R_scripts_AMRIWA_metagenomes

Melina Markkanen

27/1/2022

Contents

Set working directory

```
setwd("~/Desktop/Git/AMRIWA/RFiles")
```

Load required libraries

```
library(phyloseq)
library(stringr)
library(vegan)
library(RColorBrewer)
library(ggplot2)
library(knitr)
library(ggpubr)
library(pheatmap)
library(MASS)
library(gplots)
library(grid)
library(cowplot)
library(DESeq2)
library(multcomp)
library(ggrepel)
library(ggcorrplot)
library(dplyr)
library(VennDiagram)
library(psych)
library(usefun)
library(patchwork)
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
library(ggspatial)
library(rgeos)
library(maps)
library(Hmisc)
```

Load data

metadata

```
metadata <- read.table("metadata.txt", sep = "\t", header = T, row.names = 1, fill = 1,
    dec = ".", na.strings = "NA")

metadata$DNA_ng_µl <- as.numeric(gsub(",", ".", gsub("\\.", "", metadata$DNA_ng_µl)))
metadata$A260_280 <- as.numeric(gsub(",", ".", gsub("\\.", "", metadata$A260_280)))
metadata$M_Seqs_trimmed <- as.numeric(gsub(",", ".", gsub("\\.", "", metadata$M_Seqs_trimmed)))
metadata$lat <- as.numeric(gsub(",", ".", gsub("\\.", "", metadata$lat)))
metadata$long <- as.numeric(gsub(",", ".", gsub("\\.", "", metadata$long)))
```

Metaxa2 results

```
metaxa_genus <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/metaxa_genus.txt")

# Create OTU table
OTU_metaxa <- metaxa_genus[, -1]
# Match sample ID order with metadata file
match <- match(rownames(metadata), colnames(OTU_metaxa))
OTU_metaxa <- OTU_metaxa[, match]
all(colnames(OTU_metaxa) == rownames(metadata))
# Create tax table
tax_table_metaxa <- data.frame(str_split_fixed(data.frame(metaxa_genus)[, 1], ";",
    6))
colnames(tax_table_metaxa) <- c("Domain", "Phylum", "Class", "Order", "Family", "Genus")
# Check if samples are in order
identical(rownames(metadata), colnames(OTU_metaxa))
# Combine into phyloseq object
metaxa_PHY <- phyloseq(otu_table(OTU_metaxa, taxa_are_rows = TRUE), tax_table(as.matrix(tax_table_metaxa)),
    sample_data(metadata))

# Exclude taxa 'Unknown', 'Unclassified', 'Eukaryota', 'Mitochondria',
# 'Archaea', 'Chloroplast'
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Unknown"))
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Unclassified"))
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Eukaryota"))
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Mitochondria"))
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Archaea"))
metaxa_PHY <- subset_taxa(metaxa_PHY, !Domain %in% c("Chloroplast"))

# Add SSU counts to metadata
metadata$SSU_counts <- sample_sums(metaxa_PHY)

## Exclude biological / technical replicates
metaxa_PHY <- subset_samples(metaxa_PHY, alias != "BH31" & alias != "BH33" & alias !=
    "BH34B" & alias != "BH10" & alias != "BFH38B" & alias != "FH8" & alias != "BH45" &
    alias != "BH59" & alias != "BH62")

# Create phyloseq object with only HWW samples
metaxa_PHY_stat <- subset_samples(metaxa_PHY, category == "WA hospital effluent" |
    category == "North Eu hospital effluent")

# Create phyloseq objects (x 3) with equal group sizes for the statistical
# testing
alias = data.frame(metaxa_PHY_stat@sam_data[["alias"]])
colnames(alias) = "sample"
BH <- data.frame(alias[grepl("BH.", alias$sample), ])
colnames(BH) <- c("sample")
## Include 8 random samples per country
```

```
random_BH_1 <- sample_n(BH, 8)
random_BH_3 <- sample_n(BH, 8)
random_BH_3 <- sample_n(BH, 8)

# Create phyloseq objects (x 3) with equal group sizes for the statistical
# testing
alias = data.frame(metaxa_PHY_stat@sam_data[["alias"]])
colnames(alias) = "sample"
BFH <- data.frame(alias[grepl("BFH.", alias$sample), ])
colnames(BFH) <- c("sample")
## Include 8 random samples per country
colnames(BFH) <- c("sample")
random_BFH_1 <- sample_n(BFH, 8)
random_BFH_3 <- sample_n(BFH, 8)
random_BFH_3 <- sample_n(BFH, 8)

# Sample set 1
metaxa_PHY_stat_equal1 <- subset_samples(metaxa_PHY, alias == paste(random_BFH_1$sample[1]) |
    alias == paste(random_BFH_1$sample[2]) | alias == paste(random_BFH_1$sample[3]) |
    alias == paste(random_BFH_1$sample[4]) | alias == paste(random_BFH_1$sample[5]) |
    alias == paste(random_BFH_1$sample[6]) | alias == paste(random_BFH_1$sample[7]) |
    alias == paste(random_BFH_1$sample[8]) | alias == paste(random_BH_1$sample[1]) |
    alias == paste(random_BH_1$sample[2]) | alias == paste(random_BH_1$sample[3]) |
    alias == paste(random_BH_1$sample[4]) | alias == paste(random_BH_1$sample[5]) |
    alias == paste(random_BH_1$sample[6]) | alias == paste(random_BH_1$sample[7]) |
    alias == paste(random_BH_1$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 2
metaxa_PHY_stat_equal2 <- subset_samples(metaxa_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 3
metaxa_PHY_stat_equal3 <- subset_samples(metaxa_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")
```

rpoB

```
HMM_RESULT_TABLE <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/HMM_RESULT_TABLE.txt",
    row.names = 1)
HMM_RESULT_TABLE$SUM = rowSums(HMM_RESULT_TABLE[, c(2, 3)])

# Sum of counts for R1 and R1 reads Reorder samples to match metadata and add
```

```
# to metadata
match <- match(rownames(metadata), rownames(HMM_RESULT_TABLE))
rpoB_counts <- HMM_RESULT_TABLE[match, ]
metadata$rpoB_counts <- rpoB_counts$SUM

# Only R1 reads Reorder samples to match metadata
match <- match(rownames(metadata), rownames(HMM_RESULT_TABLE))
R1_rpoB_counts <- HMM_RESULT_TABLE[match, ]
metadata$R1_rpoB_counts <- rpoB_counts$R1
```

Metaphlan3 results

```
OTU_metaphlan <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/mod_merged_abundance_table_species.txt",
    header = T)

# Match sample order
tax_table_metaphlan <- read.table("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/tax_table_metaphlan",
    quote = "\"", comment.char = "")
identical(tax_table_metaphlan$V1, OTU_metaphlan$clade_name)

tax_table_metaphlan <- read.csv("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/tax_table_metaphlan",
    header = FALSE, sep = ";")
colnames(tax_table_metaphlan) <- c("Kingdom", "Phylum", "Class", "Order", "Family",
    "Genus", "Species")
# Remove '__'
tax_table_metaphlan <- apply(tax_table_metaphlan, 2, function(y) (gsub(".__", "",
    y)))

match <- match(rownames(metadata), colnames(OTU_metaphlan))
OTU_metaphlan <- OTU_metaphlan[, match]
all(rownames(metadata) == colnames(OTU_metaphlan))

# Combine into phyloseq object
metaphlan_PHY <- phyloseq(otu_table(OTU_metaphlan, taxa_are_rows = TRUE), tax_table(as.matrix(tax_table_metaphlan)),
    sample_data(metadata))
# Check that sums are ~100 sample_sums(metaphlan_PHY)

# Exclude Viruses, Eukaryota & Archaea
metaphlan_PHY <- subset_taxa(metaphlan_PHY, Kingdom != "Viruses" & Kingdom != "Eukaryota" &
    Kingdom != "Archaea")

## Exclude biological / technical replicates
metaphlan_PHY <- subset_samples(metaphlan_PHY, alias != "BH31" & alias != "BH33" &
    alias != "BH34B" & alias != "BH10" & alias != "BFH38B" & alias != "FH8" & alias !=
    "BH45" & alias != "BH59" & alias != "BH62")

# Create phyloseq object with only HWW samples
metaphlan_PHY_stat <- subset_samples(metaphlan_PHY, category == "WA hospital effluent" |
    category == "North Eu hospital effluent")

# Create phyloseq objects (x 3) with equal group sizes for the statistical
# testing Sample set 1
metaphlan_PHY_stat_equal1 <- subset_samples(metaphlan_PHY, alias == paste(random_BFH_1$sample[1]) |
    alias == paste(random_BFH_1$sample[2]) | alias == paste(random_BFH_1$sample[3]) |
    alias == paste(random_BFH_1$sample[4]) | alias == paste(random_BFH_1$sample[5]) |
    alias == paste(random_BFH_1$sample[6]) | alias == paste(random_BFH_1$sample[7]) |
    alias == paste(random_BFH_1$sample[8]) | alias == paste(random_BH_1$sample[1]) |
    alias == paste(random_BH_1$sample[2]) | alias == paste(random_BH_1$sample[3]) |
    alias == paste(random_BH_1$sample[4]) | alias == paste(random_BH_1$sample[5]) |
    alias == paste(random_BH_1$sample[6]) | alias == paste(random_BH_1$sample[7]) |
    alias == paste(random_BH_1$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
```

```
    "FH9")

# Sample set 2
metaphlan_PHY_stat_equal2 <- subset_samples(metaphlan_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 3
metaphlan_PHY_stat_equal3 <- subset_samples(metaphlan_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")
```

ResFinder results

```
OTU_resfinder <- as.matrix(read.table("ARG_genemat.txt", header = T, check.names = F,
    row.names = 1))

# Reorder to match metadata
match <- match(rownames(metadata), colnames(OTU_resfinder))
OTU_resfinder <- OTU_resfinder[, match]
all(colnames(OTU_resfinder) == rownames(metadata))

# Tax_table
clusters_tax_table_resfinder <- read.csv("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/clusters_tax_table.txt",
    header = FALSE, sep = ";")
colnames(clusters_tax_table_resfinder) <- c("Gene", "Cluster_name", "Class")
# Reorder columns
col_order <- c("Class", "Cluster_name", "Gene")
clusters_tax_table_resfinder <- clusters_tax_table_resfinder[, col_order]

# Reorder tax_table to match
match <- match(rownames(OTU_resfinder), clusters_tax_table_resfinder$Gene)
clusters_tax_table_resfinder <- clusters_tax_table_resfinder[match, ]
all(rownames(OTU_resfinder) == clusters_tax_table_resfinder$Gene)

# Divide by ARG gene lengths
resfinder_lengths <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/resfinder_lengths.txt",
    header = FALSE, comment.char = "#")
all(rownames(clusters_tax_table_resfinder$Gene) == resfinder_lengths$V1)
OTU_resfinder_length_norm <- OTU_resfinder/resfinder_lengths[, 2]

# Normalization with Metaxa2 SSU counts
OTU_resfinder_length_SSU_norm <- t(t(OTU_resfinder_length_norm)/metadata$SSU_counts) *
    1540
all(rownames(metadata) == colnames(OTU_resfinder_length_SSU_norm))
identical((OTU_resfinder_length_norm[3, 5]/metadata$SSU_counts[5]) * 1540, OTU_resfinder_length_SSU_norm[3,
```

```
    5])
all(rownames(OTU_resfinder_length_norm) == clusters_tax_table_resfinder$Gene)

# Hide rownames
dim(OTU_resfinder_length_SSU_norm)
rownames(OTU_resfinder_length_SSU_norm) <- c(1:3104)

dim(clusters_tax_table_resfinder)
rownames(clusters_tax_table_resfinder) <- c(1:3104)

# Combine to phyloseq object
resfinder_PHY <- phyloseq(otu_table(OTU_resfinder_length_SSU_norm, taxa_are_rows = TRUE),
    sample_data(metadata), tax_table(as.matrix(clusters_tax_table_resfinder)))

## Exclude biological / technical replicates
resfinder_PHY <- subset_samples(resfinder_PHY, alias != "BH31" & alias != "BH33" &
    alias != "BH34B" & alias != "BH10" & alias != "BFH38B" & alias != "FH8" & alias !=
    "BH45" & alias != "BH59" & alias != "BH62")

# Create phyloseq object with only hospital WW samples sequenced here
resfinder_PHY_stat <- subset_samples(resfinder_PHY, category == "WA hospital effluent" |
    category == "North Eu hospital effluent")

# Create phyloseq objects (x 3) with equal group sizes for the statistical
# testing Sample set 1
resfinder_PHY_stat_equal1 <- subset_samples(resfinder_PHY, alias == paste(random_BFH_1$sample[1]) |
    alias == paste(random_BFH_1$sample[2]) | alias == paste(random_BFH_1$sample[3]) |
    alias == paste(random_BFH_1$sample[4]) | alias == paste(random_BFH_1$sample[5]) |
    alias == paste(random_BFH_1$sample[6]) | alias == paste(random_BFH_1$sample[7]) |
    alias == paste(random_BFH_1$sample[8]) | alias == paste(random_BH_1$sample[1]) |
    alias == paste(random_BH_1$sample[2]) | alias == paste(random_BH_1$sample[3]) |
    alias == paste(random_BH_1$sample[4]) | alias == paste(random_BH_1$sample[5]) |
    alias == paste(random_BH_1$sample[6]) | alias == paste(random_BH_1$sample[7]) |
    alias == paste(random_BH_1$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 2
resfinder_PHY_stat_equal2 <- subset_samples(resfinder_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 3
resfinder_PHY_stat_equal3 <- subset_samples(resfinder_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")
```

