Plasmid Network Analysis

Alice Risely

21/01/2021

A wastewater plasmid-host network is dominated by specialist plasmids

Alice Risely, Benno I. Simmons, Angus Buckling, Dirk Sanders*

Abstract

Plasmids are ubiquitous and important vectors for horizontal gene transfer; however, little is known about the structure of interaction networks between plasmids and their hosts in natural environments. Here we analyse a natural host-plasmid network extracted from wastewater samples. We found that plasmids were highly specific to their bacterial hosts, yet a small number were super generalists that connected the entire network, allowing inter-class horizontal gene transfer and indirect interactions cross broad taxonomic scales. Beta and Gamma proteobacteria exhibited more generalist interactions with plasmids, and this may explain the greater number of antimicrobial resistance genes associated with these classes.

Load packages

```
library(phyloseq)
library(ape)
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(bmotif)
library(ggcorrplot)
library(viridis)
library(tinytex)
```

Import data

This imports the association table between bacteria and plasmids (without any filters), information on host taxonomy, and the phylogenetic tree for the 191 bacteria included in this study. Association tables and taxonomic information are stored together in a phyloseq object (package phyloseq).

```
phylo_merged<-readRDS("DATA/plasmid_50pc_97pc_unmerged.RDS") #phyloseq object with host-plasmid count t
taxonomy <- read.csv("DATA/taxonomy_phylophlan.csv", sep=",") #taxonomy for genome clusters
tr<-read.tree("DATA/phylophlan2.tre.treefile") #phylo tree for genome clusters
```

Add taxonomic classification to the phyloseq object.

```
sample_data(phylo_merged)$Phylum<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_column
sample_data(phylo_merged)$Class<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_column =
sample_data(phylo_merged)$Order<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_column =
sample_data(phylo_merged)$Family<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_column
sample_data(phylo_merged)$Genus<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_column =
sample_data(phylo_merged)$Species<-vlookup(sample_data(phylo_merged)$feature.id, taxonomy, lookup_colum

phylo_merged
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2770 taxa and 191 samples ]
## sample_data() Sample Data:       [ 191 samples by 24 sample variables ]
## tax_table()   Taxonomy Table:    [ 2770 taxa by 7 taxonomic ranks ]
```

```
# 2770 plasmid contigs and 191 bacterial hosts
```

The full interaction dataset contains 2770 potential plasmid contigs, but many of these are represented by just a few interactions and may be due to error. We therefore filter dataset for interactions represented by at least 50 known connections to focus on 249 plasmids that are commonly represented - this does not change overall outcome.

```
phylo_filtered<-prune_taxa(taxa_sums(phylo_merged) > 50, phylo_merged)

phylo_filtered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 249 taxa and 191 samples ]
## sample_data() Sample Data:       [ 191 samples by 24 sample variables ]
## tax_table()   Taxonomy Table:    [ 249 taxa by 7 taxonomic ranks ]
```

```
# This leaves 249 plasmids that are highly reliable
```

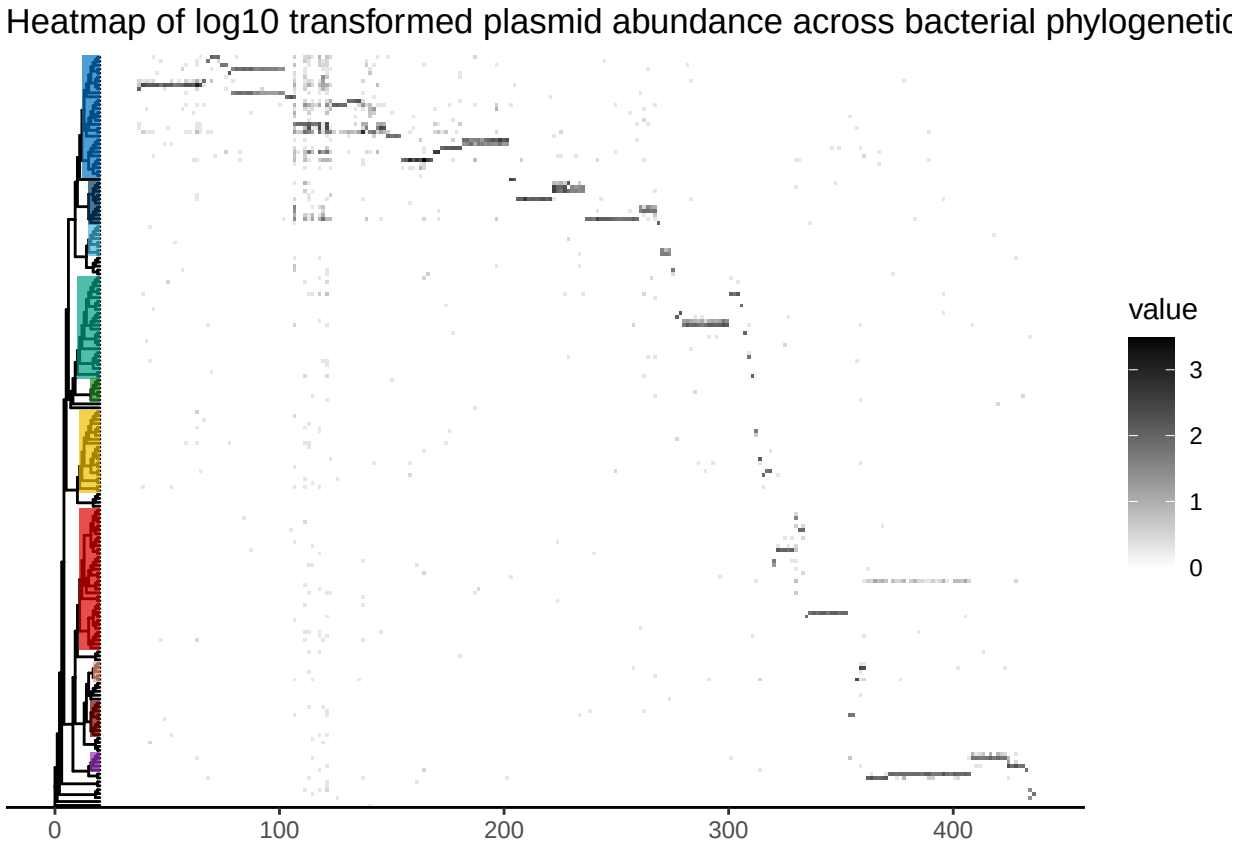
Visualising raw data

```
## extract count table
otu_table<-data.frame(phylo_filtered@otu_table@.Data)

### remove tips that don't have data
droptips<-data.frame(tr$tip.label)
droptips$present<-tr$tip.label %in% row.names(otu_table)
droptips<-subset(droptips, present==F)
droptips<-as.character(droptips$tr.tip.label)
tr1<-drop.tip(tr, droptips)
tree_rooted<-root(tr1, outgroup = "cluster.2") # root tree with archaea MAG

## first order plasmid contigs by which genome cluster they were assigned to
## and order these by the order of phylo tree tips

TAX<-data.frame(phylo_filtered@tax_table@.Data)
tree_tips_order<-get_tips_in_ape_plot_order(tree_rooted)

### for unmerged dataset
tree_tips_order<-tree_tips_order[tree_tips_order %in% unique(TAX$Assigned_genome_cluster)]
TAX$Assigned_genome_cluster<-factor(TAX$Assigned_genome_cluster, levels = tree_tips_order) # for
TAX<-TAX[order(TAX$Assigned_genome_cluster),]

colorder<-as.character(TAX$contig_id)
otu_table = otu_table %>% select(colorder)

#colours
mypal =     pal_jco("default", alpha = 1)(8)
mypal1 =    pal_locuszoom("default", alpha = 1)(8)
mypal2 =    pal_npg("nrc", alpha = 1)(8)
mypal3 =    pal_uchicago("dark", alpha = 1)(8)


# generate tree with classes labelled as different colours
p <- ggtree(tree_rooted, branch.length = "none") +
  theme_tree2()+
  geom_hilight(node=199, fill=mypal[1], alpha=0.7) + #betaproteobacteri
  geom_hilight(node=202, fill=mypal[6], alpha=0.7) + #gamma pr
  geom_hilight(node=347, fill=mypal1[4], alpha=0.7) + # epsilon

  geom_hilight(node=316, fill=mypal2[3], alpha=0.7) + #Bacteroidaceae
  geom_hilight(node=341, fill= "forestgreen", alpha=0.7) + #Flavobacteri

  geom_hilight(node=219, fill=mypal2[8], alpha=0.7) + # clostridia
  geom_hilight(node=271, fill=mypal2[5], alpha=0.7)+ #bacilli
  geom_hilight(node=258, fill=mypal3[1], alpha=0.7)+ #Negativicutes

  geom_hilight(node=290, fill=mypal[2], alpha=0.7) + #actinobacteria
```

```
  geom_hilight(node=280, fill=mypal1[6], alpha=0.7) #fusobacteria

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

## heatmap

otu_table1<-log10(otu_table+1)
phylo_pa<-phyloseq_standardize_otu_abundance(phylo_filtered, method = "pa")
otu_table3<-data.frame(phylo_pa@otu_table@.Data)
otu_table3 = otu_table3 %>% select(colorder)
```

