Host-plasmid network analysis in R

Alice Risely

23/10/2023

Contents

Information

Below is the R code for the data analysis associated with the manuscript:

Host- plasmid network structure in wastewater is linked to antimicrobial resistance genes

by Alice Risely, Arthur Newbury, Thibault Stalder, Benno I. Simmons, Eva M. Top, Angus Buckling, Dirk Sanders

Load packages

```
library(phyloseq)
library(ape)
library(phangorn)
library(phylosignal)
library(bipartite )
library(bipartiteD3)
library(reshape2)
library(expss)
library(ggsci)
library(tidyverse)
library(metagMisc)
library(igraph)
library(network)
library(intergraph)
library(scales)
library(qgraph)
library(ggpubr)
library(gridExtra)
library(jntools)
library(ggtree)
library(ggplotify)
library(gtable)
library(grid)
library(RColorBrewer)
library(forcats)
library(ggridges)
library(viridis)
library(here)
library(ggrepel)
library(ggstatsplot)
library(ggthemes)
library(ggnetwork)
library(performance)
library(sjPlot)
library(picante)
```

Import data

```
plasmid_ps_clustered<-readRDS( "C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnal

phylo_merged<-plasmid_ps_clustered  # rename
```

Bacteria-Plasmid association matrix and metadata

```
#import tree

tree_phylophlan<-read.tree("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis

## root tree

tree_phylophlan_rooted<- phangorn::midpoint (tree_phylophlan) # root tree with chloroflexi MAG
```

Phylogenetic tree

```
taxonomy<-read.csv("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Update
```

Bacterial taxonomic data

```
### plasmid & AMR metadata ##
### plasmid & AMR metadata ##



cluster_df<-readRDS("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Updat

DT::datatable(cluster_df)
```

Show 10 entries Search:

	name	ClusterMembership	Resistance	Gene	Resistance_cl
165	k141_526738	1	false		true
177	k141_1145966	1	false		true
180	k141_1856898	1	false		true
187	k141_971894	1	false		true
199	k141_190742	1	false		true
202	k141_973004	1	false		true
211	k141_515203	1	true	tet(Q)	true
2031	k141_418614	1	true	tet(Q)	true
2251	k141_1009467	1	false		true
2261	k141_1484400	1	false		true

Showing 1 to 10 of 379 entries Previous 1 2 3 4 5 … 38

Plasmid cluster metadata

```
#head(cluster_df)
```

```r
resistance_clusters2<-subset(cluster_df, Resistance==T)

resistance_clusters<-unique(resistance_clusters2$Cluster)

resistance_clusters3<-distinct(resistance_clusters2, ClusterMembership, Gene)


resistance_clusters4<-resistance_clusters3 %>%
  pivot_wider(ClusterMembership,
              names_from = Gene,
              values_from = Gene,
              values_fn = list(Gene = length),
              values_fill = list(Gene = 0))

DT::datatable(resistance_clusters4)
```

Show 10 entries Search:

	ClusterMembership	tet(Q)	blaAER	aph(6)-Id	msr(E)	aadA27	aadS	qnrS2	qacEdelta1	tet(A)	blaMC
1	1	1	0	0	0	0	0	0	0	0	
2	2	0	1	1	0	0	0	0	0	0	
3	3	0	0	0	1	0	0	0	0	0	
4	4	0	0	0	1	1	0	0	0	0	
5	5	0	0	0	0	0	1	0	0	0	
6	6	0	0	0	1	0	0	0	0	0	
7	8	0	0	0	1	1	0	0	0	0	
8	9	0	0	0	1	0	0	0	0	0	
9	12	0	0	0	1	0	0	0	0	0	
10	14	0	0	0	1	0	0	0	0	0	

Showing 1 to 10 of 32 entries Previous 1 2 3 4 Next

```
### lightly filter - a few MAGs don't appear in the phylogenetic tree so filter these out

phylo_filtered<-prune_samples(tree_phylophlan_rooted$tip.label, phylo_merged)

phylo_filtered
```

Filter

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 109 taxa and 374 samples ]
## sample_data() Sample Data:       [ 374 samples by 15 sample variables ]
## tax_table()   Taxonomy Table:    [ 109 taxa by 2 taxonomic ranks ]
```

Generate network statistics

```
net.v<-data.frame(phylo_filtered@otu_table@.Data)
  # this will estimate all common network metrics but may take a while
  network_metrics<-data.frame(networklevel(net.v, index = c( "NODF", "number of compartments")))
names(network_metrics)[1]<-"Stat"

## no AMR plasmids

ps_noAMR<-subset_taxa(phylo_filtered, ta2 == FALSE)
ps_noAMR<-prune_samples(sample_sums(ps_noAMR)>0, ps_noAMR)
net.noAMR<-data.frame(ps_noAMR@otu_table@.Data)

network_metrics_noAMR<-data.frame(networklevel(net.noAMR, index = c("NODF", "number of compartments")))
 names(network_metrics_noAMR)[1]<-"Stat"

## just AMR plasmids


ps_AMR<-subset_taxa(phylo_filtered, ta2 == TRUE)
ps_AMR<-prune_samples(sample_sums(ps_AMR)>0, ps_AMR)
net.AMR<-data.frame(ps_AMR@otu_table@.Data)
network_metrics_AMR<-data.frame(networklevel(net.AMR, index = c( "NODF", "number of compartments")))
names(network_metrics_AMR)[1]<-"Stat"


network_metrics$Stat<-round(network_metrics$Stat, 2)
network_metrics$Measure<-row.names(network_metrics)
network_metrics<-data.table(network_metrics)
network_metrics<-network_metrics[,c(2,1)]
network_metrics$Measure<-c("# Compartments", "Nestedness")

network_metrics_noAMR$Stat<-round(network_metrics_noAMR$Stat, 2)
network_metrics_noAMR$Measure<-row.names(network_metrics_noAMR)
network_metrics_noAMR<-data.table(network_metrics_noAMR)
network_metrics_noAMR<-network_metrics_noAMR[,c(2,1)]
```

```
network_metrics_noAMR$Measure<-c("# Compartments", "Nestedness")

network_metrics_AMR$Stat<-round(network_metrics_AMR$Stat, 2)
network_metrics_AMR$Measure<-row.names(network_metrics_AMR)
network_metrics_AMR<-data.table(network_metrics_AMR)
network_metrics_AMR<-network_metrics_AMR[,c(2,1)]
network_metrics_AMR$Measure<-c("# Compartments", "Nestedness")
```

Fig 2a: Full network

```
mypal1<-brewer.pal(12,"Paired")
mypal2<-brewer.pal(12,"Dark2")
mypal3<-c(mypal1, mypal2)
net.v$Plasmid<-row.names(net.v)


net_long<-reshape2::melt(net.v, id.vars=c("Plasmid"))
#head(net_long)
names(net_long)[2]<-"Bacteria"
net_long<-subset(net_long, value>0)

# generate network object

igraph_net <- igraph::graph.data.frame(net_long[,c('Bacteria','Plasmid')])


E(igraph_net)$weight<-net_long$value
igraph_net<-as.undirected(igraph_net)
V(igraph_net)$degree<-igraph::degree(igraph_net)


### network modularity
wtc <- cluster_walktrap(igraph_net)
modularity(wtc)
```

```
## [1] 0.7713455
```

```
modularity(igraph_net, membership(wtc))
```

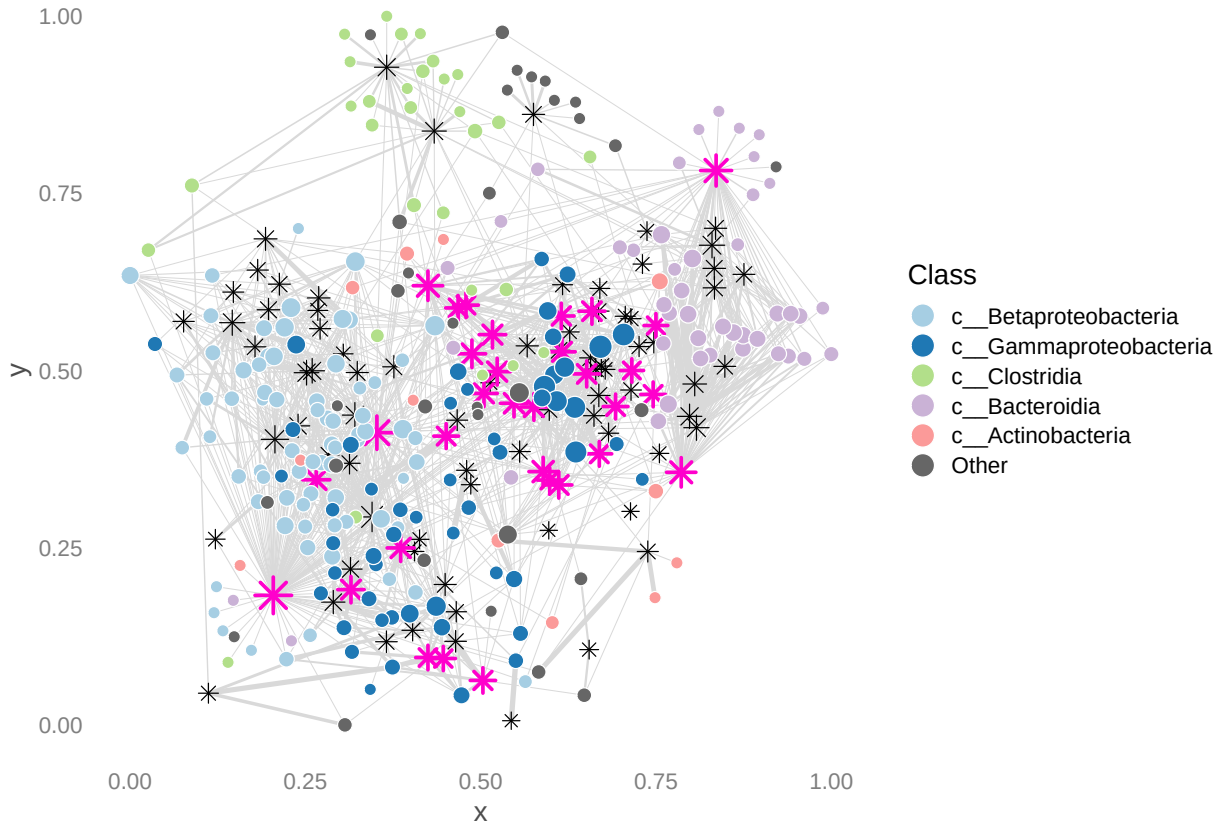
```
## [1] 0.4697711
```

```
wt <- fastgreedy.community(igraph_net)
modularity(wt)
```

```
## [1] 0.7887126
```

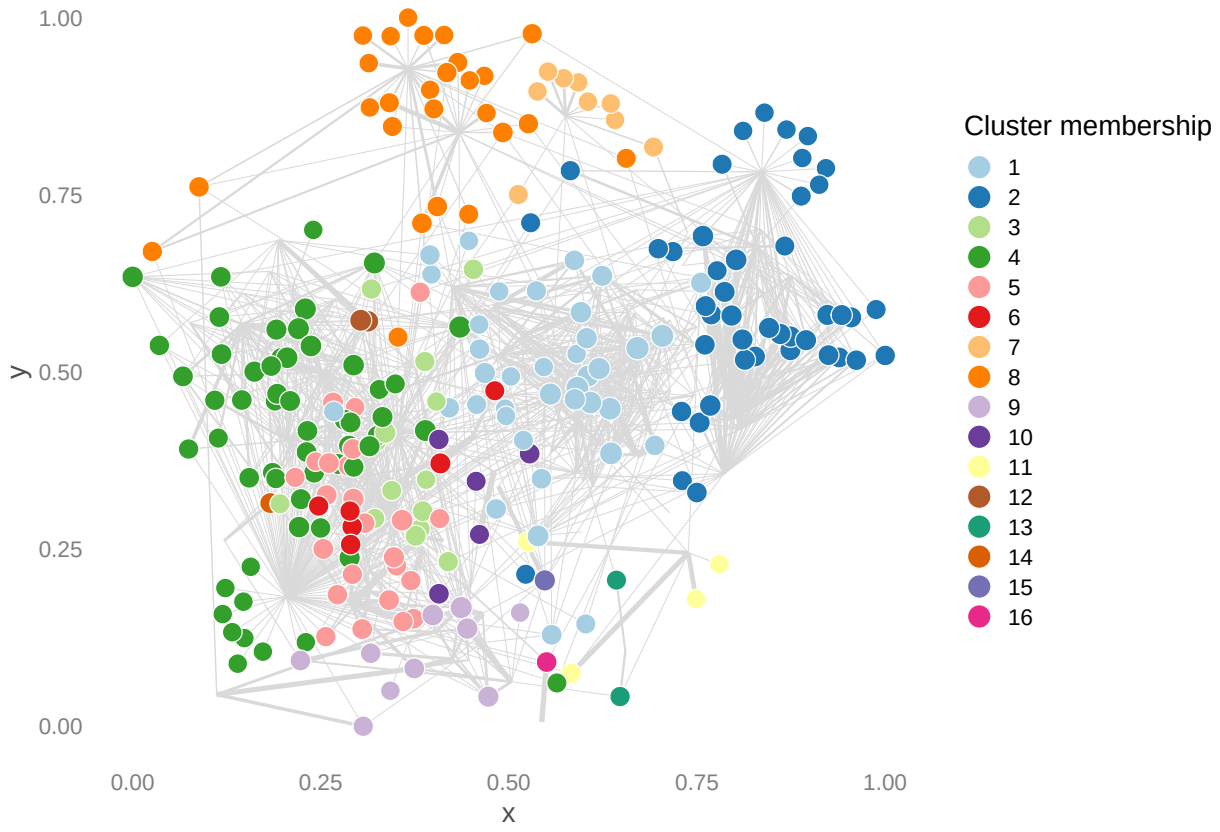
```
cluster.membership<-membership(wt)

# add cluster to graph metadata
V(igraph_net)$ClusterMembership<-as.factor(membership(wt))


network_metrics<-data.frame(network_metrics)
network_metrics[3,]<-NA
network_metrics[3,1]<-"Modularity"
network_metrics[3,2]<-modularity(igraph_net, membership(wtc))
network_metrics$Stat<-round(network_metrics$Stat, 2)

## convert to ggnetwork object

layout_dh <- layout_with_dh(igraph_net,  weight.edge.lengths = edge_density(igraph_net)/1)


gg_net<-ggnetwork(igraph_net, layout = layout_dh)


# populate metadata
plasmid_names<-taxa_names(phylo_filtered)
gg_net$Identity<-ifelse(gg_net$name %in% plasmid_names, "PLASMID", "BACTERIA")