MGE results

```
OTU_MGE <- as.matrix(read.table("cp_MGE_genemat.txt", header = T, check.names = F,
    row.names = 1))

# Reorder to match metadata
match <- match(rownames(metadata), colnames(OTU_MGE))
OTU_MGE <- OTU_MGE[, match]
all(colnames(OTU_MGE) == rownames(metadata))

# Tax table
MGE_tax_table_trim <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/MGE_tax_table_trim.txt",
    header = FALSE)
colnames(MGE_tax_table_trim) <- c("Gene", "Element", "Class")

# Reorder tax_table to match
match <- match(rownames(OTU_MGE), MGE_tax_table_trim$Gene)
MGE_tax_table_trim <- MGE_tax_table_trim[match, ]
all(rownames(OTU_MGE) == MGE_tax_table_trim$Gene)

# Normalization to MGE lengths
MGE_lengths <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/MGE_lengths.txt",
    header = FALSE, comment.char = "#", check.names = F)
match <- match(rownames(OTU_MGE), MGE_lengths$V1)
MGE_lengths <- MGE_lengths[match, ]
all(rownames(MGE_tax_table_trim$Gene) == MGE_lengths$V1)
OTU_MGE_length_norm <- OTU_MGE/MGE_lengths[, 2]

# Normalization with Metaxa2 SSU counts
OTU_MGE_length_SSU_norm <- t(t(OTU_MGE_length_norm)/metadata$SSU_counts) * 1540
all(rownames(metadata) == colnames(OTU_MGE_length_SSU_norm))
all(rownames(OTU_MGE_length_SSU_norm) == MGE_tax_table_trim$Gene)

# Hide rownames
dim(OTU_MGE_length_SSU_norm)
rownames(OTU_MGE_length_SSU_norm) <- c(1:2709)

dim(MGE_tax_table_trim)
rownames(MGE_tax_table_trim) <- c(1:2709)

# Combine to phyloseq object
MGE_PHY <- phyloseq(otu_table(OTU_MGE_length_SSU_norm, taxa_are_rows = TRUE), sample_data(metadata),
    tax_table(as.matrix(MGE_tax_table_trim)))

## Exclude biological / technical replicates
MGE_PHY <- subset_samples(MGE_PHY, alias != "BH31" & alias != "BH33" & alias != "BH34B" &
    alias != "BH10" & alias != "BFH38B" & alias != "FH8" & alias != "BH45" & alias !=
    "BH59" & alias != "BH62")

# Create phyloseq object with only hospital WW samples sequenced here
MGE_PHY_stat <- subset_samples(MGE_PHY, category == "WA hospital effluent" | category ==
    "North Eu hospital effluent")

# Create phyloseq object with equal group for the statistical analysis Sample
# set 1
MGE_PHY_stat_equal1 <- subset_samples(MGE_PHY, alias == paste(random_BFH_1$sample[1]) |
    alias == paste(random_BFH_1$sample[2]) | alias == paste(random_BFH_1$sample[3]) |
    alias == paste(random_BFH_1$sample[4]) | alias == paste(random_BFH_1$sample[5]) |
    alias == paste(random_BFH_1$sample[6]) | alias == paste(random_BFH_1$sample[7]) |
    alias == paste(random_BFH_1$sample[8]) | alias == paste(random_BH_1$sample[1]) |
    alias == paste(random_BH_1$sample[2]) | alias == paste(random_BH_1$sample[3]) |
    alias == paste(random_BH_1$sample[4]) | alias == paste(random_BH_1$sample[5]) |
    alias == paste(random_BH_1$sample[6]) | alias == paste(random_BH_1$sample[7]) |
    alias == paste(random_BH_1$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
```

```
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 2
MGE_PHY_stat_equal2 <- subset_samples(MGE_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Sample set 3
MGE_PHY_stat_equal3 <- subset_samples(MGE_PHY, alias == paste(random_BFH_3$sample[1]) |
    alias == paste(random_BFH_3$sample[2]) | alias == paste(random_BFH_3$sample[3]) |
    alias == paste(random_BFH_3$sample[4]) | alias == paste(random_BFH_3$sample[5]) |
    alias == paste(random_BFH_3$sample[6]) | alias == paste(random_BFH_3$sample[7]) |
    alias == paste(random_BFH_3$sample[8]) | alias == paste(random_BH_3$sample[1]) |
    alias == paste(random_BH_3$sample[2]) | alias == paste(random_BH_3$sample[3]) |
    alias == paste(random_BH_3$sample[4]) | alias == paste(random_BH_3$sample[5]) |
    alias == paste(random_BH_3$sample[6]) | alias == paste(random_BH_3$sample[7]) |
    alias == paste(random_BH_3$sample[8]) | alias == "FH1" | alias == "FH2" | alias ==
    "FH3" | alias == "FH4" | alias == "FH5" | alias == "FH6" | alias == "FH7" | alias ==
    "FH9")

# Get class 1 integrons
MGE_PHY_int <- tax_glom(MGE_PHY, taxrank = "Class")
MGE_PHY_int <- subset_taxa(MGE_PHY_int, Class == "intI1")

MGE_PHY_int_stat <- tax_glom(MGE_PHY_stat, taxrank = "Class")
MGE_PHY_int_stat <- subset_taxa(MGE_PHY_int_stat, Class == "intI1")

MGE_PHY_qac_stat <- tax_glom(MGE_PHY_stat, taxrank = "Class")
MGE_PHY_qac_stat <- subset_taxa(MGE_PHY_qac_stat, Class == "qacEdelta")
```

Correlation between SSU & rpoB counts

```
SSU_counts <- data.frame(sample_data(resfinder_PHY_stat)$SSU_counts)
R1_rpoB_counts <- data.frame(sample_data(resfinder_PHY_stat)$R1_rpoB_counts)
bacterial_counts <- cbind(SSU_counts, R1_rpoB_counts)
colnames(bacterial_counts) <- c("SSU_counts", "R1_rpoB_counts")

p <- ggplot(bacterial_counts, aes(x = SSU_counts, y = R1_rpoB_counts)) + geom_point(size = 7,
    shape = 19, color = "#3110D2") + geom_smooth(method = "lm", se = TRUE, fullrange = FALSE,
    level = 0.95, color = "#FB2A38", fill = "#8A91F8") + theme_bw() + theme(axis.title = element_text(size = 30,
    family = "Times"), axis.text = element_text(size = 32, family = "Times"), plot.title = element_text(size = 36,
    family = "Times"), plot.subtitle = element_text(size = 28, family = "Times")) +
    xlab("16s rRNA counts") + ylab("R1 rpoB counts") + labs(title = "Correlation of 16s rRNA and rpoB counts",
    subtitle = "Hospital WWs in Benin (25), BF (34) and Finland (8)")
cor <- p + stat_cor(method = "pearson", label.x = 1e+05, label.y = 1000, )

correl <- corr.test(SSU_counts, R1_rpoB_counts, use = "pairwise", method = "pearson",
    adjust = "fdr", alpha = 0.05, ci = TRUE)

r <- data.frame(correl$r)
p <- data.frame(correl$p)
p.ad <- data.frame(correl$p.adj)
```

```
# ggsave(filename = 'SSU_rpoB_cor_new.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)
```

Modelling ARG abundance

Gather data into data frame

```
df <- data.frame(ARG_SUM = sample_sums(resfinder_PHY_stat), intI1_SUM = sample_sums(MGE_PHY_int_stat),
    MGE_SUM = sample_sums(MGE_PHY_stat), hospital_section = as.factor(sample_data(resfinder_PHY_stat)$hospital_secti
    SSU_counts = as.factor(sample_data(resfinder_PHY_stat)$SSU_counts), rpoB_counts = as.factor(sample_data(resfinde
    hospital = as.factor(sample_data(resfinder_PHY_stat)$hospital), country = as.factor(sample_data(resfinder_PHY_st
    no_of_beds = as.factor(sample_data(resfinder_PHY_stat)$no_of_beds), long = as.factor(sample_data(resfinder_PHY_s
    lat = as.factor(sample_data(resfinder_PHY_stat)$lat), A260_280 = as.numeric(sample_data(resfinder_PHY_stat)$A260
    DNA_ng_µl = as.numeric(sample_data(resfinder_PHY_stat)$DNA_ng_µl), M_Seqs_trimmed = as.numeric(sample_data(resfi

df$SSU_counts <- as.character(df$SSU_counts)
df$SSU_counts <- as.numeric(df$SSU_counts)
df$rpoB_counts <- as.character(df$rpoB_counts)
df$rpoB_counts <- as.numeric(df$rpoB_counts)
df$no_of_beds <- as.character(df$no_of_beds)
df$no_of_beds <- as.numeric(df$no_of_beds)
```

Draw maps

```
# Plot maps for sample sites in Benin and Burkina Faso
world <- ne_countries(scale = "medium", returnclass = "sf")
class(world)

gps0 <- metadata[!duplicated(metadata[, c("lat", "long")]), ]
gps0 <- gps0[, c("country", "lat", "long", "hospital")]
gps0 <- subset(gps0, country == "Benin" | country == "Burkina Faso")
gps <- data.frame("Burkina Faso", "12.500000", "-1.666670", "H")
rownames(gps) <- "BFH13_S131"
colnames(gps) <- c("country", "lat", "long", "hospital")
gps <- rbind(gps0, gps)

# Add important cities
gps_labels <- data.frame(country = c("country_name", "Benin", "Benin", "country_name",
    "Burkina Faso", "Burkina Faso", "ocean"), lat = c("10.544904033009432", "6.3676953",
    "9.3400159", "13.740788326149952", "12.3681873", "11.1757783", "4.944956754100344"),
    long = c("2.3165032566686428", "2.4252507", "2.6278258", "-1.0794179365270806",
        "-1.5270944", "-4.2957591", "2.376996878456601"), hospital = c("nd", "nd",
        "nd", "nd", "nd", "nd", "nd"))

rownames(gps_labels) <- c("Benin", "Cotonou", "Parakou", "Burkina Faso", "Ouagadougou",
    "Bobo Dioulasso", "Gulf of Guinea")

gps_data <- rbind(gps, gps_labels)
gps_data$Label <- c("nd", "nd", "nd", "nd", "nd", "nd", "nd", "nd", "nd", "nd", "Benin",
    "Cotonou", "Parakou", "Burkina Faso", "Ouagadougou", "Bobo Dioulasso", "Gulf of Guinea")
gps_data$lat <- as.numeric(gps_data$lat)
gps_data$long <- as.numeric(gps_data$long)

# Add sampling sites
p_map1 <- ggplot(data = world) + geom_sf() + borders("world", colour = "black", fill = "wheat1") +
    theme(panel.background = element_rect(fill = "azure1", colour = "azure1")) +
    geom_point(data = subset(gps_data, Label == "nd"), aes(x = long, y = lat), size = 4,
        shape = 16, color = "#B2182B") + geom_text_repel(data = subset(gps_data,
    Label == "nd"), mapping = aes(x = long, y = lat, label = hospital, family = "Times"),
```

```
    size = 11, point.padding = 1e-06) + coord_sf(ylim = c(4.5, 14.75), xlim = c(-6,
    3.95), expand = T) + theme(axis.text = element_text(family = "Times", size = 16),
    axis.title = element_blank()) + annotation_scale(location = "bl", width_hint = 0.2,
    height = unit(0.3, "cm"))

# Add countries
p_map2 <- p_map1 + geom_point(data = subset(gps_data, Label == "Benin" | Label ==
    "Burkina Faso" | Label == "Gulf of Guinea"), aes(x = long, y = lat), size = 0,
    shape = 16, color = "black") + geom_text_repel(data = subset(gps_data, Label ==
    "Benin" | Label == "Burkina Faso" | Label == "Gulf of Guinea"), aes(x = long,
    y = lat, label = Label), color = "#4C4B49", size = 16, family = "Times")

# Add cities
p_map3 <- p_map2 + geom_point(data = subset(gps_data, Label == "Porto Novo" | Label ==
    "Cotonou" | Label == "Parakou" | Label == "Ouagadougou" | Label == "Bobo Dioulasso"),
    aes(x = long, y = lat), size = 5, shape = 9, color = "black") + geom_label_repel(data = subset(gps_data,
    Label == "Porto Novo" | Label == "Cotonou" | Label == "Parakou" | Label == "Ouagadougou" |
        Label == "Bobo Dioulasso"), aes(x = long, y = lat, label = Label), color = "black",
    size = 8, family = "Times", box.padding = 1.75)

# Save with or without the city labels p_map2 ggsave(filename =
# 'p_map_notext.png', width = 16, height = 13, dpi = 300, units = 'in',
# device='png', scale = 1) p_map3 ggsave(filename = 'p_map.png', width = 16,
# height = 13, dpi = 300, units = 'in', device='png', scale = 1)

# Plot maps for sample sites in Finland
gps <- metadata[!duplicated(metadata[, c("lat", "long")]), ]
gps <- gps[, c("country", "lat", "long")]
gps_Fin <- subset(gps, country == "Finland")

Fin_map <- ggplot(data = world) + geom_sf() + borders("world", colour = "black",
    fill = "wheat1") + theme(panel.background = element_rect(fill = "azure1", colour = "azure1")) +
    geom_point(data = subset(gps_Fin), aes(x = long, y = lat), size = 4, shape = 16,
        color = "#B2182B") + coord_sf(ylim = c(60, 67), xlim = c(18, 33), expand = T) +
    theme(axis.text = element_text(family = "Times", size = 16), axis.title = element_blank()) +
    annotation_scale(location = "bl", width_hint = 0.1)

Fin_map <- Fin_map + theme(plot.margin = ggplot2::margin(0, 0, 0, 0, "cm"))
```