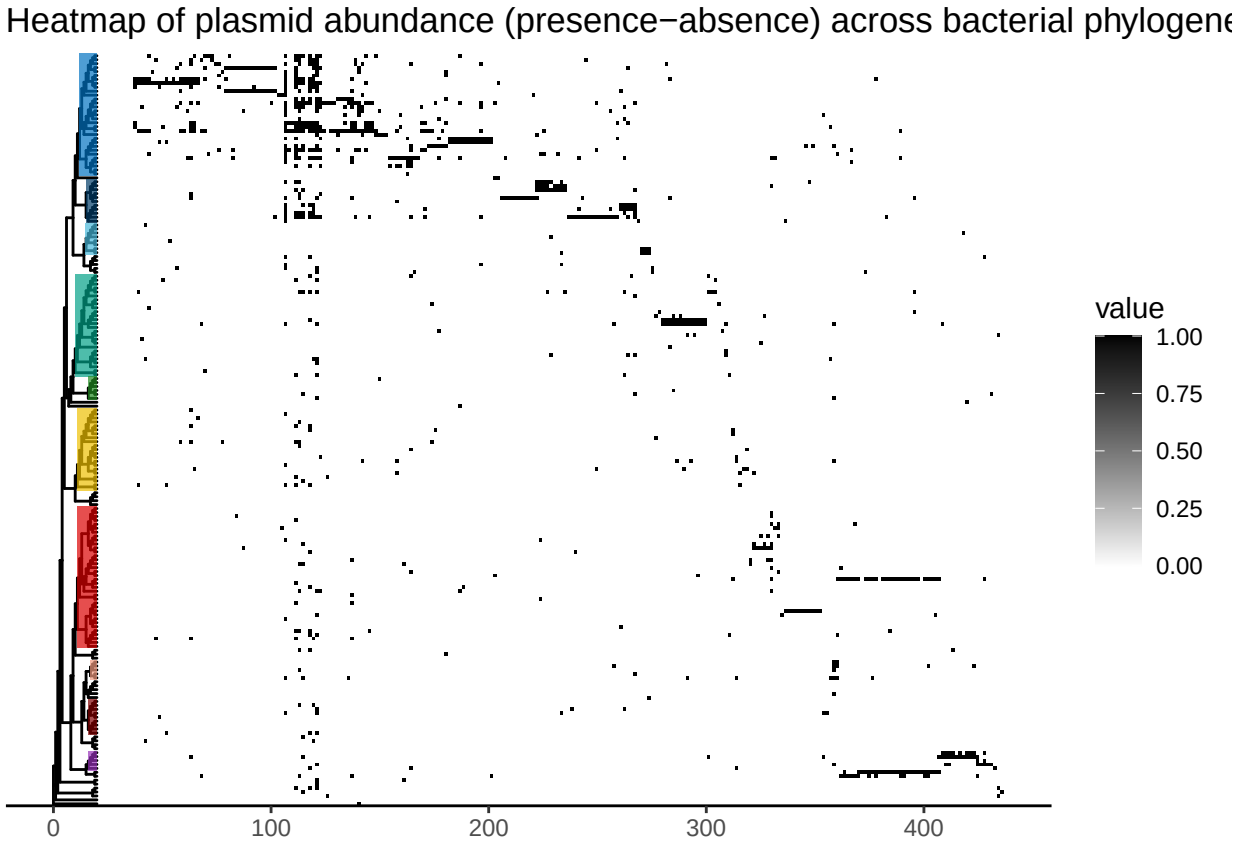
Heatmap (log transformed plasmids abundance)

```
gheatmap(p, otu_table1, offset=16, width=20,
             low= "white", high = "black",
             colnames=FALSE,
             color = NULL) +
  ggtitle("Heatmap of log10 transformed plasmid abundance across bacterial phylogenetic tree")
```



Heatmap (presence-absence)

```
gheatmap(p, otu_table3, offset=16, width=20,
             low= "white", high = "black",
             colnames=FALSE,
             color = NULL) +
  ggtitle("Heatmap of plasmid abundance (presence-absence) across bacterial phylogenetic tree")
```



Cluster networks

Make list of phyloseq objects per class so these can be analysed seperately.

```
### look for the most common classes in the data

Class_freq<-data.frame(table(sample_data(phylo_filtered)$Class))
Class_freq<-Class_freq[order(-Class_freq$Freq),]
head(Class_freq, 10) # most common bacterial classes
```

```
##                       Var1 Freq
## 24           c__Clostridia   36
## 5     c__Betaproteobacteria   31
## 4           c__Bacteroidia   24
## 1        c__Actinobacteria   21
```

```
## 30   c__Gammaproteobacteria   11
## 26 c__Epsilonproteobacteria    8
## 28        c__Flavobacteriia    8
## 3               c__Bacilli    6
## 33         c__Negativicutes    6
## 29         c__Fusobacteriia    5
```

```
## keep only classes with 10 or more taxa

classes_to_keep<-subset(Class_freq, Freq >10 )
classes_to_keep<-as.character(classes_to_keep$Var1)
sample_data(phylo_filtered)$MajorClass<-sample_data(phylo_filtered)$Class %in% classes_to_keep
phylo_classes<-subset_samples(phylo_filtered, MajorClass == TRUE)

## remove taxa that no longer occur
phylo_classes<-prune_taxa(taxa_sums(phylo_classes) > 0, phylo_classes)

### now split phyloseq object into list of x different objects, by class
phylo_by_class<-metagMisc::phyloseq_sep_variable(phylo_classes, "Class", drop_zeroes = T)

#list of new object seperated by class
phylo_by_class
```

```
## $c__Actinobacteria
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 37 taxa and 21 samples ]
## sample_data() Sample Data:       [ 21 samples by 25 sample variables ]
## tax_table()   Taxonomy Table:    [ 37 taxa by 7 taxonomic ranks ]
##
## $c__Bacteroidia
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 54 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 25 sample variables ]
## tax_table()   Taxonomy Table:    [ 54 taxa by 7 taxonomic ranks ]
##
## $c__Betaproteobacteria
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 135 taxa and 31 samples ]
## sample_data() Sample Data:       [ 31 samples by 25 sample variables ]
## tax_table()   Taxonomy Table:    [ 135 taxa by 7 taxonomic ranks ]
##
## $c__Clostridia
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 78 taxa and 36 samples ]
## sample_data() Sample Data:       [ 36 samples by 25 sample variables ]
## tax_table()   Taxonomy Table:    [ 78 taxa by 7 taxonomic ranks ]
##
## $c__Gammaproteobacteria
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 64 taxa and 11 samples ]
## sample_data() Sample Data:       [ 11 samples by 25 sample variables ]
## tax_table()   Taxonomy Table:    [ 64 taxa by 7 taxonomic ranks ]
```

