samples of a mother-pup pair as being truly independent. The two individuals of the pairs that were determined to be unrelated by parentage analyses are

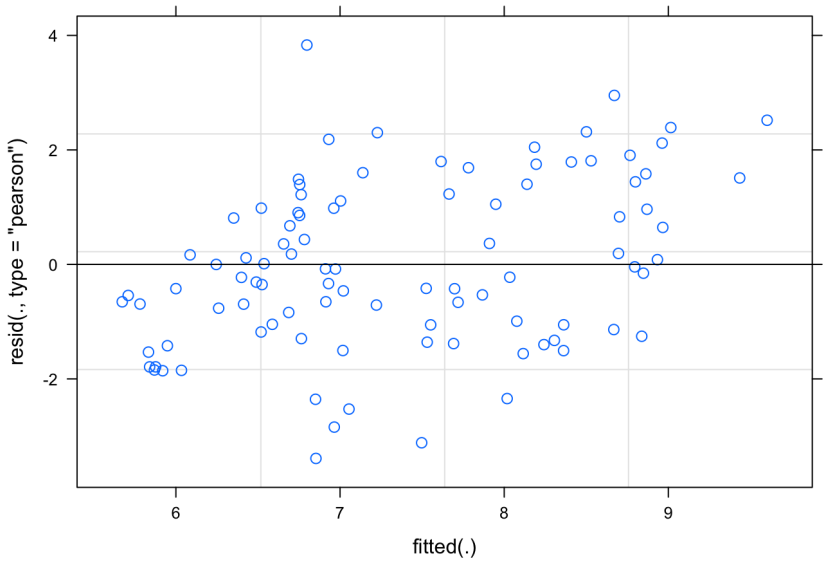
assigned different pair IDs.

```
library(lme4)
library(MuMIn)

## Combine the meta data with the diversity table
meta2.tab <- cbind(meta.tab, SampleID = row.names(meta.tab)
)
alpha_div2 <- cbind(alpha_div, SampleID = row.names(alpha_d
iv))
alpha_model.tab <- dplyr::left_join(meta2.tab, alpha_div2,
by = "SampleID")

## In the model below PairID is used as a random effect to
account for non-independence of a mother-pup pair. Parentag
e analysis revealed five unrelated pairs at SSB. To avoid r
emoving 10 data points from one beach we simply assign diff
erent unique PairIDs to these individuals. Pairs: "M-P49","
M-P46","M-P15","M-P13","M-P11"

## Copy the PairID column
alpha_model_unrel.tab <- cbind(alpha_model.tab, PairID2 = a
lpha_model.tab$PairID)
## Change the PairID of the pups by appending a p to the en
d of the ID
for (i in c("P49","P46","P15","P13","P11")) {
    levels(alpha_model_unrel.tab$PairID2) <- c(levels(alpha
_model_unrel.tab$PairID2),   paste0(alpha_model_unrel.tab$P
airID2[alpha_model_unrel.tab$SampleID==(i)],"p"))
    alpha_model_unrel.tab$PairID2[alpha_model_unrel.tab$Sam
pleID==(i)] <-  paste0(paste0(alpha_model_unrel.tab$PairID2
[alpha_model_unrel.tab$SampleID==(i)],"p"))
}

## Test if alpha diversity is different between the two bea
ches and between mothers and pups
## jost1 = response variable, beach and age = predictors
## PairID used as a random effect to account for non-indepe
ndence of a pair
model1 <- lmer(sqrt(jost1_all) ~ Beach + Age + (1|PairID2),
 data=alpha_model_unrel.tab)
## Get the model output
summary(model1)
```

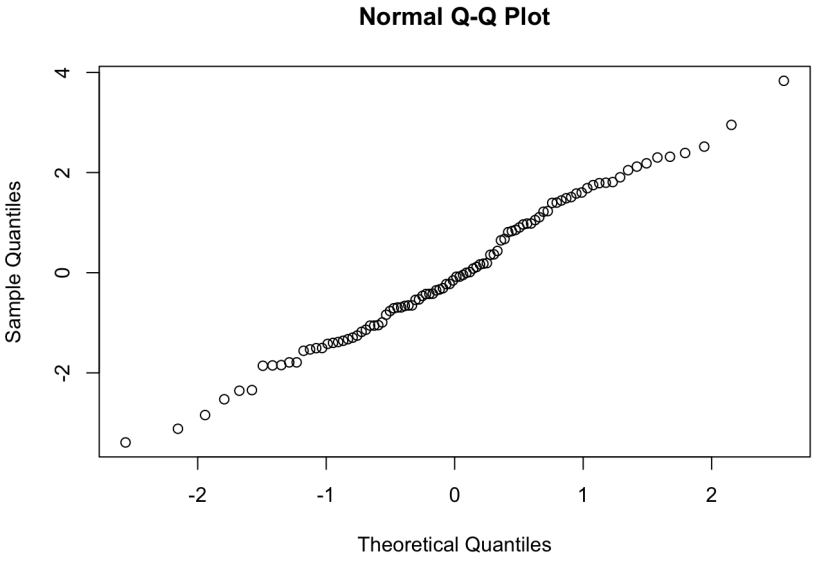
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sqrt(jost1_all) ~ Beach + Age + (1 | PairID2)
##    Data: alpha_model_unrel.tab
##
## REML criterion at convergence: 392.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.03922 -0.64774 -0.07021  0.68341  2.30587
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  PairID2  (Intercept) 0.9049   0.9513
##  Residual             2.7621   1.6619
## Number of obs: 96, groups:  PairID2, 53
```

```
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   8.2820     0.3532  23.446
## BeachSSB     -1.6417     0.4311  -3.808
## AgeP         -0.1669     0.3437  -0.486
##
## Correlation of Fixed Effects:
##          (Intr) BchSSB
## BeachSSB -0.625
## AgeP     -0.486  0.000
```

```
## Examine the residual plot and qqplot for violation of mo
del assumptions
plot(model1)
```




```
qqnorm(resid(model1))
```




```
## Calculate the coefficient of determination (outputs the
```

```
marginal and conditional R squared)
r.squaredGLMM(model1)
```

```
##       R2m       R2c
## 0.1579665 0.3657625
```

```
## Perform likelihood ratio tests to obtain p-values
## The REML=FALSE specification is necessary for comparing
models using the likelihood ratio test
modelFull <- lmer(sqrt(jost1_all) ~ Beach + Age + (1|PairID
2), data=alpha_model_unrel.tab, REML=FALSE)
modelAge <- lmer(sqrt(jost1_all) ~ Age + (1|PairID2), data=
alpha_model_unrel.tab, REML=FALSE)
modelBeach <- lmer(sqrt(jost1_all) ~ Beach + (1|PairID2), d
ata=alpha_model_unrel.tab, REML=FALSE)

anova(modelAge,modelFull)
```

```
## Data: alpha_model_unrel.tab
## Models:
## modelAge: sqrt(jost1_all) ~ Age + (1 | PairID2)
## modelFull: sqrt(jost1_all) ~ Beach + Age + (1 | PairID2)
##           Df    AIC    BIC  logLik deviance  Chisq Chi D
f Pr(>Chisq)
## modelAge   4 412.61 422.86 -202.30   404.61

## modelFull  5 401.42 414.24 -195.71   391.42 13.185
1  0.0002822 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
```

```
anova(modelBeach,modelFull)
```

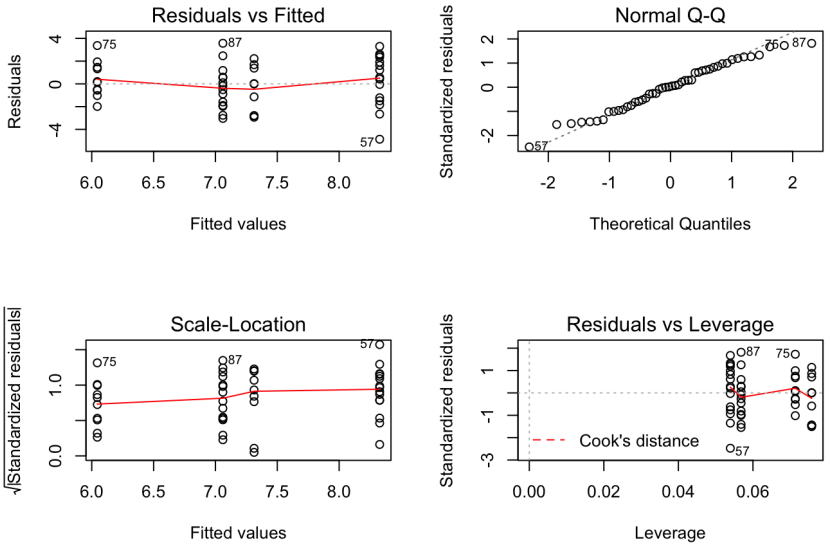
```
## Data: alpha_model_unrel.tab
## Models:
## modelBeach: sqrt(jost1_all) ~ Beach + (1 | PairID2)
## modelFull: sqrt(jost1_all) ~ Beach + Age + (1 | PairID2)
##            Df    AIC    BIC  logLik deviance  Chisq Chi
Df Pr(>Chisq)
## modelBeach  4 399.66 409.92 -195.83   391.66

## modelFull   5 401.42 414.24 -195.71   391.42 0.2393
 1     0.6247
```

```
## Test if alpha diversity is different between female and
male pups
## jost1 = response variable, beach and sex = predictors
model1 <- lm(sqrt(jost1_all) ~ Beach + Sex, data=subset(alp
ha_model.tab, Age=="P"))
summary(model1)
```

```
##
## Call:
## lm(formula = sqrt(jost1_all) ~ Beach + Sex, data = subse
```

```
t(alpha_model.tab,
##     Age == "P"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8638 -1.5132  0.0624  1.4983  3.5700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3279     0.4697   17.73   <2e-16 ***
## BeachSSB     -1.2678     0.5843   -2.17   0.0353 *
## SexM         -1.0155     0.5974   -1.70   0.0961 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
##
## Residual standard error: 2.022 on 45 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.112
## F-statistic: 3.963 on 2 and 45 DF,  p-value: 0.02599
```

```
par(mfrow=c(2,2))
plot(model1)
```



We find that estimates of alpha diversity are significantly higher at freshwater beach compared to special study beach but no significant difference is found between the age groups.

```
library(gridExtra)
library(ggplot2)

## Plot alpha diversity distribution for each beach seperat
ed by age (mothers & pups)
beach <- ggplot(alpha_model.tab, aes(x = Beach, y=jost1_all
, fill=Beach)) +
              geom_boxplot()+
              scale_x_discrete(name="", labels=c(Freshwater
="FWB", SSB="SSB"))+
              scale_fill_manual(name="Beach", values=c("dod
```

```
gerblue3","firebrick2"), labels=c("FWB", "SSB"))+
              theme_bw()+
              theme(legend.position=c(0.80,0.87), legend.ti
tle = element_blank())+
              ylab("Effective no. of species (Shannon entro
py)")+
              theme(axis.text.x=element_text(size=10, margi
n = unit(c(0.3, 0, 0, 0), "cm")))+
              theme(panel.grid.major = element_blank(), pan
el.grid.minor = element_blank())


age <- ggplot(alpha_model.tab, aes(x = Beach, y=jost1_all,
fill=Age)) +
            geom_boxplot()+
            scale_x_discrete(name="", labels=c(Freshwater="
FWB", SSB="SSB"))+
            theme_bw()+
            scale_fill_manual(name="Age", values=c("bisque4
","bisque1"), labels=c( M = "Mothers", P = "Pups"))+
            theme(legend.position=c(0.80,0.87), legend.titl
e = element_blank())+
            theme(axis.text.x=element_text(size=10, margin
= unit(c(0.3, 0, 0, 0), "cm")))+
            theme(axis.title.y = element_blank())+
            theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank())

## Use the pup subset of the data to look for differences i
n sex
model_pups.tab <- subset(alpha_model.tab, Age=="P")

## Plot alpha diversity distribution for each beach seperat
ed by pup sex
sex <- ggplot(alpha_model.tab, aes(x=Beach, y=jost1_all,fil
l=Sex)) +
            geom_boxplot(position=position_dodge())+
            theme_bw()+
            scale_x_discrete(name="", labels=c(Freshwater="
FWB", SSB="SSB"))+
            scale_fill_manual(name="Sex of Pup", values=c("
darkorchid3","darkseagreen2"), labels=c( M = "Male", F = "F
emale"))+
            theme(legend.position=c(0.80,0.87), legend.titl
e = element_blank())+
            theme(axis.text.x=element_text(size=10, margin
= unit(c(0.3, 0, 0, 0), "cm")))+
            theme(axis.title.y = element_blank())+
            theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank())

grid.arrange(beach, age, sex, ncol=3)
```

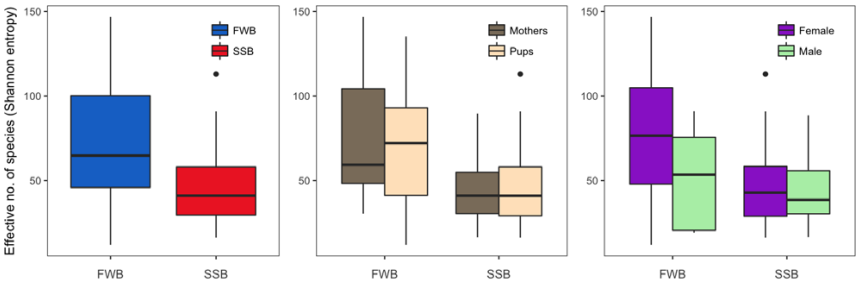


Figure 6. Boxplots of alpha diversity estimates for the two breeding colonies - freshwater beach (FWB), special study beach (SSB) - (left panel), the two age groups (center panel), and female and male pups (right panel).

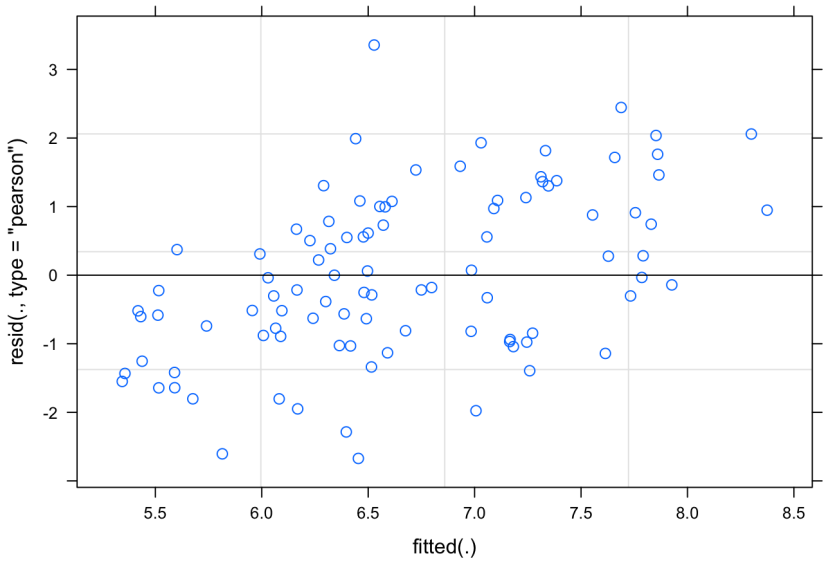
A freshwater stream is traversing through freshwater beach which could contribute to the bacterial diversity at this breeding colony through the input of environmental bacteria that are not present at special study beach. We thus want to investigate if alpha diversity is also elevated at freshwater beach when considering only the four most dominant phyla (Proteobacteria, Bacteroidetes, Firmicutes, and Actinobacteria).

```
library(dplyr)
library(lme4)
library(MuMIn)

## first make list of OTUs that belong to the 4 main phyla

mainphyla_otus.obj <-  subset_taxa(phylo.obj, Phylum %in% c
("Proteobacteria", "Bacteroidetes", "Firmicutes", "Actinoba
cteria"))
mainphyla_otus.tab <- otu_table(mainphyla_otus.obj)
mainphyla_otus_list <- row.names(mainphyla_otus.tab)

## Alpha diversity for the selected OTUs is calculated in U
search.
## Load resulting alpha diversity table
alpha_div_mainphyla.tab <- read.table("./AFSmicrobiome_SI_a
lphaDiversity_MainPhyla_Rinput_DatasetS13.txt", header=T, s
ep= "\t", row.names=1, na.strings=c("","NA"))

## Add the rownames as a column called SampleID
alpha_div_mainphyla_2.tab <- cbind(alpha_div_mainphyla.tab,
 SampleID = row.names(alpha_div_mainphyla.tab))
## Merge the tables with the meta data table
alpha_div_mainphyla_2.tab <- dplyr::left_join(meta2.tab, al
pha_div_mainphyla_2.tab, by = "SampleID")

alpha_div_mainphyla_2_unrel.tab <- cbind(alpha_div_mainphyl
a_2.tab, PairID2 = alpha_div_mainphyla_2.tab$PairID)
## Change the PairID of the pups by appending a p to the en
d of the ID
for (i in c("P49","P46","P15","P13","P11")) {
  levels(alpha_div_mainphyla_2_unrel.tab$PairID2) <- c(leve
ls(alpha_div_mainphyla_2_unrel.tab$PairID2),   paste0(alpha
_div_mainphyla_2_unrel.tab$PairID2[alpha_div_mainphyla_2_un
rel.tab$SampleID==(i)],"p"))
  alpha_div_mainphyla_2_unrel.tab$PairID2[alpha_div_mainphy
la_2_unrel.tab$SampleID==(i)] <-  paste0(paste0(alpha_div_m
```

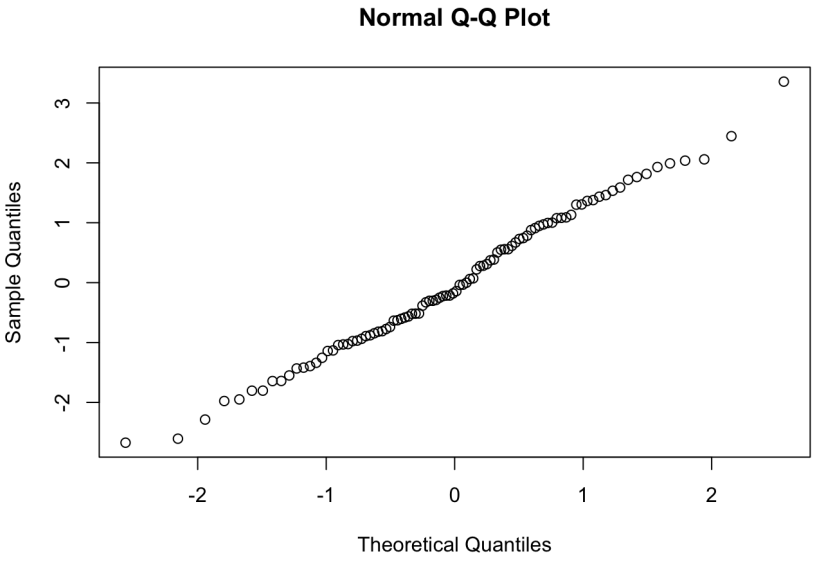
```
ainphyla_2_unrel.tab$PairID2[alpha_div_mainphyla_2_unrel.ta
b$SampleID==(i)],"p"))
}

## Test if alpha diversity is different between the two bea
ches and between mothers and pups
## jost = response variable, beach and age = predictors
## PairID used as a random effect to account for non-indepe
ndence of a pair
mod <- lmer(sqrt(jost) ~ Beach + Age + (1|PairID2), data=al
pha_div_mainphyla_2_unrel.tab)
## Get the model output
summary(mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sqrt(jost) ~ Beach + Age + (1 | PairID2)
##    Data: alpha_div_mainphyla_2_unrel.tab
##
## REML criterion at convergence: 365
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9091 -0.6306 -0.1150  0.6814  2.3977
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  PairID2  (Intercept) 0.7895   0.8885
##  Residual             1.9595   1.3998
## Number of obs: 96, groups:  PairID2, 53
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  7.08871    0.30783  23.028
## BeachSSB    -0.91041    0.37851  -2.405
## AgeP         0.07419    0.29011   0.256
##
## Correlation of Fixed Effects:
##          (Intr) BchSSB
## BeachSSB -0.633
## AgeP     -0.471  0.000
```

```
## Examine the residual plot and qqplot for violation of mo
del assumptions
plot(mod)
```




```
qqnorm(resid(mod))
```




```
## Calculate the coefficient of determination (outputs the
marginal and conditional R squared)
r.squaredGLMM(mod)
```

```
##        R2m        R2c
## 0.07121833 0.33795176
```

```
## Perform likelihood ratio tests to obtain p-values
## The REML=FALSE specification is necessary for comparing
models using the likelihood ratio test
modFull <- lmer(sqrt(jost) ~ Beach + Age + (1|PairID2), dat
a=alpha_div_mainphyla_2_unrel.tab, REML=FALSE)
modAge <- lmer(sqrt(jost) ~ Age + (1|PairID2), data=alpha_d
iv_mainphyla_2_unrel.tab, REML=FALSE)
modBeach <- lmer(sqrt(jost) ~ Beach + (1|PairID2), data=alp
ha_div_mainphyla_2_unrel.tab, REML=FALSE)
```

```
anova(modAge,modFull)
```

```
## Data: alpha_div_mainphyla_2_unrel.tab
## Models:
## modAge: sqrt(jost) ~ Age + (1 | PairID2)
## modFull: sqrt(jost) ~ Beach + Age + (1 | PairID2)
##         Df    AIC    BIC  logLik deviance  Chisq Chi Df
Pr(>Chisq)
## modAge   4 376.43 386.69 -184.22   368.43

## modFull  5 372.76 385.58 -181.38   362.76 5.6709      1
   0.01725 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
```

```
anova(modBeach,modFull)
```

```
## Data: alpha_div_mainphyla_2_unrel.tab
## Models:
## modBeach: sqrt(jost) ~ Beach + (1 | PairID2)
## modFull: sqrt(jost) ~ Beach + Age + (1 | PairID2)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df
 Pr(>Chisq)
## modBeach  4 370.83 381.08 -181.41   362.83

## modFull   5 372.76 385.58 -181.38   362.76 0.0672      1
     0.7955
```

```
library(ggplot2)

## Plot alpha diversity only for the two breeding colonies

ggplot(alpha_div_mainphyla_2_unrel.tab, aes(x = Beach, y=jo
st, fill=Beach)) +
  geom_boxplot()+
  scale_x_discrete(name="", labels=c(Freshwater="FWB", SSB=
"SSB"))+
  scale_fill_manual(name="Beach", values=c("dodgerblue3","f
irebrick2"), labels=c("FWB", "SSB"))+
  theme_bw()+
  theme(legend.position="none")+
  ylab("Effective no. of species")+
  theme(axis.text.x=element_text(size=14, margin = unit(c(0
.2, 0, 0, 0), "cm")),axis.text.y=element_text(size=14, marg
in = unit(c(0, 0.1, 0, 0.3), "cm")), axis.title=element_tex
t(size=14))+
  theme(panel.grid.major = element_blank(), panel.grid.mino
r = element_blank())
```

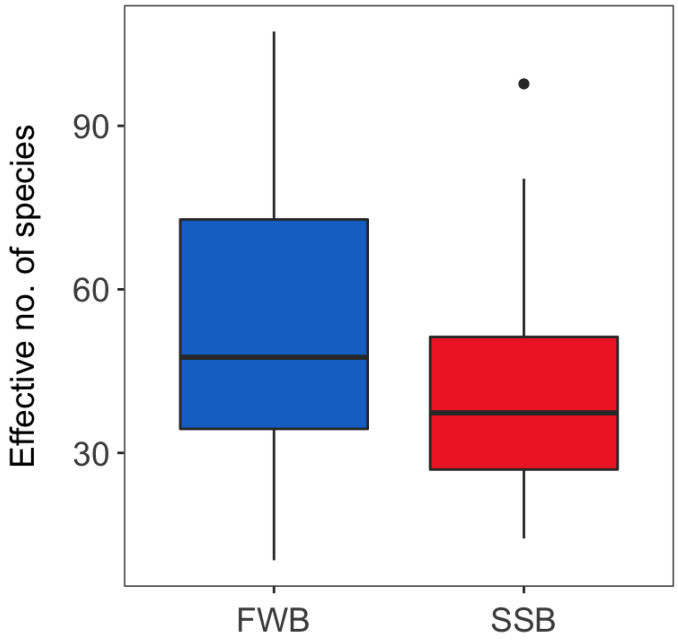


Figure 7. Boxplot of alpha diversity estimates for the two breeding colonies - freshwater beach (FWB), special study beach (SSB) based on the four dominant phyla Proteobacteria, Bacteroidetes, Firmicutes, and Actinobacteria.

Alpha diversity remains significantly higher at freshwater beach when considering only OTUs from the four most dominant phyla.

Beta diversity

To assess the difference in bacterial composition among samples (beta diversity) we calculated Bray-Curtis and weighted UniFrac dissimilarity/distance matrices for normalised and non-normalised OTU tables. To normalise the counts we performed cumulative sum scaling (CSS) with the metagenome-Seq package. UniFrac distance calculations require a phylogenetic tree. The tree was calculated with an outgroup using Pynast in QIIME v.1.9.1. The tree was rooted with an archaeal sequence as outgroup and the outgroup removed before calculation of the weighted UniFrac distance.