Data exploration using library HighstatLabv13

```
# Number of zeros in the response variable 100 * sum(df$ARG_SUM == 0) /
# nrow(df)

# Number of observations per level of a categorical covariate table(df$country)
# table(df$hospital) Only < 3 samples in some hospital groups. Let's not
# include that as a covariate.

# Let's fit a model with Gamma distribution with a log link. M0 <- glm(ARG_SUM
# ~ country, data = df, family='Gamma'(link='log'))

# MODEL VALIDATION Homogeneity Plot residuals vs fitted values F1 <- fitted(M0)
# E1 <- resid(M0, type = 'pearson') par(mfrow = c(1,1), cex.lab = 1.5, mar =
# c(5,5,2,2)) plot(x = F1, y = E1, xlab = 'Fitted values', ylab = 'Pearson
# residuals') abline(h = 0, lty = 2) No patterns, we are good.

# boxplot(E1 ~ country, data = df, ylab = 'Residuals') abline(h = 0) Looks
# good.

# Influential observations par(mfrow = c(1, 1)) plot(cooks.distance(M0), type =
# 'h', ylim = c(0, 1)) abline(h = 1) There are no influental observations
```

```
# Normality par(cex.lab = 1.5, mar = c(5,5,2,2)) E1 <- resid(M0) hist(E1,
# breaks = 15, xlab = 'Residuals', main = '')

# Independence due to model misfit df$E1 <- E1 MySel <- c('SSU_counts',
# 'intI1_SUM', 'country') MyMultipanel.ggp2(Z = df, varx = MySel, vary = 'E1',
# ylab = 'Residuals', addSmoother = TRUE, addRegressionLine = FALSE,
# addHorizontalLine = TRUE) Some / No clear non-linear patterns in these
# graphs.

# Check for spatial dependency MyCex <- 3 * abs(E1) / max(E1) MyCol <-
# ifelse(E1 > 0, 'red', 'blue') xyplot(long ~ lat, data = df, cex = MyCex, col
# = MyCol) In general, that no sig. spatial dependency can be detected.
```

Plot model (ARGs)

```
M0 <- glm(ARG_SUM ~ country, data = df, family = Gamma(link = "log"))
summary(M0)

cols <- get_palette(c("#B2182B", "#44AA99", "#2585E7"), 3)

glht.M0 <- glht(M0, mcp(country = "Tukey"))
summary(glht(glht.M0))

# Add the p values obtained above
pvalues <- tibble::tribble(~group1, ~group2, ~p, "Benin", "Burkina Faso", 0.001,
    "Benin", "Finland", 0.001, "Burkina Faso", "Finland", 0.0105)
pvalues

dfA <- cbind(df, Mean = predict(M0, newdata = df, type = "response"), SE = predict(M0,
    newdata = df, type = "response", se.fit = T)$se.fit)

resfinder_M0 <- ggplot(dfA, aes(x = country, y = Mean)) + scale_color_manual(values = cols) +
    geom_line() + geom_jitter(data = dfA, aes(x = country, y = ARG_SUM, color = country),
    size = 7.5, alpha = 1, width = 0.3) + geom_errorbar(aes(ymin = Mean - SE, ymax = Mean +
    SE), width = 0.5, lwd = 0.75) + geom_point(size = 0.9) + theme_linedraw() + theme(axis.text.x = element_text(ang
    size = 18, family = "Times", face = "bold"), axis.title.x = element_blank(),
    axis.text.y = element_text(size = 16, family = "Times"), axis.title.y = element_text(size = 16,
        family = "Times"), legend.position = "none", plot.title = element_text(size = 18,
        family = "Times", face = "bold")) + labs(y = "Normalized to 16S rRNA", x = "") +
    guides(color = "none", alpha = "none") + labs(title = "Relative sum abundance of ARGs")

ARG_sum <- resfinder_M0 + stat_pvalue_manual(pvalues, label = "p", y.position = 2.3,
    step.increase = 0.05, tip.length = 0.01, size = 5)
# ARG_sum

# ggsave(filename = 'resfinder_sum_M0.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)
```

Plot model (MGEs)

```
M1 <- glm(MGE_SUM ~ country, data = df, family = Gamma(link = "log"))
summary(M1)

glht.M1 <- glht(M1, mcp(country = "Tukey"))
summary(glht(glht.M1))

# Add the p values obtained above
pvalues <- tibble::tribble(~group1, ~group2, ~p, "Benin", "Burkina Faso", 0.991,
    "Benin", "Finland", 0.879, "Burkina Faso", "Finland", 0.911)
pvalues
```

```r
dfA <- cbind(df, Mean = predict(M1, newdata = df, type = "response"), SE = predict(M1,
    newdata = df, type = "response", se.fit = T)$se.fit)

cols <- get_palette(c("#B2182B", "#44AA99", "#2585E7"), 3)

MGE_M1 <- ggplot(dfA, aes(x = country, y = Mean)) + scale_color_manual(values = cols) +
    geom_line() + geom_jitter(data = dfA, aes(x = country, y = MGE_SUM, color = country),
    size = 7.5, alpha = 1, width = 0.3) + geom_errorbar(aes(ymin = Mean - SE, ymax = Mean +
    SE), width = 0.5, lwd = 0.75) + geom_point(size = 0.9) + theme_linedraw() + theme(axis.text.x = element_text(ang
    size = 18, family = "Times", face = "bold"), axis.title.x = element_blank(),
    axis.text.y = element_text(size = 16, family = "Times"), axis.title.y = element_blank(),
    legend.position = "none", plot.title = element_text(size = 18, family = "Times",
        face = "bold")) + labs(y = "Normalized to 16S rRNA", x = "") + guides(color = "none",
    alpha = "none") + labs(title = "Relative sum abundance of MGEs")

MGE_sum <- MGE_M1 + stat_pvalue_manual(pvalues, label = "p", y.position = 7, step.increase = 0.05,
    tip.length = 0.01, size = 5)
# MGE_sum

# ggsave(filename = 'MGE_sum_M1_new.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)
```

Plot model (intI1)

```r
M2 <- glm(intI1_SUM ~ country, data = df, family = Gamma(link = "log"))
summary(M2)

glht.M2 <- glht(M2, mcp(country = "Tukey"))
summary(glht(glht.M2))

# Add the p values obtained above
pvalues <- tibble::tribble(~group1, ~group2, ~p, "Benin", "Burkina Faso", 0.013,
    "Benin", "Finland", 0.001, "Burkina Faso", "Finland", 0.001)
pvalues

dfA <- cbind(df, Mean = predict(M2, newdata = df, type = "response"), SE = predict(M2,
    newdata = df, type = "response", se.fit = T)$se.fit)

cols <- get_palette(c("#B2182B", "#44AA99", "#2585E7"), 3)

intI1_M2 <- ggplot(dfA, aes(x = country, y = Mean)) + scale_color_manual(values = cols) +
    geom_line() + geom_jitter(data = dfA, aes(x = country, y = intI1_SUM, color = country),
    size = 7.5, alpha = 1, width = 0.3) + geom_errorbar(aes(ymin = Mean - SE, ymax = Mean +
    SE), width = 0.5, lwd = 0.75) + geom_point(size = 0.9) + theme_linedraw() + theme(axis.text.x = element_text(ang
    size = 18, family = "Times", face = "bold"), axis.title.x = element_blank(),
    axis.text.y = element_text(size = 16, family = "Times"), axis.title.y = element_blank(),
    legend.position = "none", plot.title = element_text(size = 18, family = "Times",
        face = "bold")) + labs(y = "Normalized to 16S rRNA", x = "") + guides(color = "none",
    alpha = "none") + labs(title = "Relative sum abundance of intI1")

intI1_sum <- intI1_M2 + stat_pvalue_manual(pvalues, label = "p", y.position = 1.05,
    step.increase = 0.05, tip.length = 0.01, size = 5)
# intI1_sum

# ggsave(filename = 'intI1_sum_M2_new.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)
```

Plot figures in grids

```
design <- "
###
ABC
###
"
# ARG_sum + MGE_sum + intI1_sum + plot_layout(design = design) +
# plot_annotation(tag_levels = c('A', 'B', 'C')) & theme(plot.tag =
# element_text(size = 24, family = 'Times'))

# ggsave(filename = 'sums_grid.png', width = 16, height = 13, dpi = 300, units
# = 'in', device='png', scale = 1)
```

Plot models by hospitals / hospital sections (ARGs)

```
# Benin
resfinder_PHY_stat_Ben <- subset_samples(resfinder_PHY_stat, country == "Benin")
df <- data.frame(ARG_SUM = sample_sums(resfinder_PHY_stat_Ben), hospital = as.factor(sample_data(resfinder_PHY_stat_

# Fit model
M3 <- glm(ARG_SUM ~ hospital, data = df, family = Gamma(link = "log"))
summary(M3)

glht.M3 <- glht(M3, mcp(hospital = "Tukey"))
summary(glht(glht.M3))

# BF
resfinder_PHY_stat_BF <- subset_samples(resfinder_PHY_stat, country == "Burkina Faso")
df <- data.frame(ARG_SUM = sample_sums(resfinder_PHY_stat_BF), hospital = as.factor(sample_data(resfinder_PHY_stat_B

# Fit model
M3 <- glm(ARG_SUM ~ hospital, data = df, family = Gamma(link = "log"))
summary(M3)

glht.M3 <- glht(M3, mcp(hospital = "Tukey"))
summary(glht(glht.M3))

# Finland
resfinder_PHY_stat_Fin <- subset_samples(resfinder_PHY_stat, country == "Finland")
df <- data.frame(ARG_SUM = sample_sums(resfinder_PHY_stat_Fin), hospital = as.factor(sample_data(resfinder_PHY_stat_

# Fit model
M3 <- glm(ARG_SUM ~ hospital, data = df, family = Gamma(link = "log"))
summary(M3)

glht.M3 <- glht(M3, mcp(hospital = "Tukey"))
summary(glht(glht.M3))

# Hospital section
df <- data.frame(ARG_SUM = sample_sums(resfinder_PHY_stat), hospital_section = as.factor(sample_data(resfinder_PHY_s

# Fit model
M4 <- glm(ARG_SUM ~ hospital_section, data = df, family = Gamma(link = "log"))
summary(M4)

glht.M4 <- glht(M4, mcp(hospital_section = "Tukey"))
summary(glht(glht.M4))
```

Ordinations

ARGs (ResFinder)

```
resfinder_PHY_ord <- ordinate(resfinder_PHY_stat, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(resfinder_PHY_stat, resfinder_PHY_ord, color = "country")
resfinder.p_ord <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) +
    geom_point(size = 3.5) + stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = alias),
    size = 4, family = "Times", hjust = 1.2, vjust = 0.3) + theme_minimal() + labs(title = "Resistome",
    subtitle = "Hospital WWs in Benin, Burkina Faso and Finland") + theme(plot.title = element_text(size = 36,
    family = "Times", face = "bold"), plot.subtitle = element_text(size = 20, family = "Times"),
    legend.text = element_text(size = 18, family = "Times"), legend.title = element_blank(),
    axis.title = element_text(size = 36, family = "Times"), axis.text = element_text(size = 18,
        family = "Times")) + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))

# leg_ord <- get_legend(resfinder.p_ord)

# Convert to a ggplot and print as_ggplot(leg_ord)

# Save ggsave(filename = 'ord_resfinder_new.png', width = 16, height = 13, dpi
# = 300, units = 'in', device='png', scale = 1)

# Test significance using pair-wise adonis resfinder_temp <-
# subset_samples(resfinder_PHY_stat_equal1, (country == 'Benin' | country ==
# 'Finland')) resfinder_dist <- vegdist(t(otu_table(resfinder_temp)), dist =
# 'horn') adonis(resfinder_dist ~ country, data =
# data.frame(sample_data(resfinder_temp), permutations = 9999))

# resfinder_temp <- subset_samples(resfinder_PHY_stat_equal1, (country ==
# 'Benin' | country == 'Burkina Faso')) resfinder_dist <-
# vegdist(t(otu_table(resfinder_temp)), dist = 'horn') adonis(resfinder_dist ~
# country, data = data.frame(sample_data(resfinder_temp), permutations = 9999))

# resfinder_temp <- subset_samples(resfinder_PHY_stat_equal1, (country ==
# 'Burkina Faso' | country == 'Finland')) resfinder_dist <-
# vegdist(t(otu_table(resfinder_temp)), dist = 'horn') adonis(resfinder_dist ~
# country, data = data.frame(sample_data(resfinder_temp), permutations = 9999))
```