# bacterial taxonomy
gg_net$Class <-expss::vlookup(gg_net$name, taxonomy, lookup_column = "Bin", result_column = "Class")
gg_net$Phylum <-expss::vlookup(gg_net$name, taxonomy, lookup_column = "Bin", result_column = "Phylum")


### format class variable

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
gg_net$Class_plot<- ifelse(gg_net$Class %in% classes_to_keep, as.character(gg_net$Class), "Other")
gg_net$Class_plot<-factor(gg_net$Class_plot, levels = c(classes_to_keep, "Other"))

## format phylum variable for plotting

phyla_to_keep<-c("p__Proteobacteria", "p__Firmicutes", "p__Bacteroidetes", "p__Actinobacteria")
gg_net$Phylum_plot<-ifelse(gg_net$Phylum %in% phyla_to_keep, as.character(gg_net$Phylum), "Other")
gg_net$Phylum_plot<-factor(gg_net$Phylum_plot, levels = c(phyla_to_keep, "Other"))

# plasmid resistance

gg_net$Resistance<-gg_net$name %in% resistance_clusters
#head(gg_net)

### code for plasmid

gg_net$Phylum <-ifelse(is.na(gg_net$Phylum), "Plasmid", as.character(gg_net$Phylum))


gg_net$ClusterMembership<-factor(gg_net$ClusterMembership)
```

```r
# coloured by class

gg_net<-merge(gg_net, resistance_clusters3, by.x = "name", by.y = "ClusterMembership", all.x = T)

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

class_network<-ggplot(gg_net, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 16)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net,Phylum == "Plasmid"), pch = 8, col = "black", aes(size = degree))+

  geom_nodes(data = subset(gg_net, Resistance == TRUE), pch = 8, stroke = 1.5, col = "#ff00cc", aes(siz

  scale_size(range = c(3,6))+

  guides(size = "none")+

  labs(fill = "Class")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), pch = 1
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 6)))+
  guides(size = "none")+
  scale_size(range = c(3,6))


class_network
```




```
# coloured by network cluster ###
# coloured by network cluster ###
# coloured by network cluster ###
# coloured by network cluster ###


ggplot(gg_net, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 16)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net,Phylum == "Plasmid"), pch = 8, col = "black", alpha = 0, aes(size = d

  geom_nodes(data = subset(gg_net, Resistance == TRUE), pch = 8, stroke = 1.5,alpha = 0, col = "#ff00cc

  scale_size(range = c(3,6))+

  guides(size = "none")+

  labs(fill = "Cluster membership")+
```

```
#bacterial nodes
ggnewscale::new_scale("size")+

geom_nodes(data = subset(gg_net, Phylum != "Plasmid"), aes(fill = ClusterMembership, size = degree ),
scale_fill_manual(values = mypal3)+
guides(fill= guide_legend(ncol=1, override.aes = list(size = 6)))+
guides(size = "none")+
scale_size(range = c(5,6))
```




```
## clusters

full_network_clusters<-gg_net[,c(1,5,9, 14)]

full_network_clusters<-distinct(full_network_clusters, name, .keep_all = T)

dim(full_network_clusters)
```

```
## [1] 352   4
```

```
names(full_network_clusters)
```

```
## [1] "name"              "ClusterMembership" "Identity"
## [4] "Resistance"
```

```
bacterial_clusters<-full_network_clusters

bacterial_clusters<-merge(bacterial_clusters, taxonomy, by.x = "name", by.y = "Bin", all.x = T)

df<-subset(bacterial_clusters, Phylum == "p__Proteobacteria" | Resistance == TRUE)

#head(df)

df2<-data.frame(table(df$ClusterMembership, df$Identity))


table(subset(bacterial_clusters, Phylum == "p__Proteobacteria")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
## 23  3 10 50 21  6  0  0  9  5  0  2  0  1  1  1
```

```
table(subset(bacterial_clusters, Identity=="PLASMID")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
## 46 12  3 19  6  2  2  2  9  2  1  1  1  1  1  1
```

```
table(subset(bacterial_clusters, Class == "c__Gammaproteobacteria")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
## 19  2  3  4 10  3  0  0  7  3  0  0  0  0  1  1
```

```
table(subset(bacterial_clusters, Class == "c__Betaproteobacteria")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  1  0  5 46 11  3  0  0  1  2  0  2  0  1  0  0
```

```
table(subset(bacterial_clusters, Class == "c__Clostridia")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  5  0  1  1  0  0  0 23  0  0  0  0  0  0  0  0
```

```
table(subset(bacterial_clusters, Class == "c__Bacteroidia")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  2 39  1  2  0  0  0  0  0  0  0  0  0  0  0  0
```

```
table(subset(bacterial_clusters, Class == "c__Actinobacteria")$ClusterMembership)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  4  1  2  1  1  0  0  0  0  0  3  0  0  0  0  0
```

Fig. S3: Network with AMR genes

```
ggplot(gg_net, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 16)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net,Phylum == "Plasmid"), pch = 8, col = "black", aes(size = degree))+
  geom_nodes(data = subset(gg_net, Resistance == TRUE), pch = 8, stroke = 1.5, col = "#ff00cc", aes(siz
  scale_size(range = c(3,6))+
  guides(size = "none")+
    labs(fill = "Class")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), pch = 1
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 6)))+
  guides(size = "none")+
  scale_size(range = c(4,7))+
  # labels
  ggrepel::geom_label_repel(data = subset(gg_net, Identity == "PLASMID"), aes(label = Gene), size = 3, m
```

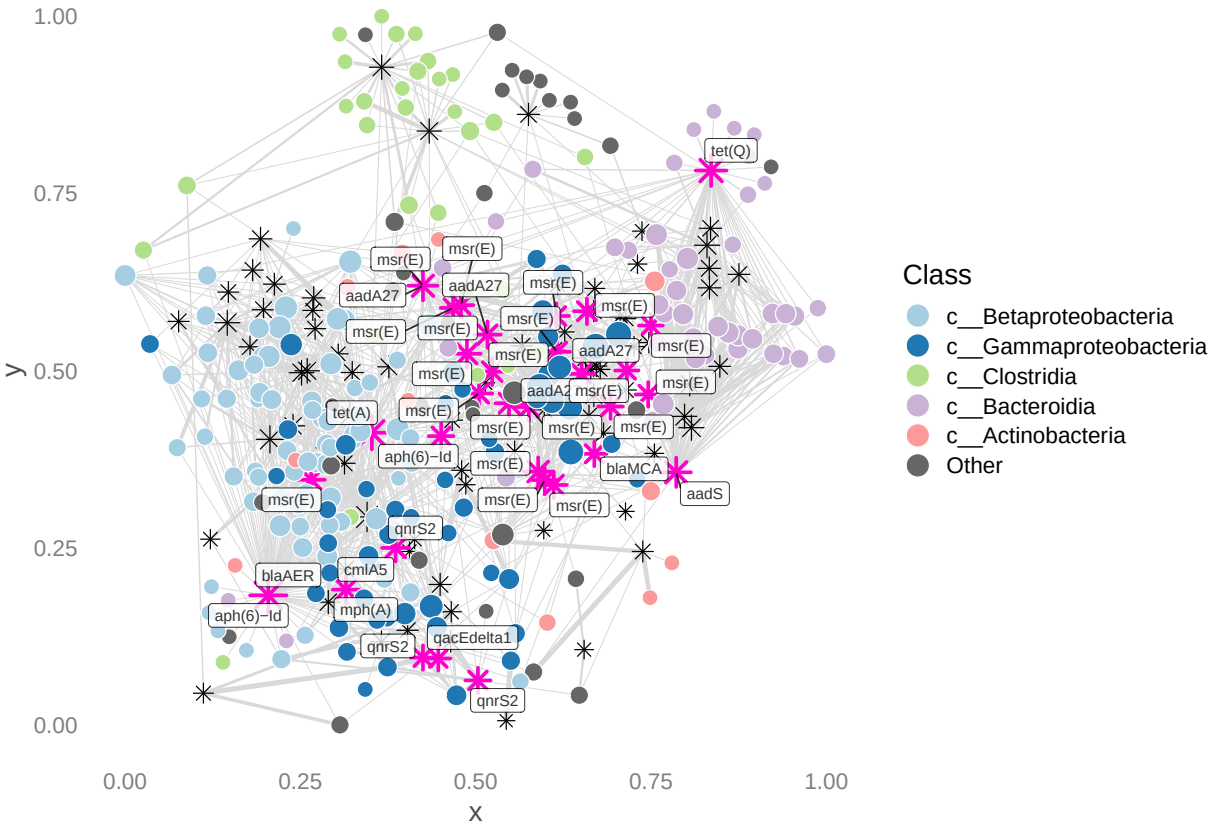


Fig 2b: Bipartite full network

```
#phylo_core<-prune_samples(sample_sums(phylo_filtered)>200, phylo_filtered)
#phylo_core<-microbiome::core(phylo_filtered, detection = 0, prevalence = 0.02)

############ keep only prevalent plasmids

prevalence <- function(physeq, add_tax = TRUE){

  ## Check if taxa are rows
  trows <- taxa_are_rows(physeq)

  ## Extract OTU table
  otutab <- as.data.frame(otu_table(physeq))

  ## Transpose OTU table (species should be arranged by rows)
  if(trows == FALSE){
    otutab <- t(otutab)
  }

  ## Estimate prevalence (number of samples with OTU present)
  prevdf <- apply(X = otutab,
                  #MARGIN = ifelse(trows, yes = 1, no = 2),  # for a non-transposed data
```

```
                  MARGIN = 1,
                  FUN = function(x){sum(x > 0)})

  ## Add total and average read counts per OTU
  prevdf <- data.frame(Prevalence = prevdf,
                       TotalAbundance = taxa_sums(physeq),
                       MeanAbundance = rowMeans(otutab),
                       MedianAbundance = apply(otutab, 1, median))

  ## Add taxonomy table
  if(add_tax == TRUE && !is.null(tax_table(physeq, errorIfNULL = F))){
    prevdf <- cbind(prevdf, tax_table(physeq))
  }
  return(prevdf)
}


prev_df<-prevalence(phylo_filtered)
prev_df<-prev_df %>% arrange(-Prevalence)
#head(prev_df)

########### work out best order for bacterial and plasmids for plot ###
########### work out best order for bacterial and plasmids for plot ###
########### work out best order for bacterial and plasmids for plot ###
########### work out best order for bacterial and plasmids for plot ###

# order bins by class

metadata<-data.frame(sample_data(phylo_filtered))
classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
metadata$Class_plot<- ifelse(metadata$Class %in% classes_to_keep, as.character(metadata$Class), "Other"

metadata$Class_plot<-factor(metadata$Class_plot, levels = c(classes_to_keep, "Other"))
metadata$Plasmid_richness<- sample_sums(phylo_filtered)

mypal1<-brewer.pal(12,"Paired")
mypal2<-brewer.pal(12,"Dark2")
mypal3<-c(mypal1, mypal2)

### colour bacteria by class

metadata<-metadata %>%
  mutate(Bin_colour = case_when(
    Class_plot == "c__Betaproteobacteria" ~ mypal3[1],
    Class_plot == "c__Gammaproteobacteria" ~ mypal3[2],
    Class_plot == "c__Clostridia" ~ mypal3[3],
    Class_plot == "c__Bacteroidia" ~ mypal3[9],
    Class_plot == "c__Actinobacteria" ~ mypal3[5],
    Class_plot == "Other" ~ mypal3[20]))

metadata<- metadata %>% arrange(Class_plot,  -Plasmid_richness)
bin_order<-metadata$cluster_id
bin_col<-metadata$Bin_colour
```

```
### plasmid order
### plasmid order
### plasmid order

# order plasmids by who they are mostly attached to
data<-data.frame(otu_table(phylo_filtered))
data_long<-reshape2::melt(as.matrix(data))
#data_long<-subset(data_long, value>0)
#head(data_long)
data_long$Class<-vlookup(data_long$Var2, metadata, lookup_column = "cluster_id", result_column = "Class
#head(data_long)
table(data_long$Class)
```

```
##
##  c__Betaproteobacteria c__Gammaproteobacteria          c__Clostridia
##                   8720                   6431                   6867
##         c__Bacteroidia      c__Actinobacteria                  Other
##                   6322                   2725                   9701
```

```
data_long %>%
  group_by(Var1, Class) %>%
  summarise(Total=sum(value))-> plasmid_sum


plasmid_sum2<-plasmid_sum %>% group_by(Var1) %>% top_n(1, Total)

#plasmid_sum2$Class<-factor(plasmid_sum2$Class)

plasmid_sum2$Class<-factor(plasmid_sum2$Class, levels = c(classes_to_keep, "Other"))

# add total abundance
plasmid_sums<-data.frame(taxa_sums(phylo_filtered))

plasmid_sums$Plasmid<-row.names(plasmid_sums)
names(plasmid_sums)[1]<-"TotalAbundance"

plasmid_sum2<-merge(plasmid_sum2, plasmid_sums, by.x = "Var1", by.y = "Plasmid", all.x = T)

####


plasmid_sum2 %>% arrange(Class,desc(TotalAbundance))-> plasmid_sum2


plasmid_order<-as.character(plasmid_sum2$Var1)


# colour plasmids by whether they have AMR genes

amr_df<-data.frame(tax_table(phylo_filtered))
head(amr_df)
```

```
##    ta1   ta2
## 2    2  TRUE
## 5    5  TRUE
## 30  30  TRUE
## 1    1  TRUE
## 3    3  TRUE
## 7    7 FALSE
```

```
plasmid_sum2$AMR<- vlookup(plasmid_sum2$Var1, amr_df, lookup_column = "ta1", result_column = "ta2")

plasmid_sum2<- plasmid_sum2 %>% mutate(Plasmid_colour = case_when(AMR == TRUE ~ "#ff00cc", AMR == FALSE

plasmid_col<-as.character(plasmid_sum2$Plasmid_colour)


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

data<-data.frame(otu_table(phylo_filtered))

#plasmid_order<-as.character(plasmid_order)

data[1:5,1:5]
```

```
##    bin_1 bin_10 bin_103 bin_104 bin_105
## 2      0      8       7       0      15
## 5     17      0       0     418       0
## 30     6   1059       0       0      52
## 1      0      0       0     241       0
## 3      0      0      49       0       0
```

```
data<-data[plasmid_order,bin_order]