```
library(ape)

## Calculate beta diversity and differential OTU abundance
for the two beaches and mother and pup groups
## 1. non-normalised OTU table -> Bray-Curtis and weighted
UniFrac

## For wUniFrac distance calculation a phylogeneic tree is
needed. The tree was calculated with an outgroup using pyna
st in QIIME
## Import phylogeny (pynast with outgroup)
tree_py_out.file <- read.tree(file="./AFSmicrobiome_SI_outg
roup_pynastAligned_filtered_Rinput_DatasetS14.tre")
## Root tree and trim outgroup from tree (label: U11044_V3V
4)
# tree_py_out.file$tip.label # For checking tip labels
## Root tree at outgroup
pytree_rooted.file <- root(tree_py_out.file, outgroup="U110
44 V3V4",resolve.root = TRUE)
```

```
## Remove outgroup
pynast.tre <- drop.tip(pytree_rooted.file, "U11044_V3V4")

## Add a column to the meta data table that combines the be
ach and age variables for easier plotting of groups
meta.tab$BeachAge <- paste(meta.tab$Beach,meta.tab$Age)
meta.tab$SampleNames <- rownames(meta.tab)

## Combine all files into a phyloseq object
otu.obj <- otu_table(otu.tab, taxa_are_rows = TRUE)
tax.obj <- tax_table(tax.tab)
meta.obj <- sample_data(meta.tab)
## Make a phyloseq object and add tree file
phylo.obj <- phyloseq(otu.obj, tax.obj, meta.obj)
phylo.obj <- merge_phyloseq(phylo.obj,pynast.tre)
```

First, we calculate the Bray-Curtis and weighted UniFrac distances for the non-normalised OTU table and visualise them with principal coordinate analysis (PCoA).

```
## Calulate PCoA for non-normalised OTU table with Bray-Cur
tis.
afs_pcoa_bray <- ordinate(physeq = phylo.obj, method = "PCo
A", distance = "bray")

## Plot PCoA
b <- plot_ordination(physeq = phylo.obj, ordination = afs_p
coa_bray, color = "BeachAge", shape = "BeachAge") +
          geom_point(size = 3.5) +
          scale_color_manual(values = c("dodgerblue3","dodg
erblue3","firebrick2","firebrick2"),name="",breaks=c("Fresh
water M", "Freshwater P", "SSB M", "SSB P"), labels=c("FWB
mothers","FWB pups", "SSB mothers", "SSB pups"))+
          scale_shape_manual(values = c(19,1,15,0),name="",
breaks=c("Freshwater M", "Freshwater P", "SSB M", "SSB P"),
 labels=c("FWB mothers","FWB pups", "SSB mothers", "SSB pup
s"))+
          theme_bw()+
          ggtitle("non-normalised Bray-Curtis")+
          theme(legend.position="none")+
          theme(legend.background = element_rect(size=0.3,l
inetype="solid", colour ="black"))+
          theme(panel.grid.major = element_blank(),panel.gr
id.minor = element_blank())
## The plot above plots the points twice. To remove the sec
ond layer do:
b$layers <- b$layers[-1]

## Calulate PCoA for non-normalised OTU table with weighted
 Unifrac.
afs_pcoa_uni <- ordinate(physeq = phylo.obj, method = "PCoA
", distance = "wunifrac")

## Plot PCoA
u <- plot_ordination(physeq = phylo.obj, ordination = afs_p
coa_uni, color = "BeachAge", shape = "BeachAge") +
          geom_point(size = 3.5) +
          scale_color_manual(values = c("dodgerblue3","dodg
erblue3","firebrick2","firebrick2"),name="",breaks=c("Fresh
```

```
water M", "Freshwater P", "SSB M", "SSB P"), labels=c("FWB
mothers","FWB pups", "SSB mothers", "SSB pups"))+
          scale_shape_manual(values = c(19,1,15,0),name="",
breaks=c("Freshwater M", "Freshwater P", "SSB M", "SSB P"),
 labels=c("FWB mothers","FWB pups", "SSB mothers", "SSB pup
s"))+
          theme_bw()+
          ggtitle("non-normalised weighted UniFrac")+
          theme(legend.position=c(0.83,0.19), legend.title
= element_blank())+
          theme(legend.background = element_rect(size=0.3,l
inetype="solid", colour ="black"))+
          theme(panel.grid.major = element_blank(),panel.gr
id.minor = element_blank())
u$layers <- u$layers[-1]

grid.arrange(b, u, ncol=2)
```

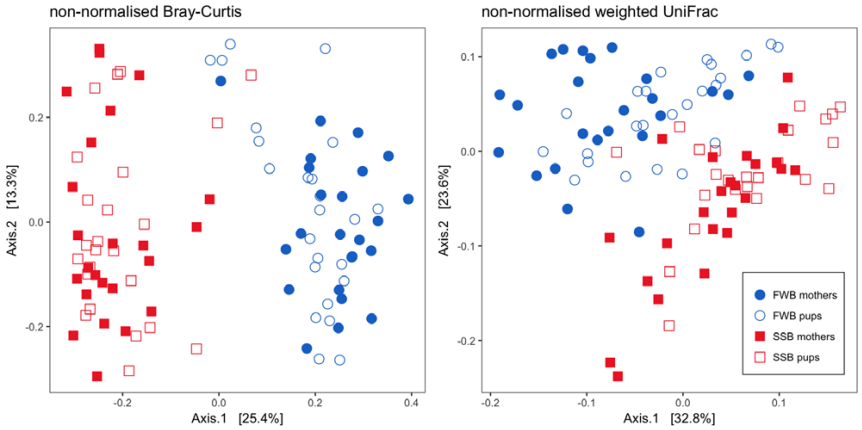


Figure 8. Principal coordinate analysis (PCoA) based on Bray-Curtis dissimliarities (left panel) and weighted UniFrac distances (right panel) calculated from the non-normalised OTU table.

Prepare an OTU table normalised with cumulative sum scaling (CSS) using the metagenomeSeq package and following the MetagenomeSeq vignette.

```
library("metagenomeSeq")

## Convert the phyloseq object to a metagenomeSeq object (M
Rexperiment).
## The Phyloseq_to_metagenomeSeq function is included in th
e phyloseq package.
metagenome.obj <- phyloseq_to_metagenomeSeq(phylo.obj)
## Calculate the proper percentile by which to normalize co
unts
cNstat <- metagenomeSeq::cumNormStatFast(metagenome.obj)
# cNstat  #0.5
## Normalise counts
metagenome.obj <- metagenomeSeq::cumNorm(metagenome.obj, p
= cNstat)
## Export the normalised count table
metag.norm.counts <- metagenomeSeq::MRcounts(metagenome.obj
, norm = TRUE)
## Add a pseudocount of 0.0001 to the table and log transfo
rm
metag.norm.counts_log <- log(metag.norm.counts+0.0001)
```

```
## Substract the value from log of pseudocount to preserve
zeros of the original counts
metag.norm.counts_log2 <- metag.norm.counts_log-(log(0.0001
))

## Make a new phyloseq object with with the new OTU table
otu_normMG.obj <- otu_table(metag.norm.counts_log2, taxa_ar
e_rows = TRUE)
phylo_normMG.obj <- phyloseq(otu_normMG.obj, tax.obj, meta.
obj)
phylo_normMG.obj <- merge_phyloseq(phylo_normMG.obj, pynast
.tre)
```

Now we can calculate the Bray-Curtis and weighted UniFrac distances for the CSS normalised OTU table.

```
library(gridExtra)
library(ggplot2)

##  Calulate PCoA for CSS normalised OTU table with Bray-Cu
rtis.
afs_pcoa_css_bray <- ordinate(physeq = phylo_normMG.obj, me
thod = "PCoA", distance = "bray")

## Plot PCoA
b <- plot_ordination(physeq = phylo_normMG.obj, ordination
= afs_pcoa_css_bray, color = "BeachAge", shape = "BeachAge"
) +
          geom_point(size = 3.5) +
          scale_color_manual(values = c("dodgerblue3","dodg
erblue3","firebrick2","firebrick2"),name="",breaks=c("Fresh
water M", "Freshwater P", "SSB M", "SSB P"), labels=c("FWB
mothers","FWB pups", "SSB mothers", "SSB pups"))+
          scale_shape_manual(values = c(19,1,15,0),name="",
breaks=c("Freshwater M", "Freshwater P", "SSB M", "SSB P"),
 labels=c("FWB mothers","FWB pups", "SSB mothers", "SSB pup
s"))+
          theme_bw()+
          ggtitle("CSS normalised Bray-Curtis")+
          theme(legend.position=c(0.17,0.80), legend.title
= element_blank())+
          theme(legend.background = element_rect(size=0.3,l
inetype="solid", colour ="black"))+
          theme(panel.grid.major = element_blank(), panel.g
rid.minor = element_blank())
b$layers <- b$layers[-1]

##  Calulate PCoA for CSS normalised OTU table with weighte
d Unifrac.
afs_pcoa_css_uni <- ordinate(physeq = phylo_normMG.obj, met
hod = "PCoA", distance = "wunifrac")

## Plot PCoA
u <- plot_ordination(physeq = phylo_normMG.obj, ordination
= afs_pcoa_css_uni, color = "BeachAge", shape = "BeachAge")
 +
          geom_point(size = 3.5) +
          scale_color_manual(values = c("dodgerblue3","dodg
erblue3","firebrick2","firebrick2"),name="",breaks=c("Fresh
water M", "Freshwater P", "SSB M", "SSB P"), labels=c("FWB
```

```
mothers","FWB pups", "SSB mothers", "SSB pups"))+
          scale_shape_manual(values = c(19,1,15,0), name=""
, breaks=c("Freshwater M", "Freshwater P", "SSB M", "SSB P"
), labels=c("FWB mothers","FWB pups", "SSB mothers", "SSB p
ups"))+
          theme_bw()+
          ggtitle("CSS normalised weighted UniFrac")+
          theme(legend.position="none")+
          theme(legend.background = element_rect(size=0.3,l
inetype="solid", colour ="black"))+
          theme(panel.grid.major = element_blank(), panel.g
rid.minor = element_blank())
u$layers <- u$layers[-1]

grid.arrange(b, u, ncol=2)
```

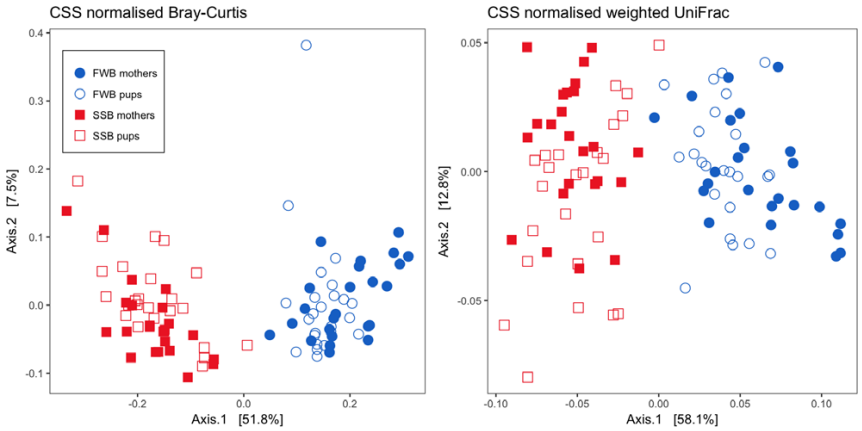


Figure 9. Principal coordinate analysis (PCoA) based on Bray-Curtis dissimliarity (left panel) and weighted UniFrac distance (right panel) calculated from the CSS normalised OTU table.

To visually examine the similarity between mothers and their pups we can use different colour and shapes for each pair.

```
library(RColorBrewer)
library(gridExtra)
library(ggplot2)

## Make plot with different colours and shapes for the pair
s
## Plot PCoA
u <- plot_ordination(physeq = phylo_normMG.obj, ordination
= afs_pcoa_css_uni, color = "BeachAge", shape = "BeachAge")
 +
        geom_point(size = 3.5) +
        scale_color_manual(values = c("dodgerblue3","dodger
blue3","firebrick2","firebrick2"),name="",breaks=c("Freshwa
ter M", "Freshwater P", "SSB M", "SSB P"), labels=c("FWB mo
thers","FWB pups", "SSB mothers", "SSB pups"))+
        scale_shape_manual(values = c(19,1,15,0), name="",
breaks=c("Freshwater M", "Freshwater P", "SSB M", "SSB P"),
 labels=c("FWB mothers","FWB pups", "SSB mothers", "SSB pup
s"))+
        theme_bw()+
        ggtitle("CSS normalised weighted UniFrac")+
        theme(legend.position="none")+
```

```
        theme(legend.position=c(0.84,0.18), legend.title =
element_blank())+
        theme(legend.background = element_rect(size=0.3,lin
etype="solid", colour ="black"))+
        theme(panel.grid.major = element_blank(), panel.gri
d.minor = element_blank())
u$layers <- u$layers[-1]

## Define colours and shapes for the pairs
# brewer.pal(8,"Set1")
colorPairs=rep(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3"
, "darkgray" ,"gold" ,"#A65628", "#F781BF"),6)
shapePairs=c(0,0,0,0,1,1,1,1,2,2,2,2,5,5,5,5,3,3,3,3,4,4,4,
4,6,6,6,6,8,8,8,8,15,15,15,15,16,16,16,16,17,17,17,17,18,18
,18,18)

x <- plot_ordination(physeq = phylo_normMG.obj, ordination
= afs_pcoa_css_uni, color = "PairID", shape = "PairID") +
          geom_point(size=3.5, stroke=1) +
          scale_color_manual(values = colorPairs,name="",br
eaks=sample_data(phylo_normMG.obj)$PairID, labels=sample_da
ta(phylo_normMG.obj)$PairID)+
          scale_shape_manual(values = shapePairs,name="",br
eaks=sample_data(phylo_normMG.obj)$PairID, labels=sample_da
ta(phylo_normMG.obj)$PairID)+
          theme_bw()+
          theme(legend.position="none")+
          ggtitle("CSS normalised weighted UniFrac pairs")+
          theme(panel.grid.major = element_blank(), panel.g
rid.minor = element_blank())
x$layers <- x$layers[-1]

grid.arrange(u,x,ncol=2)
```

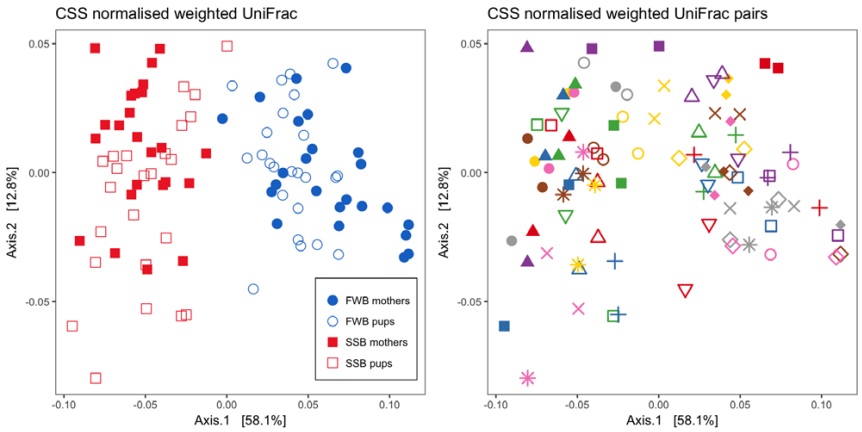


Figure 10. Principal coordinate analysis (PCoA) plots based on weighted UniFrac distance calculated from the CSS normalised OTU table. In the right panel the two individuals of a mother-pup pair are labelled with the same symbol and colour.

We can also visualise the differences in Bray-Curtis and weighted UniFrac distances within and among breeding colonies, age groups and mother-pup pairs using boxplots.

```
library(phyloseq)
library(reshape2)
library(gridExtra)
```

```
library(ggplot2)

## Calculate distance matrices
unifracDist <- phyloseq::distance(phylo_normMG.obj, method
= "wunifrac")
brayDist <- phyloseq::distance(phylo_normMG.obj, method = "
bray")
## Convert dist element into a matrix
mx_unifrac <- as.matrix(unifracDist)
mx_bray <- as.matrix(brayDist)
## Convert from pairwise matrix to pairwise table
df_unifrac <- subset(melt(mx_unifrac), value!=0)
df_bray <- subset(melt(mx_bray), value!=0)

## Use data frame that contains information about the unrel
ated pairs
alpha_model_unrel_2.tab <- alpha_model_unrel.tab

## Collect meta data. Two data frames are needed to match t
he two samples in each row
df_meta <- subset(alpha_model_unrel_2.tab, select=c("Beach"
, "Age", "PairID2", "SampleID"))
df_meta2 <- subset(alpha_model_unrel_2.tab, select=c("Beach
","Age", "PairID2", "SampleID"))
## Rename columns
colnames(df_meta) <- c("Beach", "Age", "PairID2_1","Var1")

colnames(df_meta2) <- c("Beach2","Age2", "PairID2_2", "Var2
")
## Join the data frames
df_unifrac_meta <- dplyr::left_join(df_unifrac, df_meta, by
="Var1")
df_unifrac_meta <- dplyr::left_join(df_unifrac_meta, df_met
a2, by="Var2")
df_bray_meta <- dplyr::left_join(df_bray, df_meta, by="Var1
")
df_bray_meta <- dplyr::left_join(df_bray_meta, df_meta2, by
="Var2")
## Add columns indicating if the individuals from the same
breeding colony, age group or the same pair
df_unifrac_meta  <- cbind(df_unifrac_meta,BeachMatch = as.f
actor(ifelse(df_unifrac_meta$Beach==df_unifrac_meta$Beach2,
0,1)))
df_unifrac_meta  <- cbind(df_unifrac_meta,AgeMatch = as.fac
tor(ifelse(df_unifrac_meta$Age==df_unifrac_meta$Age2,0,1)))
df_unifrac_meta  <- cbind(df_unifrac_meta,PairMatch = as.fa
ctor(ifelse(df_unifrac_meta$PairID2_1==df_unifrac_meta$Pair
ID2_2,0,1)))
df_bray_meta  <- cbind(df_bray_meta,BeachMatch = as.factor(
ifelse(df_bray_meta$Beach==df_bray_meta$Beach2,0,1)))
df_bray_meta  <- cbind(df_bray_meta,AgeMatch = as.factor(if
else(df_bray_meta$Age==df_bray_meta$Age2,0,1)))
df_bray_meta  <- cbind(df_bray_meta,PairMatch = as.factor(i
felse(df_bray_meta$PairID2_1==df_bray_meta$PairID2_2,0,1)))

## Draw plot
BeachUni <- ggplot(data = df_unifrac_meta, aes(x=BeachMatch
, y=value)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
```

```
                    scale_x_discrete(name="", labels=c("0"
="within", "1" ="among"))+
                    ylab("weighted UniFrac")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())

AgeUni <- ggplot(data = df_unifrac_meta, aes(x=AgeMatch, y=
value)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
                    scale_x_discrete(name="", labels=c("0"
="within", "1" ="among"))+
                    ylab("")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())
PairUni <- ggplot(data = df_unifrac_meta, aes(x=PairMatch,
y=value)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
                    scale_x_discrete(name="", labels=c("0"
="pairs", "1" ="unrelated"))+
                    ylab("")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())
BeachBray <- ggplot(data = df_bray_meta, aes(x=BeachMatch,
y=value)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
                    scale_x_discrete(name="breeding colonie
s", labels=c("0" ="within", "1" ="among"))+
                    ylab("Bray-Curtis")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())

AgeBray <- ggplot(data = df_bray_meta, aes(x=AgeMatch, y=va
lue)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
                    scale_x_discrete(name="age groups", lab
els=c("0" ="within", "1" ="among"))+
                    ylab("")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
```

```
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())
PairBray <- ggplot(data = df_bray_meta, aes(x=PairMatch, y=
value)) +
                    geom_boxplot(fill="lightgray")+
                    theme_bw()+
                    scale_x_discrete(name="", labels=c("0"
="pairs", "1" ="unrelated"))+
                    ylab("")+
                    theme(axis.text.x=element_text(size=12,
 margin = unit(c(0.3, 0, 0, 0), "cm")),axis.text.y=element_
text(size=12), axis.title.x=element_text(size=14),axis.titl
e.y=element_text(size=14))+
                    theme(panel.grid.major = element_blank(
), panel.grid.minor = element_blank())

grid.arrange(BeachUni,AgeUni,PairUni,BeachBray,AgeBray,Pair
Bray, nrow=2, ncol=3)
```

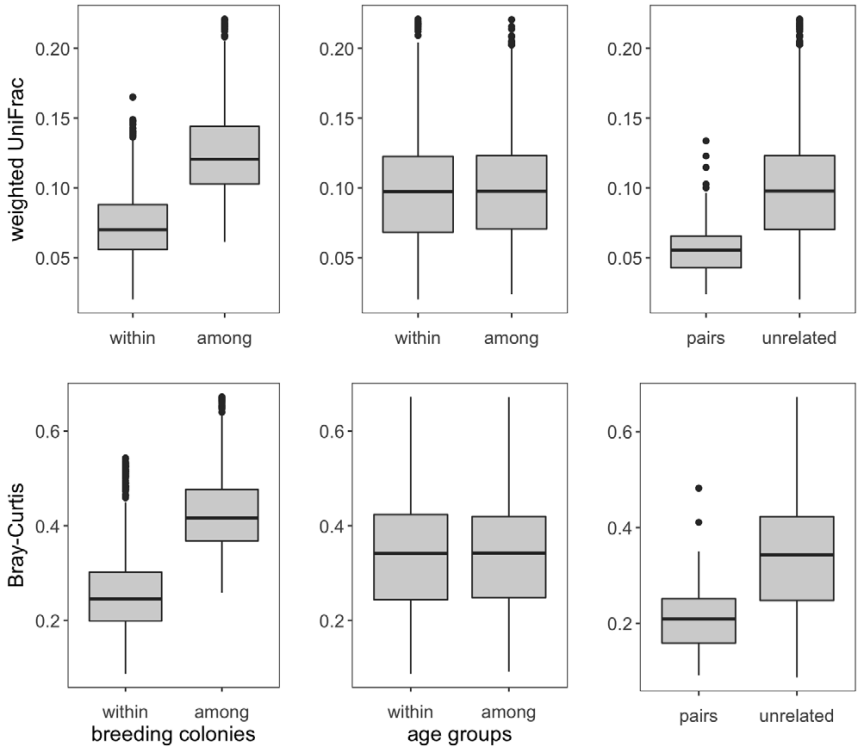


Figure 11. Weighted UniFrac and Bray-Curtis distances within and among breeding colonies, age groups and mother-pup pairs.

ANOSIM – Analysis of similarities

Analysis of similarities can be used to test for similarity/dissimilarity of bacterial communities between the breeding colonies, age groups, and mother-pup pair groups. As input we use the CSS normalised OTU table from above.
Testing for differences in microbial composition between the two breeding sites, overall and separately for mothers and pups.