Taxa (Metaphlan3)

```
PHY = transform_sample_counts(metaphlan_PHY_stat, function(x) 1e+06 * x/sum(x))

metaphlan_PHY_ord <- ordinate(PHY, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(PHY, metaphlan_PHY_ord, color = "country")
metaphlan.p_ord <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) +
    geom_point(size = 3.5) + stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = alias),
    size = 4, family = "Times", hjust = 1.2, vjust = 0.3) + theme_minimal() + labs(title = "Taxonomical composition"
    subtitle = "Hospital WWs in Benin, Burkina Faso and Finland") + theme(plot.title = element_text(size = 36,
    family = "Times", face = "bold"), plot.subtitle = element_text(size = 20, family = "Times"),
    legend.text = element_text(size = 50, family = "Times"), legend.title = element_blank(),
    axis.title = element_text(size = 36, family = "Times"), axis.text = element_text(size = 18,
        family = "Times")) + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))

# Save ggsave(filename = 'ord_metaphlan_new.png', width = 16, height = 13, dpi
# = 300, units = 'in', device='png', scale = 1)

# Test significance using pair-wise adonis metaphlan_temp <-
# subset_samples(metaphlan_PHY_stat_equal1, (country == 'Benin' | country ==
# 'Finland')) metaphlan_dist <- vegdist(t(otu_table(metaphlan_temp)), dist =
```

```
# 'horn') adonis(metaphlan_dist ~ country, data =
# data.frame(sample_data(metaphlan_temp), permutations = 9999))

# metaphlan_temp <- subset_samples(metaphlan_PHY_stat_equal1, (country ==
# 'Benin' | country == 'Burkina Faso')) metaphlan_dist <-
# vegdist(t(otu_table(metaphlan_temp)), dist = 'horn') adonis(metaphlan_dist ~
# country, data = data.frame(sample_data(metaphlan_temp), permutations = 9999))

# metaphlan_temp <- subset_samples(metaphlan_PHY_stat_equal1, (country ==
# 'Burkina Faso' | country == 'Finland')) metaphlan_dist <-
# vegdist(t(otu_table(metaphlan_temp)), dist = 'horn') adonis(metaphlan_dist ~
# country, data = data.frame(sample_data(metaphlan_temp), permutations = 9999))
```

MGEs

```
MGE_PHY_ord <- ordinate(MGE_PHY_stat, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(MGE_PHY_stat, MGE_PHY_ord, color = "country")
MGE.p_ord <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) +
    geom_point(size = 3.5) + stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = alias),
    size = 4, family = "Times", hjust = 1.2, vjust = 0.3) + theme_minimal() + labs(title = "Mobilome",
    subtitle = "Hospital WWs in Benin, Burkina Faso and Finland") + theme(plot.title = element_text(size = 36,
    family = "Times", face = "bold"), plot.subtitle = element_text(size = 20, family = "Times"),
    legend.text = element_text(size = 50, family = "Times"), legend.title = element_blank(),
    axis.title = element_text(size = 36, family = "Times"), axis.text = element_text(size = 18,
        family = "Times")) + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))

# Save ggsave(filename = 'ord_mge_new.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)

# Test significance using pair-wise adonis MGE_temp <-
# subset_samples(MGE_PHY_stat_equal1, (country == 'Benin' | country ==
# 'Finland')) MGE_dist <- vegdist(t(otu_table(MGE_temp)), dist = 'horn')
# adonis(MGE_dist ~ country, data = data.frame(sample_data(MGE_temp),
# permutations = 9999))

# MGE_temp <- subset_samples(MGE_PHY_stat_equal1, (country == 'Benin' | country
# == 'Burkina Faso')) MGE_dist <- vegdist(t(otu_table(MGE_temp)), dist =
# 'horn') adonis(MGE_dist ~ country, data = data.frame(sample_data(MGE_temp),
# permutations = 9999))

# MGE_temp <- subset_samples(MGE_PHY_stat_equal1, (country == 'Burkina Faso' |
# country == 'Finland')) MGE_dist <- vegdist(t(otu_table(MGE_temp)), dist =
# 'horn') adonis(MGE_dist ~ country, data = data.frame(sample_data(MGE_temp),
# permutations = 9999))
```

Plot ordinations in figure panel

```
resfinder_PHY_ord <- ordinate(resfinder_PHY_stat, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(resfinder_PHY_stat, resfinder_PHY_ord, color = "country")
arg <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) +
    geom_point(size = 2) + stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = hospital),
    size = 4, family = "Times", hjust = 1.2) + theme_minimal() + labs(title = "Resistome") +
    theme(plot.title = element_text(size = 20, family = "Times", face = "bold"),
        legend.position = "none", axis.title = element_text(size = 18, family = "Times"),
        axis.text = element_text(size = 18, family = "Times")) + theme(plot.margin = unit(c(0.1,
    0.1, 0.1, 1), "cm")) + coord_fixed() + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))
```

```r
# Data into counts
PHY = transform_sample_counts(metaphlan_PHY_stat, function(x) 1e+06 * x/sum(x))

metaphlan_PHY_ord <- ordinate(PHY, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(PHY, metaphlan_PHY_ord, color = "country")
mp <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) + geom_point(size = 2) +
    stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = hospital),
    size = 4, family = "Times", hjust = 1.2) + theme_minimal() + labs(title = "Taxonomical composition") +
    theme(plot.title = element_text(size = 20, family = "Times", face = "bold", hjust = 0.5),
        legend.position = "none", axis.title = element_text(size = 18, family = "Times"),
        axis.text = element_text(size = 18, family = "Times")) + theme(plot.margin = unit(c(0.1,
    0.1, 0.1, 0.1), "cm")) + coord_fixed() + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))

MGE_PHY_ord <- ordinate(MGE_PHY_stat, method = "PCoA", distance = "horn")
p_ord <- plot_ordination(MGE_PHY_stat, MGE_PHY_ord, color = "country")
mge <- p_ord + scale_color_manual(values = c("#B2182B", "#72D39D", "#2585E7")) +
    geom_point(size = 2) + stat_ellipse(level = 0.9, linetype = 1) + geom_text_repel(mapping = aes(label = hospital)
    size = 4, family = "Times", hjust = 1.2) + theme_minimal() + labs(title = "Mobilome") +
    theme(plot.title = element_text(size = 20, family = "Times", face = "bold", hjust = 0.5),
        legend.position = "none", axis.title = element_text(size = 18, family = "Times"),
        axis.text = element_text(size = 18, family = "Times")) + theme(plot.margin = unit(c(0.1,
    0.1, 0.1, 0.1), "cm")) + coord_fixed() + guides(fill = guide_legend(override.aes = list(linetype = 0)),
    color = guide_legend(override.aes = list(linetype = 0, size = 5)))

p <- arg + mp + mge + plot_layout(nrow = 2) + plot_annotation(tag_levels = list(c("A",
    "B", "C"))) & theme(plot.tag = element_text(size = 24, family = "Times"), plot.tag.position = c(0,
    1))

# p_leg <- p + inset_element(leg_ord, left = 1, bottom = 1, right = 1.7, top =
# 0)

# Save ggsave(filename = 'ord_patch_supp.png', width = 16, height = 13, dpi =
# 300, units = 'in', device='png', scale = 1)
```

DESeq2

ARGs

Finland-Benin

```r
OTU_resfinder <- as.matrix(read.table("ARG_genemat.txt", header = T, check.names = F,
    row.names = 1))

# Reorder to match metadata
match <- match(rownames(metadata), colnames(OTU_resfinder))
OTU_resfinder <- OTU_resfinder[, match]
all(colnames(OTU_resfinder) == rownames(metadata))

# Tax_table
clusters_tax_table_resfinder <- read.csv("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/clusters_tax_table.txt",
    header = FALSE, sep = ";")
colnames(clusters_tax_table_resfinder) <- c("Gene", "Cluster_name", "Class")
# Reorder columns
col_order <- c("Class", "Cluster_name", "Gene")
clusters_tax_table_resfinder <- clusters_tax_table_resfinder[, col_order]

# Reorder tax_table to match
match <- match(rownames(OTU_resfinder), clusters_tax_table_resfinder$Gene)
clusters_tax_table_resfinder <- clusters_tax_table_resfinder[match, ]
all(rownames(OTU_resfinder) == clusters_tax_table_resfinder$Gene)
```

```r
# Divide by ARG gene lengths
resfinder_lengths <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/resfinder_lengths.txt",
    header = FALSE, comment.char = "#")
all(rownames(clusters_tax_table_resfinder$Gene) == resfinder_lengths$V1)
OTU_resfinder_length_norm <- OTU_resfinder/resfinder_lengths[, 2]

# Normalization with Metaxa2 SSU counts
deseq_OTU_resfinder <- t(t(OTU_resfinder_length_norm)/metadata$SSU_counts) * 1540
all(rownames(metadata) == colnames(deseq_OTU_resfinder))
identical(OTU_resfinder_length_norm[2025, 5]/metadata$SSU_counts[5], deseq_OTU_resfinder[2025,
    5])
all(rownames(OTU_resfinder_length_norm) == clusters_tax_table_resfinder$Gene)

# Deseq
deseq_OTU <- deseq_OTU_resfinder[, ] * 10^5 + 1

# Hide rownames
dim(deseq_OTU)
rownames(deseq_OTU) <- c(1:3104)

dim(clusters_tax_table_resfinder)
rownames(clusters_tax_table_resfinder) <- c(1:3104)

resfinder_deseq <- phyloseq(otu_table(deseq_OTU, taxa_are_rows = T), sample_data(metadata),
    tax_table(as.matrix(clusters_tax_table_resfinder)))

## Exclude biological / technical replicates
resfinder_deseq <- subset_samples(resfinder_deseq, alias != "BH31" & alias != "BH33" &
    alias != "BH34B" & alias != "BH10" & alias != "BFH38B" & alias != "FH8" & alias !=
    "BH45" & alias != "BH59" & alias != "BH62")

# Create phyloseq object with only hospital WW samples sequenced here
resfinder_deseq_stat <- subset_samples(resfinder_deseq, category == "WA hospital effluent" |
    category == "North Eu hospital effluent")

# Take pair wise comparisons
deseq_PHY = subset_samples(resfinder_deseq_stat, country == "Benin" | country ==
    "Finland")

# hist(log10(apply(otu_table(deseq_PHY), 1, var)), xlab = 'log10(variance)')

# Let's set a threashold for the variance
varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Benin", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)
res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_resfinder = res[which(res$padj < alpha), ]
sigtab_resfinder = cbind(as(sigtab_resfinder, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_resfinder),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))
```

```
sigtab_resfinder = merge(sigtab_resfinder, as.data.frame(n), by = 0)

sorted_sigtab <- sigtab_resfinder[order(-sigtab_resfinder$log2FoldChange), ]
# write.table(sorted_sigtab,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/DESeq2_Ben_Fin.txt',
# row.names=T, sep = '\t', col.names = T)

Fin_Ben <- subset(sorted_sigtab, log2FoldChange >= 0)
Ben_Fin <- subset(sorted_sigtab, log2FoldChange <= 0)
Ben_Fin <- Ben_Fin[order(Ben_Fin$log2FoldChange), ]
```

Burkina Faso-Finland

```
# Take pair wise comparisons
deseq_PHY = subset_samples(resfinder_deseq_stat, country == "Burkina Faso" | country ==
    "Finland")

# hist(log10(apply(otu_table(deseq_PHY), 1, var)), xlab = 'log10(variance)')

# Let's set a threashold for the variance
varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Burkina Faso", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)

res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
# resultsNames(dds)

res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_resfinder = res[which(res$padj < alpha), ]

sigtab_resfinder = cbind(as(sigtab_resfinder, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_resfinder),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))

sigtab_resfinder = merge(sigtab_resfinder, as.data.frame(n), by = 0)

sorted_sigtab <- sigtab_resfinder[order(-sigtab_resfinder$log2FoldChange), ]

# Save BF write.table(sorted_sigtab,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/DESeq2_BF_Fin.txt',
# row.names=T, sep = '\t', col.names = T)

Fin_BF <- subset(sorted_sigtab, log2FoldChange >= 0)
BF_Fin <- subset(sorted_sigtab, log2FoldChange <= 0)
BF_Fin <- BF_Fin[order(BF_Fin$log2FoldChange), ]
# head(Fin_BF) head(BF_Fin)
```

Taxa (Metaphlan3), Species, Benin-Finland

```
OTU_metaphlan <- read.delim("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/mod_merged_abundance_table_species.txt",
    header = T)

# Match sample order
tax_table_metaphlan <- read.table("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/tax_table_metaphlan",
    quote = "\"", comment.char = "")
identical(tax_table_metaphlan$V1, OTU_metaphlan$clade_name)

tax_table_metaphlan <- read.csv("~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/tax_table_metaphlan",
    header = FALSE, sep = ";")
colnames(tax_table_metaphlan) <- c("Kingdom", "Phylum", "Class", "Order", "Family",
    "Genus", "Species")
# Remove '__'
tax_table_metaphlan <- apply(tax_table_metaphlan, 2, function(y) (gsub(".__", "",
    y)))

match <- match(rownames(metadata), colnames(OTU_metaphlan))
OTU_metaphlan <- OTU_metaphlan[, match]
all(rownames(metadata) == colnames(OTU_metaphlan))

OTU_metaphlan_deseq = OTU_metaphlan

# Multiply with SSU counts from Metaxa2
vec <- as.vector(metadata$SSU_counts)
deseq_OTU <- mapply(FUN = `*`, as.data.frame(OTU_metaphlan_deseq), vec)

metaphlan_deseq <- phyloseq(otu_table(deseq_OTU, taxa_are_rows = T), sample_data(metadata),
    tax_table(as.matrix(tax_table_metaphlan)))

## Exclude biological / technical replicates
metaphlan_deseq_stat <- subset_samples(metaphlan_deseq, alias != "BH31" & alias !=
    "BH33" & alias != "BH34B" & alias != "BH10" & alias != "BFH38B" & alias != "FH8" &
    alias != "BH45" & alias != "BH59" & alias != "BH62")

# Create phyloseq object with only hospital WW samples sequenced here
metaphlan_deseq_stat <- subset_samples(metaphlan_deseq_stat, category == "WA hospital effluent" |
    category == "North Eu hospital effluent")

metaphlan_deseq_stat <- prune_taxa(taxa_sums(metaphlan_deseq_stat) > 0, metaphlan_deseq_stat)

# Take pair wise comparisons
deseq_PHY = subset_samples(metaphlan_deseq_stat, country == "Benin" | country ==
    "Finland")

varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Benin", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)

res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
# resultsNames(dds)

res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_metaphlan = res[which(res$padj < alpha), ]
```

```
sigtab_metaphlan = cbind(as(sigtab_metaphlan, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_metaphlan),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))

sigtab_metaphlan = merge(sigtab_metaphlan, as.data.frame(n), by = 0)

sorted_sigtab <- sigtab_metaphlan[order(-sigtab_metaphlan$log2FoldChange), ]
# head(sorted_sigtab) write.table(sorted_sigtab,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/metaphlan3_DESeq2_Ben_Fin_s.txt',
# row.names=T, sep = '\t', col.names = T)
```