#head(data)



plotweb(data, text.rot=90,

        col.interaction= rep(plasmid_col, each =length(names(data))), # change number to how many bins
        method = "normal",
        col.high = bin_col,
        col.low = plasmid_col,
        bor.col.interaction = NA,
        bor.col.high = NA,
        bor.col.low = plasmid_col,
       low.spacing = 0.0044,
       high.spacing = 0.0022,
       high.lablength= 0,
       low.lablength = 0)
```

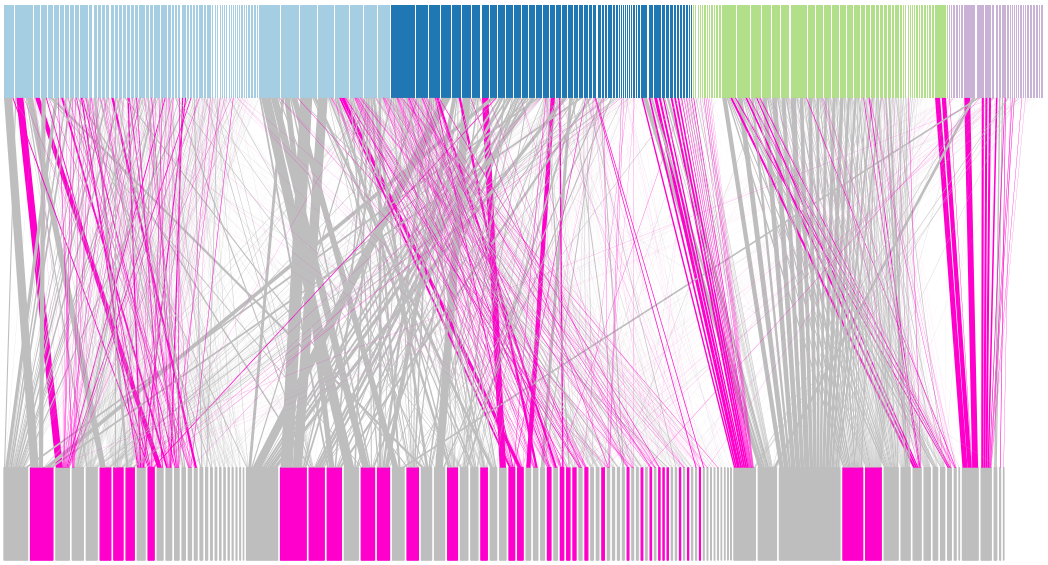
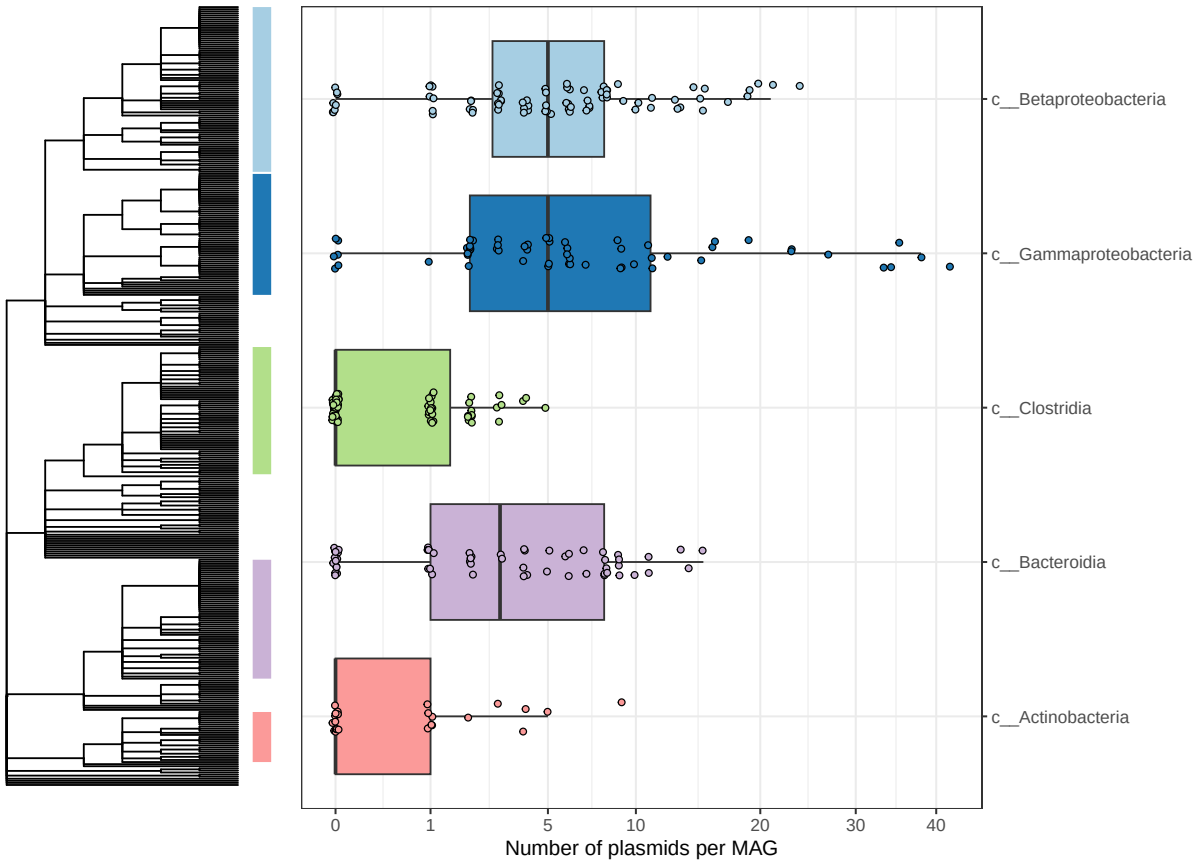


Fig. 3a: Bacterial phylogenetic tree with boxplot

*Edit tree so ordered by taxonomy

```
## edit tree

taxonomy$Phylum<-factor(taxonomy$Phylum)
taxonomy$Class<-factor(taxonomy$Class)
taxonomy$Order<-factor(taxonomy$Order)
taxonomy$Family<-factor(taxonomy$Family)
taxonomy$Genus<-factor(taxonomy$Genus)
taxonomy$Bin<-factor(taxonomy$Bin)

frm <- ~Phylum/Class/Order/Family/Genus/Bin
tree_tax <- as.phylo(frm, data = taxonomy, collapse=FALSE)
tree_tax$edge.length <- rep(1, nrow(tree_tax$edge))


droptips<-tree_tax$tip.label[!tree_tax$tip.label %in% sample_names(phylo_filtered)]

tree_tax<-drop.tip(tree_tax, tip = droptips)

# colours
```

```r
mypal1<-brewer.pal(12,"Paired")
mypal2<-brewer.pal(12,"Dark2")
mypal3<-c(mypal1, mypal2)


tree_tax$Class<-vlookup(tree_tax$tip.label, taxonomy, lookup_column = "Bin", result_column = "Class")
tree_tax$Class<-as.character(tree_tax$Class)
tree_tax$Class<-ifelse(is.na(tree_tax$Class), "Unknown", tree_tax$Class)


## tree figure
tree_class<-ggtree(tree_tax, branch.length = "none")+
  geom_cladelabel(node=415,  col=mypal3[1], label ="", barsize=5, offset = 0.5, align=T, angle=270, offs
  geom_cladelabel(node=440, col=mypal3[2],  label ="",barsize=5, offset = 0.5, align=T, angle=270, offs
  geom_cladelabel(node=464, col=mypal3[9],  label ="", barsize=5, offset = 0.5,align=T, angle=270, offs
  geom_cladelabel(node=379, col=mypal3[3],  label ="", barsize=5, offset = 0.5,align=T, angle=270, offs
  geom_cladelabel(node=482, col=mypal3[5],  label ="", barsize=5, offset = 0.5, align=T, angle=270, offs
   theme(plot.margin = margin(0.2, 0.2, 1.6, 0.2, "cm"))

######### calculate how many  plasmids connected to each MAG

bacteria_alpha<-phyloseq::estimate_richness(phylo_filtered, measures = c("Observed"))
bacteria_alpha$id<-row.names(bacteria_alpha)

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",

bacteria_alpha<-merge(bacteria_alpha, taxonomy, by.x = "id", by.y = "Bin")
bacteria_alpha_filt<-subset(bacteria_alpha, Class %in% classes_to_keep)
bacteria_alpha_filt$Class<-factor(bacteria_alpha_filt$Class, levels = rev(classes_to_keep))

#head(bacteria_alpha_filt)

barplot<-ggplot(bacteria_alpha_filt, aes(y = Class, x = Observed, group = Class, fill = Class))+
  geom_boxplot(aes(fill = Class), outlier.shape = NA)+
  scale_fill_manual(values =rev(mypal3[c(1,2,3,9,5)]) )+
  geom_jitter(height = 0.1, pch = 21)+
 # scale_x_sqrt()+
  theme_bw(base_size = 12) +
  scale_y_discrete(position = "right")+
  scale_x_sqrt(breaks = c(0,1, 5, 10,20,30,40,50))+
  theme(legend.position = "none")+
  xlab("Number of plasmids per MAG")+
  ylab("")

fig3a<-ggarrange(tree_class, barplot, widths = c(1,3))

fig3a
```




```
## krskal wallace test

kruskal.test(Observed~Class, bacteria_alpha_filt)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Observed by Class
## Kruskal-Wallis chi-squared = 88.73, df = 4, p-value < 2.2e-16
```

Generate plasmid statistics

```
plasmid_prev_df<-data.frame(microbiome::prevalence(phylo_filtered, count = T))
names(plasmid_prev_df)<-"Prev"
plasmid_prev_df$Abundance<-taxa_sums(phylo_filtered)
plasmid_prev_df$Rel_Abundance<-plasmid_prev_df$Abundance/(sum(plasmid_prev_df$Abundance))
plasmid_prev_df<-plasmid_prev_df %>% arrange(-Prev)
#head(plasmid_prev_df)
plasmid_prev_df$Cluster<-row.names(plasmid_prev_df)
plasmid_prev_df$Resistance <- plasmid_prev_df$Cluster %in% resistance_clusters
plasmid_prev_df$Prev_proportion <- plasmid_prev_df$Prev/374
```

```r
plasmid_prev_df<-plasmid_prev_df %>% arrange(-Prev)


summary(plasmid_prev_df$Prev)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00   10.00   13.61   16.00  112.00
```

```r
## add new name

plasmid_prev_df$Name<- paste("Pl.", 1:nrow(plasmid_prev_df), sep = "")


plasmid_prev_df<-merge(plasmid_prev_df, resistance_clusters4, by.x = "Cluster", by.y = "ClusterMembershi

plasmid_prev_df<-plasmid_prev_df%>%arrange(-Prev)

head(plasmid_prev_df, 10)
```

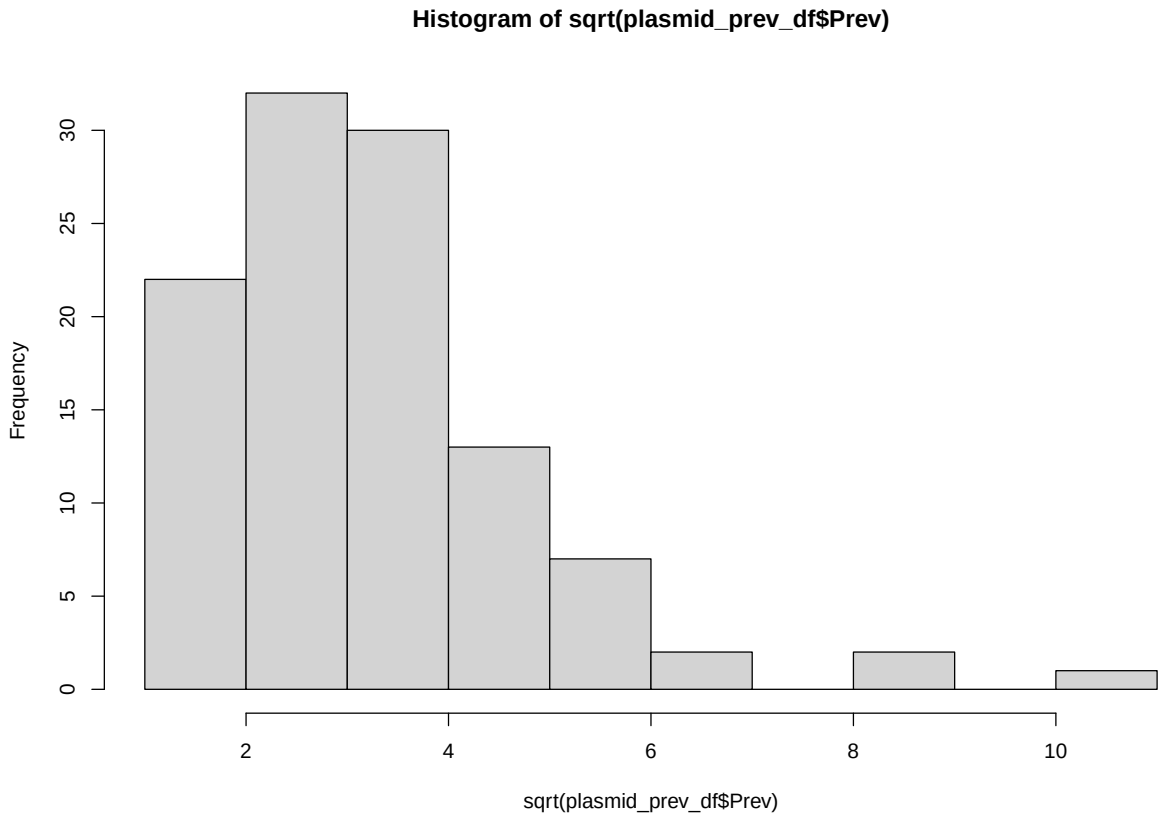
```
##    Cluster Prev Abundance Rel_Abundance Resistance Prev_proportion  Name tet(Q)
## 1        2  112      4205    0.01369404       TRUE      0.29946524  Pl.1      0
## 2       30   71      3712    0.01208853       TRUE      0.18983957  Pl.2      0
## 3       40   68      4908    0.01598343      FALSE      0.18181818  Pl.3     NA
## 4        1   47      7047    0.02294931       TRUE      0.12566845  Pl.4      1
## 5       11   46      4998    0.01627653      FALSE      0.12299465  Pl.5     NA
## 6        4   34      4433    0.01443654       TRUE      0.09090909  Pl.6      0
## 7        5   33      8959    0.02917595       TRUE      0.08823529  Pl.7      0
## 8       37   29      4095    0.01333581      FALSE      0.07754011  Pl.8     NA
## 9       39   29      6319    0.02057850      FALSE      0.07754011  Pl.9     NA
## 10      46   29     10668    0.03474149      FALSE      0.07754011 Pl.10     NA
##    blaAER aph(6)-Id msr(E) aadA27 aadS qnrS2 qacEdelta1 tet(A) blaMCA mph(A)
## 1       1         1      0      0    0     0          0      0      0      0
## 2       0         0      0      0    0     0          0      1      0      0
## 3      NA        NA     NA     NA   NA    NA         NA     NA     NA     NA
## 4       0         0      0      0    0     0          0      0      0      0
## 5      NA        NA     NA     NA   NA    NA         NA     NA     NA     NA
## 6       0         0      1      1    0     0          0      0      0      0
## 7       0         0      0      0    1     0          0      0      0      0
## 8      NA        NA     NA     NA   NA    NA         NA     NA     NA     NA
## 9      NA        NA     NA     NA   NA    NA         NA     NA     NA     NA
## 10     NA        NA     NA     NA   NA    NA         NA     NA     NA     NA
##    cmlA5
## 1      0
## 2      0
## 3     NA
## 4      0
## 5     NA
## 6      0
## 7      0
## 8     NA
## 9     NA
## 10    NA
```

Fig. 3b: Plasmid degree