```
library(phyloseq)
library(vegan)
```

```
## Vegan's anosim takes a matrix as input with columns repr
esenting OTUs and rows representing samples.
## The OTU table can be transposed and exported from the ph
yloseq object as follows:
OTU1 <- as(otu_table(phylo_normMG.obj), "matrix")
## transpose if necessary
if(taxa_are_rows(phylo_normMG.obj)){OTU1 <- t(OTU1)}
## Coerce to data.frame
otu_table.tab <- as.data.frame(OTU1)
## Extract the meta data from the phyloseq object
meta_data.tab <- as(sample_data(phylo_normMG.obj), "data.fr
ame")

## Perform ANOSIM to test for dissimilarity between beaches
x <- vegan::anosim(dat = otu_table.tab, grouping = meta_dat
a.tab$Beach, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab, grouping = meta_data.
tab$Beach,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.8765
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0175 0.0290 0.0414 0.0574
##
## Dissimilarity ranks between and within classes:
##              0%     25%    50%     75% 100%    N
## Between    1277 2661.50 3337.5 3937.25 4560 2304
## Freshwater    1  282.75  575.5 1056.25 4339 1128
## SSB         296 1165.00 1641.0 2126.25 4190 1128
```

```
# ANOSIM statistic R: 0.8765
# Significance: 9.999e-05

## Perform ANOSIM to test for dissimilarity between mother
groups of the two beaches
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Age ==
 "M", ], grouping = meta_data.tab[meta_data.tab$Age == "M",
 ]$Beach, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Age == "
M", ],      grouping = meta_data.tab[meta_data.tab$Age == "
M", ]$Beach,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
```

```
## ANOSIM statistic R: 0.946
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0373 0.0622 0.0875 0.1198
##
## Dissimilarity ranks between and within classes:
##             0%    25%   50%    75% 100%   N
## Between    365 680.75 836.5 984.25 1128 576
## Freshwater   1  69.75 138.5 252.00  735 276
## SSB        140 284.75 393.5 494.25  968 276
```

```
# ANOSIM statistic R: 0.946
# Significance: 9.999e-05

## Perform ANOSIM to test for dissimilarity between pup gro
ups of the two beaches
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Age ==
 "P", ], grouping = meta_data.tab[meta_data.tab$Age == "P",
 ]$Beach, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Age == "
P", ],      grouping = meta_data.tab[meta_data.tab$Age == "
P", ]$Beach,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.8076
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0324 0.0561 0.0759 0.1088
##
## Dissimilarity ranks between and within classes:
##             0%    25%   50%    75% 100%   N
## Between    318 639.75 806.5 959.25 1128 576
## Freshwater   1  69.75 141.5 253.25 1080 276
## SSB         93 287.50 413.5 550.25 1051 276
```

```
# ANOSIM statistic R: 0.8076
# Significance: 9.999e-05
```

Testing for differences in microbial composition between the two age groups, overall and separately for each beach.

```
library(vegan)

## Perform ANOSIM to test for dissimilarity between age gro
```

```
ups (mothers and pups)
x <- vegan::anosim(dat = otu_table.tab, grouping = meta_dat
a.tab$Age, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab, grouping = meta_data.
tab$Age,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.006069
##       Significance: 0.21588
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0178 0.0293 0.0411 0.0577
##
## Dissimilarity ranks between and within classes:
##         0%     25%  50%    75% 100%    N
## Between  2 1166.25 2286 3377.0 4559 2304
## M        1 1124.75 2230 3590.0 4560 1128
## P        6 1109.25 2309 3331.5 4540 1128
```

```
# ANOSIM statistic R: 0.006069
# Significance: 0.20998

## Perform ANOSIM to test for dissimilarity between age gro
ups at Freshwater beach only
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach
== "Freshwater", ], grouping = meta_data.tab[meta_data.tab$
Beach == "Freshwater",]$Age, distance = "bray", permutation
s = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach ==
 "Freshwater",      ], grouping = meta_data.tab[meta_data.t
ab$Beach == "Freshwater",      ]$Age, permutations = 10000,
 distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.08401
##       Significance: 0.0039996
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0259 0.0374 0.0498 0.0647
##
## Dissimilarity ranks between and within classes:
##         0%    25%   50%    75% 100%   N
```

```
## Between  2 318.50 602.5 861.50 1128 576
## M        1 282.00 558.5 827.25 1081 276
## P        6 226.75 489.0 815.00 1127 276
```

```
# ANOSIM statistic R: 0.08401
# Significance: 0.0045995

## Perform ANOSIM to test for dissimilarity between age gro
ups at Special study beach only
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach
== "SSB", ], grouping = meta_data.tab[meta_data.tab$Beach =
= "SSB",]$Age, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach ==
 "SSB",      ], grouping = meta_data.tab[meta_data.tab$Beac
h == "SSB",      ]$Age, permutations = 10000, distance = "b
ray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.08145
##       Significance: 0.0015998
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0249 0.0352 0.0445 0.0581
##
## Dissimilarity ranks between and within classes:
##         0%    25%   50%    75% 100%   N
## Between  3 315.75 603.5 871.75 1127 576
## M       12 257.50 490.5 733.50 1126 276
## P        1 258.25 596.5 886.25 1128 276
```

```
# ANOSIM statistic R: 0.08145
# Significance: 0.0014999
```

Testing for differences in microbial composition between the two sexes (only for pups), overall and separately for each beach.

```
library(vegan)

## Make data frames with pup information only
otu_pups.tab <- otu_table.tab[meta_data.tab$Age == "P", ]
meta_pups.tab <- meta_data.tab[meta_data.tab$Age == "P", ]

## Perform ANOSIM to test for dissimilarity between male an
d female pups
x <- vegan::anosim(dat = otu_pups.tab, grouping = meta_pups
.tab$Sex, distance = "bray", permutations = 10000)
summary(x)
```

```
##
```

```
## Call:
## vegan::anosim(dat = otu_pups.tab, grouping = meta_pups.t
ab$Sex,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: -0.007055
##       Significance: 0.50145
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0500 0.0741 0.0998 0.1302
##
## Dissimilarity ranks between and within classes:
##         0%    25%   50%    75% 100%   N
## Between  2 284.00 579.0 842.50 1128 551
## F        1 269.25 554.5 856.75 1127 406
## M       29 305.00 585.0 848.00 1117 171
```

```
# ANOSIM statistic R: -0.007055
# Significance: 0.50545

## Perform ANOSIM to test for dissimilarity between sexes a
t Freshwater beach only
x <- vegan::anosim(dat = otu_pups.tab[meta_pups.tab$Beach =
= "Freshwater", ], grouping = meta_pups.tab[meta_pups.tab$B
each == "Freshwater",]$Sex, distance = "bray", permutations
 = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_pups.tab[meta_pups.tab$Beach ==
"Freshwater",      ], grouping = meta_pups.tab[meta_pups.ta
b$Beach == "Freshwater",      ]$Sex, permutations = 10000,
distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.1165
##       Significance: 0.10099
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.117 0.161 0.203 0.258
##
## Dissimilarity ranks between and within classes:
##         0%    25% 50%    75% 100%   N
## Between  2  86.00 148 211.50  270 135
## F        1  43.00 102 197.00  276 105
## M       29 137.75 172 214.25  248  36
```

```
# ANOSIM statistic R: 0.1165
# Significance: 0.092991
```

```
## Perform ANOSIM to test for dissimilarity between sexes a
t Special study beach only
x <- vegan::anosim(dat = otu_pups.tab[meta_pups.tab$Beach =
= "SSB", ], grouping = meta_pups.tab[meta_pups.tab$Beach ==
 "SSB",]$Sex, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_pups.tab[meta_pups.tab$Beach ==
"SSB",      ], grouping = meta_pups.tab[meta_pups.tab$Beach
 == "SSB",      ]$Sex, permutations = 10000, distance = "br
ay")
## Dissimilarity: bray
##
## ANOSIM statistic R: -0.04223
##       Significance: 0.71183
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0851 0.1166 0.1484 0.1896
##
## Dissimilarity ranks between and within classes:
##         0%   25%   50%    75% 100%   N
## Between  1 67.75 137.5 203.25  276 140
## F        3 83.00 170.0 215.00  275  91
## M        9 65.00 103.0 138.00  271  45
```

```
# ANOSIM statistic R: -0.04223
# Significance: 0.72273
```

Testing for differences in microbial composition between different mother-pup pairs, overall and separately for each beach.

```
library(vegan)

## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pair groups
x <- vegan::anosim(dat = otu_table.tab, grouping =meta_data
.tab$PairID, distance = "bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab, grouping = meta_data.
tab$PairID,      permutations = 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.6145
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0909 0.1174 0.1380 0.1660
##
## Dissimilarity ranks between and within classes:
##           0%     25%    50%     75% 100%    N
## Between    1 1163.75 2300.5 3431.25 4560 4512
## Pair1    416  416.00  416.0  416.00  416    1
## Pair10    23   23.00   23.0   23.00   23    1
## Pair11  2385 2385.00 2385.0 2385.00 2385    1
## Pair12   174  174.00  174.0  174.00  174    1
## Pair13  1738 1738.00 1738.0 1738.00 1738    1
## Pair14   338  338.00  338.0  338.00  338    1
## Pair15   823  823.00  823.0  823.00  823    1
## Pair16   286  286.00  286.0  286.00  286    1
## Pair17  1268 1268.00 1268.0 1268.00 1268    1
## Pair18   767  767.00  767.0  767.00  767    1
## Pair19    45   45.00   45.0   45.00   45    1
## Pair2    198  198.00  198.0  198.00  198    1
## Pair20   751  751.00  751.0  751.00  751    1
## Pair21   862  862.00  862.0  862.00  862    1
## Pair22  1232 1232.00 1232.0 1232.00 1232    1
## Pair23   795  795.00  795.0  795.00  795    1
## Pair24  3983 3983.00 3983.0 3983.00 3983    1
## Pair25   704  704.00  704.0  704.00  704    1
## Pair26   218  218.00  218.0  218.00  218    1
## Pair27    53   53.00   53.0   53.00   53    1
## Pair28   434  434.00  434.0  434.00  434    1
## Pair29   403  403.00  403.0  403.00  403    1
## Pair3      8    8.00    8.0    8.00    8    1
## Pair30  1123 1123.00 1123.0 1123.00 1123    1
## Pair31   953  953.00  953.0  953.00  953    1
## Pair32   219  219.00  219.0  219.00  219    1
## Pair33   660  660.00  660.0  660.00  660    1
## Pair34   121  121.00  121.0  121.00  121    1
## Pair35   276  276.00  276.0  276.00  276    1
## Pair37   837  837.00  837.0  837.00  837    1
## Pair38  1103 1103.00 1103.0 1103.00 1103    1
## Pair39  2404 2404.00 2404.0 2404.00 2404    1
## Pair4    108  108.00  108.0  108.00  108    1
## Pair40  1507 1507.00 1507.0 1507.00 1507    1
## Pair41   784  784.00  784.0  784.00  784    1
## Pair42  1518 1518.00 1518.0 1518.00 1518    1
## Pair43  3255 3255.00 3255.0 3255.00 3255    1
## Pair44  1321 1321.00 1321.0 1321.00 1321    1
## Pair45   450  450.00  450.0  450.00  450    1
## Pair46  1094 1094.00 1094.0 1094.00 1094    1
## Pair47  1831 1831.00 1831.0 1831.00 1831    1
## Pair48   677  677.00  677.0  677.00  677    1
## Pair49   975  975.00  975.0  975.00  975    1
## Pair5   2042 2042.00 2042.0 2042.00 2042    1
## Pair6   1301 1301.00 1301.0 1301.00 1301    1
## Pair7      2    2.00    2.0    2.00    2    1
## Pair8    129  129.00  129.0  129.00  129    1
## Pair9    332  332.00  332.0  332.00  332    1
```

```
# ANOSIM statistic R: 0.6145
# Significance: 9.999e-05
```

```
## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pair groups from Freshwater Beach
## First remove levels from the grouping factor, otherwise
R will give an error message (but the calculations are corr
ect either way)
pairsFW <- meta_data.tab[meta_data.tab$Beach == "Freshwater
",]$PairID
pairsFW <- droplevels(pairsFW)
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach
== "Freshwater",], grouping = pairsFW, distance = "bray", p
ermutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach ==
 "Freshwater",      ], grouping = pairsFW, permutations = 1
0000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.3494
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0904 0.1185 0.1424 0.1696
##
## Dissimilarity ranks between and within classes:
##           0%     25%    50%     75% 100%    N
## Between    1  290.75  569.5  849.25 1128 1104
## Pair10    23   23.00   23.0   23.00   23    1
## Pair12   174  174.00  174.0  174.00  174    1
## Pair16   286  286.00  286.0  286.00  286    1
## Pair19    45   45.00   45.0   45.00   45    1
## Pair2    198  198.00  198.0  198.00  198    1
## Pair20   694  694.00  694.0  694.00  694    1
## Pair21   753  753.00  753.0  753.00  753    1
## Pair22   911  911.00  911.0  911.00  911    1
## Pair23   717  717.00  717.0  717.00  717    1
## Pair24  1089 1089.00 1089.0 1089.00 1089    1
## Pair26   218  218.00  218.0  218.00  218    1
## Pair27    53   53.00   53.0   53.00   53    1
## Pair28   430  430.00  430.0  430.00  430    1
## Pair29   400  400.00  400.0  400.00  400    1
## Pair3      8    8.00    8.0    8.00    8    1
## Pair31   803  803.00  803.0  803.00  803    1
## Pair32   219  219.00  219.0  219.00  219    1
## Pair34   121  121.00  121.0  121.00  121    1
## Pair35   276  276.00  276.0  276.00  276    1
## Pair4    108  108.00  108.0  108.00  108    1
## Pair6    932  932.00  932.0  932.00  932    1
## Pair7      2    2.00    2.0    2.00    2    1
## Pair8    129  129.00  129.0  129.00  129    1
## Pair9    330  330.00  330.0  330.00  330    1
```

```
# ANOSIM statistic R: 0.3494
```

```
# Significance: 9.999e-05

## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pairs from Special Study Beach
pairsSSB <- meta_data.tab[meta_data.tab$Beach == "SSB",]$Pa
irID
pairsSSB <- droplevels(pairsSSB)
x <- vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach
== "SSB",], grouping = pairsSSB, distance = "bray", permuta
tions = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table.tab[meta_data.tab$Beach ==
 "SSB",      ], grouping = pairsSSB, permutations = 10000,
distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.4081
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0849 0.1110 0.1317 0.1523
##
## Dissimilarity ranks between and within classes:
##           0%     25%    50%     75% 100%    N
## Between    1  290.75  570.5  849.25 1128 1104
## Pair1      4    4.00    4.0    4.00    4    1
## Pair11   934  934.00  934.0  934.00  934    1
## Pair13   630  630.00  630.0  630.00  630    1
## Pair14     3    3.00    3.0    3.00    3    1
## Pair15    90   90.00   90.0   90.00   90    1
## Pair17   345  345.00  345.0  345.00  345    1
## Pair18    64   64.00   64.0   64.00   64    1
## Pair25    42   42.00   42.0   42.00   42    1
## Pair30   251  251.00  251.0  251.00  251    1
## Pair33    32   32.00   32.0   32.00   32    1
## Pair37    97   97.00   97.0   97.00   97    1
## Pair38   237  237.00  237.0  237.00  237    1
## Pair39   942  942.00  942.0  942.00  942    1
## Pair40   491  491.00  491.0  491.00  491    1
## Pair41    72   72.00   72.0   72.00   72    1
## Pair42   498  498.00  498.0  498.00  498    1
## Pair43  1099 1099.00 1099.0 1099.00 1099    1
## Pair44   377  377.00  377.0  377.00  377    1
## Pair45     6    6.00    6.0    6.00    6    1
## Pair46   234  234.00  234.0  234.00  234    1
## Pair47   688  688.00  688.0  688.00  688    1
## Pair48    36   36.00   36.0   36.00   36    1
## Pair49   163  163.00  163.0  163.00  163    1
## Pair5    807  807.00  807.0  807.00  807    1
```

```
# ANOSIM statistic R: 0.4081
# Significance: 9.999e-05
```

We know that some of the pairs are unrelated, thus we will repeat the analysis without these unrelated pairs.

```
library(vegan)

## Because the parentage analysis revealed that several mot
her-pup pairs are infact unrelated, we repeat the analysis
removing these unrelated pairs (Pairs49,46,15,13,11).
unrelated <- c("M49","M46","M15","M13","M11","P49","P46","P
15","P13","P11")
otu_table_rel.tab <- subset(otu_table.tab, !(rownames(otu_t
able.tab) %in% unrelated))
meta_data_rel.tab <- subset(meta_data.tab, !(rownames(meta_
data.tab) %in% unrelated))
meta_data_rel.tab$PairID <- droplevels(meta_data_rel.tab$Pa
irID)

## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pair groups
x <- vegan::anosim(dat = otu_table_rel.tab, grouping =meta_
data_rel.tab$PairID, distance = "bray", permutations = 1000
0)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table_rel.tab, grouping = meta_d
ata_rel.tab$PairID,      permutations = 10000, distance = "
bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.6016
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0924 0.1190 0.1402 0.1661
##
## Dissimilarity ranks between and within classes:
##           0%     25%    50%     75% 100%    N
## Between    1  933.75 1846.5 2751.25 3655 3612
## Pair1    415  415.00  415.0  415.00  415    1
## Pair10    23   23.00   23.0   23.00   23    1
## Pair12   174  174.00  174.0  174.00  174    1
## Pair14   337  337.00  337.0  337.00  337    1
## Pair16   286  286.00  286.0  286.00  286    1
## Pair17  1128 1128.00 1128.0 1128.00 1128    1
## Pair18   741  741.00  741.0  741.00  741    1
## Pair19    45   45.00   45.0   45.00   45    1
## Pair2    198  198.00  198.0  198.00  198    1
## Pair20   726  726.00  726.0  726.00  726    1
## Pair21   819  819.00  819.0  819.00  819    1
## Pair22  1101 1101.00 1101.0 1101.00 1101    1
## Pair23   765  765.00  765.0  765.00  765    1
## Pair24  3158 3158.00 3158.0 3158.00 3158    1
```

```
## Pair25   685  685.00  685.0  685.00  685    1
## Pair26   218  218.00  218.0  218.00  218    1
## Pair27    53   53.00   53.0   53.00   53    1
## Pair28   433  433.00  433.0  433.00  433    1
## Pair29   402  402.00  402.0  402.00  402    1
## Pair3      8    8.00    8.0    8.00    8    1
## Pair30  1024 1024.00 1024.0 1024.00 1024    1
## Pair31   895  895.00  895.0  895.00  895    1
## Pair32   219  219.00  219.0  219.00  219    1
## Pair33   644  644.00  644.0  644.00  644    1
## Pair34   121  121.00  121.0  121.00  121    1
## Pair35   276  276.00  276.0  276.00  276    1
## Pair37   798  798.00  798.0  798.00  798    1
## Pair38  1008 1008.00 1008.0 1008.00 1008    1
## Pair39  1953 1953.00 1953.0 1953.00 1953    1
## Pair4    108  108.00  108.0  108.00  108    1
## Pair40  1306 1306.00 1306.0 1306.00 1306    1
## Pair41   757  757.00  757.0  757.00  757    1
## Pair42  1315 1315.00 1315.0 1315.00 1315    1
## Pair43  2599 2599.00 2599.0 2599.00 2599    1
## Pair44  1171 1171.00 1171.0 1171.00 1171    1
## Pair45   448  448.00  448.0  448.00  448    1
## Pair47  1548 1548.00 1548.0 1548.00 1548    1
## Pair48   660  660.00  660.0  660.00  660    1
## Pair5   1705 1705.00 1705.0 1705.00 1705    1
## Pair6   1155 1155.00 1155.0 1155.00 1155    1
## Pair7      2    2.00    2.0    2.00    2    1
## Pair8    129  129.00  129.0  129.00  129    1
## Pair9    331  331.00  331.0  331.00  331    1
```