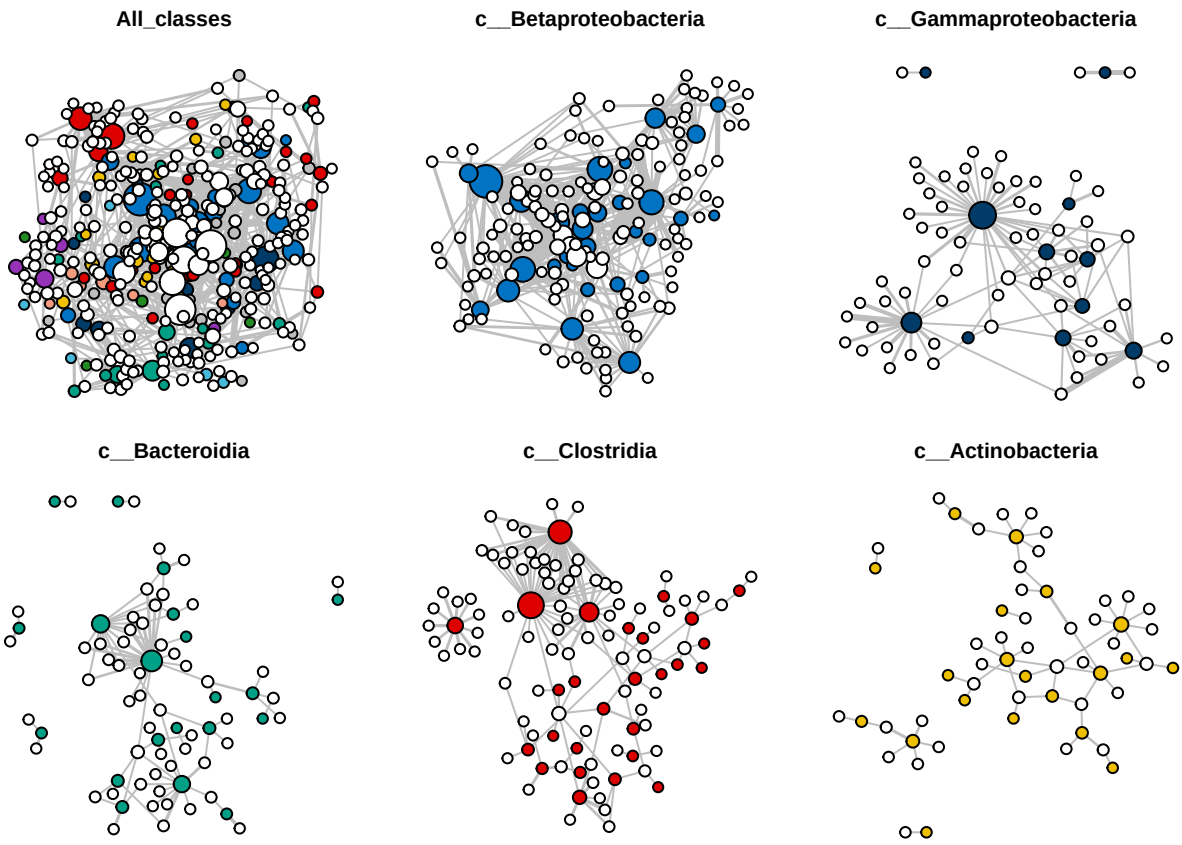
```
######## order by phylum

phylo_by_class<-phylo_by_class[c("c__Betaproteobacteria",  "c__Gammaproteobacteria",
  "c__Bacteroidia",
  "c__Clostridia",
  "c__Actinobacteria")]

phylo_by_class$All_classes<-phylo_filtered # add full network to list

phylo_by_class<-phylo_by_class[c("All_classes", "c__Betaproteobacteria",  "c__Gammaproteobacteria",
  "c__Bacteroidia",
  "c__Clostridia",
  "c__Actinobacteria")]
```