```
plasmid_prev_df %>% group_by(Resistance) %>% summarize(mean = mean(Prev), median = median(Prev))
```

```
## # A tibble: 2 x 3
##   Resistance  mean median
##   <lgl>      <dbl>  <dbl>
## 1 FALSE       11.0    8
## 2 TRUE        19.8   14.5
```

```
hist(sqrt(plasmid_prev_df$Prev))
```




```
wilcox.test(Prev ~ Resistance, data=plasmid_prev_df)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Prev by Resistance
## W = 757, p-value = 0.001558
## alternative hypothesis: true location shift is not equal to 0
```

```
plasmid_prev_df%>% group_by(Resistance) %>% summarise(mean = mean(Prev), median = median(Prev))
```

```
## # A tibble: 2 x 3
##   Resistance  mean median
##   <lgl>      <dbl>  <dbl>
## 1 FALSE       11.0    8
## 2 TRUE        19.8   14.5
```

```
t.test(sqrt(Prev) ~ Resistance, data=plasmid_prev_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  sqrt(Prev) by Resistance
## t = -2.9213, df = 46.043, p-value = 0.005384
## alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0
## 95 percent confidence interval:
##  -1.7595333 -0.3239538
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            3.034127            4.075870
```

```
plasmid_prev_df %>% group_by(Resistance) %>% summarise(median = median(Prev), mean = mean(Prev))
```

```
## # A tibble: 2 x 3
##   Resistance median  mean
##   <lgl>       <dbl> <dbl>
## 1 FALSE         8    11.0
## 2 TRUE         14.5  19.8
```

```
wilcox.test(Prev ~ Resistance, data=plasmid_prev_df)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Prev by Resistance
## W = 757, p-value = 0.001558
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(sqrt(Prev) ~ Resistance, data=plasmid_prev_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  sqrt(Prev) by Resistance
## t = -2.9213, df = 46.043, p-value = 0.005384
## alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0
## 95 percent confidence interval:
##  -1.7595333 -0.3239538
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            3.034127            4.075870
```

```
table(plasmid_prev_df$Resistance)
```

```
##
## FALSE  TRUE
##    77    32
```

```
fig3b<-ggplot(plasmid_prev_df, aes(y = Resistance, x = Prev))+
        geom_boxplot(alpha = 0.5, outlier.shape = NA, fill = "skyblue")+
        geom_jitter(  height = 0.2, width = 0.01, pch = 8,  size = 1.2, stroke = 0.5,  aes(col = Resist
        xlab("Number of MAGs per plasmid")+
        ylab("AMR gene presence")+
        theme(strip.background =element_rect(fill="white"))+
  scale_x_sqrt(breaks = c(1,5,10,20,30,50,75,100))+
 theme_bw(base_size = 12)+
#  theme(legend.position = "none")+
  scale_color_manual(values = c("black", "#ff00cc"))+
  scale_y_discrete(labels=c("AMR absent", "AMR present")) +
  labs(col = "Plasmid \nAMR resistance")+
  theme(axis.title.y = element_blank())+
 theme(plot.margin=margin(5,2,2,20))


fig3b
```

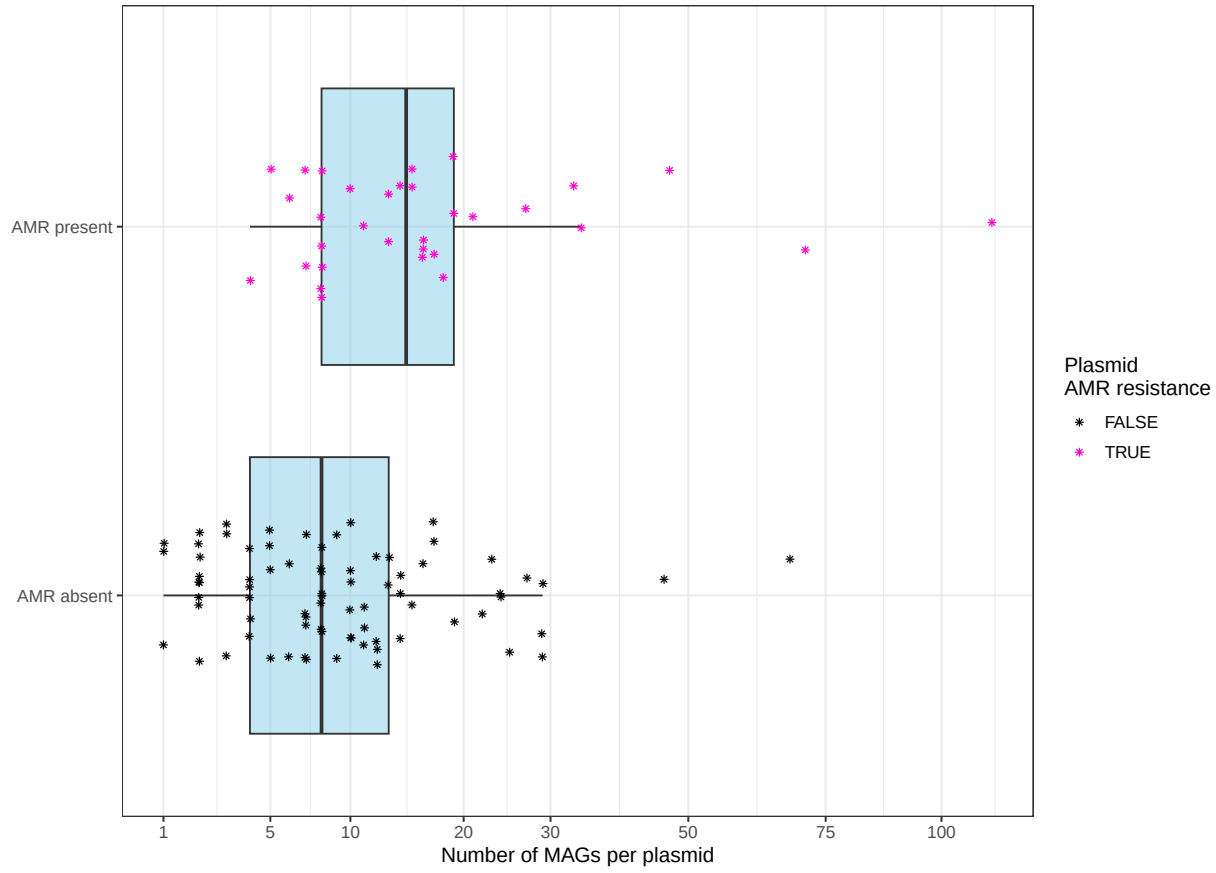


Fig. 3c: Phylogenetic breadth of bacterial hosts

```
uniq<-plasmid_prev_df$Cluster

phylo_signal_list<-list()

for (i in 1:length(uniq)){

  print(i)

plasmid<-prune_taxa(uniq[i], phylo_filtered)
#plasmid<-prune_taxa(uniq[1], phylo_filtered)

plasmid_i<-prune_samples(sample_sums(plasmid)>0, plasmid)

host_id<-sample_names(plasmid_i)


# Filter the tree to include only branches connecting the infected host species
subtree <- keep.tip(tree_phylophlan_rooted, host_id)

# Sum the branch lengths within the subtree
total_branch_length <- mean(branching.times(subtree))
#total_branch_length <- mean(adephylo::distTips(subtree, tips = host_id))

results<-data.frame(total_branch_length, uniq[i])



#barplot.phylo4d(p4d, tree.type = "phylo", tree.ladderize = T, bar.lwd = 1, show.tip = F, center = F, s


phylo_signal_list[[i]]<-results

}
```

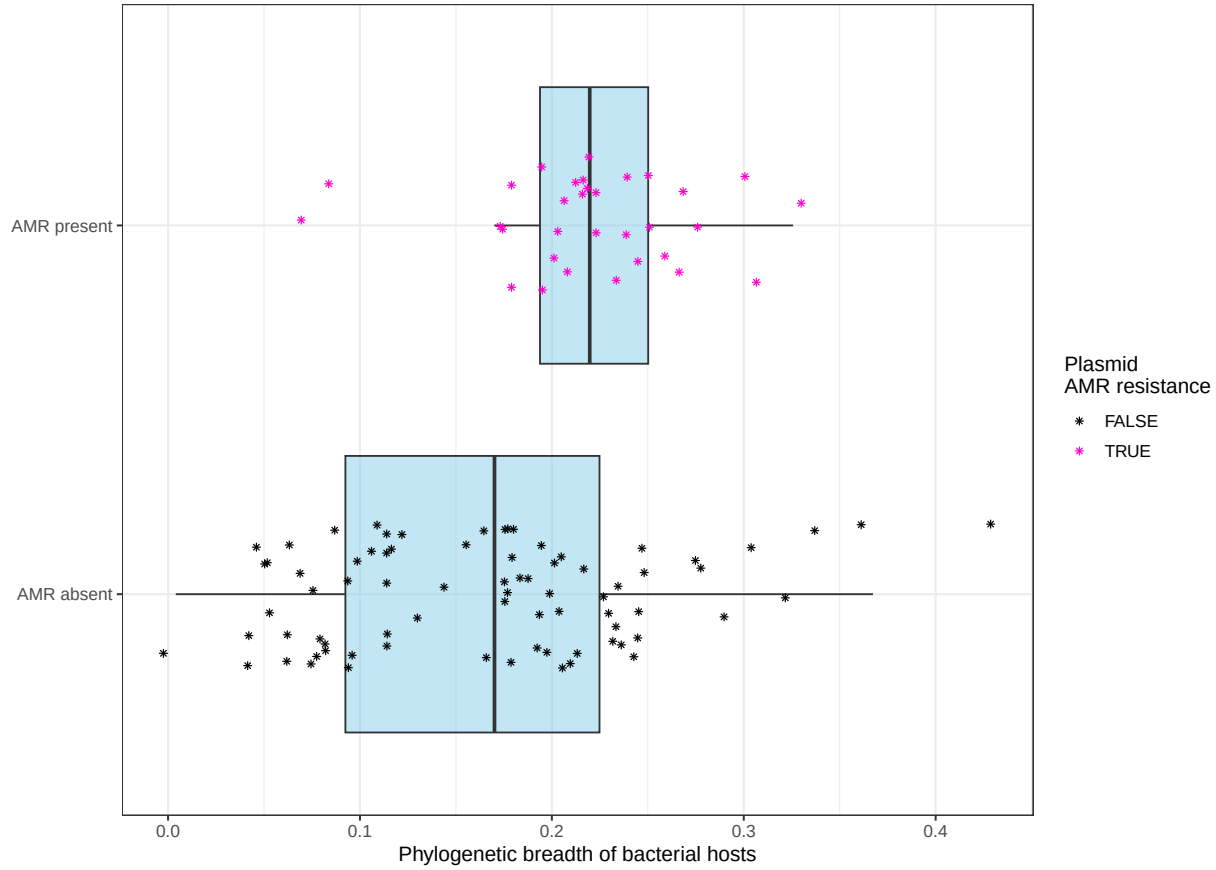
```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
```

```
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
```

```
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
## [1] 101
## [1] 102
## [1] 103
## [1] 104
## [1] 105
## [1] 106
## [1] 107
## [1] 108
## [1] 109
```

```
phylo_signal_df<-do.call(rbind, phylo_signal_list)

names(phylo_signal_df)<-c("BranchLength", "Cluster")

plasmid_prev_df$BranchLength<-vlookup(plasmid_prev_df$Cluster, phylo_signal_df, lookup_column = "Cluste

#head(plasmid_prev_df)
#tail((plasmid_prev_df)

Fig3c<-ggplot(plasmid_prev_df, aes(y = Resistance, x = BranchLength))+
   geom_boxplot(alpha = 0.5, outlier.shape = NA, fill = "skyblue")+
  geom_jitter(  height = 0.2, width = 0.01, pch = 8,  size = 1.2, stroke = 0.5,  aes(col = Resistance))
        xlab("Phylogenetic breadth of bacterial hosts")+
        ylab("AMR gene presence")+
```

```
        theme(strip.background =element_rect(fill="white"))+
 # scale_x_sqrt(breaks = c(1,5,10,20,30,50,75,100))+
 theme_bw(base_size = 12)+
#  theme(legend.position = "none")+
  scale_color_manual(values = c("black", "#ff00cc"))+
  scale_y_discrete(labels=c("AMR absent", "AMR present")) +
  labs(col = "Plasmid \nAMR resistance")+
  theme(axis.title.y = element_blank())+
 theme(plot.margin=margin(5,2,2,20))

Fig3c
```




```
wilcox.test(BranchLength~Resistance, data = plasmid_prev_df)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  BranchLength by Resistance
## W = 677, p-value = 0.0002246
## alternative hypothesis: true location shift is not equal to 0
```

Fig 3d and e: Plot networks with and without AMR

```
set.seed(3)
### without AMR

net.noAMR$Plasmid<-row.names(net.noAMR)

net_long<-reshape2::melt(net.noAMR, id.vars=c("Plasmid"))
#head(net_long)
names(net_long)[2]<-"Bacteria"
net_long<-subset(net_long, value>0)

# generate network object

igraph_net <- igraph::graph.data.frame(net_long[,c('Bacteria','Plasmid')])


E(igraph_net)$weight<-net_long$value
igraph_net<-as.undirected(igraph_net)
V(igraph_net)$degree<-igraph::degree(igraph_net)

### network modularity
wtc <- cluster_walktrap(igraph_net)
modularity(wtc)
```

```
## [1] 0.796895
```

```
modularity(igraph_net, membership(wtc))
```

```
## [1] 0.5813803
```

```
#wt <- walktrap.community(igraph_net, weights = E(igraph_net)$weight, steps = 10)
wt <- fastgreedy.community(igraph_net)
cluster.membership<-membership(wt)