```
# ANOSIM statistic R: 0.6016
# Significance: 9.999e-05

## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pair groups from Freshwater Beach
## First remove levels from the grouping factor, otherwise
R will give an error message (but the calculations are corr
ect either way)
pairsFW <- meta_data_rel.tab[meta_data_rel.tab$Beach == "Fr
eshwater",]$PairID
pairsFW <- droplevels(pairsFW)
x <- vegan::anosim(dat = otu_table_rel.tab[meta_data_rel.ta
b$Beach == "Freshwater",], grouping = pairsFW, distance = "
bray", permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table_rel.tab[meta_data_rel.tab$
Beach ==      "Freshwater", ], grouping = pairsFW, permutat
ions = 10000,      distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.3494
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0879 0.1129 0.1335 0.1584
##
## Dissimilarity ranks between and within classes:
##           0%     25%    50%     75% 100%    N
## Between    1  290.75  569.5  849.25 1128 1104
## Pair10    23   23.00   23.0   23.00   23    1
## Pair12   174  174.00  174.0  174.00  174    1
## Pair16   286  286.00  286.0  286.00  286    1
## Pair19    45   45.00   45.0   45.00   45    1
## Pair2    198  198.00  198.0  198.00  198    1
## Pair20   694  694.00  694.0  694.00  694    1
## Pair21   753  753.00  753.0  753.00  753    1
## Pair22   911  911.00  911.0  911.00  911    1
## Pair23   717  717.00  717.0  717.00  717    1
## Pair24  1089 1089.00 1089.0 1089.00 1089    1
## Pair26   218  218.00  218.0  218.00  218    1
## Pair27    53   53.00   53.0   53.00   53    1
## Pair28   430  430.00  430.0  430.00  430    1
## Pair29   400  400.00  400.0  400.00  400    1
## Pair3      8    8.00    8.0    8.00    8    1
## Pair31   803  803.00  803.0  803.00  803    1
## Pair32   219  219.00  219.0  219.00  219    1
## Pair34   121  121.00  121.0  121.00  121    1
## Pair35   276  276.00  276.0  276.00  276    1
## Pair4    108  108.00  108.0  108.00  108    1
## Pair6    932  932.00  932.0  932.00  932    1
## Pair7      2    2.00    2.0    2.00    2    1
## Pair8    129  129.00  129.0  129.00  129    1
## Pair9    330  330.00  330.0  330.00  330    1
```

```
# ANOSIM statistic R: 0.3494
# Significance: 9.999e-05

## Perform ANOSIM to test for dissimilarity between differe
nt mother pup pairs from Special Study Beach
pairsSSB <- meta_data_rel.tab[meta_data_rel.tab$Beach == "S
SB",]$PairID
pairsSSB <- droplevels(pairsSSB)
x <- vegan::anosim(dat = otu_table_rel.tab[meta_data_rel.ta
b$Beach == "SSB",], grouping = pairsSSB, distance = "bray",
 permutations = 10000)
summary(x)
```

```
##
## Call:
## vegan::anosim(dat = otu_table_rel.tab[meta_data_rel.tab$
Beach ==      "SSB", ], grouping = pairsSSB, permutations =
 10000, distance = "bray")
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.4561
##       Significance: 9.999e-05
##
## Permutation: free
## Number of permutations: 10000
##
## Upper quantiles of permutations (null model):
```

```
##    90%    95%  97.5%    99%
## 0.0994 0.1294 0.1547 0.1851
##
## Dissimilarity ranks between and within classes:
##          0%    25%   50%    75% 100%   N
## Between   1 182.75 357.5 530.25  703 684
## Pair1     3   3.00   3.0   3.00    3   1
## Pair14    2   2.00   2.0   2.00    2   1
## Pair17  205 205.00 205.0 205.00  205   1
## Pair18   38  38.00  38.0  38.00   38   1
## Pair25   23  23.00  23.0  23.00   23   1
## Pair30  152 152.00 152.0 152.00  152   1
## Pair33   16  16.00  16.0  16.00   16   1
## Pair37   58  58.00  58.0  58.00   58   1
## Pair38  142 142.00 142.0 142.00  142   1
## Pair39  579 579.00 579.0 579.00  579   1
## Pair40  297 297.00 297.0 297.00  297   1
## Pair41   45  45.00  45.0  45.00   45   1
## Pair42  302 302.00 302.0 302.00  302   1
## Pair43  678 678.00 678.0 678.00  678   1
## Pair44  227 227.00 227.0 227.00  227   1
## Pair45    4   4.00   4.0   4.00    4   1
## Pair47  428 428.00 428.0 428.00  428   1
## Pair48   19  19.00  19.0  19.00   19   1
## Pair5   506 506.00 506.0 506.00  506   1
```

```
# ANOSIM statistic R: 0.4561
# Significance: 9.999e-05
```

Beta diversity correlations

We now want to test if beta diversity is correlated with the genetic relatedness of individuals. For this, we use the Wang relatedness estimates and a Bray-Curtis dissimilarity matrix calculated form the CSS normalised OTU table and run Mantel tests.

```
library(reshape2)
library(dplyr)
library(vegan)
library(phyloseq)

## Save the relatedness values from the previous analysis i
n a data frame
df <- relvals[,c(2,3,4)]

## To perform mantel tests the data frame has to be transfo
rmed into a distance matrix
## First, we collect the sample names from the phyloseq obj
ect and remove P22 for which we don't have relatedness esti
mates
snames <- sample_names(phylo_normMG.obj)
snames <- snames[-which(snames=="P22")]
## Create an empty matrix and fill it with the relateness e
stimates
M <- array(0, c(length(snames), length(snames)), list(sname
s, snames))
```

```
i <- match(df$ind1.id, snames)
j <- match(df$ind2.id, snames)
M[cbind(i,j)] <- M[cbind(j,i)] <- df$wang

##Also remove sample P22 from the phyloseq object
phylo_normMG_sub.obj <- subset_samples(phylo_normMG.obj, Sa
mpleNames != "P22")
#phylo_sub.obj <- subset_samples(phylo.obj, SampleNames !=
"P22")

## Extract OTU table from phyloseq object
OTU1 = as(otu_table(phylo_normMG_sub.obj), "matrix")
## Transpose the otu table
if(taxa_are_rows(phylo_normMG_sub.obj)){OTU1 <- t(OTU1)}
## Coerce to data.frame
OTUdf = as.data.frame(OTU1)

## Calulate bray-curtis distance
otu_dist_bray <- as.matrix(vegan::vegdist(as.matrix(OTUdf),
 method = "bray", diag=TRUE, upper=TRUE))

## Perform the mantel test
vegan::mantel(otu_dist_bray,M, method = "spearman", permuta
tion = 1000)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray, ydis = M, method = "
spearman",      permutations = 1000)
##
## Mantel statistic r: 0.0179
##       Significance: 0.2018
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0277 0.0360 0.0430 0.0497
## Permutation: free
## Number of permutations: 1000
```

Testing for differences in microbial composition between the age groups, overall and separately for each beach.

```
library(vegan)

## Perform mantel test separately for pups and mothers
## Get the sample names for all mothers and for all pups
snames_M <- sample_names(subset_samples(phylo_normMG.obj, A
ge=="M" ))
snames_P <- sample_names(subset_samples(phylo_normMG.obj, A
ge=="P" & SampleNames != "P22"))
## Perform the mantel test for mothers
vegan::mantel(otu_dist_bray[snames_M,snames_M],M[snames_M,s
names_M], method = "spearman", permutation = 1000)
```

```
##
```

```
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[snames_M, snames_M],
ydis = M[snames_M,      snames_M], method = "spearman", per
mutations = 1000)
##
## Mantel statistic r: 0.03241
##       Significance: 0.18581
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0462 0.0575 0.0653 0.0788
## Permutation: free
## Number of permutations: 1000
```

```
## Perform the mantel test for pups
vegan::mantel(otu_dist_bray[snames_P,snames_P], M[snames_P,
snames_P], method = "spearman", permutation = 1000)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[snames_P, snames_P],
ydis = M[snames_P,      snames_P], method = "spearman", per
mutations = 1000)
##
## Mantel statistic r: 0.0385
##       Significance: 0.15984
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0466 0.0579 0.0699 0.0820
## Permutation: free
## Number of permutations: 1000
```

```
## Perform mantel test separately for pups and mothers at e
ach beach
## Get the sample names for all mothers and for all pups
sname_M_FWB <- sample_names(subset_samples(phylo_normMG.obj
, BeachAge=="Freshwater M" ))
sname_M_SSB <- sample_names(subset_samples(phylo_normMG.obj
, BeachAge=="SSB M" ))
sname_P_FWB <- sample_names(subset_samples(phylo_normMG.obj
, BeachAge=="Freshwater P" & SampleNames != "P22" ))
sname_P_SSB <- sample_names(subset_samples(phylo_normMG.obj
, BeachAge=="SSB P" ))

## Perform the mantel test
vegan::mantel(otu_dist_bray[sname_M_FWB, sname_M_FWB],M[sna
me_M_FWB, sname_M_FWB], method = "spearman", permutation =
1000)
```

```
##
```

```
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_M_FWB, sname_M_
FWB],      ydis = M[sname_M_FWB, sname_M_FWB], method = "sp
earman",      permutations = 1000)
##
## Mantel statistic r: -0.1672
##       Significance: 0.98302
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.107 0.128 0.143 0.166
## Permutation: free
## Number of permutations: 1000
```

```
vegan::mantel(otu_dist_bray[sname_M_SSB, sname_M_SSB],M[sna
me_M_SSB, sname_M_SSB], method = "spearman", permutation =
1000)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_M_SSB, sname_M_
SSB],      ydis = M[sname_M_SSB, sname_M_SSB], method = "sp
earman",      permutations = 1000)
##
## Mantel statistic r: 0.0305
##       Significance: 0.36364
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0955 0.1157 0.1277 0.1525
## Permutation: free
## Number of permutations: 1000
```

```
vegan::mantel(otu_dist_bray[sname_P_FWB, sname_P_FWB],M[sna
me_P_FWB, sname_P_FWB], method = "spearman", permutation =
1000)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_P_FWB, sname_P_
FWB],      ydis = M[sname_P_FWB, sname_P_FWB], method = "sp
earman",      permutations = 1000)
##
## Mantel statistic r: -0.004504
##       Significance: 0.53546
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
```

```
## 0.100 0.127 0.150 0.169
## Permutation: free
## Number of permutations: 1000
```

```
vegan::mantel(otu_dist_bray[sname_P_SSB, sname_P_SSB],M[sna
me_P_SSB, sname_P_SSB], method = "spearman", permutation =
1000)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_P_SSB, sname_P_
SSB],      ydis = M[sname_P_SSB, sname_P_SSB], method = "sp
earman",      permutations = 1000)
##
## Mantel statistic r: -0.002328
##       Significance: 0.53846
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0919 0.1100 0.1301 0.1541
## Permutation: free
## Number of permutations: 1000
```

We find no correlation between the genetic relatedness of individuals and the similarity of their microbial communities.

For special study beach pupping locations have been recorded in form of x-y coordinates in a grid layout. Thus we can test if individuals that are in closer geographical proximity also share a more similar bacterial community composition. Similar to genetic relatedness we can test for correlation between geographical distance on the beach and Bray-Curtis dissimilarity using Mantel tests.

```
library(vegan)

## Correlation between geographical distance of SSB individ
uals and their microbiome similarity (beta diversity)
## Geographical locations for pupping events were collected
 from the viewing platform and coded as X,Y coordinates in
a grid system
## Read the data file
locs <- read.table("./AFSmicrobiome_SI_PuppingLocations_Rin
put_DatasetS15.txt", sep = "\t", header = T)
## Subset the dataframe for mothers
locs_M <- locs[,c(1,3,4)]
rownames(locs_M) <- locs_M$motherID
locs_M <- locs_M[,-1]
## Subset the dataframe for pups
locs_P <- locs[,2:4]
rownames(locs_P) <- locs_P$pupID
locs_P <- locs_P[,-1]

## Calculate a euclidean distance matrix from the geographi
```

```
c coordinates
geo_dist_M <- as.matrix(dist(locs_M, method = "euclidean"))
geo_dist_P <- as.matrix(dist(locs_P, method = "euclidean"))
## We need to sort the Bray matrix according to the order o
f the geo_dist matrix first
sname_M_SSB_sorted <- rownames(geo_dist_M)
sname_P_SSB_sorted <- rownames(geo_dist_P)
## Perform the mantel tests
vegan::mantel(otu_dist_bray[sname_M_SSB_sorted, sname_M_SSB
_sorted],geo_dist_M, method = "spearman", permutation = 100
0)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_M_SSB_sorted, s
name_M_SSB_sorted],      ydis = geo_dist_M, method = "spear
man", permutations = 1000)
##
## Mantel statistic r: 0.003016
##       Significance: 0.47453
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.141 0.188 0.231 0.261
## Permutation: free
## Number of permutations: 1000
```

```
vegan::mantel(otu_dist_bray[sname_P_SSB_sorted, sname_P_SSB
_sorted],geo_dist_P, method = "spearman", permutation = 100
0)
```

```
##
## Mantel statistic based on Spearman's rank correlation rh
o
##
## Call:
## vegan::mantel(xdis = otu_dist_bray[sname_P_SSB_sorted, s
name_P_SSB_sorted],      ydis = geo_dist_P, method = "spear
man", permutations = 1000)
##
## Mantel statistic r: -0.01026
##       Significance: 0.49451
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.145 0.179 0.231 0.270
## Permutation: free
## Number of permutations: 1000
```

There is no relationship between geographical distance on the beach and beta diversity.

Differential abundance analysis

We now want to statistically test which OTUs show differential abundance between the beaches and age groups. We use the DESeq2 extension in the phyloseq package to identify these differentially abundant OTUs.

```
library(DESeq2)
library(phyloseq)

## Make the DESeq object using the phyloseq function. Use b
each as variable.
dsBeach <- phyloseq_to_deseq2(phylo.obj, ~ Beach)
## Run test for differential abundance using the negative b
inomial Wald test.
dsBeachtest <- DESeq(dsBeach, test="Wald", fitType="paramet
ric")
## Extract the result table that contains log2FC and adjust
ed p-values (FDR corrected)
res_beach <- results(dsBeachtest,cooksCutoff = FALSE)
## Use an alpha cutoff of 0.01
alpha <- 0.01
sigtab_beach <- res_beach[which(res_beach$padj < alpha), ]
sigtab_beach <- cbind(as(sigtab_beach, "data.frame"), as(ta
x_table(phylo.obj)[rownames(sigtab_beach), ], "matrix"))

paste( "Overall, we find", length(sigtab_beach$log2FoldChan
ge)," significantly differentially abundant OTUs with", len
gth(which(sigtab_beach$log2FoldChange < 0)), "being signifi
cantly more abundant at FWB and", length(which(sigtab_beach
$log2FoldChange > 0)), "at SSB.")
```

```
## [1] "Overall, we find 655  significantly differentially
abundant OTUs with 380 being significantly more abundant at
 FWB and 275 at SSB."
```

```
## For plotting remove entries for which the phylum level c
lassification is not available
sigtab_beach <- sigtab_beach[-which(is.na(sigtab_beach$Phyl
um)), ]

## Order results by the largest fold change
x_beach <- tapply(sigtab_beach$log2FoldChange, sigtab_beach
$Phylum, function(x_beach) max(x_beach))
x_beach <- sort(x_beach, TRUE)
sigtab_beach$Phylum <- factor(as.character(sigtab_beach$Phy
lum), levels=names(x_beach))

## Repeat analyis with age as variable.
dsAge = phyloseq_to_deseq2(phylo.obj, ~ Age)
dsAgetest = DESeq(dsAge, test="Wald", fitType="parametric")
res_age <- results(dsAgetest,cooksCutoff = FALSE)
alpha <- 0.01
sigtab_age <- res_age[which(res_age$padj < alpha), ]
sigtab_age <- cbind(as(sigtab_age, "data.frame"), as(tax_ta
ble(phylo.obj)[rownames(sigtab_age), ], "matrix"))

paste( "Overall, we find", length(sigtab_age$log2FoldChange
)," significantly differentially abundant OTUs with", lengt
```

```
h(which(sigtab_age$log2FoldChange < 0)), "being significant
ly more abundant in mothers and", length(which(sigtab_age$l
og2FoldChange > 0)), "in pups.")
```

```
## [1] "Overall, we find 155  significantly differentially
abundant OTUs with 138 being significantly more abundant in
 mothers and 17 in pups."
```

```
sigtab_age <- sigtab_age[-which(is.na(sigtab_age$Phylum)),
]
x_age <- tapply(sigtab_age$log2FoldChange, sigtab_age$Phylu
m, function(x_age) max(x_age))
x_age <- sort(x_age, TRUE)
sigtab_age$Phylum <- factor(as.character(sigtab_age$Phylum)
, levels=names(x_age))
```

We find more differentially abundant OTUs between the two breeding colonies and than between the two age groups. We can plot the results to further examine the magnitude of the fold changes and which phyla the differentially abundant OTUs belong to.

```
library(ggplot2)

## Assign colours to the phyla (matching those from the rel
ative abundance plot)
phylcols <- c(Acidobacteria = "#673770",Actinobacteria = "#
5F7FC7", Armatimonadetes = "#ffe119", Bacteroidetes = "oran
ge", BRC1 = "#808000",Candidatus_Saccharibacteria = "#DA572
4", Chloroflexi = "#3cb44b", Cyanobacteria = "#508578",Dein
ococcus_Thermus = "#CD9BCD",Firmicutes = "#AD6F3B",Fusobact
eria = "#CBD588",Gemmatimonadetes = "#fabebe",Ignavibacteri
ae = "#aaffc3",Microgenomates = "#808080",Planctomycetes =
"#D14285",Proteobacteria = "#652926",SR1 = "#000080",Synerg
istetes = "#46f0f0",Tenericutes = "#C84248",Verrucomicrobia
 = "#8569D5")

## Change name of Candidatus_Saccharibacteria and Deinococc
us_Thermus back to the original names used in the table
names(phylcols)[which(names(phylcols)=="Candidatus_Sacchari
bacteria")] <- "Candidatus Saccharibacteria"
names(phylcols)[which(names(phylcols)=="Deinococcus_Thermus
")] <- "Deinococcus-Thermus"

## Make the plot
ggplot(sigtab_beach, aes(x=Phylum, y=log2FoldChange, colour
=Phylum)) +
            geom_point(size=2.5) +
            geom_hline(yintercept = 0,linetype = 2, colour=
"gray44")+
            theme_bw()+
            theme(axis.text.x = element_blank(),axis.ticks.
x = element_blank(),axis.title.x = element_blank(), axis.ti
tle.y = element_text(size=14), axis.text.y = element_text(s
ize=12))+
            guides(colour = guide_legend(override.aes = lis
t(shape = 15, size = 5.5, linetype=0), ncol = 1))+
            theme(legend.text = element_text( size = 10),le
gend.title = element_text(face="bold"))+
```

```
            ylab("log2FC")+
            ggtitle("DA between beaches")+
            expand_limits(y=c(-7.5,11))+
            scale_colour_manual(values=phylcols)+
            theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank())
```

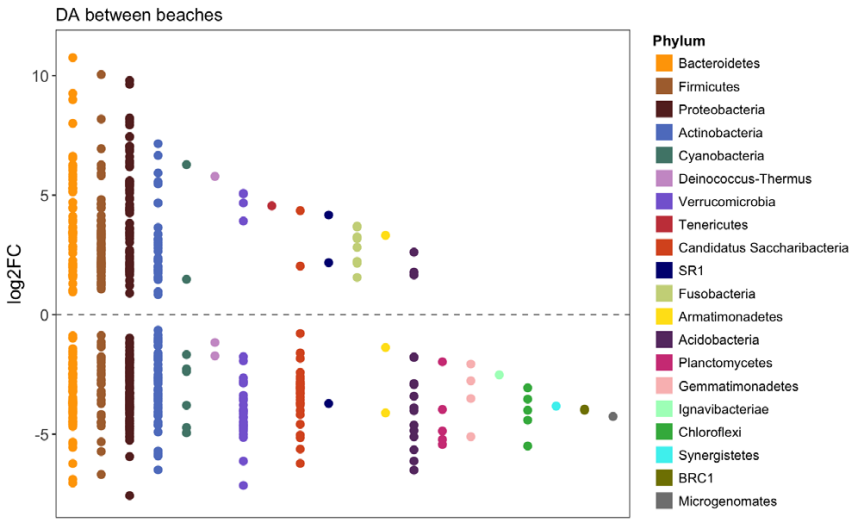


Figure 12. Differential abundance of OTUs between the two breeding colonies. OTU phylum memberships are represented by the different colours. OTUs above 0 are significantly more abundant at SSB and OTUs below 0 are significantly more abundant at FWB.