Taxa (Metaphlan3), Genus, Benin-Finland

```
# Take pair wise comparisons
deseq_PHY = subset_samples(metaphlan_deseq_stat, country == "Benin" | country ==
    "Finland")

# Get genus
deseq_PHY <- tax_glom(deseq_PHY, taxrank = "Genus")

varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Benin", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)

res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
# resultsNames(dds)

res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_metaphlan = res[which(res$padj < alpha), ]

sigtab_metaphlan = cbind(as(sigtab_metaphlan, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_metaphlan),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))

sigtab_metaphlan = merge(sigtab_metaphlan, as.data.frame(n), by = 0)

sorted_metaphlan_sigtable <- sigtab_metaphlan[order(-sigtab_metaphlan$log2FoldChange),
    ]
# head(sorted_metaphlan_sigtable) write.table(sorted_metaphlan_sigtable,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/metaphlan3_DESeq2_Ben_Fin_g.txt',
# row.names=T, sep = '\t', col.names = T)
```

Taxa (Metaphlan3), Species, Burkina Faso-Finland

```
# Take pair wise comparisons
deseq_PHY = subset_samples(metaphlan_deseq_stat, country == "Burkina Faso" | country ==
    "Finland")

varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Burkina Faso", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)

res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
# resultsNames(dds)

res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_metaphlan = res[which(res$padj < alpha), ]

sigtab_metaphlan = cbind(as(sigtab_metaphlan, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_metaphlan),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))

sigtab_metaphlan = merge(sigtab_metaphlan, as.data.frame(n), by = 0)

sorted_sigtab <- sigtab_metaphlan[order(-sigtab_metaphlan$log2FoldChange), ]
# head(sorted_sigtab) write.table(sorted_sigtab,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/metaphlan3_DESeq2_BF_Fin_s.txt',
# row.names=T, sep = '\t', col.names = T)
```

Taxa (Metaphlan3), Genus, Burkina Faso-Finland

```
# Take pair wise comparisons
deseq_PHY = subset_samples(metaphlan_deseq_stat, country == "Burkina Faso" | country ==
    "Finland")

# Get genus
deseq_PHY <- tax_glom(deseq_PHY, taxrank = "Genus")

varianceThreshold = 50
keepOTUs = apply(otu_table(deseq_PHY), 1, var) > varianceThreshold
deseq_PHY = prune_taxa(keepOTUs, deseq_PHY)
deseq_PHY

dds = phyloseq_to_deseq2(deseq_PHY, ~country)

dds$category <- relevel(dds$country, "Burkina Faso", "Finland")

dds = DESeq(dds, fitType = "mean", test = "Wald", betaPrior = FALSE)

res = results(dds, cooksCutoff = FALSE, alpha = 0.05)
# resultsNames(dds)
```

```
res = res[order(res$padj, na.last = NA), ]

alpha = 0.05
sigtab_metaphlan = res[which(res$padj < alpha), ]

sigtab_metaphlan = cbind(as(sigtab_metaphlan, "data.frame"), as(tax_table(deseq_PHY)[rownames(sigtab_metaphlan),
    ], "matrix"))

otu_table(deseq_PHY)[otu_table(deseq_PHY) == 1] <- 0
otu_table(deseq_PHY)[otu_table(deseq_PHY) > 0] <- 1

n <- rowSums(otu_table(deseq_PHY))

sigtab_metaphlan = merge(sigtab_metaphlan, as.data.frame(n), by = 0)

sorted_metaphlan_sigtable <- sigtab_metaphlan[order(-sigtab_metaphlan$log2FoldChange),
    ]
# head(sorted_metaphlan_sigtable) write.table(sorted_metaphlan_sigtable,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/metaphlan3_Genus_DESeq2_BF_Fin_g.txt',
# row.names=T, sep = '\t', col.names = T)
```

Heatmap for clinically relevant taxa

```
metaphlan_PHY_Species <- tax_glom(metaphlan_PHY_stat, taxrank = "Species")

# ESKAPEEc and other relevant taxa
selected <- subset_taxa(metaphlan_PHY_Species, Species == "Acinetobacter_baumannii" |
    Species == "Acinetobacter_nosocomialis" | Species == "Acinetobacter_bouvetii" |
    Species == "Acinetobacter_johnsonii" | Species == "Acinetobacter_radioresistens" |
    Species == "Acinetobacter_lwoffii" | Species == "Acinetobacter_calcoaceticus" |
    Species == "Acinetobacter_haemolyticus" | Species == "Acinetobacter_bereziniae" |
    Species == "Acinetobacter_venetianus" | Species == "Acinetobacter_calcoaceticus" |
    Species == "Acinetobacter_pittii" | Species == "Acinetobacter_guillouiae" | Species ==
    "Acinetobacter_schindleri" | Species == "Acinetobacter_bereziniae" | Species ==
    "Acinetobacter_kyonggiensis" | Species == "Enterobacter_cloacae_complex" | Species ==
    "Enterococcus_faecium" | Species == "Klebsiella_pneumoniae" | Species == "Staphylococcus_aureus" |
    Species == "Pseudomonas_aeruginosa_group" | Species == "Escherichia_coli")

# Filter out low abundance taxa
selected <- subset_taxa(selected, taxa_sums(selected) != 0)

# OTU matrix
heat_OTU = as(otu_table(selected), "matrix")
# Coerce to data.frame
heat.df = as.data.frame(heat_OTU)

# Tax table matrix
heat_tax = as(tax_table(selected), "matrix")

# Swap colnames
match <- match(rownames(heat.df), rownames(heat_tax))
temp <- heat_tax[match, ]
all(rownames(temp) == rownames(heat_OTU))
all(rownames(temp) == rownames(heat_tax))
rownames(heat.df) <- temp[, 7]

new_df <- heat.df[order(row.names(heat.df)), ]
new_tax = heat_tax
rownames(new_tax) <- paste(selected@tax_table[, 7])
new_tax[order(row.names(new_tax)), ]
```

```
# Col annotation
country <- as.matrix(sample_data(selected)[["country"]])
country <- as.factor(country)
country <- data.frame(country)
colnames(country) <- c("country")
rownames(country) <- as.matrix(colnames(otu_table(selected)))
country$country <- gsub(" ", "_", country$country)

ann_colors <- list(country = c(Benin = "#B2182B", Burkina_Faso = "#44AA99", Finland = "#2166AC"))

colnames(new_df) <- gsub(pattern = "_[A-Z].*", replacement = "_", colnames(new_df))
rownames(new_df) <- gsub(patter = "_", replacement = " ", rownames(new_df))

## Plot log
newnames <- lapply(rownames(new_df), function(x) bquote(italic(.(x))))

# Plot heat <- pheatmap(sqrt(new_df), cluster_rows = F, cluster_cols = T,
# border_color = 'grey', colorRampPalette(brewer.pal(9, 'Blues'))(100), main =
# 'Relative abundance of clinically relevant species\n (Metaphlan3, square root
# transformed)', angle_col = 90, legend = TRUE, fontsize_row = 11, labels_row =
# as.expression(newnames), filename = 'eskape_heat.png', annotation_col =
# country, clustering_distance_cols = 'euclidean', show_colnames = T, cellwidth
# = 13, cellheight = 26, gaps_row = rep(c(12)), annotation_colors = ann_colors)
```

15 most abundant ARGs in HWWs from each country

```
# Benin
resfinder_PHY_stat_Ben <- subset_samples(resfinder_PHY_stat, country == "Benin")
resfinder_PHY_stat_Ben_abun <- tax_glom(resfinder_PHY_stat_Ben, taxrank = "Gene")

# Take 15 most abundant
resfinder_PHY_stat_Ben_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_stat_Ben_abun),
    TRUE)[1:15]), resfinder_PHY_stat_Ben_abun)

# BF
resfinder_PHY_stat_BF <- subset_samples(resfinder_PHY_stat, country == "Burkina Faso")
resfinder_PHY_stat_BF_abun <- tax_glom(resfinder_PHY_stat_BF, taxrank = "Gene")

# Take 15 most abundant
resfinder_PHY_stat_BF_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_stat_BF_abun),
    TRUE)[1:15]), resfinder_PHY_stat_BF_abun)

# Finland
resfinder_PHY_stat_Fin <- subset_samples(resfinder_PHY_stat, country == "Finland")
resfinder_PHY_stat_Fin_abun <- tax_glom(resfinder_PHY_stat_Fin, taxrank = "Gene")

# Take 15 most abundant
resfinder_PHY_stat_Fin_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_stat_Fin_abun),
    TRUE)[1:15]), resfinder_PHY_stat_Fin_abun)

# Create dataframe
Benin <- data.frame(resfinder_PHY_stat_Ben_abun@tax_table)$Gene
BF <- data.frame(resfinder_PHY_stat_BF_abun@tax_table)$Gene
Finland <- data.frame(resfinder_PHY_stat_Fin_abun@tax_table)$Gene

top_ARGs <- data.frame(Benin, BF, Finland)
```

Most abundant ARGs in other than HWW samples

```
## Sample sums feces
resfinder_PHY_feces <- subset_samples(resfinder_PHY, alias == "BH20" | alias == "BH22" |
    alias == "BH24" | alias == "BH25")
resfinder_PHY_feces <- tax_glom(resfinder_PHY_feces, taxrank = "Gene")
# Take 15 most abundant
resfinder_PHY_feces_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_feces),
    TRUE)[1:15]), resfinder_PHY_feces)
resfinder_PHY_feces_abun@tax_table

# drinking
resfinder_PHY_ben_drink <- subset_samples(resfinder_PHY, alias == "BSE100" | alias ==
    "BSE74" | alias == "BSE79" | alias == "BSE93" | alias == "BH11")
resfinder_PHY_ben_drink <- subset_taxa(resfinder_PHY_ben_drink, taxa_sums(resfinder_PHY_ben_drink) !=
    0)
resfinder_PHY_ben_drink <- tax_glom(resfinder_PHY_ben_drink, taxrank = "Gene")
# Take 15 most abundant
resfinder_PHY_ben_drink_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_ben_drink),
    TRUE)[1:15]), resfinder_PHY_ben_drink)
resfinder_PHY_ben_drink_abun@tax_table

# other, Benin
resfinder_PHY_ben_other <- subset_samples(resfinder_PHY, alias == "BH13" | alias ==
    "BH14" | alias == "BH32" | alias == "BH52")
resfinder_PHY_ben_other <- subset_taxa(resfinder_PHY_ben_other, taxa_sums(resfinder_PHY_ben_other) !=
    0)
resfinder_PHY_ben_other <- tax_glom(resfinder_PHY_ben_other, taxrank = "Gene")
# Take 15 most abundant
resfinder_PHY_ben_other_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_ben_other),
    TRUE)[1:15]), resfinder_PHY_ben_other)
resfinder_PHY_ben_other_abun@tax_table

# other, BF
resfinder_PHY_BF_other <- subset_samples(resfinder_PHY, alias == "BFH27" | alias ==
    "BFH42" | alias == "BFH26")
resfinder_PHY_BF_other <- subset_taxa(resfinder_PHY_BF_other, taxa_sums(resfinder_PHY_BF_other) !=
    0)
resfinder_PHY_BF_other <- tax_glom(resfinder_PHY_BF_other, taxrank = "Gene")
# Take 15 most abundant
resfinder_PHY_BF_other_abun <- prune_taxa(names(sort(taxa_sums(resfinder_PHY_BF_other),
    TRUE)[1:15]), resfinder_PHY_BF_other)
resfinder_PHY_BF_other_abun@tax_table
```

Interesting ARGs

MCR

```
# Save sums
resfinder_PHY_mcr <- subset_taxa(resfinder_PHY_stat, Class == "Polymyxin")
resfinder_PHY_mcr <- tax_glom(resfinder_PHY_mcr, taxrank = "Cluster_name")
name <- data.frame(unique(resfinder_PHY_mcr@tax_table))

resfinder_PHY_mcr_1 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-1.11_1_clust")
mcr <- data.frame(sample_sums(resfinder_PHY_mcr_1))
resfinder_PHY_mcr_2 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-2.1_1_clust")
mcr$"mcr-2.1_1_clust" <- data.frame(sample_sums(resfinder_PHY_mcr_2))
resfinder_PHY_mcr_3.1 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-3.1_1_clust")
mcr$"mcr-3.1_1_clust" <- data.frame(sample_sums(resfinder_PHY_mcr_3.1))
resfinder_PHY_mcr_3.17 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-3.17_1")
```

```
mcr$"mcr-3.17_1" <- data.frame(sample_sums(resfinder_PHY_mcr_3.17))
resfinder_PHY_mcr_4 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-4.1_1_clust")
mcr$"mcr-4.1_1_clust" <- data.frame(sample_sums(resfinder_PHY_mcr_4))
resfinder_PHY_mcr_5 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-5.1_1_clust")
mcr$"mcr-5.1_1_clust" <- data.frame(sample_sums(resfinder_PHY_mcr_5))
resfinder_PHY_mcr_6 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-6.1_1")
mcr$"mcr-6.1_1" <- data.frame(sample_sums(resfinder_PHY_mcr_6))
resfinder_PHY_mcr_7 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-7.1_1")
mcr$"mcr-7.1_1" <- data.frame(sample_sums(resfinder_PHY_mcr_7))
resfinder_PHY_mcr_8 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-8_1")
mcr$"mcr-8_1" <- data.frame(sample_sums(resfinder_PHY_mcr_8))
resfinder_PHY_mcr_9 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-9_1")
mcr$"mcr-9_1" <- data.frame(sample_sums(resfinder_PHY_mcr_9))
resfinder_PHY_mcr_10 <- subset_taxa(resfinder_PHY_mcr, Cluster_name == "mcr-10_1")
mcr$"mcr-10_1" <- data.frame(sample_sums(resfinder_PHY_mcr_10))
```