# add cluster to graph metadata
V(igraph_net)$ClusterMembership<-as.factor(membership(wt))

network_metrics_noAMR<-data.frame(network_metrics_noAMR)
network_metrics_noAMR[3,]<-NA
network_metrics_noAMR[3,1]<-"Modularity"
network_metrics_noAMR[3,2]<-modularity(igraph_net, membership(wtc))
network_metrics_noAMR$Stat<-round(network_metrics_noAMR$Stat, 2)



## convert to ggnetwork object


#gg_net_noAMR<-ggnetwork(igraph_net, layout = igraph::with_dh())
```

```
layout_dh <- layout_with_dh(igraph_net,  weight.edge.lengths = edge_density(igraph_net)/1)

gg_net_noAMR<-ggnetwork(igraph_net, layout = layout_dh)


# populate metadata
plasmid_names<-taxa_names(phylo_filtered)
gg_net_noAMR$Identity<-ifelse(gg_net_noAMR$name %in% plasmid_names, "PLASMID", "BACTERIA")

# bacterial taxonomy
gg_net_noAMR$Class <-expss::vlookup(gg_net_noAMR$name, taxonomy, lookup_column = "Bin", result_column =
gg_net_noAMR$Phylum <-expss::vlookup(gg_net_noAMR$name, taxonomy, lookup_column = "Bin", result_column =

#data.frame(table(gg_net_noAMR$Class))

### format class variable

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
gg_net_noAMR$Class_plot<- ifelse(gg_net_noAMR$Class %in% classes_to_keep, as.character(gg_net_noAMR$Clas
gg_net_noAMR$Class_plot<-factor(gg_net_noAMR$Class_plot, levels = c(classes_to_keep, "Other"))


### code for plasmid

gg_net_noAMR$Phylum <-ifelse(is.na(gg_net_noAMR$Phylum), "Plasmid", as.character(gg_net_noAMR$Phylum))


###### plot
tt3 <- ttheme_minimal(
  core=list(bg_params = list(fill = blues9[1:4], col=NA),
            fg_params=list(fontface=3)),
  colhead=list(fg_params=list(col="navyblue", fontface=4L)),
  rowhead=list(fg_params=list(col="orange", fontface=3L)))



#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############
#### with just amr ############

### without AMR

net.AMR$Plasmid<-row.names(net.AMR)

net_long<-reshape2::melt(net.AMR, id.vars=c("Plasmid"))
#head(net_long)
names(net_long)[2]<-"Bacteria"
```

```
net_long<-subset(net_long, value>0)

# generate network object

igraph_net <- igraph::graph.data.frame(net_long[,c('Bacteria','Plasmid')])


E(igraph_net)$weight<-net_long$value
igraph_net<-as.undirected(igraph_net)
V(igraph_net)$degree<-igraph::degree(igraph_net)

### network modularity
wtc <- cluster_walktrap(igraph_net)
modularity(wtc)
```

```
## [1] 0.7116615
```

```
modularity(igraph_net, membership(wtc))
```

```
## [1] 0.362929
```

```
wt <- fastgreedy.community(igraph_net)
#wt <- walktrap.community(igraph_net, weights = E(igraph_net)$weight, steps = 10)
#modularity(wt)
cluster.membership<-membership(wt)

# add cluster to graph metadata
V(igraph_net)$ClusterMembership<-as.factor(membership(wt))


network_metrics_AMR<-data.frame(network_metrics_AMR)
network_metrics_AMR[3,]<-NA
network_metrics_AMR[3,1]<-"Modularity"
network_metrics_AMR[3,2]<-modularity(igraph_net, membership(wtc))
network_metrics_AMR$Stat<-round(network_metrics_AMR$Stat, 2)


## convert to ggnetwork object

#gg_net_AMR<-ggnetwork(igraph_net, layout = igraph::with_dh())
#layout_with_dh()

layout_dh <- layout_with_dh(igraph_net,  weight.edge.lengths = edge_density(igraph_net)/1)

layout_fr<-layout_with_fr(igraph_net)

gg_net_AMR<-ggnetwork(igraph_net, layout = layout_dh)


# populate metadata
plasmid_names<-taxa_names(phylo_filtered)
gg_net_AMR$Identity<-ifelse(gg_net_AMR$name %in% plasmid_names, "PLASMID", "BACTERIA")
```

```r
# bacterial taxonomy
gg_net_AMR$Class <-expss::vlookup(gg_net_AMR$name, taxonomy, lookup_column = "Bin", result_column = "Cla
gg_net_AMR$Phylum <-expss::vlookup(gg_net_AMR$name, taxonomy, lookup_column = "Bin", result_column = "Ph

data.frame(table(gg_net_AMR$Class))
```

```
##                        Var1 Freq
## 1         c__Actinobacteria   29
## 2    c__Alphaproteobacteria    0
## 3                c__Bacilli   19
## 4           c__Bacteroidia  131
## 5     c__Betaproteobacteria  231
## 6            c__Caldilineae    2
## 7               c__CFGB1334    0
## 8               c__CFGB1340    2
## 9               c__CFGB1349    0
## 10              c__CFGB1354    0
## 11              c__CFGB1451    2
## 12              c__CFGB1464    3
## 13              c__CFGB1704    0
## 14              c__CFGB1874    0
## 15              c__CFGB2107    0
## 16              c__CFGB3005    0
## 17              c__CFGB3012    0
## 18              c__CFGB3068    3
## 19              c__CFGB3069    0
## 20              c__CFGB3072    0
## 21              c__CFGB3088    0
## 22              c__CFGB3522    0
## 23              c__CFGB3793    0
## 24              c__CFGB4353    0
## 25               c__CFGB784    2
## 26               c__CFGB971    0
## 27               c__CFGB990    0
## 28            c__Clostridia   30
## 29        c__Coriobacteriia    3
## 30   c__Deltaproteobacteria    8
## 31 c__Epsilonproteobacteria   20
## 32      c__Erysipelotrichia    0
## 33        c__Flavobacteriia    2
## 34         c__Fusobacteriia    4
## 35   c__Gammaproteobacteria  344
## 36         c__Negativicutes    4
## 37            c__Nitrospira    0
## 38           c__Oligoflexia    0
## 39          c__Tissierellia    0
## 40      c__Verrucomicrobiae    0
```

```r
### format class variable

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
gg_net_AMR$Class_plot<- ifelse(gg_net_AMR$Class %in% classes_to_keep, as.character(gg_net_AMR$Class), "
gg_net_AMR$Class_plot<-factor(gg_net_AMR$Class_plot, levels = c(classes_to_keep, "Other"))
```

```
### code for plasmid

gg_net_AMR$Phylum <-ifelse(is.na(gg_net_AMR$Phylum), "Plasmid", as.character(gg_net_AMR$Phylum))

########### no AMR #######
########### no AMR #######
########### no AMR #######
########### no AMR #######

length(unique(gg_net_noAMR$ClusterMembership))
```

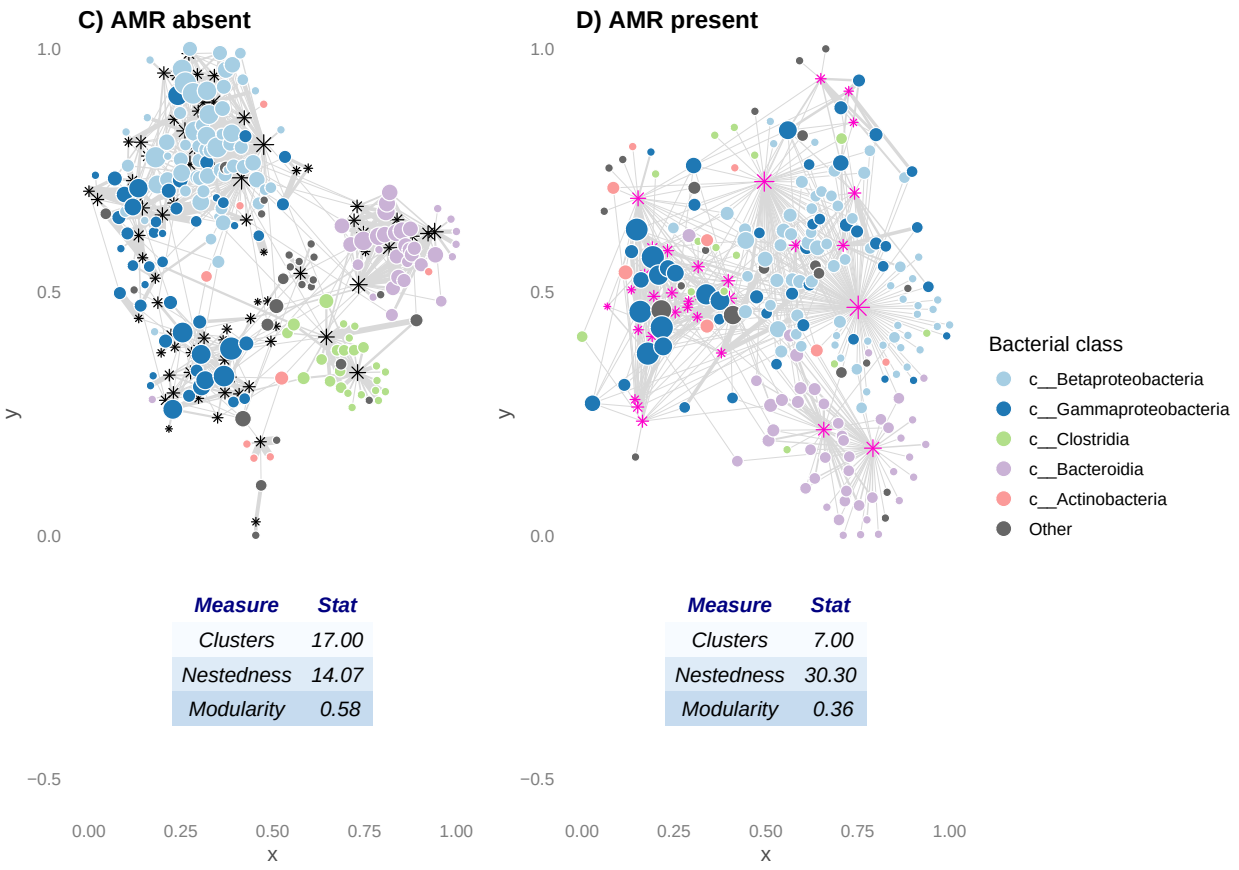
```
## [1] 17
```

```
length(unique(gg_net_AMR$ClusterMembership))
```

```
## [1] 7
```

```
network_metrics_noAMR[1,1]<-"Clusters"
network_metrics_AMR[1,1] <-"Clusters"

network_metrics_noAMR[1,2]<-length(unique(gg_net_noAMR$ClusterMembership))
network_metrics_AMR[1,2]<-length(unique(gg_net_AMR$ClusterMembership))


### plot ###
### plot ###
### plot ###

plot_noAMR<-ggplot(gg_net_noAMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,3))+
  guides(linewidth = "none")+

  # plasmid nodes
  geom_nodes(data = subset(gg_net_noAMR,Phylum == "Plasmid"), pch = 8, col = "black", aes(size = degree
    scale_size(range = c(1,4))+
  guides(size = "none")+
  labs(fill = "Bacterial class")+

  ggnewscale::new_scale("size")+

  # bacterial nodes

  geom_nodes(data = subset(gg_net_noAMR, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), p
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(2,6))+
```

```
  annotation_custom(tableGrob(network_metrics_noAMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymi
  ylim(c(-0.5, 1))

##### with AMR ###
##### with AMR ###
##### with AMR ###
##### with AMR ###

plot_AMR<-ggplot(gg_net_AMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes
  geom_nodes(data = subset(gg_net_AMR,Phylum == "Plasmid"), pch = 8, col = "#ff00cc", aes(size = degree

  scale_size(range = c(1,4))+
  guides(size = "none")+

  labs(fill = "Bacterial class")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net_AMR, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), pc
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(2,6))+

  # stats box
  annotation_custom(tableGrob(network_metrics_AMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymin=
  ylim(c(-0.5, 1))



ggarrange(plot_noAMR, plot_AMR, ncol = 2, legend = "right", common.legend = T, labels = c("C) AMR absen
```



Colour by cluster membership

```
length(unique(gg_net_noAMR$ClusterMembership))
```

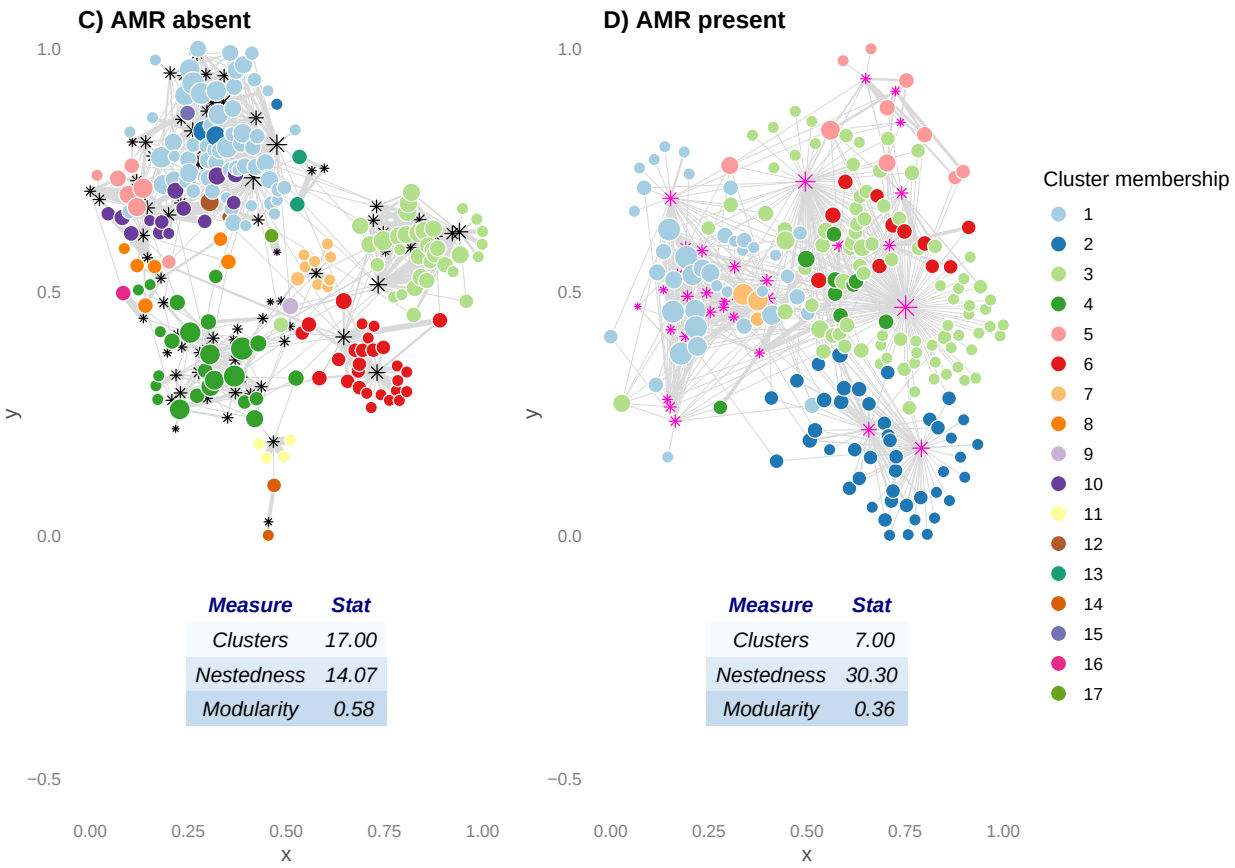
```
## [1] 17
```