```
library(ggplot2)

## Assign colours to the phyla (matching those from the rel
ative abundance plot)
phylcols <- c(Acidobacteria = "#673770",Actinobacteria = "#
5F7FC7", Armatimonadetes = "#ffe119", Bacteroidetes = "oran
ge", BRC1 = "#808000",Candidatus_Saccharibacteria = "#DA572
4", Chloroflexi = "#3cb44b", Cyanobacteria = "#508578",Dein
ococcus_Thermus = "#CD9BCD",Firmicutes = "#AD6F3B",Fusobact
eria = "#CBD588",Gemmatimonadetes = "#fabebe",Ignavibacteri
ae = "#aaffc3",Microgenomates = "#808080",Planctomycetes =
"#D14285",Proteobacteria = "#652926",SR1 = "#000080",Synerg
istetes = "#46f0f0",Tenericutes = "#C84248",Verrucomicrobia
 = "#8569D5")

## Change name of Candidatus_Saccharibacteria and Deinococc
us_Thermus back to the original names used in the table
names(phylcols)[which(names(phylcols)=="Candidatus_Sacchari
bacteria")] <- "Candidatus Saccharibacteria"
names(phylcols)[which(names(phylcols)=="Deinococcus_Thermus
")] <- "Deinococcus-Thermus"

## Make the plot
ggplot(sigtab_age, aes(x=Phylum, y=log2FoldChange, colour=P
hylum)) +
          geom_point(size=2.5) +
          geom_hline(yintercept = 0,linetype = 2, colour="g
ray44")+
          theme_bw()+
             theme(axis.text.x = element_blank(),axis.ticks
.x = element_blank(),axis.title.x = element_blank(), axis.t
```

```
itle.y = element_text(size=14), axis.text.y = element_text(
size=12))+
          guides(colour = guide_legend(override.aes = list(
shape = 15, size = 5.5, linetype=0), ncol = 1))+
          theme(legend.text = element_text( size = 10),lege
nd.title = element_text(face="bold"))+
          expand_limits(y=c(-7.5,11))+
          ylab("log2FC")+
          ggtitle("DA between age groups")+
          scale_colour_manual(values=phylcols)+
          theme(panel.grid.major = element_blank(), panel.g
rid.minor = element_blank())
```

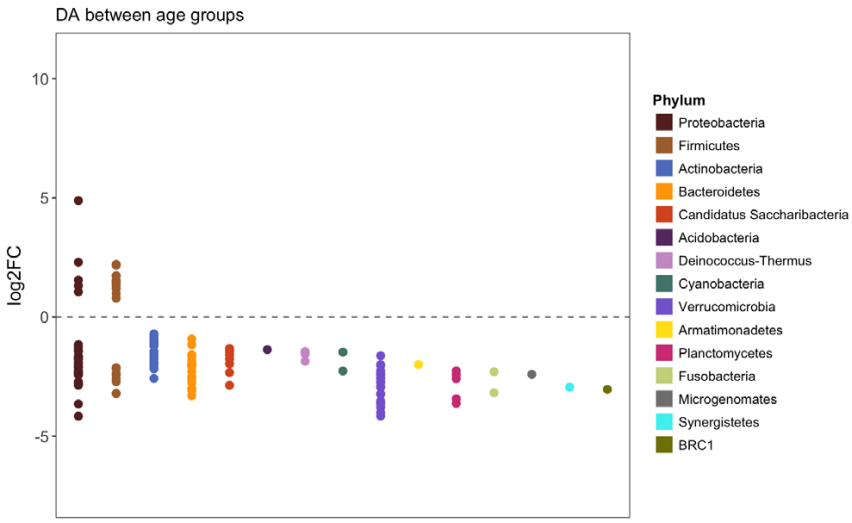


Figure 13. Differential abundance of OTUs between the two age groups. OTU phylum memberships are represented by the different colours. OTUs above 0 are significantly more abundant in pups and OTUs below 0 are significantly more abundant in mothers.

```
library(DESeq2)

## Find the differences between the age groups at each beac
h and within each age group between the beaches

## Make subsets of the data
phylo_FW.obj = subset_samples(phylo.obj, sample_data(phylo.
obj)$Beach == "Freshwater")
phylo_SSB.obj = subset_samples(phylo.obj, sample_data(phylo
.obj)$Beach == "SSB")
phylo_M.obj = subset_samples(phylo.obj, sample_data(phylo.o
bj)$Age == "M")
phylo_P.obj = subset_samples(phylo.obj, sample_data(phylo.o
bj)$Age == "P")

## Run DESeq2 for Freshwater Beach (compare age groups with
in FWB)
dsBeach_FW = phyloseq_to_deseq2(phylo_FW.obj, ~ Age)
## Run test for differential abundance using the negative b
inomial Wald test.
dsBeachtest_FW = DESeq(dsBeach_FW, test="Wald", fitType="pa
rametric")
## Extract the result table that contains logFC and adjuste
d p-values (FDR corrected)
res_FW <- results(dsBeachtest_FW,cooksCutoff = FALSE)
```

```
## Use a strict alpha cutoff of 0.01
alpha <- 0.01
sigtab_FW <- res_FW[which(res_FW$padj < alpha), ]
sigtab_FW <- cbind(as(sigtab_FW, "data.frame"), as(tax_tabl
e(phylo_FW.obj)[rownames(sigtab_FW), ], "matrix"))
# dim(sigtab_FW) #69 13
sigtab_FW <- sigtab_FW[-which(is.na(sigtab_FW$Phylum)), ]

## Order results by the largest fold change
x_FW <- tapply(sigtab_FW$log2FoldChange, sigtab_FW$Phylum,
function(x_FW) max(x_FW))
x_FW <- sort(x_FW, TRUE)
sigtab_FW$Phylum <- factor(as.character(sigtab_FW$Phylum),
levels=names(x_FW))

## Run DESeq2 for Special Study Beach (compare age groups w
ithin SSB)
dsBeach_SSB = phyloseq_to_deseq2(phylo_SSB.obj, ~ Age)
dsBeachtest_SSB = DESeq(dsBeach_SSB, test="Wald", fitType="
parametric")
res_SSB <- results(dsBeachtest_SSB,cooksCutoff = FALSE)
alpha <- 0.01
sigtab_SSB <- res_SSB[which(res_SSB$padj < alpha), ]
sigtab_SSB <- cbind(as(sigtab_SSB, "data.frame"), as(tax_ta
ble(phylo_SSB.obj)[rownames(sigtab_SSB), ], "matrix"))
# dim(sigtab_SSB) #64 13
sigtab_SSB <- sigtab_SSB[-which(is.na(sigtab_SSB$Phylum)),
]
x_SSB <- tapply(sigtab_SSB$log2FoldChange, sigtab_SSB$Phylu
m, function(x_SSB) max(x_SSB))
x_SSB <- sort(x_SSB, TRUE)
sigtab_SSB$Phylum <- factor(as.character(sigtab_SSB$Phylum)
, levels=names(x_SSB))

## Run DESeq2 for mothers (compare the two beaches for this
 age group).
dsBeach_M = phyloseq_to_deseq2(phylo_M.obj, ~ Beach)
dsBeachtest_M = DESeq(dsBeach_M, test="Wald", fitType="para
metric")
res_M <- results(dsBeachtest_M,cooksCutoff = FALSE)
alpha <- 0.01
sigtab_M <- res_M[which(res_M$padj < alpha), ]
sigtab_M <- cbind(as(sigtab_M, "data.frame"), as(tax_table(
phylo_M.obj)[rownames(sigtab_M), ], "matrix"))
# dim(sigtab_M) #610  13
sigtab_M <- sigtab_M[-which(is.na(sigtab_M$Phylum)), ]
x_M <- tapply(sigtab_M$log2FoldChange, sigtab_M$Phylum, fun
ction(x_M) max(x_M))
x_M <- sort(x_M, TRUE)
sigtab_M$Phylum <- factor(as.character(sigtab_M$Phylum), le
vels=names(x_M))

## Run DESeq2 for pups (compare the two beaches for this ag
e group).
dsBeach_P = phyloseq_to_deseq2(phylo_P.obj, ~ Beach)
dsBeachtest_P = DESeq(dsBeach_P, test="Wald", fitType="para
metric")
res_P <- results(dsBeachtest_P,cooksCutoff = FALSE)
alpha <- 0.01
sigtab_P <- res_P[which(res_P$padj < alpha), ]
```

```
sigtab_P <- cbind(as(sigtab_P, "data.frame"), as(tax_table(
phylo_P.obj)[rownames(sigtab_P), ], "matrix"))
# dim(sigtab_P) #487  13
sigtab_P <- sigtab_P[-which(is.na(sigtab_P$Phylum)), ]
x_P = tapply(sigtab_P$log2FoldChange, sigtab_P$Phylum, func
tion(x_P) max(x_P))
x_P = sort(x_P, TRUE)
sigtab_P$Phylum = factor(as.character(sigtab_P$Phylum), lev
els=names(x_P))

## Which groups have more abundant OTUs?
paste( "At FWB", length(which(sigtab_FW$log2FoldChange < 0)
), "OTUs are significantly more abundant in mothers and", l
ength(which(sigtab_FW$log2FoldChange > 0)), "in pups.")
```

```
## [1] "At FWB 37 OTUs are significantly more abundant in m
others and 40 in pups."
```

```
paste( "At SSB", length(which(sigtab_SSB$log2FoldChange < 0
)), "OTUs are significantly more abundant in mothers and",
length(which(sigtab_SSB$log2FoldChange > 0)), "in pups.")
```

```
## [1] "At SSB 55 OTUs are significantly more abundant in m
others and 8 in pups."
```

```
paste( "In the mother cohort", length(which(sigtab_M$log2Fo
ldChange < 0)), "OTUs are significantly more abundant at FW
B and", length(which(sigtab_M$log2FoldChange > 0)), "at SSB
.")
```

```
## [1] "In the mother cohort 337 OTUs are significantly mor
e abundant at FWB and 242 at SSB."
```

```
paste( "In the pup cohort", length(which(sigtab_P$log2FoldC
hange < 0)), "OTUs are significantly more abundant at FWB a
nd", length(which(sigtab_P$log2FoldChange > 0)), "at SSB.")
```

```
## [1] "In the pup cohort 298 OTUs are significantly more a
bundant at FWB and 164 at SSB."
```

```
library(cowplot)
library(ggplot2)

## Assign colours to the phyla (matching those from the rel
ative abundance plot)
phylcols <- c(Acidobacteria = "#673770",Actinobacteria = "#
5F7FC7", Armatimonadetes = "#ffe119", Bacteroidetes = "oran
ge", BRC1 = "#808000",Candidatus_Saccharibacteria = "#DA572
4", Chloroflexi = "#3cb44b", Cyanobacteria = "#508578",Dein
ococcus_Thermus = "#CD9BCD",Firmicutes = "#AD6F3B",Fusobact
eria = "#CBD588",Gemmatimonadetes = "#fabebe",Ignavibacteri
ae = "#aaffc3",Microgenomates = "#808080",Planctomycetes =
"#D14285",Proteobacteria = "#652926",SR1 = "#000080",Synerg
```

```
istetes = "#46f0f0",Tenericutes = "#C84248",Verrucomicrobia
 = "#8569D5")

names(phylcols)[which(names(phylcols)=="Candidatus_Sacchari
bacteria")] <- "Candidatus Saccharibacteria"
names(phylcols)[which(names(phylcols)=="Deinococcus_Thermus
")] <- "Deinococcus-Thermus"

## Make the plot
FW <- ggplot(sigtab_FW, aes(x=Phylum, y=log2FoldChange, col
our=Phylum)) +
          geom_point(size=2) +
          geom_hline(yintercept = 0,linetype = 2, colour="g
ray44")+
          theme_bw()+
          theme(axis.text.x = element_blank(),axis.ticks.x
= element_blank(),axis.title.x = element_blank(),axis.text.
y = element_text(size=10))+
          guides(colour = guide_legend(override.aes = list(
shape = 15, size = 5.5, linetype=0), ncol = 1))+
          theme(legend.text = element_text( size = 10),lege
nd.title = element_text(face="bold"))+
          ylab("log2FC")+
          expand_limits(y=c(-7.5,11))+
          theme(legend.position="none")+
          ggtitle("DA between age groups at FWB")+
          scale_colour_manual(values=phylcols)+
          theme(panel.grid.major = element_blank(),panel.gr
id.minor = element_blank())

## Make the plot
SSB <- ggplot(sigtab_SSB, aes(x=Phylum, y=log2FoldChange, c
olour=Phylum)) +
        geom_point(size=2) +
        geom_hline(yintercept = 0,linetype = 2, colour="gra
y44")+
        theme_bw()+
        theme(axis.text.x = element_blank(),axis.ticks.x =
element_blank(),axis.title.x = element_blank(),axis.text.y
= element_text(size=10))+
        guides(colour = guide_legend(override.aes = list(sh
ape = 15, size = 5.5, linetype=0), ncol = 1))+
        theme(legend.text = element_text( size = 10),legend
.title = element_text(face="bold"))+
        ylab("log2FC")+
        expand_limits(y=c(-7.5,11))+
        theme(legend.position="none")+
        ggtitle("DA between age groups at SSB")+
        scale_colour_manual(values=phylcols)+
        theme(panel.grid.major = element_blank(),panel.grid
.minor = element_blank())

## Make the plot
M <- ggplot(sigtab_M, aes(x=Phylum, y=log2FoldChange, colou
r=Phylum)) +
        geom_point(size=2) +
        geom_hline(yintercept = 0,linetype = 2, colour="gra
y44")+
        theme_bw()+
        theme(axis.text.x = element_blank(),axis.ticks.x =
```

```
element_blank(),axis.title.x = element_blank(),axis.text.y
= element_text(size=10))+
        guides(colour = guide_legend(override.aes = list(sh
ape = 15, size = 5.5, linetype=0), ncol = 1))+
        theme(legend.text = element_text( size = 10),legend
.title = element_text(face="bold"))+
        ylab("log2FC")+
        expand_limits(y=c(-7.5,11))+
        scale_colour_manual(values=phylcols)+
        ggtitle("DA in mothers between beaches")+
        theme(legend.position="none")+
        theme(panel.grid.major = element_blank(),panel.grid
.minor = element_blank())

## Make the plot
P <- ggplot(sigtab_P, aes(x=Phylum, y=log2FoldChange, colou
r=Phylum)) +
        geom_point(size=2) +
        geom_hline(yintercept = 0,linetype = 2, colour="gra
y44")+
        theme_bw()+
        theme(axis.text.x = element_blank(),axis.ticks.x =
element_blank(), axis.title.x = element_blank(),axis.text.y
 = element_text(size=10))+
        guides(colour = guide_legend(override.aes = list(sh
ape = 15, size = 5.5, linetype=0), ncol = 1))+
        theme(legend.text = element_text( size = 10),legend
.title = element_text(face="bold"))+
        ylab("log2FC")+
        expand_limits(y=c(-7.5,11))+
        scale_colour_manual(values=phylcols)+
        ggtitle("DA in pups between beaches")+
        theme(legend.position="none")+
        theme(panel.grid.major = element_blank(),panel.grid
.minor = element_blank())

plot_grid(M,FW,P,SSB, align = "v",axis="r" ,nrow=2, ncol=2)
```

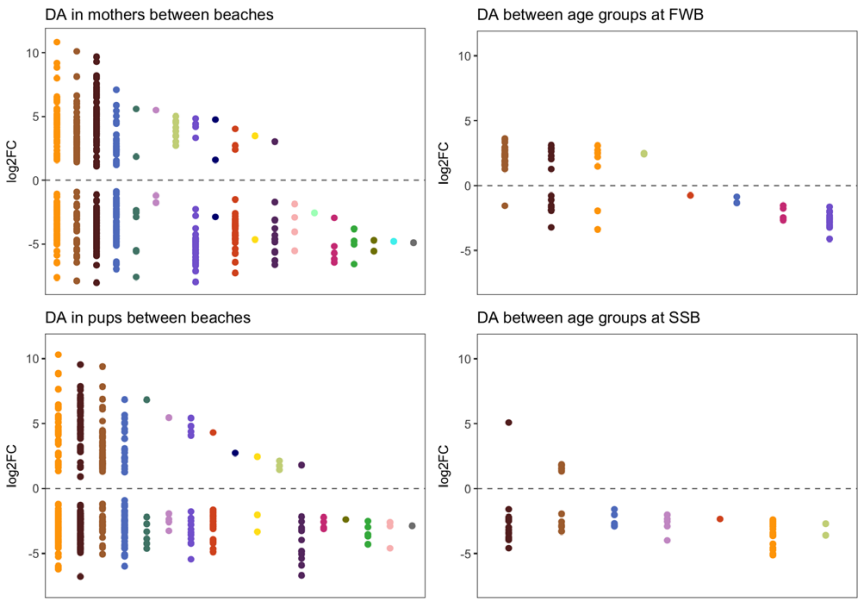


Figure 14. Differential abundance of OTUs between the two breeding colonies and the two age groups. OTU phylum memberships are represented by the different colours. OTUs above 0 are significantly more abundant at SSB/in pups and OTUs

below 0 are significantly more abundant at FWB/in mothers.

Heatmap of OTU abundance

The OTU abundance for each sample can also be visualised using a heatmap. As input for abundance we use the CSS normalised OTU counts with added pseudocount.

```
#library(phyloseq)
library(phyloseq)
library(pheatmap)
library(dplyr)

## Define colours for the heatmap and
## the colour gradient for abundance
heatcols <- c("#EFDEC0","#EFBE95", "#E48889", "#DA577C", "#C13177", "#901A81", "#640089", "#480076", "#350155")
heatmapCols <- colorRampPalette(heatcols)(50)

## Define the phyla colours
phylcols <- c(Acidobacteria = "#673770",Actinobacteria = "#5F7FC7", Armatimonadetes = "#ffe119", Bacteroidetes = "orange", BRC1 = "#808000",Candidatus_Saccharibacteria = "#DA5724", Chloroflexi = "#3cb44b", Cyanobacteria = "#508578",Deinococcus_Thermus = "#CD9BCD",Firmicutes = "#AD6F3B",Fusobacteria = "#CBD588",Gemmatimonadetes = "#fabebe",Ignavibacteriae = "#aaffc3",Microgenomates = "#808080",Planctomycetes = "#D14285",Proteobacteria = "#652926",SR1 = "#000080",Synergistetes = "#46f0f0",Tenericutes = "#C84248",Verrucomicrobia = "#8569D5",undefined = "lightgrey")

## Correct some names to match the naming in the table
names(phylcols)[which(names(phylcols)=="Candidatus_Saccharibacteria")] <- "Candidatus Saccharibacteria"
names(phylcols)[which(names(phylcols)=="Deinococcus_Thermus")] <- "Deinococcus-Thermus"

## Define the colours used for mothers and pups at each beach
samplecols <- c(FWB_mothers= "dodgerblue3",FWB_pups = "#d7e4f5", SSB_mothers = "firebrick2", SSB_pups = "#ffd6d7")

## Make a list of the colour vectors for the heatmap function
ann_colors <- list(Phylum = phylcols, BeachAge = samplecols)
## Correct the names to match the naming in the table
names(ann_colors$BeachAge)[which(names(ann_colors$BeachAge)=="FWB_mothers")] <- "FWB mothers"
names(ann_colors$BeachAge)[which(names(ann_colors$BeachAge)=="FWB_pups")] <- "FWB pups"
names(ann_colors$BeachAge)[which(names(ann_colors$BeachAge)=="SSB_mothers")] <- "SSB mothers"
names(ann_colors$BeachAge)[which(names(ann_colors$BeachAge)=="SSB_pups")] <- "SSB pups"

## The abundance will be based on the CSS normalised OTU counts with added pseudocount (input for the beta diversity a
```

```
nalysis)
## Extract the taxonomic information for OTUs from the phyl
oseq object
heatmap.tab <- as.data.frame(as(tax_table(phylo.obj)[rownam
es(metag.norm.counts_log2), ], "matrix"))

## Make a data frame that has the sample meta information a
bout beach and age of the individuals
## (The rownames have to be present to plot the heatmap)
colannot <- as.data.frame(meta_data.tab$BeachAge, row.names
 = rownames(meta_data.tab))
colnames(colannot) <- "BeachAge"
levels(colannot$BeachAge) <- c(levels(colannot$BeachAge), "
FWB mothers","SSB mothers","FWB pups", "SSB pups" )
colannot$BeachAge[which(colannot$BeachAge=="Freshwater M")]
 <- "FWB mothers"
colannot$BeachAge[which(colannot$BeachAge=="Freshwater P")]
 <- "FWB pups"
colannot$BeachAge[which(colannot$BeachAge=="SSB M")] <- "SS
B mothers"
colannot$BeachAge[which(colannot$BeachAge=="SSB P")] <- "SS
B pups"
colannot$BeachAge <- droplevels(colannot$BeachAge)

## Make a data frame that contains the phylum information f
or each OTU
## (The rownames have to be present to plot the heatmap)
rowannot <- as.data.frame(heatmap.tab[,"Phylum"])
colnames(rowannot) <- "Phylum"
levels(rowannot$Phylum) <- c(levels(rowannot$Phylum), "unde
fined")
rowannot$Phylum[is.na(rowannot$Phylum)] <- "undefined"
rowannot$Phylum <- droplevels(rowannot$Phylum)