Carbapenemases

```
resfinder_PHY_Cluster_1 <- subset_taxa(resfinder_PHY_stat, Cluster_name == "blaKPC-34_1_clust"|
          Cluster_name == "blaNDM-18_1_clust" | Cluster_name == "blaVIM-48_1_clust" |
          Cluster_name == "blaIMP-1_1_clust" | Cluster_name == "blaOXA-397_1_clust")

resfinder_PHY_Cluster_3 <- subset_taxa(resfinder_PHY_stat, Gene == "blaGES-2_1_AF326355" |
    Gene == "blaGES-4_1_AB116723" | Gene == "blaGES-5_1_DQ236171" |
    Gene == "blaGES-6_1_AY494718" | Gene == "blaGES-14_1_GU207844" |
    Gene == "blaGES-16_1_HM173356" | Gene == "blaGES-18_1_JQ028729"|
    Gene == "blaGES-20_1_JN596280" | Gene == "blaOXA-48_1_AY236073" |
    Gene == "blaOXA-162_1_GU197550" | Gene == "blaOXA-181_1_CM004561" |
    Gene == "blaOXA-199_1_JN704570" | Gene == "blaOXA-204_1_KP027885" |
    Gene == "blaOXA-232_1_JX423831" | Gene == "blaOXA-244_1_KP659189" |
    Gene == "blaOXA-245_1_JX438001" | Gene == "blaOXA-247_1_JX893517" |
    Gene == "blaOXA-247_1_JX893517" | Gene == "blaOXA-514_1_KU866382" |
    Gene == "blaOXA-515_1_KU866383" | Gene == "blaOXA-517_1_KU878974") # blaOXA-48-like

resfinder_PHY_Cluster <- merge_phyloseq(resfinder_PHY_Cluster_1, resfinder_PHY_Cluster_3)

cols <- get_palette(c("#332288", "#117733", "#52BFAD", "#88CCEE", "#DDCC77", "#FDA4B3",
                      "#F22D3D", "#882255", "#5F5E98", "#E4C960", "#FD8FD9"), 11)

hospital <- factor(c("F", "G",  "G",    "G",    "H",    "H",    "H",
    "I",    "I",    "I",    "I",    "I",    "I", "I",   "I",    "I",    "I",
    "J",    "J",    "J",    "J",    "J",    "J",    "J",    "J", "J", "J",
    "J",    "F",    "J",    "J",    "G",    "G",    "G",    "A",    "A", "A",
    "A",    "A",    "A",    "A",    "B",    "B",    "B",    "B",    "B",    "B",
    "B",    "B",    "B",    "C",    "C", "C",   "C",    "C",    "C",    "D",
    "D",    "D",    "K",    "K",    "K",    "L",    "L",    "L",    "M",    "N"))

rfc <- plot_bar(resfinder_PHY_Cluster, fill = "Cluster_name")
rfc_plot <- rfc + geom_bar(stat="identity", color = NA, size = 0) + scale_fill_manual(values = cols,
  labels = c("blaGES", "blaIMP", "blaKPC", "blaNDM", "blaOXA-48", "blaOXA-58", "blaVIM")) +
  labs(y = expression(atop(bold("ARGs/16S rRNA")))) + ggtitle("Hospital wastewaters") +
  scale_x_discrete(breaks=levels(factor(rownames(sample_data(resfinder_PHY_Cluster)))),
  labels=hospital, expression(bar("x"))) + theme_minimal() +
  theme(axis.text.x = element_text(size = 19, family = "Times", angle = 0, hjust = 0.6, vjust = 1),
  axis.text.y = element_text(size = 16, family = "Times", angle = 0),
  axis.title.y = element_blank(), axis.title.x = element_blank(),
  #legend.text = element_text(size = 14, family = "Times",  face = "italic"),   # run first with these
  #legend.title = element_blank(),                                              # to get the legend
  #legend.key = element_rect(size = 1, color = "white"),
  #legend.key.size = unit(0.5, "cm"),
  #legend.spacing.y = unit(2, "char"),
  legend.position = "none",                                                    # then with this
```

```
  panel.background = element_rect(fill = "#FFFDF9"),
  panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
  plot.title = element_text(size = 26, family = "Times", face = "bold")) +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01),
  breaks=seq(0, 0.1, 0.05)) + facet_grid(~country, scales = "free", space = "free") +
  theme(strip.text.x= element_text(size = 16,
  family = "Times", hjust = 0, vjust = 0.5, angle = 0, face = "bold"),
  strip.background = element_rect(colour = "white")) + guides(fill=guide_legend(ncol=1))

# Save legend
leg <- get_legend(rfc_plot)

# Convert to a ggplot and print
#as_ggplot(leg)

# Other than HWW
resfinder_PHY_Cluster_1 <- subset_taxa(resfinder_PHY, Cluster_name == "blaKPC-34_1_clust" |
    Cluster_name == "blaNDM-18_1_clust" | Cluster_name == "blaVIM-48_1_clust" |
    Cluster_name == "blaIMP-1_1_clust" | Cluster_name == "blaOXA-397_1_clust")

resfinder_PHY_Cluster_3 <- subset_taxa(resfinder_PHY, Gene == "blaGES-2_1_AF326355" |
    Gene == "blaGES-4_1_AB116723" | Gene == "blaGES-5_1_DQ236171" |
    Gene == "blaGES-6_1_AY494718" | Gene == "blaGES-14_1_GU207844" |
    Gene == "blaGES-16_1_HM173356" | Gene == "blaGES-18_1_JQ028729" |
    Gene == "blaGES-20_1_JN596280" | Gene == "blaOXA-48_1_AY236073" |
    Gene == "blaOXA-162_1_GU197550" | Gene == "blaOXA-181_1_CM004561" |
    Gene == "blaOXA-199_1_JN704570" | Gene == "blaOXA-204_1_KP027885" |
    Gene == "blaOXA-232_1_JX423831" | Gene == "blaOXA-244_1_KP659189" |
    Gene == "blaOXA-245_1_JX438001" | Gene == "blaOXA-247_1_JX893517" |
    Gene == "blaOXA-247_1_JX893517" | Gene == "blaOXA-514_1_KU866382" |
    Gene == "blaOXA-515_1_KU866383" | Gene == "blaOXA-517_1_KU878974") # blaOXA-48-like

resfinder_PHY_Cluster <- merge_phyloseq(resfinder_PHY_Cluster_1, resfinder_PHY_Cluster_3)
## Benin
# Feces
resfinder_PHY_feces <- subset_samples(resfinder_PHY_Cluster, alias == "BH20" | alias == "BH22" |
                                        alias == "BH24" | alias == "BH25")
df <- sample_sums(resfinder_PHY_feces)

names <- paste(resfinder_PHY_feces@sam_data$alias)

rfc <- plot_bar(resfinder_PHY_feces, fill = "Cluster_name")
rfc_plot1 <- rfc + geom_bar(stat="identity", color = NA, size = 0) +
  ggtitle("Benin") + scale_fill_manual(values = cols,
  labels = c("blaGES", "blaIMP", "blaKPC", "blaNDM", "blaOXA-48", "blaOXA-58", "blaVIM")) +
  labs(y = expression(atop(bold("ARGs/16S rRNA")))) + ggtitle("Benin") +
  scale_x_discrete(breaks=levels(factor(rownames(sample_data(resfinder_PHY_feces)))),
  labels=names, expression(bar("x"))) + theme_minimal() +
  theme(axis.text.x = element_text(size = 20, family = "Times", angle = 0, face = "bold"),
  axis.text.y = element_text(size = 16, family = "Times", angle = 0),
  axis.title.y = element_text(size = 24, family = "Times"),
  axis.title.x = element_blank(), legend.position = "none",
  panel.background = element_rect(fill = "#FFFDF9"),
  panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
  plot.title = element_text(size = 16, family = "Times", face = "bold")) +
  scale_y_continuous(limits = c(0, 0.002), labels = scales::number_format(accuracy = 0.001),
  breaks = seq(0, 0.002, by = 0.001)) + facet_grid(~plot_name,
  scales = "free", space = "free", labeller = label_wrap_gen(width = 30, multi_line = TRUE)) +
  theme(strip.text.x= element_text(size = 14, family = "Times", hjust = 0, vjust = 0.5, angle = 0),
  strip.background = element_rect(colour = "white")) + guides(fill=guide_legend(ncol=1))

# Drinking
resfinder_PHY_ben_drink <- subset_samples(resfinder_PHY_Cluster, alias == "BSE100" |
                                            alias == "BSE74" | alias == "BSE79"|
```

```
                                          alias == "BSE93"| alias ==  "BH11")
df <- sample_sums(resfinder_PHY_ben_drink)

names <- paste(resfinder_PHY_ben_drink@sam_data$alias)

rfc <- plot_bar(resfinder_PHY_ben_drink, fill = "Cluster_name")
rfc_plot2 <- rfc + geom_bar(stat="identity", color = NA, size = 0) +
  scale_fill_manual(values = cols,
  labels = c("blaGES", "blaIMP", "blaKPC", "blaNDM", "blaOXA-48", "blaOXA-58", "blaVIM")) +
  labs(y = expression(atop(bold("ARGs/16S rRNA")))) + ggtitle("Benin") +
  scale_x_discrete(breaks=levels(factor(rownames(sample_data(resfinder_PHY_ben_drink)))),
  labels=names, expression(bar("x"))) + theme_minimal() + theme(axis.text.x = element_text(size = 20,
  family = "Times", angle = 0, face = "bold"),
  axis.text.y = element_text(size = 16, family = "Times", angle = 0),
  axis.title.x = element_blank(), axis.title.y = element_blank(),
  legend.position = "none", panel.background = element_rect(fill = "#FFFDF9"),
  panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
  plot.title = element_text(size = 16, family = "Times", face = "bold")) +
  scale_y_continuous(limits = c(0, 0.002), labels = scales::number_format(accuracy = 0.001),
  breaks = seq(0, 0.002, by = 0.001)) + facet_grid(~plot_name, scales = "free", space = "free",
  labeller = label_wrap_gen(width = 20, multi_line = TRUE)) + theme(strip.text.x= element_text(size = 14,
  family = "Times", hjust = 0, vjust = 0.5, angle = 0),strip.background = element_rect(colour = "white")) +
  guides(fill=guide_legend(ncol=1))

# other
resfinder_PHY_ben_other <- subset_samples(resfinder_PHY_Cluster,
                                          alias ==  "BH13" | alias == "BH14"| alias == "BH32" | alias == "BH52")
df <- sample_sums(resfinder_PHY_ben_other)

names <- paste(resfinder_PHY_ben_other@sam_data$alias)

rfc <- plot_bar(resfinder_PHY_ben_other, fill = "Cluster_name")
rfc_plot3 <- rfc + geom_bar(stat="identity", color = NA, size = 0) + scale_fill_manual(values = cols,
  labels = c("blaGES", "blaIMP", "blaKPC", "blaNDM", "blaOXA-48", "blaOXA-58", "blaVIM")) +
  labs(y = expression(atop(bold("ARGs/16S rRNA")))) + ggtitle("Benin") +
  scale_x_discrete(breaks=levels(factor(rownames(sample_data(resfinder_PHY_ben_other)))),
  labels=names, expression(bar("x"))) + theme_minimal() + theme(axis.text.x = element_text(size = 20,
  family = "Times", angle = 0, face = "bold"),
  axis.text.y = element_text(size = 16, family = "Times", angle = 0),
  axis.title.y = element_blank(), axis.title.x = element_blank(),
  legend.position = "none", panel.background = element_rect(fill = "#FFFDF9"),
  panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
  plot.title = element_text(size = 16, family = "Times", face = "bold")) +
  scale_y_continuous(limits = c(0, 0.01), labels = scales::number_format(accuracy = 0.001),
  breaks = seq(0, 0.01, by = 0.005)) + facet_grid(~plot_name, scales = "free", space = "free",
  labeller = label_wrap_gen(width = 20, multi_line = TRUE)) +
  theme(strip.text.x= element_text(size = 14, family = "Times", hjust = 0, vjust = 0.5, angle = 0),
  strip.background = element_rect(colour = "white")) + guides(fill=guide_legend(ncol=1))

## Burkina Faso
# other
resfinder_PHY_BF_other <- subset_samples(resfinder_PHY_Cluster,
                                         alias == "BFH27" | alias ==  "BFH42" | alias == "BFH26")
df <- sample_sums(resfinder_PHY_BF_other)

names <- paste(resfinder_PHY_BF_other@sam_data$alias)

rfc <- plot_bar(resfinder_PHY_BF_other, fill = "Cluster_name")
rfc_plot4 <- rfc + geom_bar(stat="identity", size = 0, color = NA) + scale_fill_manual(values = cols,
  labels = c("blaGES", "blaIMP", "blaKPC", "blaNDM", "blaOXA-48", "blaOXA-58", "blaVIM")) +
  labs(y = expression(atop(bold("ARGs/16S rRNA")))) + ggtitle("Burkina Faso") +
  scale_x_discrete(breaks=levels(factor(rownames(sample_data(resfinder_PHY_BF_other)))),
  labels=names, expression(bar("x"))) + theme_minimal() + theme(axis.text.x = element_text(size = 20,
  family = "Times", angle = 0, face = "bold"),
```

```
  axis.text.y = element_text(size = 16, family = "Times", angle = 0),
  axis.title.y = element_blank(), axis.title.x = element_blank(),
  legend.position = "none", panel.background = element_rect(fill = "#FFFDF9"),
  panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
  plot.title = element_text(size = 16, family = "Times", face = "bold")) +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01),
  breaks=seq(0, 0.02, 0.01)) + facet_grid(~plot_name, scales = "free", space = "free",
  labeller = label_wrap_gen(width = 25, multi_line = TRUE)) + theme(strip.text.x= element_text(size = 11,
  family = "Times", hjust = 0, vjust = 0.5, angle = 0), strip.background = element_rect(colour = "white")) +
  guides(fill=guide_legend(ncol=1))

layout <- "
AAAAAA
AAAAAA
AAAAAA
BBBCCC
DDDEE#
"

p <- rfc_plot + rfc_plot1 + rfc_plot2 + rfc_plot3 + rfc_plot4 +
  plot_layout(design = layout) + plot_annotation(tag_levels = list(c("A", "B"))) &
  theme(plot.tag = element_text(size = 24, family = "Times"))

p_leg <- p + inset_element(leg, left = 1.65, bottom = 1, right = 1, top = 0)

#ggsave(filename = "carbapenemases_grid.png",
#       width = 16, height = 13, dpi = 300, units = "in", device='png', scale = 1)
```