```
length(unique(gg_net_AMR$ClusterMembership))
```

```
## [1] 7
```

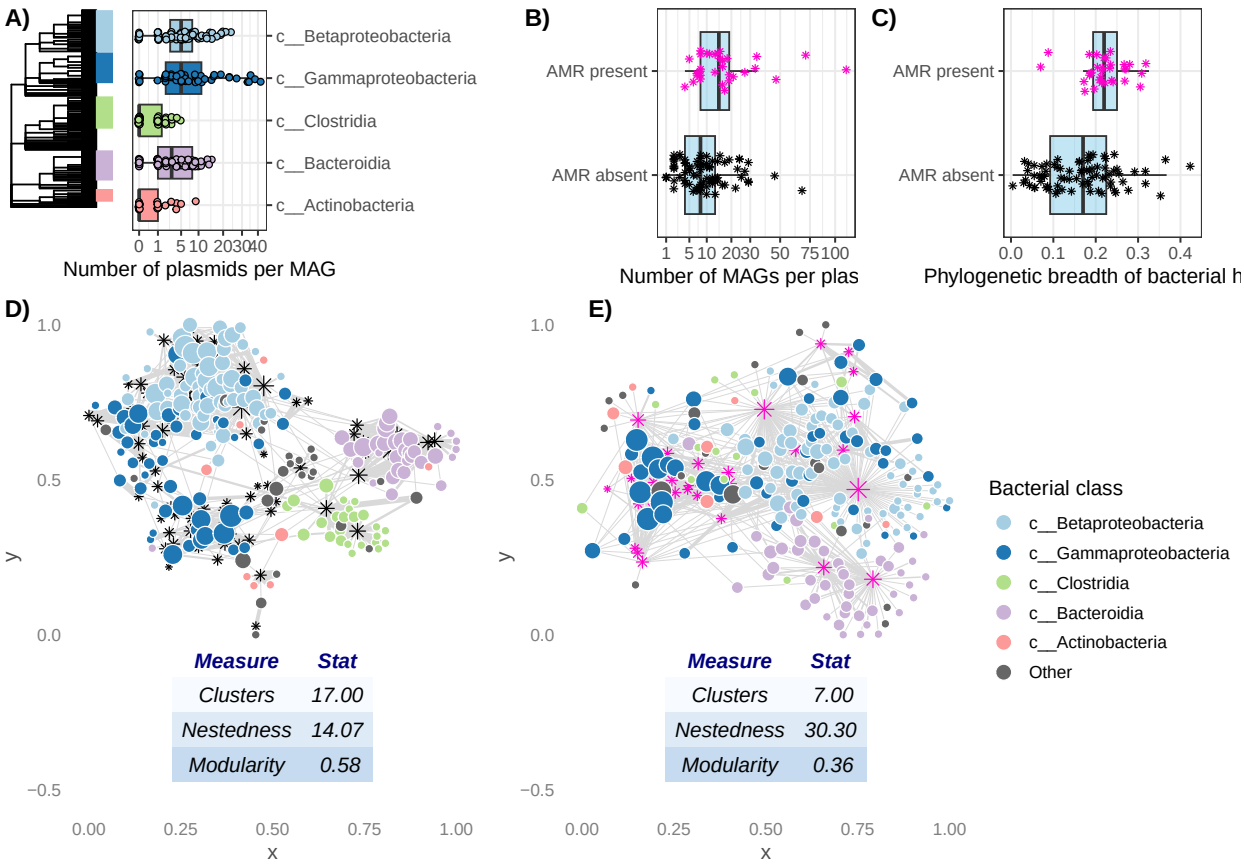
```
plot_noAMR_clusters<-ggplot(gg_net_noAMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,3))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net_noAMR,Phylum == "Plasmid"), pch = 8, col = "black", aes(size = degree
```

```
  scale_size(range = c(1,4))+

  guides(size = "none")+

  labs(fill = "Cluster membership")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net_noAMR, Phylum != "Plasmid"), aes(fill = ClusterMembership, size = degr
  scale_fill_manual(values = mypal3)+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(3,6))+
  annotation_custom(tableGrob(network_metrics_noAMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymi
  ylim(c(-0.5, 1))

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

plot_AMR_clusters<-ggplot(gg_net_AMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net_AMR,Phylum == "Plasmid"), pch = 8, col = "#ff00cc", aes(size = degree

  scale_size(range = c(1,4))+
  guides(size = "none")+
  labs(fill = "Cluster membership")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net_AMR, Phylum != "Plasmid"), aes(fill = ClusterMembership, size = degre
  scale_fill_manual(values = mypal3)+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(3,6))+

  # stats box
  annotation_custom(tableGrob(network_metrics_AMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymin=
  ylim(c(-0.5, 1))



ggarrange(plot_noAMR_clusters, plot_AMR_clusters, ncol = 2, legend = "right", common.legend = T, labels
```



- Plot all together

```
fig3abc<-ggarrange(fig3a, fig3b, Fig3c, NULL, ncol = 4, widths = c(1.5,1,1,0.1), labels = c("A)", "B)",

fig3de<- ggarrange(plot_noAMR, plot_AMR, ncol = 2, legend = "right", common.legend = T, labels = c("D)"

ggarrange(fig3abc, fig3de, ncol = 1, heights = c(1,2))
```




```
#ggsave("FIGURES_NC/Fig_3.pdf")
```

Fig S4: Proteobacteria-only network

```
proteobacteria<-subset(taxonomy, Phylum == "p__Proteobacteria")

prot_to_keep<-as.character(proteobacteria$Bin)

prot_ps <- prune_samples(prot_to_keep, phylo_filtered)

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

table(sample_data(prot_ps)$Class)
```

```
##
##   c__Alphaproteobacteria    c__Betaproteobacteria             c__CFGB1704
##                        5                       80                       2
##              c__CFGB990   c__Deltaproteobacteria c__Epsilonproteobacteria
##                        1                        8                       7
##   c__Gammaproteobacteria          c__Oligoflexia
##                       59                       1
```

```
######################
######################
######################
######################

net.v<-data.frame(prot_ps@otu_table@.Data)
  network_metrics<-data.frame(networklevel(net.v, index = c( "NODF", "number of compartments")))
names(network_metrics)[1]<-"Stat"

ps_noAMR<-subset_taxa(prot_ps, ta2 == FALSE)
ps_noAMR<-prune_samples(sample_sums(ps_noAMR)>0, ps_noAMR)
net.noAMR<-data.frame(ps_noAMR@otu_table@.Data)

network_metrics_noAMR<-data.frame(networklevel(net.noAMR, index = c("NODF", "number of compartments")))
 names(network_metrics_noAMR)[1]<-"Stat"

## just AMR

ps_AMR<-subset_taxa(prot_ps, ta2 == TRUE)
ps_AMR<-prune_samples(sample_sums(ps_AMR)>0, ps_AMR)
net.AMR<-data.frame(ps_AMR@otu_table@.Data)
network_metrics_AMR<-data.frame(networklevel(net.AMR, index = c( "NODF", "number of compartments")))
names(network_metrics_AMR)[1]<-"Stat"

network_metrics_AMR
```

```
##                            Stat
## number of compartments  1.00000
## NODF                   50.40679
```

```
#head(net.AMR)
```

Plot Proteobacteria networks with and without AMR

```
### without AMR

net.noAMR$Plasmid<-row.names(net.noAMR)

net_long<-reshape2::melt(net.noAMR, id.vars=c("Plasmid"))
#head(net_long)
names(net_long)[2]<-"Bacteria"
net_long<-subset(net_long, value>0)

# generate network object

igraph_net <- igraph::graph.data.frame(net_long[,c('Bacteria','Plasmid')])


E(igraph_net)$weight<-net_long$value
igraph_net<-as.undirected(igraph_net)
V(igraph_net)$degree<-igraph::degree(igraph_net)
```

```
### network modularity
wtc <- cluster_walktrap(igraph_net)
modularity(wtc)
```

```
## [1] 0.7282715
```

```
modularity(igraph_net, membership(wtc))
```

```
## [1] 0.3960505
```

```
network_metrics_noAMR<-data.frame(network_metrics_noAMR)
network_metrics_noAMR[3,]<-NA
row.names(network_metrics_noAMR)[3]<-"Modularity"
network_metrics_noAMR[3,1]<-modularity(igraph_net, membership(wtc))
network_metrics_noAMR$Stat<-round(network_metrics_noAMR$Stat, 2)

network_metrics_noAMR$Measure <- row.names(network_metrics_noAMR)
network_metrics_noAMR<-network_metrics_noAMR[,c(2,1)]

network_metrics_noAMR$Measure<-c("# compartments", "Nestedness", "Modularity")


## convert to ggnetwork object

#gg_net_noAMR<-ggnetwork(igraph_net, layout = igraph::with_dh())
layout_dh <- layout_with_dh(igraph_net,  weight.edge.lengths = edge_density(igraph_net)/1)

gg_net_noAMR<-ggnetwork(igraph_net, layout = layout_dh)


# populate metadata
plasmid_names<-taxa_names(phylo_filtered)
gg_net_noAMR$Identity<-ifelse(gg_net_noAMR$name %in% plasmid_names, "PLASMID", "BACTERIA")

# bacterial taxonomy
gg_net_noAMR$Class <-expss::vlookup(gg_net_noAMR$name, taxonomy, lookup_column = "Bin", result_column =
gg_net_noAMR$Phylum <-expss::vlookup(gg_net_noAMR$name, taxonomy, lookup_column = "Bin", result_column =

#data.frame(table(gg_net_noAMR$Class))

### format class variable

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
gg_net_noAMR$Class_plot<- ifelse(gg_net_noAMR$Class %in% classes_to_keep, as.character(gg_net_noAMR$Clas
gg_net_noAMR$Class_plot<-factor(gg_net_noAMR$Class_plot, levels = c(classes_to_keep, "Other"))


### code for plasmid

gg_net_noAMR$Phylum <-ifelse(is.na(gg_net_noAMR$Phylum), "Plasmid", as.character(gg_net_noAMR$Phylum))
```

```
###### plot
tt3 <- ttheme_minimal(
  core=list(bg_params = list(fill = blues9[1:4], col=NA),
            fg_params=list(fontface=3)),
  colhead=list(fg_params=list(col="navyblue", fontface=4L)),
  rowhead=list(fg_params=list(col="orange", fontface=3L)))




#### with just amr ###########
#### with just amr ###########
#### with just amr ###########
#### with just amr ###########

### without AMR

net.AMR$Plasmid<-row.names(net.AMR)

net_long<-reshape2::melt(net.AMR, id.vars=c("Plasmid"))
#head(net_long)
names(net_long)[2]<-"Bacteria"
net_long<-subset(net_long, value>0)

# generate network object

igraph_net <- igraph::graph.data.frame(net_long[,c('Bacteria','Plasmid')])


E(igraph_net)$weight<-net_long$value
igraph_net<-as.undirected(igraph_net)
V(igraph_net)$degree<-igraph::degree(igraph_net)

### network modularity
wtc <- cluster_walktrap(igraph_net)
modularity(wtc)
```

```
## [1] 0.6673206
```

```
modularity(igraph_net, membership(wtc))
```

```
## [1] 0.3218583
```

```
network_metrics_AMR<-data.frame(network_metrics_AMR)
network_metrics_AMR[3,]<-NA
row.names(network_metrics_AMR)[3]<-"Modularity"
network_metrics_AMR[3,1]<-modularity(igraph_net, membership(wtc))
network_metrics_AMR$Stat<-round(network_metrics_AMR$Stat, 2)
network_metrics_AMR$Measure <- row.names(network_metrics_AMR)
network_metrics_AMR<-network_metrics_AMR[,c(2,1)]
network_metrics_AMR$Measure<-c("# compartments", "Nestedness", "Modularity")
```

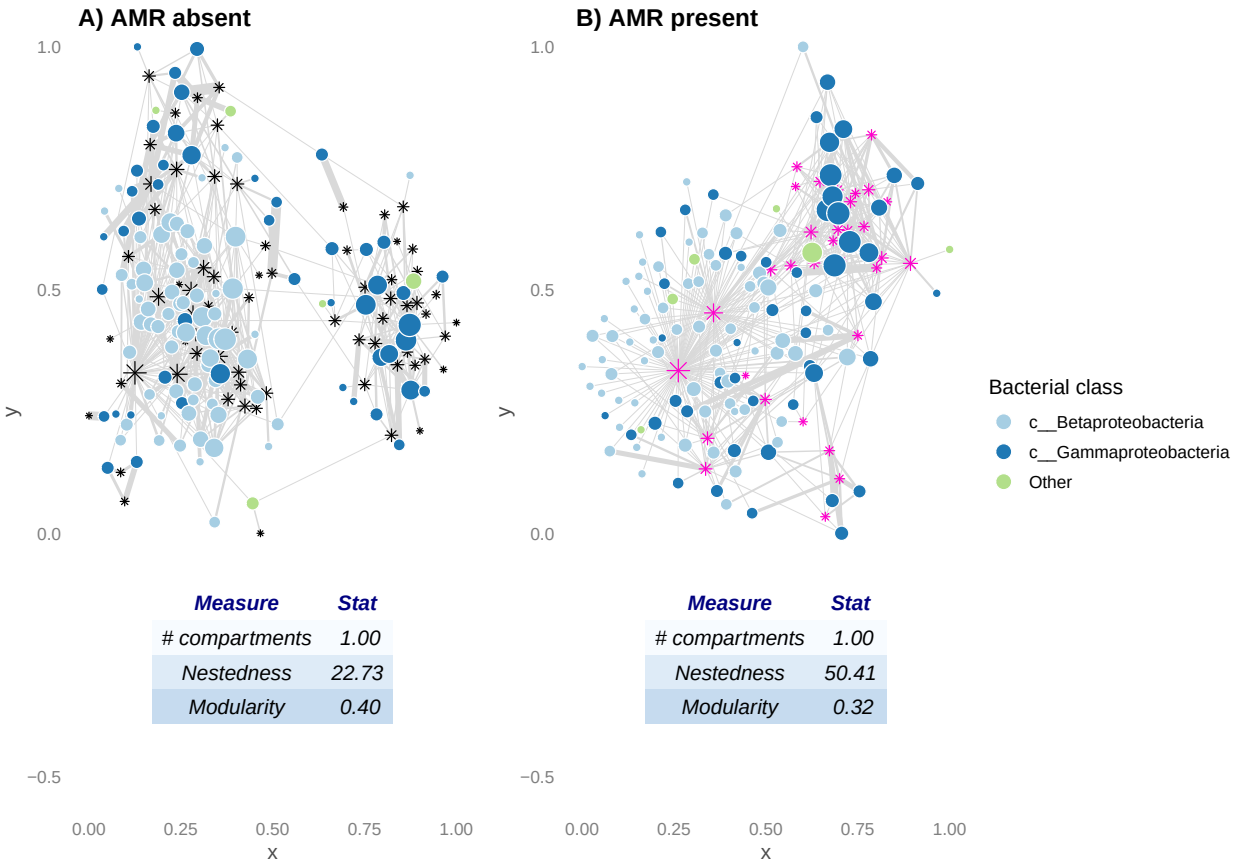
```
## convert to ggnetwork object