## Plot the heatmap
pheatmap::pheatmap(metag.norm.counts_log2, color=heatmapCol
s, annotation_col=colannot, annotation_row=rowannot, show_r
ownames = FALSE, annotation_colors=ann_colors, drop_levels=
 TRUE, fontsize_row = 1, fontsize_col = 4, annotation_names
_row=FALSE, annotation_names_col=FALSE)
```

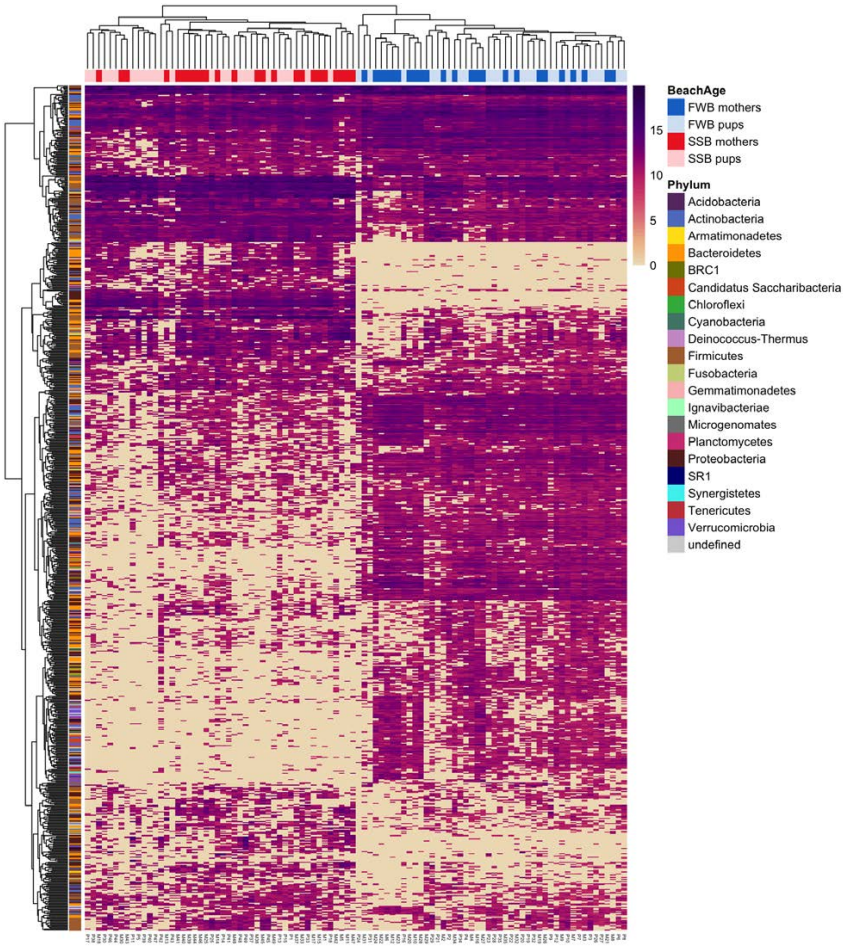


Figure 15. Heatmap of OTU abundance. Each column corresponds to one individual and each row corresponds to an OTUs. Abundance is represented by the log transformed CSS normalised OTU counts with added pseudocounts. The horizontal bar above the plot indicates which breeding colony and age group an individual belongs to. The vertical bar on the left-hand side of the plot represents the phylum membership of each OTU.

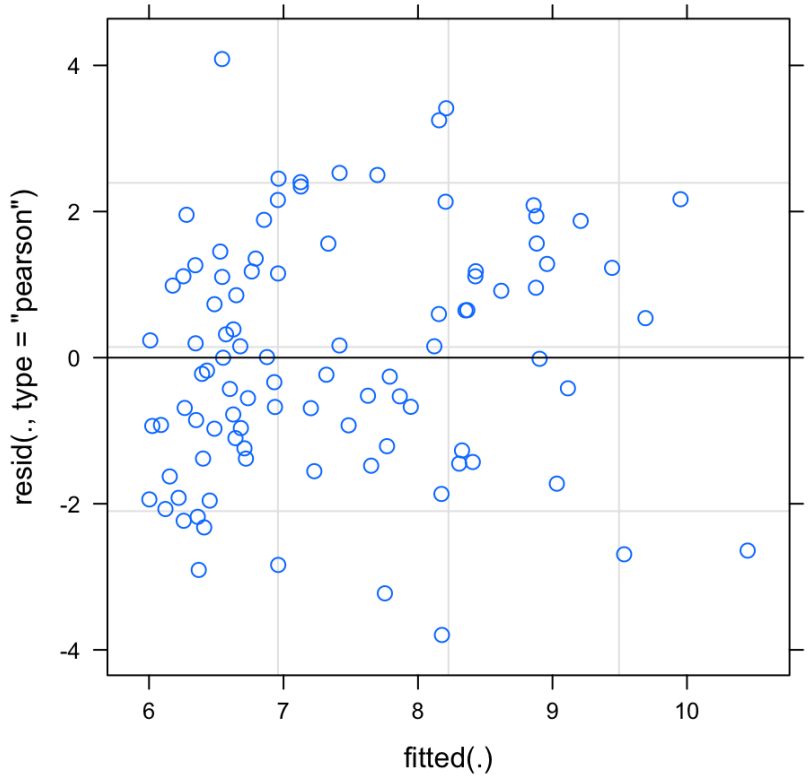
Heterozygosity & bacterial diversity

We want to explore the relationship between heterozygosity and bacterial diversity as it has been suggested that the host can excert some control over its microbial community. We hypothesise that the strength of control over the microbiota depends on the heterozygosity of an individual. Standardised multilocus heterozygosity (sMLH, total number of heterozygous loci in an individual divided by the sum of average observed heterozygosities in the population over the subset of loci successfully typed in the focal individual) was calculated for each individual with inbreedR.

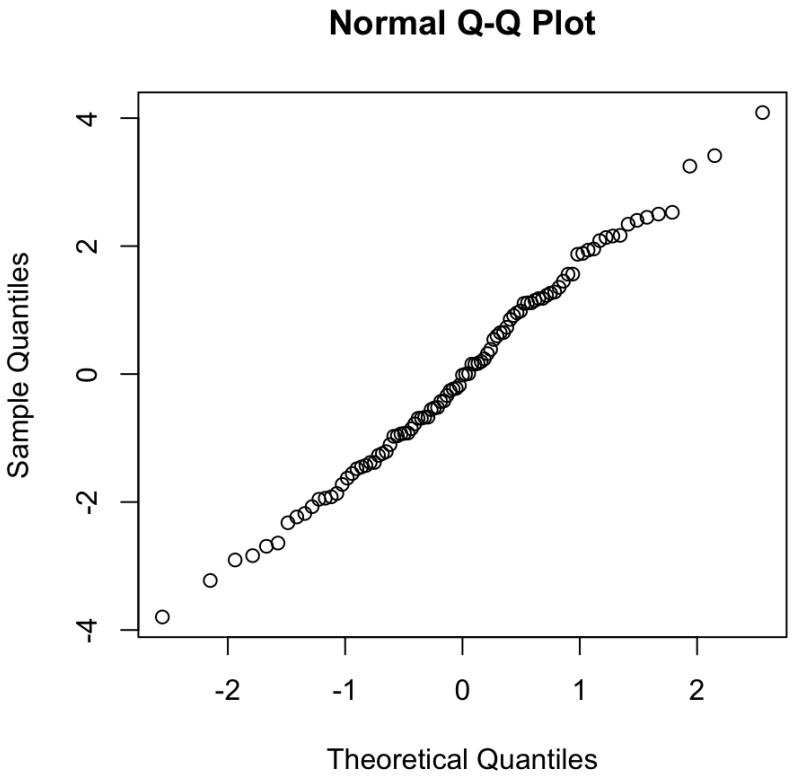
We use LMMs and include interaction terms to investigate whether the effect of an individual's heterozygosity on alpha diversity is different between the age classes and breeding colonies.

```
library(inbreedR)
library(dplyr)
library(ggplot2)
library(lme4)
library(MuMIn)
```

```
## Calculate individual heterozygozity and correlate with alpha diversity

## Import the preformatted microsatellite table and remove NAs
msats <- read.table("./AFSmicrobiome_SI_MicrosatelliteGenotypes50_P22removed_colnames_Rinput_DatasetS5.1.txt",header=TRUE,row.names = 1, sep= "\t", na.strings=c("","NA"))
is.na(msats) <- !msats
## Convert to inbreedR format
genos <- inbreedR::convert_raw(msats)
## Calculate heterozygosity (sMLH)
heterozygosity <- inbreedR::sMLH(genos)

## Use the alpha diversity estimates from the alpha_model_unrel.tab
het_alpha.tab <- as.data.frame(subset(alpha_model_unrel.tab, select=c("Beach","Age","PairID","SampleID","jost1_all","PairID2")))
heterozygosity <- as.data.frame(heterozygosity)
heterozygosity["SampleID"] <- rownames(heterozygosity)
het_alpha.tab <- left_join(het_alpha.tab, heterozygosity, by="SampleID")
het_alpha.tab["BeachAge"] <- paste(het_alpha.tab$Beach,het_alpha.tab$Age )

## Run LMMs to examine the relationship between heterozygosity and bacterial alpha diversity. We include two interaction terms to investigate if the effect of an individual's heterozygosity on alpha diversity is different between the breeding colonies and the age classes.

## Centre heterozygosity
het_alpha.tab <- cbind(het_alpha.tab, sHeteroz = scale(het_alpha.tab$heterozygosity,scale=FALSE,center = TRUE))

## Run the full model for FWB
model_full <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age + sHeteroz*Beach + sHeteroz*Age + (1|PairID2), data = het_alpha.tab)
## Check the residual plots
plot(model_full)
```




```
qqnorm(resid(model_full))
```




```
## Examine the model output
summary(model_full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sqrt(jost1_all) ~ sHeteroz + Beach + Age + sHet
eroz * Beach +
##     sHeteroz * Age + (1 | PairID2)
##    Data: het_alpha.tab
##
## REML criterion at convergence: 367.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1589 -0.6977 -0.0091  0.6718  2.3245
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  PairID2  (Intercept) 0.3123   0.5588
##  Residual             3.0917   1.7583
## Number of obs: 95, groups:  PairID2, 53
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)         8.3744     0.3350  24.995
## sHeteroz           -9.8477     3.8534  -2.556
## BeachSSB           -1.6848     0.3936  -4.281
## AgeP               -0.2879     0.3682  -0.782
## sHeteroz:BeachSSB   7.7464     4.9056   1.579
## sHeteroz:AgeP      -0.3200     4.9871  -0.064
##
## Correlation of Fixed Effects:
##             (Intr) sHetrz BchSSB AgeP   sH:BSS
## sHeteroz    -0.104
## BeachSSB    -0.593  0.001
## AgeP        -0.536  0.103 -0.010
## sHtrz:BcSSB -0.020 -0.483 -0.001 -0.001
## sHeterz:AgP  0.092 -0.482  0.001  0.008 -0.230
```

```
## Calculates the marginal (only for fixed effects) and con
ditional (for all effects) R squared for the LMM
r.squaredGLMM(model_full)
```

```
##       R2m       R2c
## 0.2354759 0.3056198
```

```
## Likelihood ratio tests
## anova does not work if missing data are present. Heteroz
ygosity could not be calculated for one individual due to m
issing genotypes. This invididual will be removed from the
table first. This does not change the results of the model
above.
het_alpha2.tab <- het_alpha.tab
het_alpha2.tab <- het_alpha2.tab[-which(is.na(het_alpha2.ta
b$heterozygosity)),]

full <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age + + s
Heteroz*Beach + sHeteroz*Age + (1|PairID2),  data = het_alp
ha2.tab, REML=FALSE)
## test interaction first. If not significant remove from m
odel
```

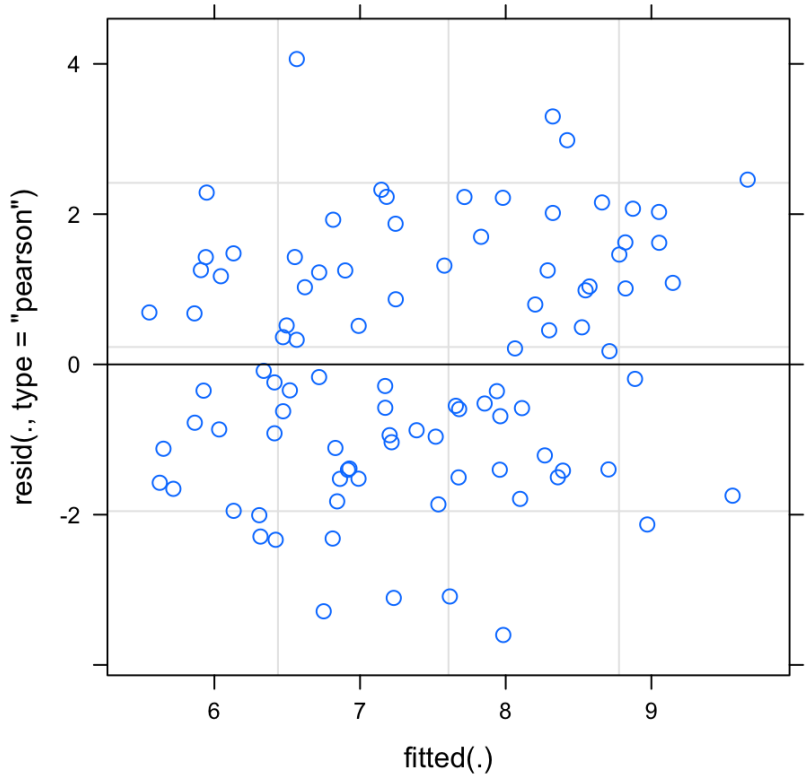
```
ageint <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age + s
Heteroz*Beach + (1|PairID2),  data = het_alpha2.tab, REML=F
ALSE)
beachint <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age +
 sHeteroz*Age + (1|PairID2),  data = het_alpha2.tab, REML=F
ALSE)

anova(full,beachint)
```

```
## Data: het_alpha2.tab
## Models:
## beachint: sqrt(jost1_all) ~ sHeteroz + Beach + Age + sHe
teroz * Age + (1 |
## beachint:     PairID2)
## full: sqrt(jost1_all) ~ sHeteroz + Beach + Age + +sHeter
oz * Beach +
## full:     sHeteroz * Age + (1 | PairID2)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df
 Pr(>Chisq)
## beachint  7 396.03 413.91 -191.01   382.03

## full      8 395.41 415.84 -189.71   379.41 2.6177      1
     0.1057
```

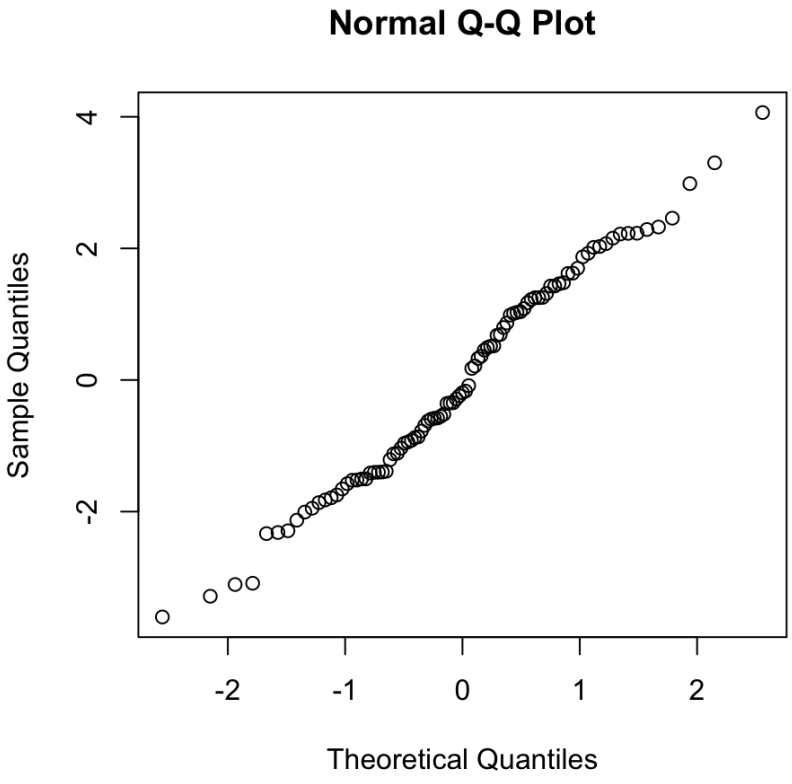
```
anova(full,ageint)
```

```
## Data: het_alpha2.tab
## Models:
## ageint: sqrt(jost1_all) ~ sHeteroz + Beach + Age + sHete
roz * Beach +
## ageint:     (1 | PairID2)
## full: sqrt(jost1_all) ~ sHeteroz + Beach + Age + +sHeter
oz * Beach +
## full:     sHeteroz * Age + (1 | PairID2)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df P
r(>Chisq)
## ageint  7 393.42 411.30 -189.71   379.42

## full    8 395.41 415.84 -189.71   379.41 0.0051      1
   0.9432
```

```
## refit without interactions
model_full <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age
 + (1|PairID2), data = het_alpha.tab)
## Check the residual plots
plot(model_full)
```




```
qqnorm(resid(model_full))
```




```
## Examine the model output
summary(model_full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: sqrt(jost1_all) ~ sHeteroz + Beach + Age + (1 |
 PairID2)
##    Data: het_alpha.tab
##
## REML criterion at convergence: 380.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0646 -0.7981 -0.1106  0.7185  2.3301
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  PairID2  (Intercept) 0.3902   0.6247
##  Residual             3.0426   1.7443
## Number of obs: 95, groups:  PairID2, 53
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   8.3737     0.3363  24.901
## sHeteroz     -6.0578     2.4273  -2.496
## BeachSSB     -1.6819     0.3987  -4.218
## AgeP         -0.2863     0.3658  -0.783
##
## Correlation of Fixed Effects:
##          (Intr) sHetrz BchSSB
## sHeteroz -0.094
## BeachSSB -0.600  0.001
## AgeP     -0.531  0.172 -0.010
```

```
## Calculates the marginal (only for fixed effects) and con
ditional (for all effects) R squared for the LMM
r.squaredGLMM(model_full)
```

```
##       R2m       R2c
## 0.2153902 0.3045733
```

```
## New full model without interactions
full <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + Age + (1|
PairID2),  data = het_alpha2.tab, REML=FALSE)
#LRT
het <- lmer(sqrt(jost1_all) ~ Beach + Age + (1|PairID2),  d
ata = het_alpha2.tab, REML=FALSE)
beach <- lmer(sqrt(jost1_all) ~ sHeteroz + Age + (1|PairID2
),  data = het_alpha2.tab, REML=FALSE)
age <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach + (1|PairID2
),  data = het_alpha2.tab, REML=FALSE)

anova(full,het)
```

```
## Data: het_alpha2.tab
## Models:
## het: sqrt(jost1_all) ~ Beach + Age + (1 | PairID2)
## full: sqrt(jost1_all) ~ sHeteroz + Beach + Age + (1 | Pa
irID2)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(
```

```
>Chisq)
## het   5 397.83 410.60 -193.91   387.83

## full  6 394.13 409.46 -191.07   382.13 5.6931      1
0.01703 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
```

```
anova(full,beach)
```

```
## Data: het_alpha2.tab
## Models:
## beach: sqrt(jost1_all) ~ sHeteroz + Age + (1 | PairID2)
## full: sqrt(jost1_all) ~ sHeteroz + Beach + Age + (1 | Pa
irID2)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr
(>Chisq)
## beach  5 407.75 420.52 -198.88   397.75

## full   6 394.13 409.46 -191.07   382.13 15.621      1  7
.739e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
```

```
anova(full,age)
```

```
## Data: het_alpha2.tab
## Models:
## age: sqrt(jost1_all) ~ sHeteroz + Beach + (1 | PairID2)
## full: sqrt(jost1_all) ~ sHeteroz + Beach + Age + (1 | Pa
irID2)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(
>Chisq)
## age   5 392.77 405.54 -191.38   382.77

## full  6 394.13 409.46 -191.07   382.13 0.6362      1
 0.4251
```

We find that alpha diversity and heterozygosity are significantly correlated, with decreased alpha diversity in more heterozygous individuals. The non-significant interaction terms suggest that the effect does not differ between the age classes and breeding colonies.

```
library(ggplot2)

## Draw the plot with separate points and ablines for each
group (e.g. FWB mothers)
model_full_int <- lmer(sqrt(jost1_all) ~ sHeteroz + Beach +
 Age + sHeteroz*Beach + sHeteroz*Age + (1|PairID2), data =
het_alpha.tab)
```

```
#summary(model_full_int)$coefficients
#                     Estimate Std. Error     t value
# (Intercept)        8.3743789  0.3350369 24.99539266
# sHeteroz          -9.8476947  3.8534081 -2.55558055
# BeachSSB          -1.6847756  0.3935801 -4.28064240
# AgeP              -0.2879318  0.3681890 -0.78202167
# sHeteroz:BeachSSB  7.7464112  4.9055802  1.57910193
# sHeteroz:AgeP     -0.3200377  4.9871232 -0.06417281

intercept_FWB_M <- summary(model_full_int)$coefficients[1,1
]
intercept_SSB_M <- (summary(model_full_int)$coefficients[1,
1])+(summary(model_full_int)$coefficients[3,1])
intercept_FWB_P <- (summary(model_full_int)$coefficients[1,
1])+(summary(model_full_int)$coefficients[4,1])
intercept_SSB_P <- (summary(model_full_int)$coefficients[1,
1])+(summary(model_full_int)$coefficients[3,1])+(summary(mo
del_full_int)$coefficients[4,1])
slope_FWB_M <- summary(model_full_int)$coefficients[2,1]
slope_SSB_M <- (summary(model_full_int)$coefficients[2,1])+
(summary(model_full_int)$coefficients[5,1])
slope_FWB_P <- (summary(model_full_int)$coefficients[2,1])+
(summary(model_full_int)$coefficients[6,1])
slope_SSB_P <- (summary(model_full_int)$coefficients[2,1])+
(summary(model_full_int)$coefficients[5,1])+(summary(model_
full_int)$coefficients[6,1])



ggplot() +
      geom_point(aes(x=het_alpha.tab$sHeteroz[het_alpha.tab
$BeachAge == "SSB P"], y=sqrt(het_alpha.tab$jost1_all[het_a
lpha.tab$BeachAge == "SSB P"])),colour = "firebrick2",shape
=0, size = 2.5) +
     geom_segment(aes(x = -0.25, xend = 0.18, y = (intercep
t_SSB_P+slope_SSB_P*-0.25), yend = (intercept_SSB_P+slope_S
SB_P*0.18)), size = 1 ,linetype="dotdash", colour="firebric
k2") +
      geom_point(aes(x=het_alpha.tab$sHeteroz[het_alpha.tab
$BeachAge == "Freshwater P"], y=sqrt(het_alpha.tab$jost1_al
l[het_alpha.tab$BeachAge == "Freshwater P"])),colour = "dod
gerblue3",shape=1, size = 2.5) +
       geom_segment(aes(x = -0.25, xend = 0.18, y = (interc
ept_FWB_P+slope_FWB_P*-0.25), yend = (intercept_FWB_P+slope
_FWB_P*0.18)), size = 1 ,linetype="dotdash", colour="dodger
blue3") +
        geom_point(aes(x=het_alpha.tab$sHeteroz[het_alpha.t
ab$BeachAge == "SSB M"], y=sqrt(het_alpha.tab$jost1_all[het
_alpha.tab$BeachAge == "SSB M"])),colour = "firebrick2",sha
pe=15, size = 2.5) +
         geom_segment(aes(x = -0.25, xend = 0.18, y = (inte
rcept_SSB_M+slope_SSB_M*-0.25), yend = (intercept_SSB_M+slo
pe_SSB_M*0.18)), size = 1 , colour="firebrick2") +
        geom_point(aes(x=het_alpha.tab$sHeteroz[het_alpha.
tab$BeachAge == "Freshwater M"], y=sqrt(het_alpha.tab$jost1
_all[het_alpha.tab$BeachAge == "Freshwater M"])),colour = "
dodgerblue3", shape=19, size = 2.5) +
          geom_segment(aes(x = -0.25, xend = 0.18, y = (int
ercept_FWB_M+slope_FWB_M*-0.25), yend = (intercept_FWB_M+sl
ope_FWB_M*0.18)), size = 1 , colour="dodgerblue3") +
```

```
      theme_bw(base_size = 12)+
      theme(panel.grid.major = element_blank(), panel.grid.
minor = element_blank())+
      theme(axis.text.x = element_text(size=12), axis.title
.x = element_text(size=14),axis.text.y = element_text(size=
12),axis.title.y = element_text(size=14),plot.margin = unit
(c(.5, .5, .5, .5), "cm"))+
       #scale_x_continuous(breaks=c(seq(from = -0.3, to = 0
.25, by = 0.1))) +
      xlab("Centred sMLH") +
      ylab("Effective no. of species (sqrt)")
```

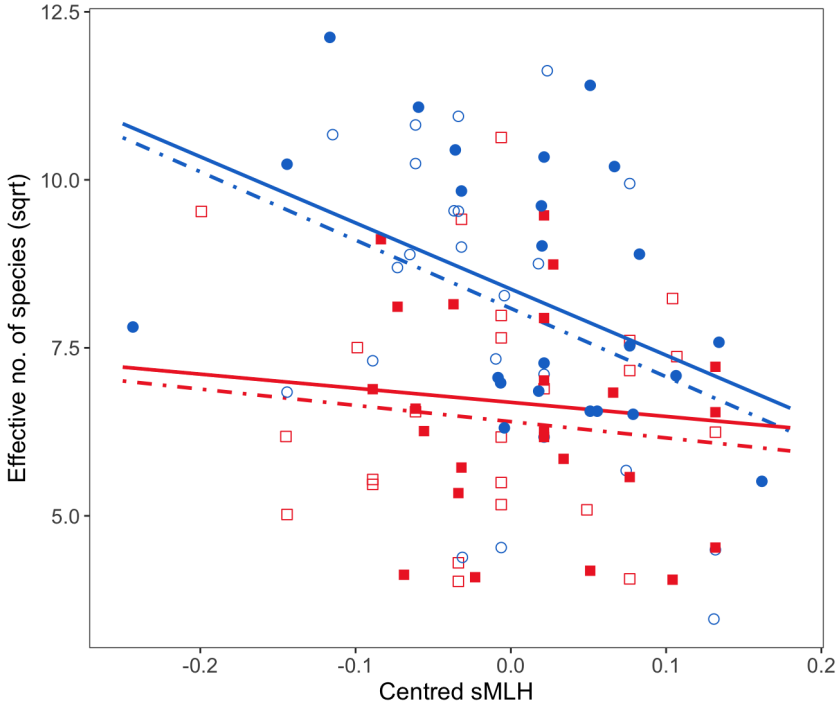


Figure 16. Relationship between bacterial alpha diversity (effective number of species, square root transformed) and individual heterozygosity (sMLH, centered around the mean). Plotted are the rawdata and regression lines from the LMM (heterozygosity regressed against alpha diversity, while controlling for breeding colony and age and including interaction terms beach x sMLH and age x sMLH). FWB mothers - blue filled circles and solid line, FWB pups - blue empty circles and dashed line, SSB mothers - red filled squares and solid line, SSB pups - red empty squares and dashed line.

We wanted to know if our results could potentially be biased by using the non-normalised OTU table for the calculation of alpha diversity values (despite the strong correlation observed between alpha diversity estimates from the non-normalised and single rarefied OTU table). To this end, we rarefied the OTU table to 10,000 reads per sample 100 times (multiple rarefaction) using the QIIME multiple_rarefactions_even_depth.py script and calculated alpha diversity for each of the rarefied tables. We can now calculate LMMs for each set of alpha diversity values to see how robust the model estimates are to rarefaction.