15 most abundant taxa

```
# 15 most abundant taxa in hospital WW in each country
metaphlan_PHY_Ben <- subset_samples(metaphlan_PHY_stat, country == "Benin")
metaphlan_PHY_BF <- subset_samples(metaphlan_PHY_stat, country == "Burkina Faso")
metaphlan_PHY_Fin <- subset_samples(metaphlan_PHY_stat, country == "Finland")

# At genus level
metaphlan_PHY_Genus <- tax_glom(metaphlan_PHY_Ben, taxrank = "Genus")
metaphlan_PHY_Genus_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Genus),
    TRUE)[1:15]), metaphlan_PHY_Genus)
# tax_table(metaphlan_PHY_Genus_abund) At species level
metaphlan_PHY_Species <- tax_glom(metaphlan_PHY_Ben, taxrank = "Species")
metaphlan_PHY_Species_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Species),
    TRUE)[1:15]), metaphlan_PHY_Species)
# tax_table(metaphlan_PHY_Species_abund)

# At genus level
metaphlan_PHY_Genus <- tax_glom(metaphlan_PHY_BF, taxrank = "Genus")
metaphlan_PHY_Genus_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Genus),
    TRUE)[1:15]), metaphlan_PHY_Genus)
# tax_table(metaphlan_PHY_Genus_abund) At species level
metaphlan_PHY_Species <- tax_glom(metaphlan_PHY_BF, taxrank = "Species")
metaphlan_PHY_Species_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Species),
    TRUE)[1:15]), metaphlan_PHY_Species)
# tax_table(metaphlan_PHY_Species_abund)

# At genus level
metaphlan_PHY_Genus <- tax_glom(metaphlan_PHY_Fin, taxrank = "Genus")
metaphlan_PHY_Genus_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Genus),
    TRUE)[1:15]), metaphlan_PHY_Genus)
# tax_table(metaphlan_PHY_Genus_abund) At species level
metaphlan_PHY_Species <- tax_glom(metaphlan_PHY_Fin, taxrank = "Species")
metaphlan_PHY_Species_abund <- prune_taxa(names(sort(taxa_sums(metaphlan_PHY_Species),
```

```
    TRUE)[1:15]), metaphlan_PHY_Species)
# tax_table(metaphlan_PHY_Species_abund)
```

Correlation between MGE/intI1 & all ARGs

```
ARG_relative_sum <- data.frame(sample_sums(resfinder_PHY_stat))
MGE_relative_sum <- data.frame(sample_sums(MGE_PHY_stat))
intI1_relative_sum <- data.frame(sample_sums(MGE_PHY_int_stat))
all(rownames(ARG_relative_sum) == rownames(MGE_relative_sum))
all(rownames(ARG_relative_sum) == rownames(intI1_relative_sum))

## MGEs Join data
mge_res <- cbind(ARG_relative_sum, MGE_relative_sum)
colnames(mge_res) <- c("ARGs", "MGEs")

# Plot
cor <- ggplot(mge_res, aes(x = ARGs, y = MGEs)) + geom_point(size = 7, shape = 19,
    color = "#3110D2") + geom_smooth(method = "lm", se = TRUE, fullrange = FALSE,
    level = 0.95, color = "#FB2A38", fill = "#8A91F8") + theme_bw() + theme(axis.title = element_text(size = 30,
    family = "Times"), axis.text = element_text(size = 28, family = "Times"), plot.title = element_text(size = 36,
    family = "Times"), plot.subtitle = element_text(size = 28, family = "Times")) +
    xlab("ARG") + ylab("MGEs") + labs(title = "Correlation of relative sums of ARGs and MGEs",
    )
cor2 <- cor + stat_cor(method = "pearson", label.x = 2, label.y = 1.5)

# ggsave(filename = 'ARG_MGE_cor_new.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)

## Intl1 Join data
intl_res <- cbind(ARG_relative_sum, intI1_relative_sum)
colnames(intl_res) <- c("ARGs", "intI1")

# Plot
cor <- ggplot(intl_res, aes(x = ARGs, y = intI1)) + geom_point(size = 7, shape = 19,
    color = "#3110D2") + geom_smooth(method = "lm", se = TRUE, fullrange = FALSE,
    level = 0.95, color = "#FB2A38", fill = "#8A91F8") + theme_bw() + theme(axis.title = element_text(size = 30,
    family = "Times"), axis.text = element_text(size = 28, family = "Times"), plot.title = element_text(size = 36,
    family = "Times"), plot.subtitle = element_text(size = 28, family = "Times")) +
    xlab("ARG") + ylab("intI1") + labs(title = "Correlation of relative sums of ARGs and Int1",
    subtitle = "Hospital WWs in Benin, Burkina Faso and Finland")
cor2 <- cor + stat_cor(method = "pearson", label.x = 1, label.y = 1.5)

# ggsave(filename = 'ARG_intl1_cor_new.png', width = 16, height = 13, dpi =
# 300, units = 'in', device='png', scale = 1)
```

Save correlation data for intI & qacEdelta and all ARGs

```
# intI1
tax <- data.frame(clusters_tax_table_resfinder)
tax$n <- rownames(tax)
tax$sp <- rep("sp", times = 3104)
rownames(tax) <- paste(tax$sp, tax$n, sep = "")
tax <- tax[c(-4, -5)]

args <- resfinder_PHY_stat
int <- MGE_PHY_int_stat

arg_matrix <- as.data.frame(otu_table(args))
arg_matrix$n <- rownames(arg_matrix)
```

```r
arg_matrix$sp <- rep("sp", times = 3104)
rownames(arg_matrix) <- paste(arg_matrix$sp, arg_matrix$n, sep = "")
arg_matrix <- arg_matrix[c(-68, -69)]

arg_matrix <- arg_matrix[which(rowSums(arg_matrix) > 0), ]

match <- match(rownames(arg_matrix), rownames(tax))
arg_tax <- tax[match, ]

rownames(arg_matrix) <- arg_tax$Gene

int_matrix <- data.frame(sample_sums(otu_table(int)))
arg_matrix <- t(arg_matrix)

correl <- corr.test(arg_matrix, int_matrix, use = "pairwise", method = "pearson",
    adjust = "fdr", alpha = 0.05, ci = TRUE)

r <- data.frame(correl$r)
p <- data.frame(correl$p)
p.ad <- data.frame(correl$p.adj)

cor_data <- data.frame(r, p, p.ad)
cor_data$Gene <- rownames(cor_data)
colnames(cor_data) <- c("r", "p", "p.ad", "Gene")

cor_data_filt <- cor_data[which(cor_data$p < 0.05), ]

pos_all <- cor_data_filt[which(cor_data_filt$r > 0), ]
neg_all <- cor_data_filt[which(cor_data_filt$r < 0), ]


# write.table(pos_all,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/pos_all.txt', row.names=F,
# sep = '\t', col.names = T)

# write.table(neg_all,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/neg_all.txt', row.names=F,
# sep = '\t', col.names = T)

# qacEdelta
tax <- data.frame(clusters_tax_table_resfinder)
tax$n <- rownames(tax)
tax$sp <- rep("sp", times = 3104)
rownames(tax) <- paste(tax$sp, tax$n, sep = "")
tax <- tax[c(-4, -5)]

args <- resfinder_PHY_stat
qac <- MGE_PHY_qac_stat

arg_matrix <- as.data.frame(otu_table(args))
arg_matrix$n <- rownames(arg_matrix)
arg_matrix$sp <- rep("sp", times = 3104)
rownames(arg_matrix) <- paste(arg_matrix$sp, arg_matrix$n, sep = "")
arg_matrix <- arg_matrix[c(-68, -69)]

arg_matrix <- arg_matrix[which(rowSums(arg_matrix) > 0), ]

match <- match(rownames(arg_matrix), rownames(tax))
arg_tax <- tax[match, ]

rownames(arg_matrix) <- arg_tax$Gene

qac_matrix <- data.frame(sample_sums(otu_table(qac)))
arg_matrix <- t(arg_matrix)
```

```
correl <- corr.test(arg_matrix, qac_matrix, use = "pairwise", method = "pearson",
    adjust = "fdr", alpha = 0.05, ci = TRUE)

r <- data.frame(correl$r)
p <- data.frame(correl$p)
p.ad <- data.frame(correl$p.adj)

cor_data <- data.frame(r, p, p.ad)
cor_data$Gene <- rownames(cor_data)
colnames(cor_data) <- c("r", "p", "p.ad", "Gene")

cor_data_filt <- cor_data[which(cor_data$p < 0.05), ]

pos_all <- cor_data_filt[which(cor_data_filt$r > 0), ]
neg_all <- cor_data_filt[which(cor_data_filt$r < 0), ]


# write.table(pos_all,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/pos_all.txt', row.names=F,
# sep = '\t', col.names = T)

# write.table(neg_all,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/neg_all.txt', row.names=F,
# sep = '\t', col.names = T)
```