#gg_net_AMR<-ggnetwork(igraph_net, layout = igraph::with_dh())
layout_dh <- layout_with_dh(igraph_net,  weight.edge.lengths = edge_density(igraph_net)/1)

gg_net_AMR<-ggnetwork(igraph_net, layout = layout_dh)


# populate metadata
plasmid_names<-taxa_names(phylo_filtered)
gg_net_AMR$Identity<-ifelse(gg_net_AMR$name %in% plasmid_names, "PLASMID", "BACTERIA")

# bacterial taxonomy
gg_net_AMR$Class <-expss::vlookup(gg_net_AMR$name, taxonomy, lookup_column = "Bin", result_column = "Cla
gg_net_AMR$Phylum <-expss::vlookup(gg_net_AMR$name, taxonomy, lookup_column = "Bin", result_column = "Ph

data.frame(table(gg_net_AMR$Class))
```

```
##                        Var1 Freq
## 1         c__Actinobacteria    0
## 2    c__Alphaproteobacteria    0
## 3                c__Bacilli    0
## 4            c__Bacteroidia    0
## 5     c__Betaproteobacteria  231
## 6            c__Caldilineae    0
## 7               c__CFGB1334    0
## 8               c__CFGB1340    0
## 9               c__CFGB1349    0
## 10              c__CFGB1354    0
## 11              c__CFGB1451    0
## 12              c__CFGB1464    0
## 13              c__CFGB1704    0
## 14              c__CFGB1874    0
## 15              c__CFGB2107    0
## 16              c__CFGB3005    0
## 17              c__CFGB3012    0
## 18              c__CFGB3068    0
## 19              c__CFGB3069    0
## 20              c__CFGB3072    0
## 21              c__CFGB3088    0
## 22              c__CFGB3522    0
## 23              c__CFGB3793    0
## 24              c__CFGB4353    0
## 25               c__CFGB784    0
## 26               c__CFGB971    0
## 27               c__CFGB990    0
## 28           c__Clostridia    0
## 29        c__Coriobacteriia    0
## 30   c__Deltaproteobacteria    8
## 31 c__Epsilonproteobacteria   20
## 32      c__Erysipelotrichia    0
## 33        c__Flavobacteriia    0
## 34         c__Fusobacteriia    0
```

```
## 35   c__Gammaproteobacteria  344
## 36         c__Negativicutes    0
## 37            c__Nitrospira    0
## 38           c__Oligoflexia    0
## 39          c__Tissierellia    0
## 40      c__Verrucomicrobiae    0
```

```
### format class variable

classes_to_keep<-c("c__Betaproteobacteria","c__Gammaproteobacteria","c__Clostridia" , "c__Bacteroidia",
gg_net_AMR$Class_plot<- ifelse(gg_net_AMR$Class %in% classes_to_keep, as.character(gg_net_AMR$Class), "
gg_net_AMR$Class_plot<-factor(gg_net_AMR$Class_plot, levels = c(classes_to_keep, "Other"))


### code for plasmid

gg_net_AMR$Phylum <-ifelse(is.na(gg_net_AMR$Phylum), "Plasmid", as.character(gg_net_AMR$Phylum))


###### plot

plot_noAMR<-ggplot(gg_net_noAMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+

  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,3))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net_noAMR,Phylum == "Plasmid"), pch = 8, col = "black", aes(size = degree


  scale_size(range = c(1,4))+

  guides(size = "none")+

  labs(fill = "Bacterial class")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net_noAMR, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), p
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(2,6))+
  annotation_custom(tableGrob(network_metrics_noAMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymi
  ylim(c(-0.5, 1))


plot_AMR<-ggplot(gg_net_AMR, aes(x = x, y = y, xend = xend, yend = yend)) +
  theme_blank(base_size = 12)+
```

```
  # edges
  geom_edges( aes(linewidth = sqrt(weight)), col = "gray85") +
  scale_linewidth(range = c(0,2))+
  guides(linewidth = "none")+

  # plasmid nodes

  geom_nodes(data = subset(gg_net_AMR,Phylum == "Plasmid"), pch = 8, col = "#ff00cc", aes(size = degree)


  scale_size(range = c(1,4))+

  guides(size = "none")+

  labs(fill = "Bacterial class")+

  #bacterial nodes
  ggnewscale::new_scale("size")+

  geom_nodes(data = subset(gg_net_AMR, Phylum != "Plasmid"), aes(fill = Class_plot, size = degree ), pch
  scale_fill_manual(values = mypal3[c(1,2,3,9,5,20)])+
  guides(fill= guide_legend(ncol=1, override.aes = list(size = 4)))+
  guides(size = "none")+
  scale_size(range = c(2,6))+

  # stats box
  annotation_custom(tableGrob(network_metrics_AMR, rows = NULL, theme = tt3), xmin=0.3, xmax=0.7, ymin=
  ylim(c(-0.5, 1))



ggarrange(plot_noAMR, plot_AMR, ncol = 2, legend = "right", common.legend = T, labels = c("A) AMR absen
```




```
#ggsave("FIGURES_NC/Proteobacteria.pdf")
```

Fig. 4: Phylogenetic distribution of top plasmids

```
mypal1<-brewer.pal(12,"Paired")
mypal2<-brewer.pal(12,"Dark2")
mypal3<-c(mypal1, mypal2)

top_plasmids<-plasmid_prev_df$Cluster[1:15]
#top_plasmids<-plasmid_prev_df$Cluster[50:59]
#top_plasmids<-plasmid_prev_df$Cluster[150:159]
#head(plasmid_prev_df, 15)
table(plasmid_prev_df$Resistance)
```

```
##
## FALSE  TRUE
##    77    32
```

```
plasmid_counts<-data.frame(t(otu_table(phylo_filtered)))
#head(plasmid_counts)
names(plasmid_counts)<-taxa_names(phylo_filtered)
```

```
top_plasmid_counts<-plasmid_counts[,top_plasmids]
#head(top_plasmid_counts)

names(top_plasmid_counts)<-paste("Pl.", 1:ncol(top_plasmid_counts), sep = "")

top_plasmid_counts$id<-row.names(top_plasmid_counts)

#head(plasmid_prev_df,10)


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


tree_class<-ggtree(tree_tax, branch.length = "none")+
  geom_cladelabel(node=415,  col=mypal3[1], label ="BETAPR.", barsize=5, align=T, angle=270, offset.tex
  geom_cladelabel(node=440, col=mypal3[2],  label ="GAMMAPR.", barsize=5, align=T, angle=270, offset.te
  geom_cladelabel(node=464, col=mypal3[9],  label ="BACTEROIDIA", barsize=5, align=T, angle=270, offset
  geom_cladelabel(node=379, col=mypal3[3],  label ="CLOSTRIDIA", barsize=5, align=T, angle=270, offset.
  geom_cladelabel(node=482, col=mypal3[5],  label ="ACTINO.", barsize=5, align=T, angle=270, offset.tex

#p2<-tree_class + theme(plot.margin=margin(3, 6, 35, 6))

p3<-facet_plot(tree_class, panel = "Plasmid 1",
           data = top_plasmid_counts[,c(16,1)],
           geom = geom_segment,
           size=1,
           color = "#ff00cc",
           mapping=aes(x=0, xend= Pl.1, y = y, yend = y))


p4<-facet_plot(p3, panel = "Plasmid 2",
           data = top_plasmid_counts[,c(16,2)],
           geom = geom_segment,
           size=1,
           color = "#ff00cc",
           mapping=aes(x=0, xend= Pl.2, y = y, yend = y))

p5<-facet_plot(p4, panel = "Plasmid 3",
           data = top_plasmid_counts[,c(16,3)],
           geom = geom_segment,
           size=1,
           color = "black",
           mapping=aes(x=0, xend= Pl.3, y = y, yend = y))


p6<-facet_plot(p5, panel = "Plasmid 4",
           data = top_plasmid_counts[,c(16,4)],
           geom = geom_segment,
           size=1,
           color = "#ff00cc",
```

```
           mapping=aes(x=0, xend= Pl.4, y = y, yend = y))

p7<-facet_plot(p6, panel = "Plasmid 5",
           data = top_plasmid_counts[,c(16,5)],
           geom = geom_segment,
           size=1,
           color = "black",
           mapping=aes(x=0, xend= Pl.5, y = y, yend = y))+
   geom_vline(data = top_plasmid_counts[,c(11,5)], aes(xintercept=0), col = "#ff00cc", linetype = "dash

##################################### 6-10

p8<-facet_plot(p7, panel = "Plasmid 6",
           data = top_plasmid_counts[,c(16,6)],
           geom = geom_segment,
           size=1,
           color = "#ff00cc",
           mapping=aes(x=0, xend= Pl.6, y = y, yend = y))

p9<-facet_plot(p8, panel = "Plasmid 7",
           data = top_plasmid_counts[,c(16,7)],
           geom = geom_segment,
           size=1,
           color = "#ff00cc",
           mapping=aes(x=0, xend= Pl.7, y = y, yend = y))

p10<-facet_plot(p9, panel = "Plasmid 8",
           data = top_plasmid_counts[,c(16,8)],
           geom = geom_segment,
           size=1,
           color = "black",
           mapping=aes(x=0, xend= Pl.8, y = y, yend = y))

p11<-facet_plot(p10, panel = "Plasmid 9",
           data = top_plasmid_counts[,c(16,9)],
           geom = geom_segment,
           size=1,
           color = "black",
           mapping=aes(x=0, xend= Pl.9, y = y, yend = y))

final<-facet_plot(p11, panel = "Plasmid 10",
           data = top_plasmid_counts[,c(16,10)],
           geom = geom_segment,
           size=1,
           color = "black",
          # widths = c(5,1,1,1,1,1,1,1,1,1,1,1),
           mapping=aes(x=0, xend= Pl.10, y = y, yend = y))+
   geom_vline(data = top_plasmid_counts[,c(11,5)], aes(xintercept=0), col = "darkgrey", linetype = "dash

# load the packaged
library(grid)
library(gtable)
```

```r
gt = ggplot_gtable(ggplot_build(final))
gtable_show_layout(gt) # will show you the layout - very handy function
gt # see plot layout in table format
```

```
## TableGrob (13 x 29) "layout": 55 grobs
##     z         cells       name                                grob
## 1   0 ( 1-13, 1-29) background       rect[plot.background..rect.1952]
## 2   1 ( 8- 8, 5- 5)  panel-1-1              gTree[panel-1.gTree.1709]
## 3   1 ( 8- 8, 7- 7)  panel-1-2              gTree[panel-2.gTree.1720]
## 4   1 ( 8- 8, 9- 9)  panel-1-3              gTree[panel-3.gTree.1731]
## 5   1 ( 8- 8,11-11)  panel-1-4              gTree[panel-4.gTree.1742]
## 6   1 ( 8- 8,13-13)  panel-1-5              gTree[panel-5.gTree.1753]
## 7   1 ( 8- 8,15-15)  panel-1-6              gTree[panel-6.gTree.1764]
## 8   1 ( 8- 8,17-17)  panel-1-7              gTree[panel-7.gTree.1775]
## 9   1 ( 8- 8,19-19)  panel-1-8              gTree[panel-8.gTree.1786]
## 10  1 ( 8- 8,21-21)  panel-1-9              gTree[panel-9.gTree.1797]
## 11  1 ( 8- 8,23-23) panel-1-10             gTree[panel-10.gTree.1808]
## 12  1 ( 8- 8,25-25) panel-1-11             gTree[panel-11.gTree.1819]
## 13  3 ( 6- 6, 5- 5)   axis-t-1                          zeroGrob[NULL]
## 14  3 ( 6- 6, 7- 7)   axis-t-2                          zeroGrob[NULL]
## 15  3 ( 6- 6, 9- 9)   axis-t-3                          zeroGrob[NULL]
## 16  3 ( 6- 6,11-11)   axis-t-4                          zeroGrob[NULL]
## 17  3 ( 6- 6,13-13)   axis-t-5                          zeroGrob[NULL]
## 18  3 ( 6- 6,15-15)   axis-t-6                          zeroGrob[NULL]
## 19  3 ( 6- 6,17-17)   axis-t-7                          zeroGrob[NULL]
## 20  3 ( 6- 6,19-19)   axis-t-8                          zeroGrob[NULL]
## 21  3 ( 6- 6,21-21)   axis-t-9                          zeroGrob[NULL]
## 22  3 ( 6- 6,23-23)  axis-t-10                          zeroGrob[NULL]
## 23  3 ( 6- 6,25-25)  axis-t-11                          zeroGrob[NULL]
## 24  3 ( 9- 9, 5- 5)   axis-b-1   absoluteGrob[GRID.absoluteGrob.1820]
## 25  3 ( 9- 9, 7- 7)   axis-b-2   absoluteGrob[GRID.absoluteGrob.1821]
## 26  3 ( 9- 9, 9- 9)   axis-b-3   absoluteGrob[GRID.absoluteGrob.1822]
## 27  3 ( 9- 9,11-11)   axis-b-4   absoluteGrob[GRID.absoluteGrob.1823]
## 28  3 ( 9- 9,13-13)   axis-b-5   absoluteGrob[GRID.absoluteGrob.1824]
## 29  3 ( 9- 9,15-15)   axis-b-6   absoluteGrob[GRID.absoluteGrob.1825]
## 30  3 ( 9- 9,17-17)   axis-b-7   absoluteGrob[GRID.absoluteGrob.1826]
## 31  3 ( 9- 9,19-19)   axis-b-8   absoluteGrob[GRID.absoluteGrob.1827]
## 32  3 ( 9- 9,21-21)   axis-b-9   absoluteGrob[GRID.absoluteGrob.1828]
## 33  3 ( 9- 9,23-23)  axis-b-10   absoluteGrob[GRID.absoluteGrob.1829]
## 34  3 ( 9- 9,25-25)  axis-b-11   absoluteGrob[GRID.absoluteGrob.1830]
## 35  3 ( 8- 8, 4- 4)   axis-l-1   absoluteGrob[GRID.absoluteGrob.1831]
## 36  3 ( 8- 8,26-26)   axis-r-1                          zeroGrob[NULL]
## 37  2 ( 7- 7, 5- 5)  strip-t-1                           gtable[strip]
## 38  2 ( 7- 7, 7- 7)  strip-t-2                           gtable[strip]
## 39  2 ( 7- 7, 9- 9)  strip-t-3                           gtable[strip]
## 40  2 ( 7- 7,11-11)  strip-t-4                           gtable[strip]
## 41  2 ( 7- 7,13-13)  strip-t-5                           gtable[strip]
## 42  2 ( 7- 7,15-15)  strip-t-6                           gtable[strip]
## 43  2 ( 7- 7,17-17)  strip-t-7                           gtable[strip]
## 44  2 ( 7- 7,19-19)  strip-t-8                           gtable[strip]
## 45  2 ( 7- 7,21-21)  strip-t-9                           gtable[strip]
## 46  2 ( 7- 7,23-23) strip-t-10                           gtable[strip]
## 47  2 ( 7- 7,25-25) strip-t-11                           gtable[strip]
```