```
library(lme4)
library(MuMIn)
library(ggplot2)
```

```
## Set the alpha diversity values for P24 and P39 to "NA" t
o make the analysis comparable to the muliple rarefied data
 set, where these two samples are missing.
het_alpha3.tab <- het_alpha.tab
het_alpha3.tab[c(which(het_alpha3.tab$SampleID=="P39"), whi
ch(het_alpha.tab$SampleID=="P24")),]$jost1_all <- NA
## Run the full model as above
model_full_int_jost1_all <- lmer(sqrt(jost1_all) ~ sHeteroz
 + Beach + Age + sHeteroz*Beach + sHeteroz*Age + (1|PairID2
), data = het_alpha3.tab)

intercept_FWB_M_j1a <- summary(model_full_int_jost1_all)$co
efficients[1,1]
intercept_SSB_M_j1a <- (summary(model_full_int_jost1_all)$c
oefficients[1,1])+(summary(model_full_int_jost1_all)$coeffi
cients[3,1])
intercept_FWB_P_j1a <- (summary(model_full_int_jost1_all)$c
oefficients[1,1])+(summary(model_full_int_jost1_all)$coeffi
cients[4,1])
intercept_SSB_P_j1a <- (summary(model_full_int_jost1_all)$c
oefficients[1,1])+(summary(model_full_int_jost1_all)$coeffi
cients[3,1])+(summary(model_full_int_jost1_all)$coefficient
s[4,1])
slope_FWB_M_j1a <- summary(model_full_int_jost1_all)$coeffi
cients[2,1]
slope_SSB_M_j1a <- (summary(model_full_int_jost1_all)$coeff
icients[2,1])+(summary(model_full_int_jost1_all)$coefficien
ts[5,1])
slope_FWB_P_j1a <- (summary(model_full_int_jost1_all)$coeff
icients[2,1])+(summary(model_full_int_jost1_all)$coefficien
ts[6,1])
slope_SSB_P_j1a <- (summary(model_full_int_jost1_all)$coeff
icients[2,1])+(summary(model_full_int_jost1_all)$coefficien
ts[5,1])+(summary(model_full_int_jost1_all)$coefficients[6,
1])

## Import the table with 100 alpha diversity estimates per
sample
multi_alpha <- read.table("./AFSmicrobiome_SI_alphaDiversit
yMultiRaref_Rinput_DatasetS16.txt", header=T, sep= "\t", ro
w.names=1, na.strings=c("","NA"))

## Join the heterozygosity table (without the non-normalise
d alpha diversity estimates) and the multi estimate table
multi_alpha2 <- cbind(multi_alpha, SampleID = row.names(mul
ti_alpha))
multi_alpha2.tab <- dplyr::left_join(het_alpha.tab[,-5], mu
lti_alpha2, by = "SampleID")

## We want to plot the model results for each alpha diversi
ty estimate to get an idea about the level of uncertainty i
ntroduced through rarefying the OTU table (i.e. the robustn
ess of the observed correlation between alpha diversity and
 heterozygosity).
## First, an empty data frame is created that will be fille
d with the model outputs.
model_outs <- data.frame(matrix(nrow = 8, ncol = 100))
rownames(model_outs) <- c("intercept_FWB_M", "intercept_SSB
_M", "intercept_FWB_P", "intercept_SSB_P", "slope_FWB_M", "
slope_SSB_M", "slope_FWB_P", "slope_SSB_P")
```

```
colnames(model_outs) <- as.character(c(0:99))

## For all the alpha diversity estimates run the model and
fill the data frame with the model estimates.
for(i in 0:99){
  jost <- paste0("jost_",i)
  model1 <- paste0("lmer(sqrt(",jost,") ~ sHeteroz + Beach
+ Age + sHeteroz*Beach + sHeteroz*Age + (1|PairID2), data =
 multi_alpha2.tab)")
  model_full_int <- eval(parse(text=model1))

  model_outs[which(rownames(model_outs)=="intercept_FWB_M")
, which(colnames(model_outs)==i)] <- summary(model_full_int
)$coefficients[1,1]
  model_outs[which(rownames(model_outs)=="intercept_SSB_M")
, which(colnames(model_outs)==i)] <- (summary(model_full_in
t)$coefficients[1,1])+(summary(model_full_int)$coefficients
[3,1])
  model_outs[which(rownames(model_outs)=="intercept_FWB_P")
, which(colnames(model_outs)==i)] <- (summary(model_full_in
t)$coefficients[1,1])+(summary(model_full_int)$coefficients
[4,1])
  model_outs[which(rownames(model_outs)=="intercept_SSB_P")
, which(colnames(model_outs)==i)] <- (summary(model_full_in
t)$coefficients[1,1])+(summary(model_full_int)$coefficients
[3,1])+(summary(model_full_int)$coefficients[4,1])
  model_outs[which(rownames(model_outs)=="slope_FWB_M"), wh
ich(colnames(model_outs)==i)] <- summary(model_full_int)$co
efficients[2,1]
  model_outs[which(rownames(model_outs)=="slope_SSB_M"), wh
ich(colnames(model_outs)==i)] <- (summary(model_full_int)$c
oefficients[2,1])+(summary(model_full_int)$coefficients[5,1
])
  model_outs[which(rownames(model_outs)=="slope_FWB_P"), wh
ich(colnames(model_outs)==i)] <- (summary(model_full_int)$c
oefficients[2,1])+(summary(model_full_int)$coefficients[6,1
])
  model_outs[which(rownames(model_outs)=="slope_SSB_P"), wh
ich(colnames(model_outs)==i)] <- (summary(model_full_int)$c
oefficients[2,1])+(summary(model_full_int)$coefficients[5,1
])+(summary(model_full_int)$coefficients[6,1])
}

## First plot the results from the model with non-normalise
d alpha diversity estimates
hetplot <- ggplot() +
  geom_point(aes(x=het_alpha3.tab$sHeteroz[het_alpha3.tab$B
eachAge == "SSB P"], y=sqrt(het_alpha3.tab$jost1_all[het_al
pha3.tab$BeachAge == "SSB P"])),colour = "firebrick2",shape
=0, size = 2.5) +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (intercept_S
SB_P_j1a+slope_SSB_P_j1a*-0.25), yend = (intercept_SSB_P_j1
a+slope_SSB_P_j1a*0.18)), size = 1 ,linetype="dotdash", col
our="firebrick2") +
  geom_point(aes(x=het_alpha3.tab$sHeteroz[het_alpha3.tab$B
eachAge == "Freshwater P"], y=sqrt(het_alpha3.tab$jost1_all
[het_alpha3.tab$BeachAge == "Freshwater P"])),colour = "dod
gerblue3",shape=1, size = 2.5) +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (intercept_F
WB_P_j1a+slope_FWB_P_j1a*-0.25), yend = (intercept_FWB_P_j1
```

```
a+slope_FWB_P_j1a*0.18)), size = 1 ,linetype="dotdash", col
our="dodgerblue3") +
  geom_point(aes(x=het_alpha3.tab$sHeteroz[het_alpha3.tab$B
eachAge == "SSB M"], y=sqrt(het_alpha3.tab$jost1_all[het_al
pha3.tab$BeachAge == "SSB M"])),colour = "firebrick2",shape
=15, size = 2.5) +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (intercept_S
SB_M_j1a+slope_SSB_M_j1a*-0.25), yend = (intercept_SSB_M_j1
a+slope_SSB_M_j1a*0.18)), size = 1 , colour="firebrick2") +
  geom_point(aes(x=het_alpha3.tab$sHeteroz[het_alpha3.tab$B
eachAge == "Freshwater M"], y=sqrt(het_alpha3.tab$jost1_all
[het_alpha3.tab$BeachAge == "Freshwater M"])),colour = "dod
gerblue3", shape=19, size = 2.5) +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (intercept_F
WB_M_j1a+slope_FWB_M_j1a*-0.25), yend = (intercept_FWB_M_j1
a+slope_FWB_M_j1a*0.18)), size = 1 , colour="dodgerblue3")
+
  theme_bw(base_size = 12)+
  theme(panel.grid.major = element_blank(), panel.grid.mino
r = element_blank())+
  theme(axis.text.x = element_text(size=12), axis.title.x =
 element_text(size=14),axis.text.y = element_text(size=12),
axis.title.y = element_text(size=14),plot.margin = unit(c(.
5, .5, .5, .5), "cm"))+
  #scale_x_continuous(breaks=c(seq(from = -0.3, to = 0.25,
by = 0.1))) +
  xlab("Centred sMLH") +
  ylab("Effective no. of species (sqrt)")

## Add the results for the 100 alpha diversity estimates ca
lculated for the multiple rarefactions using thin grey line
s
for(i in 1:100){
hetplot <- hetplot +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (model_outs[
"intercept_SSB_P",i]+model_outs["slope_SSB_P",i]*-0.25), ye
nd = (model_outs["intercept_SSB_P",i]+model_outs["slope_SSB
_P",i]*0.18)), size = 0.6, linetype="dotdash", colour="ligh
tgrey") +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (model_outs[
"intercept_FWB_P",i]+model_outs["slope_FWB_P",i]*-0.25), ye
nd = (model_outs["intercept_FWB_P",i]+model_outs["slope_FWB
_P",i]*0.18)), size = 0.6, linetype="dotdash", colour="ligh
tgrey") +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (model_outs[
"intercept_SSB_M",i]+model_outs["slope_SSB_M",i]*-0.25), ye
nd = (model_outs["intercept_SSB_M",i]+model_outs["slope_SSB
_M",i]*0.18)), size = 0.6, colour="lightgrey") +
  geom_segment(aes(x = -0.25, xend = 0.18, y = (model_outs[
"intercept_FWB_M",i]+model_outs["slope_FWB_M",i]*-0.25), ye
nd = (model_outs["intercept_FWB_M",i]+model_outs["slope_FWB
_M",i]*0.18)), size = 0.6, colour="lightgrey")
}

## Show plot
hetplot
```

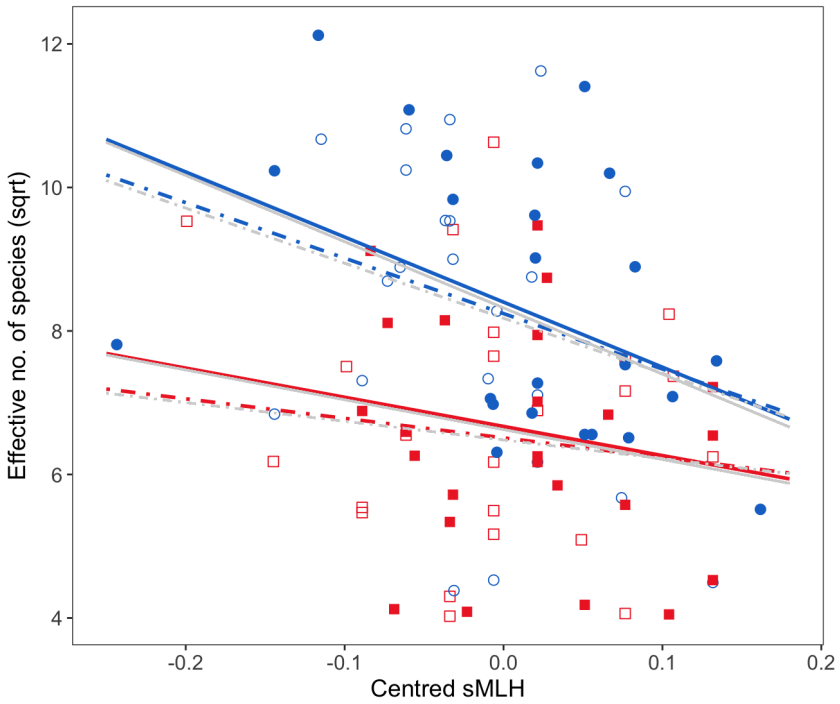


Figure 17. Relationship between bacterial alpha diversity (effective number of species, square root transformed) and individual heterozygosity (sMLH, centered around the mean). Plotted are the rawdata and regression lines from the LMM (heterozygosity regressed against alpha diversity, while controlling for breeding colony and age and including interaction terms beach x sMLH and age x sMLH). Light grey lines represent regression lines from 100 LMMs for which alpha diversity was calculated for 100 rarefied OTU tables. FWB mothers - blue filled circles and solid line, FWB pups - blue empty circles and dashed line, SSB mothers - red filled squares and solid line, SSB pups - red empty squares and dashed line.

We can see that the estimates derived from the 100 additional models are very similar to the original results, thus rarefying the OTU table has very little influence on the correlation between alpha diversity and heterozygosity. Excluding the two individuals with less than 10,000 reads (P24, P39) has a stronger effect on the estimates but does not change the overall results.

Identity disequilibrium g2

We computed the two-locus heterozygosity disequilibrium g2, which assesses the covariance of heterozygosity between markers and tells us something about the correlations between heterozygosity and inbreeding. We calculated g2 using the inbreedR package.

```
library(inbreedR)
library(grid)

## Calculate g2 from the genotype data
g2 <- inbreedR::g2_microsats(genos, nperm = 10000, nboot =
10000, CI = 0.95, verbose=FALSE)
g2
```

```
##
## Data: 95 observations at 50 markers
## Function call = inbreedR::g2_microsats(genotypes = genos
```

```
, nperm = 10000, nboot = 10000,     CI = 0.95, verbose = FA
LSE)
##
## g2 = 0.001142894, se = 0.001263793
##
## confidence interval
##         2.5%        97.5%
## -0.001145066  0.003806479
##
## p (g2 > 0) = 0.1195 (based on 10000 permutations)
```

```
## Estimate g2 from increasing number of randomly subsample
d loci
## Define the function
resample_loc_g2 <- function(genos, niter) {
  nloc <- ncol(genos)
  all_g2 <- matrix(data = NA, nrow = niter, ncol = nloc-1)

  for (i in 2:nloc) {
    for (k in 1:niter) {
      ind <- sample(1:50, i)
      gene_sub <- genos[ind]
      all_g2[k, i-1] <- g2_microsats(gene_sub)$g2
    }
  }
  all_g2
}
## Perform the resampling
resampling_g2 <- resample_loc_g2(genos, niter = 1000)
## Define a function to summerise the results
sum_results <- function(resampling_output) {
  mean_cor <- apply(resampling_output,2,mean, na.rm=T)
  sd_cor <- apply(resampling_output,2,sd, na.rm=T)
  se_cor <- sd_cor/(sqrt(nrow(resampling_output)))
  sum_results <- data.frame(locnum = 1:ncol(resampling_outp
ut),
                            cormean = mean_cor, corsd = sd_
cor, corse = se_cor)
}
## Perform the summerising of results
results_g2 <- sum_results(resampling_g2)

## Plot the results
ggplot2::ggplot(results_g2, aes(x = locnum, y = cormean)) +
        geom_line(size = 0.6, colour = "black") +
        geom_errorbar(aes(ymin = cormean-corsd, ymax = corm
ean+corsd),
                      width=0.8, alpha=0.7, size = 0.8, col
our = "black") +
        geom_point(size = 2, shape = 16) +
        theme_bw()+
        theme(panel.grid.major = element_blank(), panel.gri
d.minor = element_blank())+
        geom_hline(yintercept = 0,linetype = 2, colour="gra
y44")+
        theme(axis.title.x = element_text(vjust= -2 ,size =
 14), axis.title.y = element_text(vjust=3,size = 14), axis.
text.x = element_text(size = 12), axis.text.y = element_tex
t(size = 12)) +
```

```
        ylab("g2") +
        xlab("Number of loci") +
        labs(title = "g2 estimated from an increasing numbe
r of loci")
```

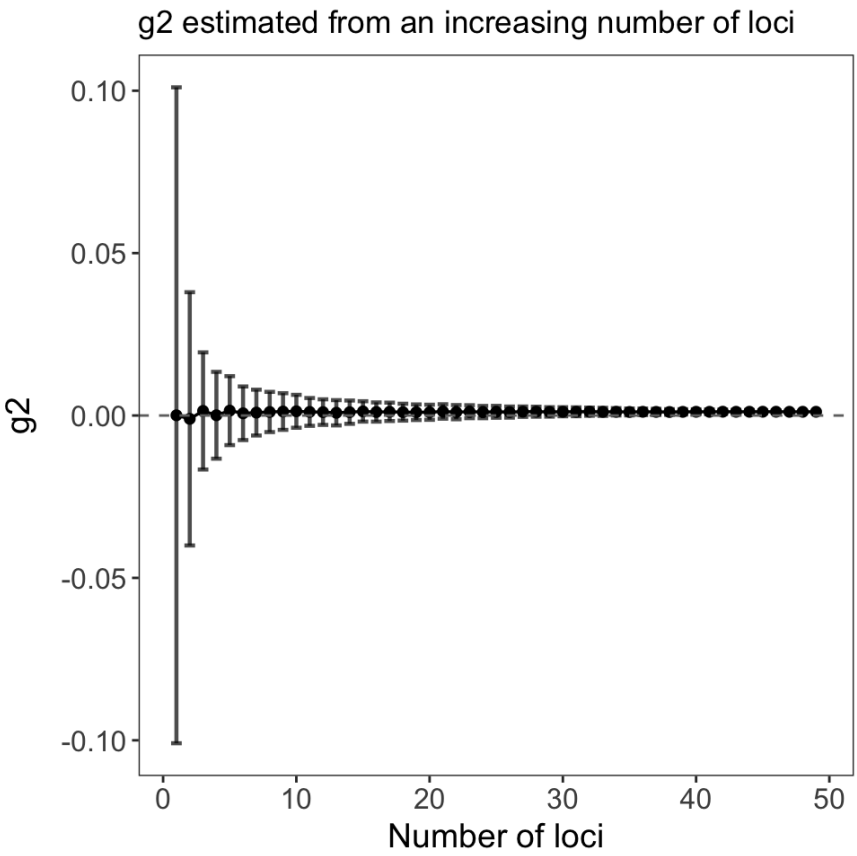


Figure 18. Estimation of the two-loci identity disequilibrium g2 from an increasing random subset of loci.

Locus specific effects (local effects)

We also tested for possible local effects following Szulkin et al. (2010). Using an F-ratio test we compare a model of alpha diversity containing multi locus heterozygosity (MLH – the sum of all single locus heterozygosities over all loci) with a model in which MLH was replaced by separate terms for the heterozygosity of each of the 50 microsatellite loci. Local effects can be identified if the second model explains significantly more variance than the first model. For missing genotypes we replaced specific heterozygosity values with the sample average.