Make list of cluster networks per class and plot

```
uniq<-names(phylo_by_class)

net_list<-list()
network_df_vertices_list<-list()

par(mfrow=c(2,3))

for (i in 1:length(uniq)){
 phylo<- phylo_by_class[[i]]
  otu_table<-data.frame(phylo@otu_table@.Data)
  otu_matrix<-as.matrix(otu_table)
  row.names(otu_matrix)<-row.names(otu_table)
  otu_long<-melt(otu_matrix, na.rm = T)
  otu_long<-subset(otu_long, value !=0)
  head(otu_long)
  names(otu_long)<-c("host", "plasmid", "count")

   ### nodes
    sources <- otu_long %>%
    distinct(host) %>%
    rename(label = host)

  destinations <- otu_long%>%
    distinct(plasmid) %>%
    rename(label = plasmid)

  nodes <- full_join(sources, destinations, by = "label")
  nodes <- nodes %>% rowid_to_column("id")
  nodes$label<-as.character(nodes$label)

  ##edges
  per_route <- otu_long[,c(1:3)]
  names(per_route)[3]<-"weight"

  edges <- per_route %>%
    left_join(nodes, by = c("host" = "label")) %>%
    rename(from = id)
```

```
edges <- edges %>%
  left_join(nodes, by = c("plasmid" = "label")) %>%
  rename(to = id)

names_reference<-edges
 edges <- select(edges, from, to, weight)

#################### make into network object

routes_network <- network(edges, vertex.attr = nodes, matrix.type = "edgelist", ignore.eval = FALSE)
igraph_net<-intergraph::asIgraph(routes_network)
igraph_net<-as.undirected(igraph_net)

### add metadata

V(igraph_net)$contig_id<-vlookup(V(igraph_net)$id, nodes, lookup_column = "id", result_column = "label
V(igraph_net)$Type<-ifelse(V(igraph_net)$contig_id %in% sources$label, "Host", "Plasmid")

##### add network stats per node

V(igraph_net)$closeness <- igraph::closeness(igraph_net) #closeness centrality
V(igraph_net)$betweenness <- igraph::betweenness(igraph_net) #betweenness centrality
V(igraph_net)$degree<-igraph::degree(igraph_net) #degree
V(igraph_net)$w_degree<-igraph::strength(igraph_net) # weighted degree
V(igraph_net)$hubbinessnet.hs <- igraph::hub_score(igraph_net)$vector


 ### add taxonomy

taxonomy<-data.frame(sample_data(phylo))
V(igraph_net)$class<-as.character(vlookup(V(igraph_net)$contig_id, taxonomy, lookup_column = "feature
V(igraph_net)$Species<-as.character(vlookup(V(igraph_net)$contig_id, taxonomy, lookup_column = "featu
V(igraph_net)$Genus<-as.character(vlookup(V(igraph_net)$contig_id, taxonomy, lookup_column = "feature


## dataframe for vertices only

network_df_vertices<-as_data_frame(igraph_net, what = "vertices")
network_df_vertices$Network<-uniq[i]
network_df_vertices_list[[i]]<-network_df_vertices

############ set colour and shapes


mypal =   pal_jco("default", alpha = 1)(8)
mypal1 =      pal_locuszoom("default", alpha = 1)(8)
mypal2 =      pal_npg("nrc", alpha = 1)(8)
mypal3 =      pal_uchicago("dark", alpha = 1)(8)

V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Betaproteobacteria", mypal[1], NA)
V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Gammaproteobacteria", mypal[6], V(igraph_net)$
V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Epsilonproteobacteria",  mypal1[4], V(igraph_n
V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Alphaproteobacteria",  "gray", V(igraph_net)$c
```

```
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Bacteroidia",  mypal2[3], V(igraph_net)$colour
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Flavobacteriia",  "forestgreen", V(igraph_net)
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Clostridia",  mypal2[8], V(igraph_net)$colour
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Bacilli",  mypal2[5], V(igraph_net)$colour)
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Negativicutes",  mypal3[1], V(igraph_net)$colo
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Actinobacteria",  mypal[2], V(igraph_net)$colo
  V(igraph_net)$colour<-ifelse(V(igraph_net)$class == "c__Fusobacteriia",  mypal1[6], V(igraph_net)$colo
  V(igraph_net)$colour<-ifelse(is.na(V(igraph_net)$colour),  "gray", V(igraph_net)$colour)
  V(igraph_net)$colour<-ifelse(V(igraph_net)$Type=="Plasmid",  "white", V(igraph_net)$colour)


  par(mar=c(0,0,2,0)+0.1)

  plot(igraph_net,
    layout=layout_with_dh,
    vertex.size = (V(igraph_net)$degree/3) +6,
    vertex.label = NA,
    vertex.color = V(igraph_net)$colour,
    edge.width=(E(igraph_net)$weight/200)+1,
    edge.color = "gray")
  title(paste(uniq[i]),cex.main=1)


  ### add network to list

  net_list[[i]]<-igraph_net

}
```




```
#net_list

### list of networks
names(net_list)<-uniq
names(network_df_vertices_list)<-uniq
```

Generate table of network stats per major class

```
network_stats_list<-list()
uniq<-names(phylo_by_class)

#par(mfrow=c(2,2))

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

  phylo<- phylo_by_class[[i]]
  net.v<-data.frame(phylo@otu_table@.Data)
  # this will estimate all common network metrics but may take a while
  network_metrics<-data.frame(networklevel(net.v))
  network_metrics$Class<- uniq[i]
  # network_metrics$Phylum<- sample_data(phylo)$Phylum[1]
  network_metrics$Metric<-row.names(network_metrics)
  #network_metrics<-network_metrics[,c(4,1,2,3)]
```

```
  names(network_metrics)[2]<-"Stat"
  network_stats_list[[i]]<-network_metrics
}

network_stats_df<-do.call(rbind, network_stats_list)
network_stats_df<-network_stats_df[,c(3,2,1)]
names(network_stats_df)[3]<-"Stat"
names(network_stats_df)[2]<-"Class"

row.names(network_stats_df)<-1:nrow(network_stats_df)


network_stats_table<-subset(network_stats_df, Metric == "connectance" |Metric == "NODF" |Metric == "numl
row.names(network_stats_table)<-1:nrow(network_stats_table)

# make short format
network_stats_final <- spread(network_stats_table, Metric, Stat)
network_stats_final$connectance<-round(network_stats_final$connectance, 2)
network_stats_final$generality.HL<-round(network_stats_final$generality.HL, 2)
network_stats_final$NODF<-round(network_stats_final$NODF, 2)
names(network_stats_final)[5]<-"No.compartments"

network_stats_final
```

```
##                    Class connectance generality.HL  NODF No.compartments
## 1            All_classes        0.03          1.82  8.51               1
## 2      c__Actinobacteria        0.07          1.20  7.90               5
## 3         c__Bacteroidia        0.08          1.33 12.85               7
## 4  c__Betaproteobacteria        0.11          1.71 18.94               1
## 5          c__Clostridia        0.07          1.19 12.61               4
## 6 c__Gammaproteobacteria        0.15          1.22 22.89               3
```

Figure 1

```
network<-net_list$All_classes
network_stats<-as_data_frame(network, what = c("vertices"))
network_bacteria<-subset(network_stats, Type == "Host")

degree_df<-network_bacteria[,c("label", "degree")]
names(degree_df)[1]<-"id"


w_degree_df<-network_bacteria[,c("label", "w_degree")]
names(w_degree_df)[1]<-"id"
```