```
## 48  4 ( 5- 5, 5-25)     xlab-t                          zeroGrob[NULL]
## 49  5 (10-10, 5-25)     xlab-b                          zeroGrob[NULL]
## 50  6 ( 8- 8, 3- 3)     ylab-l                          zeroGrob[NULL]
## 51  7 ( 8- 8,27-27)     ylab-r                          zeroGrob[NULL]
## 52  8 ( 4- 4, 5-25)   subtitle zeroGrob[plot.subtitle..zeroGrob.1948]
## 53  9 ( 3- 3, 5-25)      title    zeroGrob[plot.title..zeroGrob.1947]
## 54 10 (11-11, 5-25)    caption  zeroGrob[plot.caption..zeroGrob.1950]
## 55 11 ( 2- 2, 2- 2)        tag      zeroGrob[plot.tag..zeroGrob.1949]
```

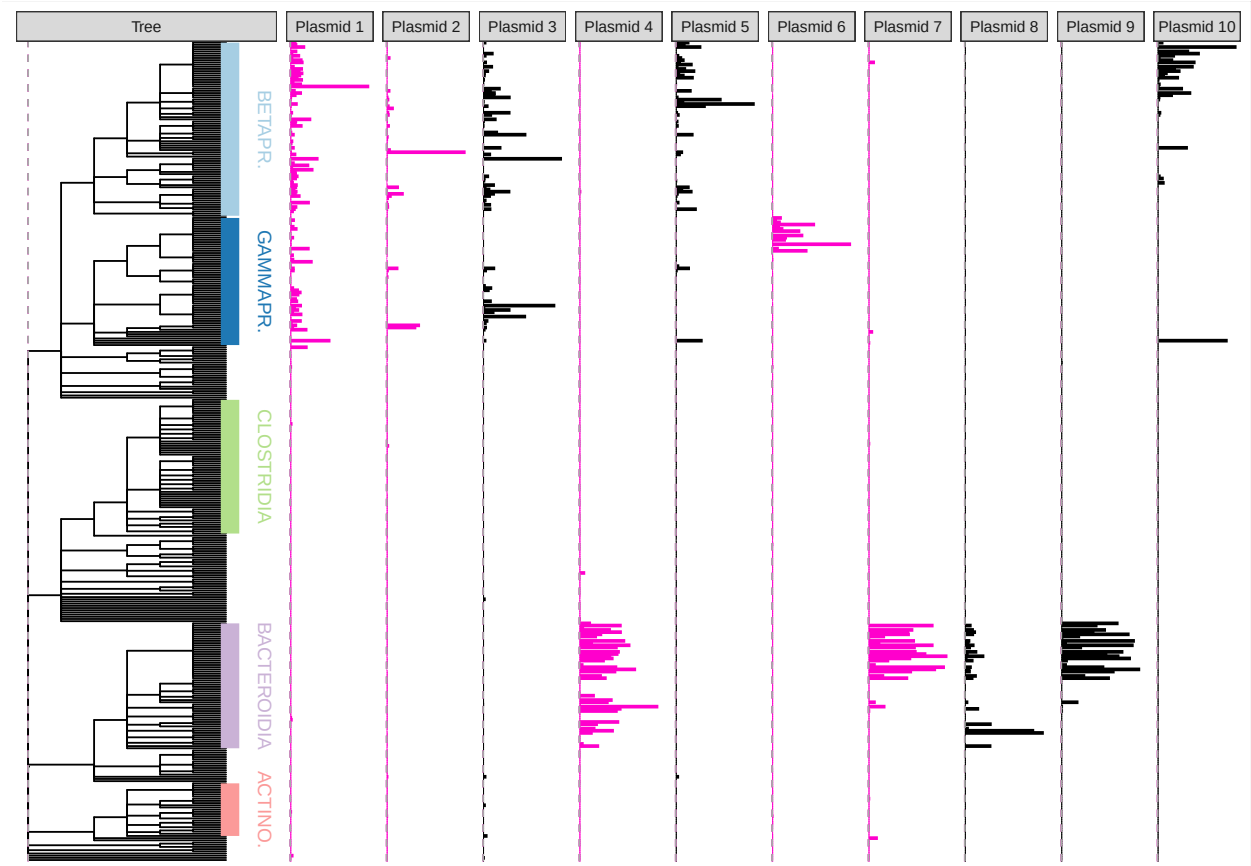
```
str(gt)
```

```
## gtable, containing
## grobs (55) : chr [1:55] "rect[plot.background..rect.1952]" "gTree[panel-1.gTree.1709]" ...
## layout :
## 'data.frame':    55 obs. of  7 variables:
##  $ t   : num  1 8 8 8 8 8 8 8 8 ...
##  $ l   : num  1 5 7 9 11 13 15 17 19 21 ...
##  $ b   : num  13 8 8 8 8 8 8 8 8 ...
##  $ r   : num  29 5 7 9 11 13 15 17 19 21 ...
##  $ z   : num  0 1 1 1 1 1 1 1 1 ...
##  $ clip: chr  "on" "on" "on" "on" ...
##  $ name: chr  "background" "panel-1-1" "panel-1-2" "panel-1-3" ...
## widths :
## unit vector of length 29
## heights :
## unit vector of length 13
## respect :
##  logi FALSE
## rownames :
##  NULL
## name :
##  chr "layout"
## gp :
##  NULL
## vp :
##  NULL
```

```
gt$layout$l[grep('2', gt$layout$name)] # you want to find the column specific to panel-2
```

```
## [1] 7 7 7 7
```

```
gt$widths[5] = 3*gt$widths[5] # in this case it was colmun 7 - reduce the width by a half
grid.draw(gt) # plot with grid draw
```




```
#ggsave("FIGURES_NC/Fig_4.pdf")

##############################
```

Session info

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Australia.1252  LC_CTYPE=English_Australia.1252
## [3] LC_MONETARY=English_Australia.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Australia.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
```

```
##
## other attached packages:
##  [1] picante_1.8.2        nlme_3.1-152         sjPlot_2.8.11
##  [4] performance_0.10.0   ggnetwork_0.5.10     ggthemes_4.2.4
##  [7] ggstatsplot_0.9.5    ggrepel_0.9.1        here_1.0.1
## [10] viridis_0.6.2        viridisLite_0.4.1    ggridges_0.5.4
## [13] RColorBrewer_1.1-3   gtable_0.3.1         ggplotify_0.1.0
## [16] ggtree_3.0.4         jntools_0.1.0        gridExtra_2.3
## [19] ggpubr_0.4.0         qgraph_1.9.2         scales_1.2.1
## [22] intergraph_2.0-2     igraph_1.4.2         metagMisc_0.0.4
## [25] forcats_0.5.2        stringr_1.4.1        dplyr_1.0.10
## [28] purrr_0.3.4          readr_2.1.3          tidyr_1.2.1
## [31] tibble_3.1.8         ggplot2_3.4.2        tidyverse_1.3.2
## [34] ggsci_2.9            expss_0.10.7         reshape2_1.4.4
## [37] bipartiteD3_0.3.0    bipartite_2.17       sna_2.7
## [40] network_1.18.0       statnet.common_4.7.0 vegan_2.5-7
## [43] lattice_0.20-44      permute_0.9-7        phylosignal_1.3
## [46] phangorn_2.8.1       ape_5.7-1            phyloseq_1.36.0
##
## loaded via a namespace (and not attached):
##   [1] estimability_1.4.1     lavaan_0.6-12          coda_0.19-4
##   [4] knitr_1.40             multcomp_1.4-20        data.table_1.14.2
##   [7] rpart_4.1-15           RCurl_1.98-1.5         generics_0.1.3
##  [10] BiocGenerics_0.38.0    microbiome_1.14.0      cowplot_1.1.1
##  [13] TH.data_1.1-1          correlation_0.8.3      tzdb_0.3.0
##  [16] phylobase_0.8.10       webshot_0.5.4          xml2_1.3.3
##  [19] lubridate_1.8.0        httpuv_1.6.6           assertthat_0.2.1
##  [22] gargle_1.2.1           xfun_0.33              jquerylib_0.1.4
##  [25] hms_1.1.2              evaluate_0.17          promises_1.2.0.1
##  [28] fansi_0.5.0            progress_1.2.2         dbplyr_2.2.1
##  [31] readxl_1.4.1           DBI_1.1.3              htmlwidgets_1.6.2
##  [34] googledrive_2.0.0      stats4_4.1.1           paletteer_1.4.1
##  [37] ellipsis_0.3.2         crosstalk_1.2.0        ggnewscale_0.4.9
##  [40] backports_1.4.1        pbivnorm_0.6.0         insight_0.18.5
##  [43] deldir_1.0-6           websocket_1.4.1        vctrs_0.5.0
##  [46] Biobase_2.52.0         sjlabelled_1.2.0       abind_1.4-5
##  [49] cachem_1.0.6           withr_2.5.0            checkmate_2.1.0
##  [52] emmeans_1.8.1-1        treeio_1.16.2          fdrtool_1.2.17
##  [55] prettyunits_1.1.1      mnormt_2.1.0           cluster_2.1.2
##  [58] dotCall64_1.0-2        lazyeval_0.2.2         crayon_1.5.2
##  [61] labeling_0.4.2         pkgconfig_2.0.3        GenomeInfoDb_1.28.4
##  [64] statsExpressions_1.3.4 nnet_7.3-16            rlang_1.1.1
##  [67] lifecycle_1.0.3        sandwich_3.0-2         downloader_0.4
##  [70] seqinr_4.2-16          modelr_0.1.9           adegenet_2.1.8
##  [73] cellranger_1.1.0       rprojroot_2.0.3        matrixStats_0.62.0
##  [76] datawizard_0.6.2       Matrix_1.5-1           aplot_0.1.8
##  [79] carData_3.0-5          Rhdf5lib_1.14.2        boot_1.3-28
##  [82] zoo_1.8-11             reprex_2.0.2           base64enc_0.1-3
##  [85] processx_3.7.0         googlesheets4_1.0.1    png_0.1-7
##  [88] parameters_0.19.0      bitops_1.0-7           rncl_0.8.6
##  [91] spam_2.9-1             rhdf5filters_1.4.0     Biostrings_2.60.2
##  [94] jpeg_0.1-9             rstatix_0.7.0          gridGraphics_0.5-1
##  [97] ggeffects_1.1.3        S4Vectors_0.30.1       ggsignif_0.6.4
## [100] magrittr_2.0.1         plyr_1.8.6             zlibbioc_1.38.0
```

```
## [103] compiler_4.1.1         lme4_1.1-30            cli_3.4.1
## [106] ade4_1.7-18            XVector_0.32.0         ps_1.7.1
## [109] patchwork_1.1.2        pbapply_1.5-0          htmlTable_2.4.1
## [112] Formula_1.2-4          MASS_7.3-57            mgcv_1.8-36
## [115] tidyselect_1.2.0       stringi_1.7.4          highr_0.9
## [118] webshot2_0.1.0         yaml_2.3.5             latticeExtra_0.6-30
## [121] sass_0.4.2             fastmatch_1.1-3        tools_4.1.1
## [124] parallel_4.1.1         rstudioapi_0.14        uuid_1.1-0
## [127] foreach_1.5.2          foreign_0.8-81         RNeXML_2.4.7
## [130] Rtsne_0.16             farver_2.1.1           digest_0.6.29
## [133] shiny_1.7.2            quadprog_1.5-8         Rcpp_1.0.7
## [136] car_3.1-0              broom_1.0.1            later_1.3.0
## [139] httr_1.4.4             psych_2.2.9            effectsize_0.8.1
## [142] sjstats_0.18.1         colorspace_2.0-2       rvest_1.0.3
## [145] XML_3.99-0.10          fs_1.5.2               IRanges_2.26.0
## [148] splines_4.1.1          fields_14.1            yulab.utils_0.0.5
## [151] rematch2_2.1.2         tidytree_0.4.1         multtest_2.48.0
## [154] xtable_1.8-4           nloptr_2.0.3           jsonlite_1.7.2
## [157] corpcor_1.6.10         glasso_1.11            zeallot_0.1.0
## [160] ggfun_0.0.7            chromote_0.1.1         R6_2.5.1
## [163] Hmisc_4.7-1            pillar_1.8.1           htmltools_0.5.5
## [166] mime_0.12              DT_0.25                minqa_1.2.4
## [169] glue_1.4.2             fastmap_1.1.0          codetools_0.2-18
## [172] maps_3.4.0             mvtnorm_1.1-3          utf8_1.2.2
## [175] bslib_0.4.0            curl_4.3.3             gtools_3.9.3
## [178] adephylo_1.1-11        interp_1.1-3           survival_3.2-11
## [181] rmarkdown_2.17         biomformat_1.20.0      munsell_0.5.0
## [184] rhdf5_2.36.0           GenomeInfoDbData_1.2.6 iterators_1.0.14
## [187] sjmisc_2.8.9           haven_2.5.1            bayestestR_0.13.0
```