Figures for correlations for differentially abundant ARGs across countries (from DESeq2) & intI1/qacEdelta

```
intI1 <- data.frame(sample_sums(MGE_PHY_int_stat))
colnames(intI1) <- c("intI1")
qacEdelta <- data.frame(sample_sums(MGE_PHY_qac_stat))
colnames(qacEdelta) <- c("qacEdelta")

# DESeq2: Fin-Ben Benin
BenFin20 <- Ben_Fin[1:20, ]

pattern_Ben_Fin <- as.matrix(BenFin20$Row.names)

args <- data.frame(otu_table(resfinder_PHY_stat))
arg_data <- args[pattern_Ben_Fin, ]
all(rownames(arg_data) == BenFin20$Row.names)

rownames(arg_data) <- BenFin20$Gene
# shorten gene names
rownames(arg_data) <- gsub(pattern = "_[A-Z].*", replacement = "", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "-", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\(", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\)", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\'", replacement = "", rownames(arg_data))
rownames(arg_data) <- c("lnu_F_3", "qnrVC4", "qnrVC5", "aac_6_IIc", "blaCARB_2",
    "ant_2_Ia_6", "blaOXA_129", "dfrA22", "blaVEB_1_3", "blaAER_1", "ant_2_Ia_10",
    "aph_2_Id", "blaVEB_1_1", "catQ_1", "blaVEB_5", "blaCARB_11", "blaCARB_1", "sul3_2",
    "dfrA15", "cmlA1")

arg_data = t(arg_data)

df <- data.frame(arg_data, intI1, qacEdelta)

par(family = "Times New Roman", cex = 1.5)
cor <- rcorr(as.matrix(df))
M <- cor$r
p_mat <- cor$P
```

```
M1 <- M[, -c(1:20)]
M1 <- M1[-c(21:22), ]
p_mat1 <- p_mat[, -c(1:20)]
p_mat1 <- p_mat1[-c(21:22), ]

# Finland
FinBen20 <- Fin_Ben[1:20, ]
pattern_Fin_Ben <- as.matrix(FinBen20$Row.names)

args <- data.frame(otu_table(resfinder_PHY_stat))
arg_data <- args[pattern_Fin_Ben, ]
all(rownames(arg_data) == FinBen20$Row.names)

rownames(arg_data) <- FinBen20$Gene
# shorten gene names
rownames(arg_data) <- gsub(pattern = "_[A-Z].*", replacement = "", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "-", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\(", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\)", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\'", replacement = "", rownames(arg_data))
rownames(arg_data) <- c("blaOXA_211", "blaOXA_299", "blaOXA_212", "blaOXA_334", "aac_6_Ig",
    "blaOXA_373", "blaOXA_296", "blaOXA_333", "blaOXA_309", "blaOXA_427", "dfrA3",
    "VanHOX_1", "blaOXA_281", "blaOXA_280", "cphA1", "VanHAX_1", "cphA2", "blaMOX_3",
    "VanHBX_1", "tet_39")

arg_data = t(arg_data)

df <- data.frame(arg_data, intI1, qacEdelta)

par(family = "Times New Roman", cex = 1.5)
cor <- rcorr(as.matrix(df))
M <- cor$r
p_mat <- cor$P
M2 <- M[, -c(1:20)]
M2 <- M2[-c(21:22), ]
p_mat2 <- p_mat[, -c(1:20)]
p_mat2 <- p_mat2[-c(21:22), ]

# DESeq2: Fin-BF BF
BFFin20 <- BF_Fin[1:20, ]

pattern_BF_Fin <- as.matrix(BFFin20$Row.names)

args <- data.frame(otu_table(resfinder_PHY_stat))
arg_data <- args[pattern_BF_Fin, ]
all(rownames(arg_data) == BFFin20$Row.names)

rownames(arg_data) <- BFFin20$Gene
# shorten gene names
rownames(arg_data) <- gsub(pattern = "_[A-Z].*", replacement = "", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "-", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\(", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\)", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\'", replacement = "", rownames(arg_data))
rownames(arg_data) <- c("dfrB5", "blaCMY_4", "sul4", "dfrA15_2", "blaOXA_46", "blaOXA_101",
    "dfrA15_1", "blaOXA_7", "qnrVC1", "lnu_F_3", "nimA_1", "blaVIM_5", "dfrA15_4",
    "blaOXA_56", "catQ_1", "blaVIM_38", "blaCMY_130", "blaCMY_59", "qnrVC4", "blaVIM_25")

arg_data = t(arg_data)

df <- data.frame(arg_data, intI1, qacEdelta)

cor <- rcorr(as.matrix(df))
M <- cor$r
```

```
p_mat <- cor$P
M3 <- M[, -c(1:20)]
M3 <- M3[-c(21:22), ]
p_mat3 <- p_mat[, -c(1:20)]
p_mat3 <- p_mat3[-c(21:22), ]

# Finland
FinBF20 <- Fin_BF[1:20, ]
pattern_Fin_BF <- as.matrix(FinBF20$Row.names)

args <- data.frame(otu_table(resfinder_PHY_stat))
arg_data <- args[pattern_Fin_BF, ]
all(rownames(arg_data) == FinBF20$Row.names)

rownames(arg_data) <- FinBF20$Gene
# shorten gene names
rownames(arg_data) <- gsub(pattern = "_[A-Z].*", replacement = "", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "-", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\(", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\)", replacement = "_", rownames(arg_data))
rownames(arg_data) <- gsub(pattern = "\\'", replacement = "", rownames(arg_data))
rownames(arg_data) <- c("blaOXA_299", "blaOXA_334", "blaOXA_296", "blaOXA_333", "blaOXA_211",
    "aac_6_Ig", "blaOXA_281", "blaOXA_373", "blaOXA_212", "blaOXA_309", "VanHOX_1",
    "VanHAX_2", "VanHBX_1", "blaOXA_275", "aadA11", "VanHAX_1", "qnrB21", "VanC4XY_1",
    "mef_A_3", "cphA2")

arg_data = t(arg_data)

df <- data.frame(arg_data, intI1, qacEdelta)

cor <- rcorr(as.matrix(df))
M <- cor$r
p_mat <- cor$P
M4 <- M[, -c(1:20)]
M4 <- M4[-c(21:22), ]
p_mat4 <- p_mat[, -c(1:20)]
p_mat4 <- p_mat4[-c(21:22), ]

# Plot with ggcorrplot For the legend
p_mat1[is.na(p_mat1)] = 0
p_mat2[is.na(p_mat2)] = 0
p_mat3[is.na(p_mat3)] = 0
p_mat4[is.na(p_mat4)] = 0

m0 <- ggcorrplot(M1, p.mat = p_mat1, type = "full", insig = "blank", method = "square",
    ggtheme = ggplot2::theme_classic() + theme(axis.text = element_text(face = "italic",
        family = "Times", size = 9, angle = 20), plot.title = element_text(size = 9,
        face = "bold", family = "Times"), legend.title = element_blank(), legend.text = element_text(family = "Times
        size = 20), legend.key.size = unit(1.4, "cm")))

title1 <- ggdraw() + draw_label("Differentially abundant ARGs in HWWs from Benin vs. Finland",
    fontface = "bold", x = 0.32, hjust = 0.1, y = 0.35, fontfamily = "Times", size = 24)
title2 <- ggdraw() + draw_label("Differentially abundant ARGs in HWWs from Burkina Faso vs. Finland",
    fontface = "bold", x = 0.32, hjust = 0.1, y = 0.35, fontfamily = "Times", size = 24)

m1 <- ggcorrplot(M1, p.mat = p_mat1, type = "full", insig = "blank", method = "square",
    ggtheme = ggplot2::theme_classic() + theme(axis.text = element_text(face = "italic",
        family = "Times"), legend.position = "none", plot.margin = unit(c(0, 0, 0,
        0), "cm"), axis.text.y.left = element_text(angle = 0, face = "bold.italic",
        size = 20), axis.text.x.bottom = element_text(size = 16, angle = 35, face = "italic"),
        plot.title = element_text(size = 24, family = "Times"))) + ggtitle("Benin")

m2 <- ggcorrplot(M2, p.mat = p_mat2, type = "full", insig = "blank", method = "square",
    ggtheme = ggplot2::theme_classic() + theme(axis.text = element_text(face = "italic",
```

```
        family = "Times"), legend.position = "none", plot.margin = unit(c(0, 0, 0,
        0), "cm"), axis.text.y.left = element_text(angle = 0, face = "bold.italic",
        size = 20), axis.text.x.bottom = element_text(size = 16, angle = 35, face = "italic"),
        plot.title = element_text(size = 24, family = "Times"))) + ggtitle("Finland")

m3 <- ggcorrplot(M3, p.mat = p_mat3, type = "full", insig = "blank", method = "square",
    ggtheme = ggplot2::theme_classic() + theme(axis.text = element_text(face = "italic",
        family = "Times"), legend.position = "none", plot.margin = unit(c(0, 0, 0,
        0), "cm"), axis.text.y.left = element_text(angle = 0, face = "bold.italic",
        size = 20), axis.text.x.bottom = element_text(size = 16, angle = 35, face = "italic"),
        plot.title = element_text(size = 24, family = "Times"))) + ggtitle("Burkina Faso")

m4 <- ggcorrplot(M4, p.mat = p_mat4, type = "full", insig = "blank", method = "square",
    ggtheme = ggplot2::theme_classic() + theme(axis.text = element_text(face = "italic",
        family = "Times"), legend.position = "none", plot.margin = unit(c(0, 0, 0,
        0), "cm"), axis.text.y.left = element_text(angle = 0, face = "bold.italic",
        size = 20), axis.text.x.bottom = element_text(size = 16, angle = 35, face = "italic"),
        plot.title = element_text(size = 24, family = "Times"))) + ggtitle("Finland")

# Extract the legend from one of the plots legend <- get_legend(m0)

# Some inception with cowplot...
A <- plot_grid(title1, m1, m2, NULL, ncol = 1, rel_heights = c(0.5, 1, 1, 0.1))
B <- plot_grid(NULL, title2, m3, m4, ncol = 1, rel_heights = c(0.1, 0.5, 1, 1))
AB <- plot_grid(A, B, ncol = 1)

# ggsave(filename = 'ARGs_corr_deseq.png', width = 16, height = 13, dpi = 300,
# units = 'in', device='png', scale = 1)
```

"Core" resistome and unique ARGs

```
Ben_temp <- otu_table(subset_samples(resfinder_PHY_stat, country %in% c("Benin")))[rowSums(otu_table(subset_samples(
    country %in% c("Benin")))) > 0]
nrow(Ben_temp)  # 1738

BF_temp <- otu_table(subset_samples(resfinder_PHY_stat, country %in% c("Burkina Faso")))[rowSums(otu_table(subset_sa
    country %in% c("Burkina Faso")))) > 0]
nrow(BF_temp)  # 2131

Fin_temp <- otu_table(subset_samples(resfinder_PHY_stat, country %in% c("Finland")))[rowSums(otu_table(subset_sample
    country %in% c("Finland")))) > 0]
nrow(Fin_temp)  # 1555

length(intersect(row.names(Ben_temp), (row.names(BF_temp))))  # 1664
length(intersect(row.names(BF_temp), (row.names(Fin_temp))))  # 1414
length(intersect(row.names(Ben_temp), (row.names(Fin_temp))))  # 1295

# grid.newpage() ven.p <- draw.triple.venn(area1 = nrow(Ben_temp), area2 =
# nrow(BF_temp), area3 = nrow(Fin_temp), n12 =
# length(intersect(row.names(Ben_temp), (row.names(BF_temp)))), n23 =
# length(intersect(row.names(BF_temp), (row.names(Fin_temp)))), n13 =
# length(intersect(row.names(Ben_temp), (row.names(Fin_temp)))), n123 =
# length(intersect(intersect(row.names(Ben_temp), (row.names(BF_temp))),
# row.names(Fin_temp))), fontfamily = 'Times', category = c('Benin', 'Burkina
# Faso', 'Finland'), lty = 'blank', fill = c('#B2182B', '#44AA99', '#2166AC'),
# alpha = 0.75, cex = 4.5, cat.cex = 6, rotation.degree = 0, label.col =
# 'white', cat.dist = 0.05, filename = 'Venn_diagram.png', output=TRUE,
# imagetype='png', margin = 0.08) grid.draw(ven.p)

# And which ARGs are those?
tax <- data.frame(clusters_tax_table_resfinder)
```

```
tax$n <- rep(1:3104, each = 1)
colnames(tax) <- c("Class", "Cluster_name", "Gene", "n")
rownames(tax) <- paste(tax$n, sep = "")
tax <- tax[c(-4)]

match <- match(rownames(Ben_temp), rownames(tax))
Ben_names <- tax[match, ]

match <- match(rownames(BF_temp), rownames(tax))
BF_names <- tax[match, ]

match <- match(rownames(Fin_temp), rownames(tax))
Fin_names <- tax[match, ]

# write.table(Ben_names,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/counts_Ben.txt', row.names=F,
# sep = '\t', col.names = T)

# write.table(BF_names,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/counts_BF.txt', row.names=F,
# sep = '\t', col.names = T)

# write.table(Fin_names,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/counts_Fin.txt', row.names=F,
# sep = '\t', col.names = T)

# What about the unique ARGs?  Core
counts <- data.frame(otu_table(resfinder_PHY_stat))
counts[counts > 0] <- 1
core <- counts[rowSums(counts) == 67, ]

tax <- data.frame(clusters_tax_table_resfinder)
tax$n <- rep(1:3104, each = 1)
colnames(tax) <- c("Class", "Cluster_name", "Gene", "n")
rownames(tax) <- paste(tax$n, sep = "")
tax <- tax[c(-4)]

match <- match(rownames(core), rownames(tax))
core_names <- tax[match, ]

# write.table(core_names,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/core_names.txt', row.names=F,
# sep = '\t', col.names = T)

# Unique for Benin
temp1 <- intersect(row.names(Ben_temp), row.names(Fin_temp))
temp2 <- intersect(row.names(Ben_temp), row.names(BF_temp))
temp <- c(temp1, temp2)
temp <- data.frame(temp)
temp <- data.frame(unique(temp))
rownames(temp) <- temp$temp

unique_Ben <- data.frame(names = outersect(rownames(temp), rownames(Ben_temp)))
rownames(unique_Ben) <- unique_Ben$names

match <- match(rownames(unique_Ben), rownames(tax))
unique_Ben <- tax[match, ]

# write.table(unique_Ben,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/unique_Ben.txt', row.names=F,
# sep = '\t', col.names = T)

# Unique for Burkina Faso
temp1 <- intersect(row.names(BF_temp), row.names(Fin_temp))
```

```
temp2 <- intersect(row.names(BF_temp), row.names(Ben_temp))
temp <- c(temp1, temp2)
temp <- data.frame(temp)
temp <- data.frame(unique(temp))
rownames(temp) <- temp$temp

unique_BF <- data.frame(names = outersect(rownames(temp), rownames(BF_temp)))
rownames(unique_BF) <- unique_BF$names

match <- match(rownames(unique_BF), rownames(tax))
unique_BF <- tax[match, ]

# write.table(unique_BF,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/unique_BF.txt', row.names=F,
# sep = '\t', col.names = T)

# Unique for Finland
temp1 <- intersect(row.names(Fin_temp), row.names(BF_temp))
temp2 <- intersect(row.names(Fin_temp), row.names(Ben_temp))
temp <- c(temp1, temp2)
temp <- data.frame(temp)
temp <- data.frame(unique(temp))
rownames(temp) <- temp$temp

unique_Fin <- data.frame(names = outersect(rownames(temp), rownames(Fin_temp)))
rownames(unique_Fin) <- unique_Fin$names

match <- match(rownames(unique_Fin), rownames(tax))
unique_Fin <- tax[match, ]

# write.table(unique_Fin,
# '~/Documents/Metagenomes_AMRIWA/R/AMRIWA/RFiles/unique_Fin.txt', row.names=F,
# sep = '\t', col.names = T)
```