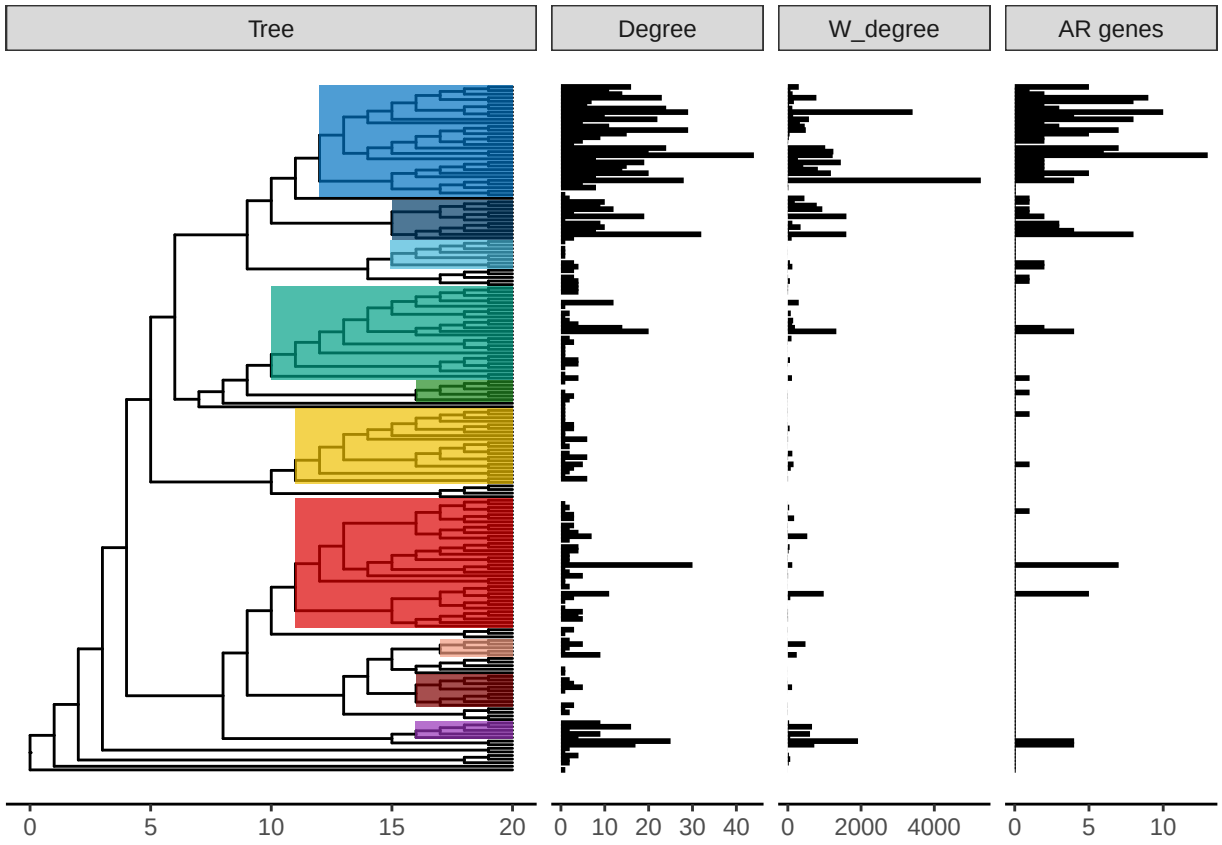
```
P2<-facet_plot(p+xlim_tree(5), panel='Degree', data = degree_df, geom=geom_segment, mapping=aes(x=0, xe
  stat = "identity", size=1,  color = "black")+theme_tree2()

P3<-facet_plot(P2+xlim_tree(5), panel='W_degree', data = w_degree_df, geom=geom_segment, mapping=aes(x=
  stat = "identity", size=1,  color = "black")+theme_tree2()
```

```
## antimicrobial genes

### here used the CARD database to run the plasmid sequences though (with loose thresholds) to identify

plasmid_gene_annotation <- read.csv("CARD_ARgene_annotation_plasmids.csv")[c(-1,-19,-20,-21)]

plasmid_gene_annotation<-plasmid_gene_annotation[,c("Contig", "contig_id", "Cut_Off", "ARO")]

plasmid_AR_count<-data.frame(table(plasmid_gene_annotation$contig_id))

plasmid_AR_count1<-data.frame(taxa_names(phylo_filtered))
names(plasmid_AR_count1)<-"PlasmidID"
plasmid_AR_count1$AR_gene_count<-vlookup(plasmid_AR_count1$PlasmidID, plasmid_AR_count, lookup_column =

plasmid_AR_count1[is.na(plasmid_AR_count1)] <- 0
plasmid_order_AR<-as.character(plasmid_AR_count1$PlasmidID)
otu_table<-data.frame(phylo_filtered@otu_table@.Data)
otu_table[1:5,1:5]
```

```
##             k141_1036310 k141_1023016 k141_2191729 k141_1554928 k141_1931562
## cluster.1              0            0            0            0            0
## cluster.10             0            0            0            0            0
## cluster.100            0            0            0            0            0
## cluster.101            0            0            0            0            0
## cluster.102            0            0            0            0            0
```

```
otu_table<-otu_table[,plasmid_order_AR] #just to make sure cols are in right order

## loop to go through otu table columns and replace ones with number of AR genes

otu_table_AR_UW<-list()
rownames<-nrow(otu_table)

for (i in 1:rownames){
  row<-otu_table[i,]
  row<-ifelse(row >1, 1, 0)
  row1<-row*plasmid_AR_count1$AR_gene_count
  otu_table_AR_UW[[i]]<-row1
}

unweighted_AR_df<-do.call(rbind, otu_table_AR_UW)

unweighted_rowsums<-data.frame(rowSums(unweighted_AR_df))
names(unweighted_rowsums)<-"AR_genes"
unweighted_rowsums$id<-row.names(unweighted_rowsums)
unweighted_rowsums<-unweighted_rowsums[,c(2,1)]


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

P4<-facet_plot(P3+xlim_tree(5), panel='AR genes', data = unweighted_rowsums, geom=geom_segment, mapping=
  stat = "identity", size=1,  color = "black")+theme_tree2()
```

```
################# figure 1a

gt = ggplot_gtable(ggplot_build(P4))
gt$widths[5] = 0.5*gt$widths[5] # in this case it was colmun 7 - reduce the width by a half
gt$widths[7] = 0.2*gt$widths[7] # in this case it was colmun 7 - reduce the width by a half
gt$widths[9] = 0.2*gt$widths[9] # in this case it was colmun 7 - reduce the width by a half
gt$widths[11] = 0.2*gt$widths[11] # in this case it was colmun 7 - reduce the width by a half
grid.draw(gt) # plot with grid draw
```



Unweighted bipartite network (Fig. 1c)

Here we use the bipartiteD3 package to generate interactive bipartite networks, but first need to massage the data into right format.

```
### generate long data table of interactions

otu_table<-data.frame(phylo_filtered@otu_table@.Data)
node_order_weighted<-OrderByCrossover(otu_table)
### long format
otu_matrix<-as.matrix(otu_table)
row.names(otu_matrix)<-row.names(otu_table)
otu_long<-melt(otu_matrix, na.rm = T)
otu_long<-subset(otu_long, value !=0) # remove zeros
```

```
names(otu_long)<-c("lower", "higher", "freq") #rename cols
head(otu_long)
```

```
##           lower       higher freq
## 83  cluster.173 k141_1036310    1
## 91  cluster.180 k141_1036310   39
## 129  cluster.41 k141_1036310    1
## 185  cluster.93 k141_1036310   20
## 199 cluster.105 k141_1023016    2
## 205 cluster.110 k141_1023016    1
```

```
### get taxonomy
taxtable<-data.frame(sample_data(phylo_filtered))[,c(1, 20)]

otu_long$webID<-vlookup(otu_long$lower, taxtable, lookup_column = "feature.id", result_column = "Class"
```

```
#### only keep main classes which have over 15 associations
Class_freq<-data.frame(table(otu_long$webID))
Class_freq<-Class_freq[order(-Class_freq$Freq),]
head(Class_freq, 15)
```

```
##                        Var1 Freq
## 5     c__Betaproteobacteria  446
## 24            c__Clostridia  145
## 30   c__Gammaproteobacteria  109
## 4             c__Bacteroidia   86
## 1          c__Actinobacteria   55
## 29          c__Fusobacteriia   40
## 3                c__Bacilli   20
## 2    c__Alphaproteobacteria   14
## 26 c__Epsilonproteobacteria   11
## 6             c__Caldilineae    8
## 34             c__Nitrospira    8
## 28         c__Flavobacteriia    7
## 33          c__Negativicutes    6
## 11               c__CFGB1464    5
## 15               c__CFGB1874    5
```

```
## keep only classes with 8 or more taxa (can edit)
classes_to_keep<-subset(Class_freq, Freq > 5)
classes_to_keep<-as.character(classes_to_keep$Var1)
otu_long$MajorClass<-otu_long$webID %in% classes_to_keep
otu_long_reduced<-subset(otu_long, MajorClass == TRUE)
head(otu_long_reduced)
```

```
##           lower       higher freq                  webID MajorClass
## 83  cluster.173 k141_1036310    1  c__Betaproteobacteria       TRUE
## 91  cluster.180 k141_1036310   39 c__Gammaproteobacteria       TRUE
## 185  cluster.93 k141_1036310   20 c__Gammaproteobacteria       TRUE
## 199 cluster.105 k141_1023016    2  c__Betaproteobacteria       TRUE
## 205 cluster.110 k141_1023016    1  c__Betaproteobacteria       TRUE
## 209 cluster.114 k141_1023016  714  c__Betaproteobacteria       TRUE
```

```
otu_long_reduced<-otu_long_reduced[,-5]
otu_long_reduced$webID<-factor(otu_long_reduced$webID)
names(otu_long_reduced)[4]<-"Class"
otu_long_reduced$webID<-"all"
```