```
## Calculate Heterozygozity and correlate with a-diversity

 library(inbreedR)
 library(dplyr)

## The header line needs to be present for this analysis (i
t was removed for the relatedness calculations).
 msats <- read.table("./AFSmicrobiome_SI_MicrosatelliteGeno
types50_P22removed_colnames_Rinput_DatasetS5.1.txt",header=
TRUE,row.names = 1, sep= "\t", na.strings=c("","NA"))
 is.na(msats) <- !msats
```

```
 ## Convert to inbreedR format
 genos <- inbreedR::convert_raw(msats)

 ## To restore column names for genos data frame use column
 names from msats data frame
 mnames <- colnames(msats)
 ## remove every second entry (in genos every marker has on
ly one column)
 mnames <- mnames[seq(1,length(mnames),2)]
 colnames(genos) <- mnames

 ## replace the missing values for each marker with the ave
rage for this marker (column average)
NA2mean <- function(x) replace(x, is.na(x), mean(x, na.rm =
 TRUE))
genosNoNA <- replace(genos, TRUE, lapply(genos, NA2mean))
## Calculate MLH as H = the sum of hi over L loci, hi = het
erozygosity at a single locus i (hi, coded as 0 or 1)
## and add to data frame
genosNoNA <- cbind(genosNoNA, MLH = rowSums(genosNoNA))
genosNoNA <- cbind(genosNoNA, Sample = rownames(genosNoNA))
## data frame containing alpha diversity estimates
alpha <- read.table("./AFSmicrobiome_SI_alphaDiversity_Rinp
ut_DatasetS12.txt", header = TRUE, sep = "\t")
## Only keep columns with sample names and jost1 estimates
for all individuals
alpha <- subset(alpha, select=c("Sample","jost1_all"))
## combine both data frames
genosNoNA2 <- left_join(genosNoNA,alpha)
row.names(genosNoNA2) <- genosNoNA2$Sample

## Model 1: Regress alpha diversity on MLH using a simple r
egression
m1 <- lm(jost1_all ~ MLH, data = genosNoNA2)
summary(m1)
```

```
##
## Call:
## lm(formula = jost1_all ~ MLH, data = genosNoNA2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.696 -21.902  -6.108  19.626  78.700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  161.194     41.438    3.89 0.000188 ***
## MLH           -2.827      1.140   -2.48 0.014942 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
##
## Residual standard error: 31.18 on 93 degrees of freedom
## Multiple R-squared:  0.06203,    Adjusted R-squared:  0.
05194
## F-statistic:  6.15 on 1 and 93 DF,  p-value: 0.01494
```

```
## Model 2: Regress alpha diversity on all single-locus het
erozygosities hi . . . hL, expressed as one or zero (L bein
```

```
g the number of loci), using a multiple regression
m2 <- lm(jost1_all ~ Pv9 + Hg6.3 + Hg8.10 + Hg1.3 + M11a +
PvcA + Zcwb07 + Agaz2 + Ag3 + Agaz6 + OrrFCB7 + Ag2 + OrrFC
B2 + Lw10 + Zcwc01 + Agaz5 + ZcwCgDhB.14 + SSL301 + Ag7 + A
gt10 + ZcwCgDh4.7 + Zcwe05 + Ag1 + OrrFCB8 + Agt47 + Zcwf07
 + ZcwD02 + ZcwCgDh1.8 + Aa4 + ZcCgDh5.8 + Agaz3 + X962.1 +
 X554.6 + Zcwa12 + PvcE + Zcwb09 + agaz10 + Mang44 + Mang36
 + Zcwe03 + Zcwe04 + X101.26 + X928.4b + X507.11 + Zcwa05 +
 Zcwe12 + ZcwCgDh3.6 + Hg6.1 + Zcwc11 + Lc28, data = genosN
oNA2)
summary(m2)
```

```
##
## Call:
## lm(formula = jost1_all ~ Pv9 + Hg6.3 + Hg8.10 + Hg1.3 +
M11a +
##     PvcA + Zcwb07 + Agaz2 + Ag3 + Agaz6 + OrrFCB7 + Ag2
+ OrrFCB2 +
##     Lw10 + Zcwc01 + Agaz5 + ZcwCgDhB.14 + SSL301 + Ag7 +
 Agt10 +
##     ZcwCgDh4.7 + Zcwe05 + Ag1 + OrrFCB8 + Agt47 + Zcwf07
 + ZcwD02 +
##     ZcwCgDh1.8 + Aa4 + ZcCgDh5.8 + Agaz3 + X962.1 + X554
.6 +
##     Zcwa12 + PvcE + Zcwb09 + agaz10 + Mang44 + Mang36 +
Zcwe03 +
##     Zcwe04 + X101.26 + X928.4b + X507.11 + Zcwa05 + Zcwe
12 +
##     ZcwCgDh3.6 + Hg6.1 + Zcwc11 + Lc28, data = genosNoNA
2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.503 -13.478  -0.988  14.669  72.130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 222.4175    65.1841   3.412  0.00139 **
## Pv9          -1.7427    11.1360  -0.156  0.87636
## Hg6.3        -6.8814    13.3745  -0.515  0.60947
## Hg8.10       -5.8589     9.0555  -0.647  0.52099
## Hg1.3       -26.0757    11.0725  -2.355  0.02305 *
## M11a         -1.6963    17.1643  -0.099  0.92172
## PvcA         -7.4104    12.2281  -0.606  0.54762
## Zcwb07       13.5809    13.7076   0.991  0.32722
## Agaz2       -15.8241    12.1362  -1.304  0.19906
## Ag3           8.9310     9.5723   0.933  0.35591
## Agaz6        -0.2008     9.9289  -0.020  0.98395
## OrrFCB7      -9.0776    13.0387  -0.696  0.48996
## Ag2           0.7049    11.9082   0.059  0.95307
## OrrFCB2      -4.7734    16.5496  -0.288  0.77437
## Lw10         -4.3254    15.3805  -0.281  0.77986
## Zcwc01       10.0046    14.3341   0.698  0.48887
## Agaz5        -7.3567     9.4066  -0.782  0.43836
## ZcwCgDhB.14   2.5961    12.1315   0.214  0.83154
## SSL301       -9.5548    15.2998  -0.625  0.53552
## Ag7           6.5474    11.0949   0.590  0.55812
## Agt10        -8.9138     9.9231  -0.898  0.37392
## ZcwCgDh4.7  -15.6473    12.8643  -1.216  0.23034
```

```
## Zcwe05       -9.4434    10.6076  -0.890  0.37817
## Ag1           4.0519    13.5991   0.298  0.76714
## OrrFCB8      -0.5235    10.0479  -0.052  0.95868
## Agt47       -14.9806    10.0315  -1.493  0.14248
## Zcwf07        1.9909    11.8394   0.168  0.86723
## ZcwD02       -5.9169    17.8801  -0.331  0.74228
## ZcwCgDh1.8    7.9141     9.8672   0.802  0.42683
## Aa4          -5.1714    10.4371  -0.495  0.62273
## ZcCgDh5.8   -30.7573    15.7172  -1.957  0.05672 .
## Agaz3        10.0092     8.7977   1.138  0.26140
## X962.1      -13.4975     9.3173  -1.449  0.15453
## X554.6        8.1116    12.7721   0.635  0.52865
## Zcwa12      -21.3043    13.9776  -1.524  0.13462
## PvcE         -2.0696    14.3056  -0.145  0.88563
## Zcwb09       -4.4766    12.3290  -0.363  0.71827
## agaz10       10.7827    10.7365   1.004  0.32073
## Mang44       -4.1525    10.3323  -0.402  0.68971
## Mang36      -17.6256    20.9464  -0.841  0.40464
## Zcwe03       -6.4964    12.0617  -0.539  0.59288
## Zcwe04       12.5628    12.6753   0.991  0.32705
## X101.26       2.1655    11.7022   0.185  0.85404
## X928.4b     -18.5645    12.6550  -1.467  0.14950
## X507.11       6.2648     9.5117   0.659  0.51356
## Zcwa05      -16.5090    15.1272  -1.091  0.28106
## Zcwe12        9.4646    12.9694   0.730  0.46940
## ZcwCgDh3.6  -20.4411    13.3432  -1.532  0.13269
## Hg6.1        -6.3384    14.0861  -0.450  0.65494
## Zcwc11       -8.5377    14.0219  -0.609  0.54574
## Lc28         -9.1522    13.7173  -0.667  0.50813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
 ' ' 1
##
## Residual standard error: 31.55 on 44 degrees of freedom
## Multiple R-squared:  0.5457, Adjusted R-squared:  0.0294
3
## F-statistic: 1.057 on 50 and 44 DF,  p-value: 0.4277
```

```
## Test whether the two models differ significantly from ea
ch other using an F-ratio test.
anova(m1,m2)
```

```
## Analysis of Variance Table
##
## Model 1: jost1_all ~ MLH
## Model 2: jost1_all ~ Pv9 + Hg6.3 + Hg8.10 + Hg1.3 + M11a
 + PvcA + Zcwb07 +
##     Agaz2 + Ag3 + Agaz6 + OrrFCB7 + Ag2 + OrrFCB2 + Lw10
 + Zcwc01 +
##     Agaz5 + ZcwCgDhB.14 + SSL301 + Ag7 + Agt10 + ZcwCgDh
4.7 +
##     Zcwe05 + Ag1 + OrrFCB8 + Agt47 + Zcwf07 + ZcwD02 + Z
cwCgDh1.8 +
##     Aa4 + ZcCgDh5.8 + Agaz3 + X962.1 + X554.6 + Zcwa12 +
 PvcE +
##     Zcwb09 + agaz10 + Mang44 + Mang36 + Zcwe03 + Zcwe04
 + X101.26 +
##     X928.4b + X507.11 + Zcwa05 + Zcwe12 + ZcwCgDh3.6 + H
g6.1 +
```

```
##     Zcwc11 + Lc28
##   Res.Df   RSS Df Sum of Sq     F Pr(>F)
## 1     93 90418
## 2     44 43794 49     46624 0.956 0.5627
```

```
# Res.Df   RSS Df Sum of Sq     F Pr(>F)
# 1     93 90418
# 2     44 43794 49     46624 0.956 0.5627
```

The second model does not explain more variance than the first model, thus we find no evidence for local effects.

PICRUst functional analysis

Lastly, we want to take a look at the potential functional capacity of the Antarctic fur seal skin microbial communities. We run PICRUSt analysis to obtain functional annotations for our 16S amplicon data. To evaluate the prediction accuracy of the PICRUSt results, first the nearest sequenced taxon index (NSTI) is calculated. The NSTI is defined as the sum of phylogenetic distances for each organism in the OTU table to its nearest relative with a sequenced reference genome (measured in substitutions per site and weighted by its frequency in the OTU table). NSTI values between 0.06-0.10 indicate that the PICRUSt predictions reasonably reflect the true functional profiles of the microbial community.

```
library(dplyr)

## Calculate average NSTI values overall and for each breed
ing colony to assess reliability of PICRUSt results
## Import NSTI table (output from PICRUSt)
nsti.tab <- read.table("./AFSmicrobiome_SI_NSTIvalues_filte
redTrimmed_rarefied_Rinput_DatasetS17.txt",header=TRUE, sep
= "\t", na.strings=c("","NA"))
colnames(nsti.tab)[which(colnames(nsti.tab) == 'Sample')] <
- 'SampleNames'
## Create a data frame with beach information for each samp
le which will be merged with the nsti table
beach.tab <- subset(meta_data.tab, select=c("Beach", "Age",
 "SampleNames"))
## Merge data frames
nsti.tab <- dplyr::left_join(nsti.tab, beach.tab, by = "Sam
pleNames")

## Caluclate overall NSTI
paste("Overall NSTI:",  round(mean(nsti.tab$Value),digits=3
),"+-",round(sd(nsti.tab$Value),digits=3),"sd")
```

```
## [1] "Overall NSTI: 0.075 +- 0.022 sd"
```

```
paste("Freshwater beach NSTI:",round(mean(nsti.tab[nsti.tab
$Beach=="Freshwater",]$Value),digits=3),"+-",round(sd(nsti.
tab[nsti.tab$Beach=="Freshwater",]$Value),digits=3),"sd")
```

```
## [1] "Freshwater beach NSTI: 0.083 +- 0.02 sd"
```

```
paste("Special study beach NSTI:", round(mean(nsti.tab[nsti
.tab$Beach=="SSB",]$Value),digits=3),"+-",round(sd(nsti.tab
[nsti.tab$Beach=="SSB",]$Value),digits=3),"sd")
```

```
## [1] "Special study beach NSTI: 0.068 +- 0.02 sd"
```

```
paste("Mother NSTI:", round(mean(nsti.tab[nsti.tab$Age=="M"
,]$Value),digits=3),"+-",round(sd(nsti.tab[nsti.tab$Age=="M
",]$Value),digits=3),"sd")
```

```
## [1] "Mother NSTI: 0.079 +- 0.02 sd"
```

```
paste("Pup NSTI:", round(mean(nsti.tab[nsti.tab$Age=="P",]$
Value),digits=3),"+-",round(sd(nsti.tab[nsti.tab$Age=="P",]
$Value),digits=3),"sd")
```

```
## [1] "Pup NSTI: 0.072 +- 0.022 sd"
```

After establishing that the functional predictions should be reliable for the Antarctic fur seal microbiome we can perform principal component analysis with the data similar to the analysis that can be done with the STAMP software.

```
library(dplyr)

## Import the table with functional predictions. The table
looks similar to the OTU table but instead for OTUs read co
unts are given for the different functional categories. Bef
ore importing the orginal output table (converted to .txt f
rom .biom) some manual adjustments were done. The first lin
e of the file as well as the "#" at the beginning of the se
cond line were deleted. Any "'" symbols from the category n
ames were removed. In the header line the last column name
"KEGG_Pathways" was replace by three column names (Level1,L
evel2,Level3). All ";" were replaced by "\t".
pi.tab <- read.table("./AFSmicrobiome_SI_Categorize_by_Func
tionL3_FilteredTrimmed_rarefied_Rinput_DatasetS18.txt", sep
 = "\t",row.names =1, header = TRUE)

## The table rownames correspond to KEGG categories at leve
l 3 (level 1-3 category names can be found in the last thre
e columns of the data frame).
## Remove rows with all 0 entries (Note: the last 3 columns
 contain the category names at levels 1-3)
pi.tab <- pi.tab[-which(rowSums(pi.tab[,1:96])==0),]
## Transpose the table so that the sample names become the
row names
piT.tab <- t(pi.tab[,1:96])
## Log-transform and add pseudocount as above for the beta
diversity analysis
pi_log.tab <- log(piT.tab+0.0001)
## Substract the log of the pseudocount
pi_log.tab <- pi_log.tab-(log(0.0001))
```

```
## Perform principal componant analysis (PCA). Centre and s
cale the data to zero mean and unit variance.
pi_pca <- prcomp(pi_log.tab, center = TRUE, scale. = TRUE)
## Look at the variance proportions of each PC
# summary(pi_pca)
## Extract the first 3 PCS and make a new data frame
pcsL3.tab <- as.data.frame(pi_pca$x[,1:3])
## Add a column with sample names to the data frame
pcsL3.tab["SampleID"] <- rownames(pcsL3.tab)
## Combine the data frame with the heterozygosity table fro
m above
pcs_metaL3.tab <- left_join(pcsL3.tab,het_alpha.tab, by="Sa
mpleID")
rownames(pcs_metaL3.tab) <- pcs_metaL3.tab$SampleID

## Repeat the PCA for level2 functional categories.
## Sum up all read counts at level 2 for each sample
pi_L2.tab<- aggregate(pi.tab[,1:96], by=list(Level2=pi.tab$
Level2), FUN=sum)
## Add rownames
row.names(pi_L2.tab) <- pi_L2.tab$Level2
## Remove the Level2 column (now rownames)
pi_L2.tab <- pi_L2.tab[,-(which(colnames(pi_L2.tab) == 'Lev
el2'))]
## All steps as before for level 3
pi_L2T.tab <- t(pi_L2.tab)
pi_L2T_log.tab <- log(pi_L2T.tab+0.0001)
pi_L2T_log.tab <- pi_L2T_log.tab-(log(0.0001))
piL2_pca <- prcomp(pi_L2T_log.tab, center = TRUE, scale. =
TRUE)
# summary(piL2_pca)
pcsL2.tab <- as.data.frame(piL2_pca$x[,1:3])
pcsL2.tab["SampleID"] <- rownames(pcsL2.tab)
pcs_metaL2.tab <- left_join(pcsL2.tab,het_alpha.tab, by="Sa
mpleID")
rownames(pcs_metaL2.tab) <- pcs_metaL2.tab$SampleID

## Repeat the PCA for level2 functional categories.
## Sum up all read counts at level 2 for each sample
pi_L1.tab<- aggregate(pi.tab[,1:96], by=list(Level1=pi.tab$
Level1), FUN=sum)
row.names(pi_L1.tab) <- pi_L1.tab$Level1
pi_L1.tab <- pi_L1.tab[,-(which(colnames(pi_L1.tab) == 'Lev
el1'))]
pi_L1T.tab <- t(pi_L1.tab)
pi_L1T_log.tab <- log(pi_L1T.tab+0.0001)
pi_L1T_log.tab <- pi_L1T_log.tab-(log(0.0001))
piL1_pca <- prcomp(pi_L1T_log.tab, center = TRUE, scale. =
TRUE)
# summary(piL1_pca)
pcsL1.tab <- as.data.frame(piL1_pca$x[,1:3])
pcsL1.tab["SampleID"] <- rownames(pcsL1.tab)
pcs_metaL1.tab <- left_join(pcsL1.tab,het_alpha.tab, by="Sa
mpleID")
rownames(pcs_metaL1.tab) <- pcs_metaL1.tab$SampleID
```

```
library(ggplot2)
library(gridExtra)
```

```
## Plot the PCA results for level 3
L3_PC12 <- ggplot(pcs_metaL3.tab, aes(x=PC1, y=PC2))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position=c(0.16,0.16),legend.t
itle = element_blank(),legend.background = element_rect(siz
e=0.3,linetype="solid", colour ="black"))+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
                labs(x = "PC1 (61.3% explained variability)
", y = "PC2 (13.7% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

L3_PC13<-ggplot(pcs_metaL3.tab, aes(x=PC1, y=PC3))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position="none")+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
                labs(x = "PC1 (61.3% explained variability)
", y = "PC3 (5.6% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

grid.arrange(L3_PC12, L3_PC13, ncol=2)
```

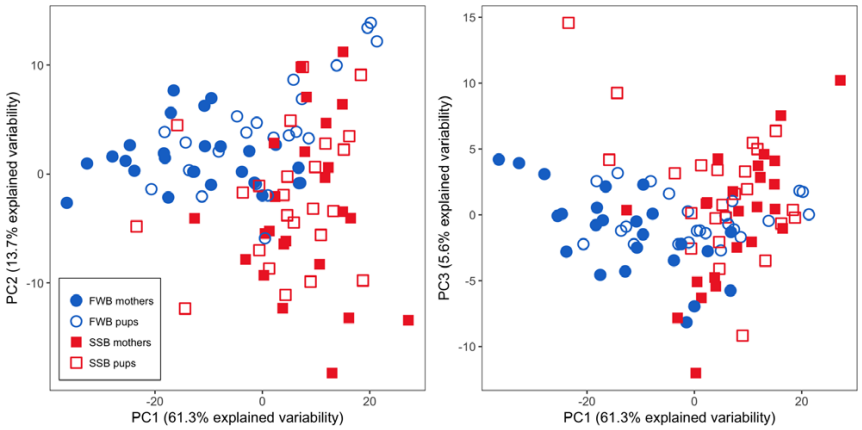


Figure 19. Principal component analysis of PICRUSt functional predictions at hierachical level 3.

```
library(ggplot2)
library(gridExtra)

## Plot the PCA results for level 2
L2_PC12<-ggplot(pcs_metaL2.tab, aes(x=PC1, y=PC2))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position="none")+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
                labs(x = "PC1 (78.0% explained variability)
", y = "PC2 (12.0% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

L2_PC13<-ggplot(pcs_metaL2.tab, aes(x=PC1, y=PC3))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position=c(0.84,0.84),legend.t
itle = element_blank(),legend.background = element_rect(siz
e=0.3,linetype="solid", colour ="black"))+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
```

```
                labs(x = "PC1 (78.0% explained variability)
", y = "PC3 (3.5% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

grid.arrange(L2_PC12, L2_PC13, ncol=2)
```

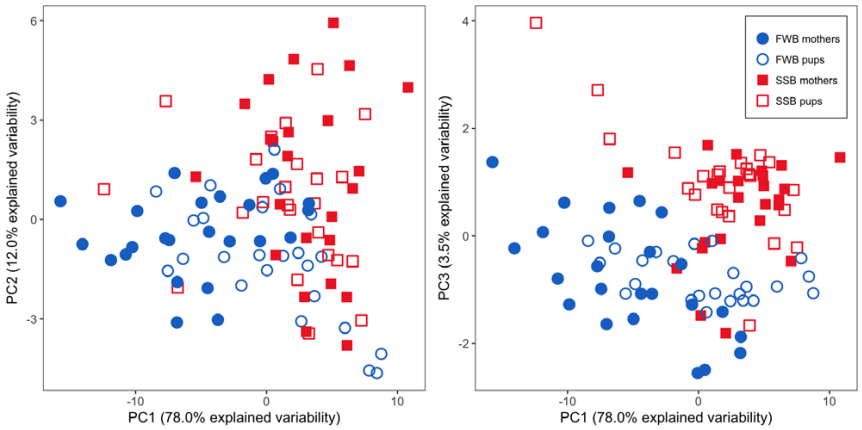


Figure 20. Principal component analysis of PICRUSt functional predictions at hierachical level 2.

```
library(ggplot2)
library(gridExtra)

## Plot the PCA results for level 1
L1_PC12<-ggplot(pcs_metaL1.tab, aes(x=PC1, y=PC2))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position="none")+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
                labs(x = "PC1 (92.1% explained variability)
", y = "PC2 (5.5% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

L1_PC13<-ggplot(pcs_metaL1.tab, aes(x=PC1, y=PC3))+
                geom_point(size=3.5, stroke=1, aes(colour=B
eachAge, shape=BeachAge))+
                scale_color_manual(values = c("dodgerblue3"
,"dodgerblue3","firebrick2","firebrick2"),name="",breaks=c(
"Freshwater M", "Freshwater P", "SSB M", "SSB P"), labels=c
("FWB mothers","FWB pups", "SSB mothers", "SSB pups"))+
                scale_shape_manual(values = c(19,1,15,0), n
ame="", breaks=c("Freshwater M", "Freshwater P", "SSB M", "
```

```
SSB P"), labels=c("FWB mothers","FWB pups", "SSB mothers",
"SSB pups"))+
                theme_bw()+
                theme(legend.position=c(0.16,0.16),legend.t
itle = element_blank(),legend.background = element_rect(siz
e=0.3,linetype="solid", colour ="black"))+
                theme(panel.grid.major = element_blank(), p
anel.grid.minor = element_blank())+
                labs(x = "PC1 (92.1% explained variability)
", y = "PC3 (1.0% explained variability)")+
                theme(axis.title.y=element_text(size=12), a
xis.title.x  = element_text(size=12), axis.text.x=element_t
ext(size=10), axis.text.y  = element_text(size=10))

grid.arrange(L1_PC12, L1_PC13, ncol=2)
```

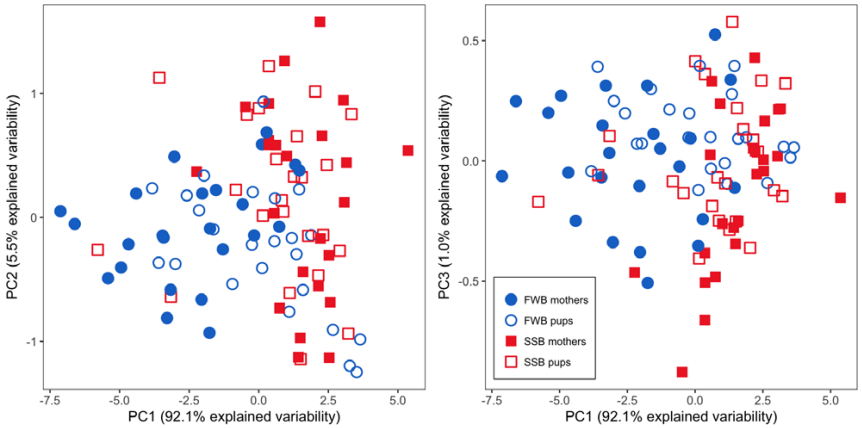


Figure 21. Principal component analysis of PICRUSt functional predictions at hierachical level 1.

References

Stoffel MA, Caspers BA, Forcada J, et al. (2015) Chemical fingerprints encode mother-offspring similarity, colony membership, relatedness, and genetic quality in fur seals. *Proceedings of the National Academy of Sciences U S A* **112**, E5005-5012.

Szulkin M, Bierne N, David P (2010) Heterozygosity-fitness correlations: a time for reappraisal. *Evolution* **64**, 1202-1217.