Plotting bipartite networks

```
############################## plotting

## get in right format for plotting
bipartite::frame2webs(otu_long_reduced)-> plasmid_network_all

## need genrate vector of plasmid and bacteria order for plotting
# bacteria should be ordered by taxonomy and then number of associations
#plasmids should be ordered by the class to which they are majorly associated with,
#and then by number of associations

df<-bipartiteD3::List2DF(plasmid_network_all)

#Primary = bacteria
#Secondary = Plasmids

df$Class<-vlookup(df$Primary, taxtable, lookup_column = "feature.id", result_column = "Class")
df$Class<-factor(df$Class)

# To sort secondary/plasmids by  bacteria class they are mostly associatied with and total size:

df %>%
  group_by(Secondary, Class) %>%
  summarise(Total=sum(all))-> SortDf_s

SortDf_s2<-SortDf_s %>% group_by(Secondary) %>% top_n(1, Total)
SortDf_s2$Class<-factor(SortDf_s2$Class)

SortDf_s2$Class<-factor(SortDf_s2$Class, levels = c("c__Betaproteobacteria",  "c__Gammaproteobacteria",
  "c__Bacteroidia",  "c__Actinobacteria", "c__Clostridia", "c__Bacilli",  "c__Fusobacteriia", "c__Caldi

SortDf_s2 %>% arrange(Class,desc(Total))-> SortDf_s2

### sort bacteria/primary by class and number of associations

###

df %>%
  group_by(Primary) %>%
  summarise(Total=sum(all))-> SortDf_p

SortDf_p$Class<-vlookup(SortDf_p$Primary, taxtable, lookup_column = "feature.id", result_column = "Clas
SortDf_p$Class<-factor(SortDf_p$Class)
unique(SortDf_p$Class)
```

```
##  [1] c__Bacteroidia           c__Actinobacteria        c__Flavobacteriia
##  [4] c__Betaproteobacteria    c__Clostridia            c__Bacilli
```

```
##  [7] c__Alphaproteobacteria   c__Gammaproteobacteria   c__Negativicutes
## [10] c__Epsilonproteobacteria c__Caldilineae           c__Fusobacteriia
## [13] c__Nitrospira
## 13 Levels: c__Actinobacteria c__Alphaproteobacteria ... c__Nitrospira
```

```
SortDf_p$Class<-factor(SortDf_p$Class, levels = c("c__Betaproteobacteria",  "c__Gammaproteobacteria", "
  "c__Bacteroidia", "c__Flavobacteriia", "c__Actinobacteria", "c__Clostridia", "c__Bacilli","c__Negativ

SortDf_p  %>% arrange(Class,desc(Total))-> SortDf_p

positions<-match(SortDf_p$Primary, node_order_weighted$PrimaryOrder)

#node_order_weighted$SecondaryOrder[positions]
```

```
#not run
bipartite_D3(plasmid_network_all,
 SortSecondary = rev(SortDf_s2$Secondary),
  SortPrimary = rev(SortDf_p$Primary),
  colouroption = 'manual',
  NamedColourVector = ColoursTaxonomy,
  ColourBy = 1,
  MainFigSize = c(1500,1000),
  IndivFigSize = c(500,1200),
  Pad = 0.5,
  BarSize = 90,
  MinWidth = 0.5,
  PercentageDecimals = 1,
  Orientation = 'horizontal',
  filename = 'all_classes')
```

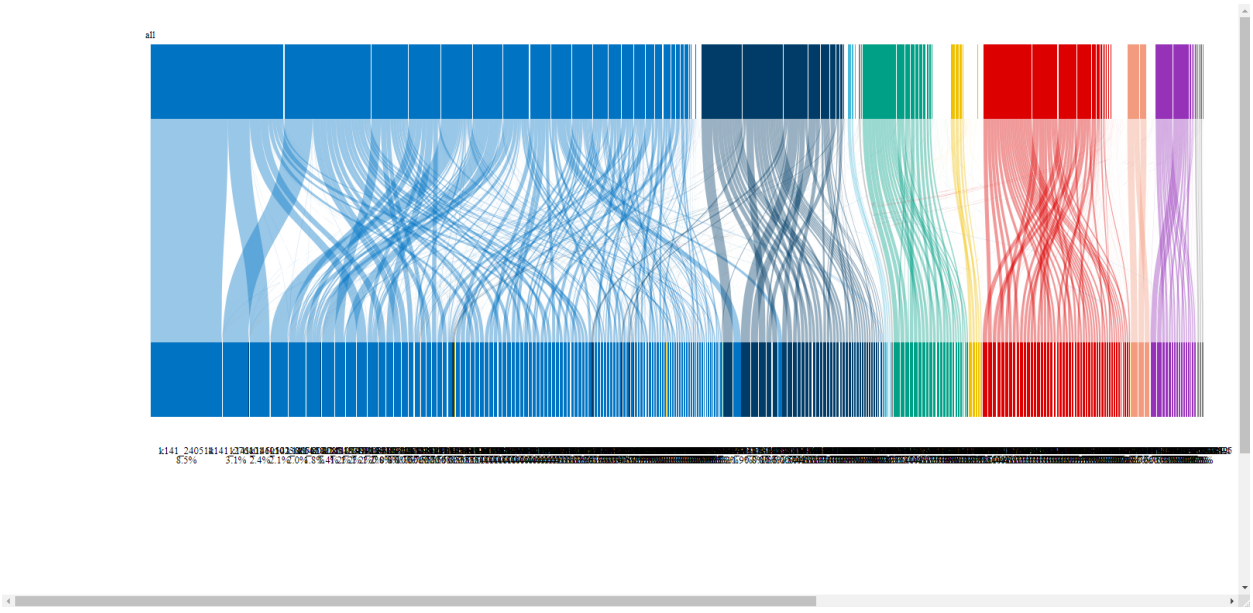


Figure 1: Weighted bipartite network

Unweighted bipartite network (Fig. 1c)

```
phylo_transformed <-metagMisc::phyloseq_standardize_otu_abundance(phylo_filtered, method = "pa")

### generate long data table of interactions
otu_table<-data.frame(phylo_transformed@otu_table@.Data)
#node_order_unweighted<-OrderByCrossover(otu_table)

### long format

otu_matrix<-as.matrix(otu_table)
row.names(otu_matrix)<-row.names(otu_table)
otu_long<-melt(otu_matrix, na.rm = T)
otu_long<-subset(otu_long, value !=0) # remove zeros
names(otu_long)<-c("lower", "higher", "freq") #rename cols

### get taxonomy

taxtable<-data.frame(sample_data(phylo_transformed))[,c(1, 20)]
otu_long$webID<-vlookup(otu_long$lower, taxtable, lookup_column = "feature.id", result_column = "Class"

#### only keep main classes which have over 15 associations
Class_freq<-data.frame(table(otu_long$webID))
Class_freq<-Class_freq[order(-Class_freq$Freq),]

## keep only classes with 8 or more taxa (can edit)
classes_to_keep<-subset(Class_freq, Freq > 5)
classes_to_keep<-as.character(classes_to_keep$Var1)
otu_long$MajorClass<-otu_long$webID %in% classes_to_keep
otu_long_reduced<-subset(otu_long, MajorClass == TRUE)
head(otu_long_reduced)
```

```
##           lower       higher freq                  webID MajorClass
## 83  cluster.173 k141_1036310    1  c__Betaproteobacteria       TRUE
## 91  cluster.180 k141_1036310    1 c__Gammaproteobacteria       TRUE
## 185  cluster.93 k141_1036310    1 c__Gammaproteobacteria       TRUE
## 199 cluster.105 k141_1023016    1  c__Betaproteobacteria       TRUE
## 205 cluster.110 k141_1023016    1  c__Betaproteobacteria       TRUE
## 209 cluster.114 k141_1023016    1  c__Betaproteobacteria       TRUE
```

```
otu_long_reduced<-otu_long_reduced[,-5]

otu_long_reduced$webID<-factor(otu_long_reduced$webID)

names(otu_long_reduced)[4]<-"Class"
otu_long_reduced$webID<-"all"

## get in right format for plotting
bipartite::frame2webs(otu_long_reduced)-> plasmid_network_all
```

```
#not run
bipartite_D3(plasmid_network_all,
 SortSecondary = rev(SortDf_s2$Secondary),
  SortPrimary = rev(SortDf_p$Primary),
  colouroption = 'manual',
  NamedColourVector = ColoursTaxonomy,
  ColourBy = 1,
  MainFigSize = c(1500,1000),
  IndivFigSize = c(500,1200),
  Pad = 0.5,
  BarSize = 90,
  MinWidth = 0.5,
  PercentageDecimals = 1,
  Orientation = 'horizontal',
  filename = 'all_classes')
```

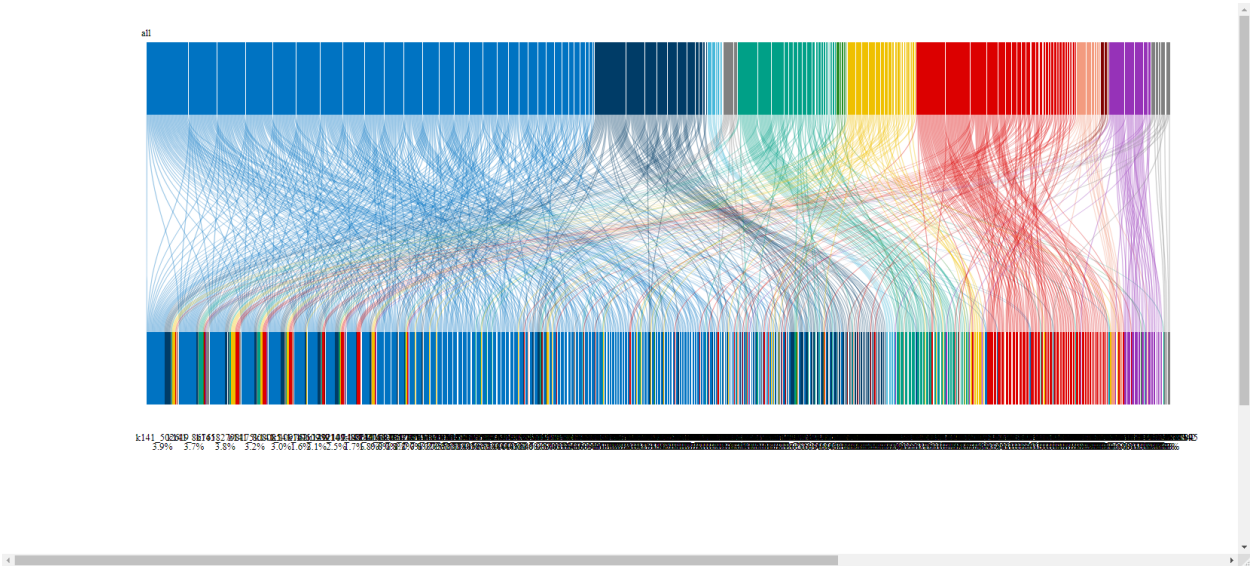


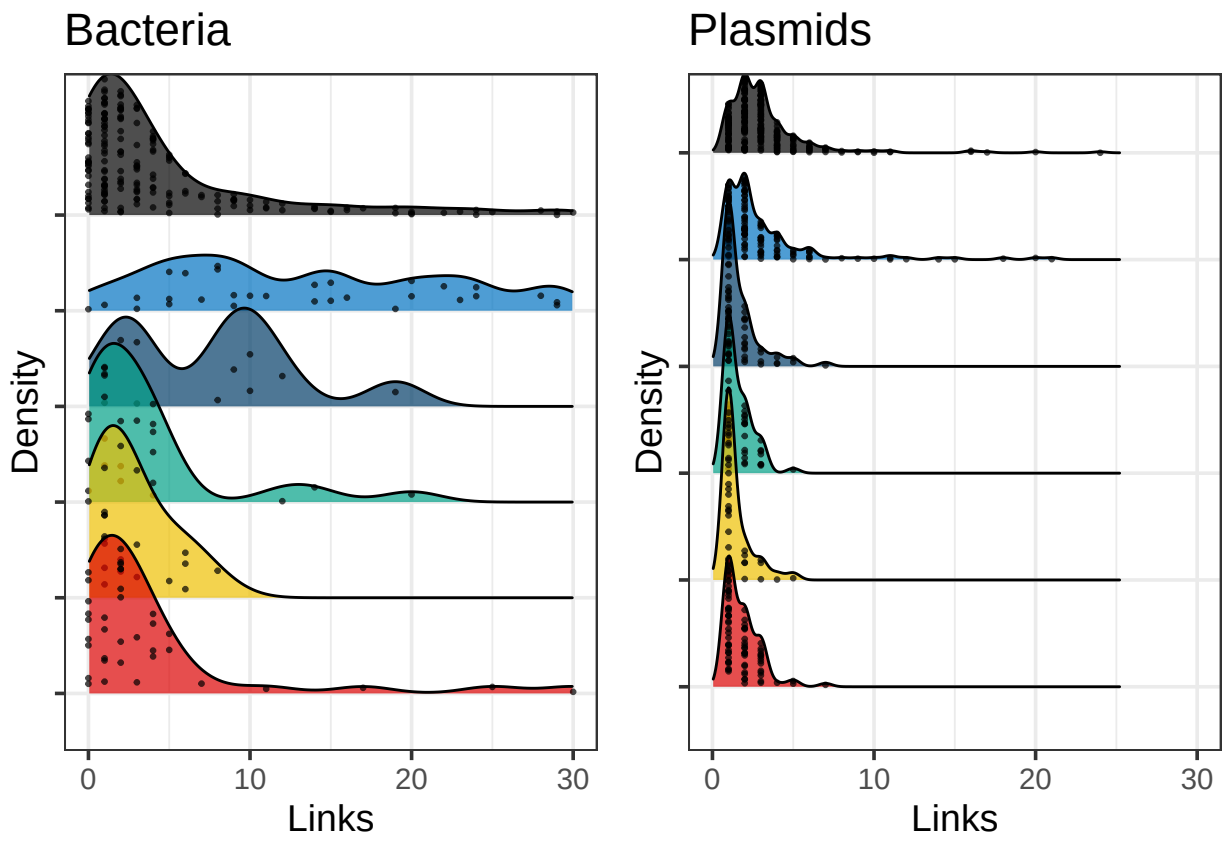
Figure 2: Unweighted bipartite network

Linkage distributions

```
#### how many hosts does each plasmid associate with?

# loop to save number of links per plasmid, across whole network and within network

uniq<-names(phylo_by_class)
list_all<-list()

for (i in 1:length(uniq)){
  phylo<- phylo_by_class[[i]]
  prev0 = apply(X = otu_table(phylo),
  MARGIN = ifelse(taxa_are_rows(phylo), yes = 1, no = 2),
  FUN = function(x){sum(x > 0)})
```

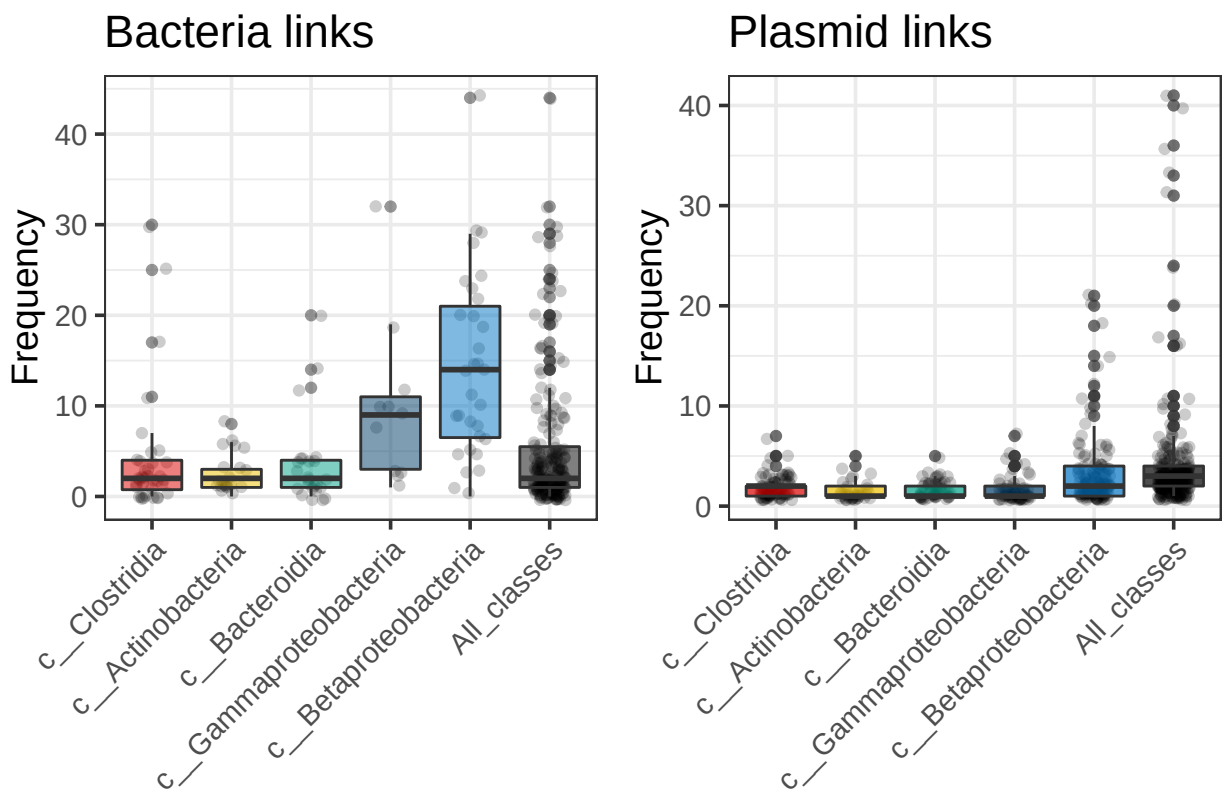
```
prevdf = data.frame(Prevalence = prev0, TotalAbundance = taxa_sums(phylo))
head(prevdf)

prevdf$Prevalence_rel<-(prevdf$Prevalence/length(unique(sample_data(phylo)$feature.id)))*100
prevdf$rel_abund<-(prevdf$TotalAbundance/(sum(prevdf$TotalAbundance)))*100
prevdf<-prevdf[order(-prevdf$Prevalence),] #sort by prevalence
head(prevdf, 20)
prevdf$class<-names(phylo_by_class)[[i]]
list_all[[i]]<-prevdf
}

names(list_all)<-uniq
prevalence_distributions<-do.call(rbind, list_all)
prevalence_distributions$class<-fct_rev(prevalence_distributions$class)


####### how many plasmids does each bacteria/host associate with?


uniq<-names(phylo_by_class)
list_alpha<-list()

for (i in 1:length(uniq)){
 phylo<- phylo_by_class[[i]]
 sample_data(phylo)$Observed<-phyloseq::estimate_richness(phylo, measures="Observed")
 sample_data(phylo)$Shannon<-phyloseq::estimate_richness(phylo, measures="Shannon")
 sample_data(phylo)$Observed<-sample_data(phylo)$Observed$Observed
 sample_data(phylo)$Shannon<-sample_data(phylo)$Shannon$Shannon

metadata<-data.frame(sample_data(phylo))[,c(1,20,26,27)]
alphadiversity<-data.frame(sample_data(phylo))[,c(1,20,26,27)]
alphadiversity<-alphadiversity[order(-alphadiversity$Observed),]
head(alphadiversity, 30)
alphadiversity$class<-uniq[i]
list_alpha[[i]]<-alphadiversity
}

names(list_alpha)<-uniq
alpha_df<-do.call(rbind, list_alpha)
alpha_df$class<-fct_rev(alpha_df$class)

linetypes<-c("dotted","dotted","dotted","dotted","dotted","dotted", "solid")

############## ggridges ##############################

mypal =     pal_jco("default", alpha = 1)(8)
mypal1 =    pal_locuszoom("default", alpha = 1)(8)
mypal2 =    pal_npg("nrc", alpha = 1)(8)
mypal3 =    pal_uchicago("dark", alpha = 1)(8)


prevalence_distributions$class<-factor(prevalence_distributions$class, levels = c("c__Clostridia", "c__
```

```
  "c__Bacteroidia",  "c__Gammaproteobacteria", "c__Betaproteobacteria", "All_classes"))

alpha_df$class<-factor(alpha_df$class, levels = c("c__Clostridia", "c__Actinobacteria",
  "c__Bacteroidia",  "c__Gammaproteobacteria", "c__Betaproteobacteria", "All_classes"))


colors<-c(mypal2[8], mypal[2], mypal2[3],  mypal[6], mypal[1], "black")

## comparison of linkage distribution with density ridges

P1<-ggplot(prevalence_distributions, aes(x = Prevalence, y = class, fill = class)) +
  geom_density_ridges(alpha = 0.7, jittered_points = T, point_size = 0.5, point_fill = "black")+
  xlim(0,30)+
  scale_linetype_manual(values = linetypes)+
  theme_bw(base_size = 14)+
  ggtitle("Plasmids")+
  scale_fill_manual(values = colors)+
  xlab("Links")+
  ylab("Density")+
  theme(axis.text.y=element_blank())+
  theme(legend.position = "none")


P2<-ggplot(alpha_df, aes(x = Observed, y = class, fill = class)) +
  geom_density_ridges(alpha = 0.7, jittered_points = T, point_size = 0.5, point_fill = "black")+
 xlim(0,30)+
  scale_linetype_manual(values = linetypes)+
  theme_bw(base_size = 14)+
  ggtitle("Bacteria")+
  scale_fill_manual(values = colors)+
  xlab("Links")+
  ylab("Density")+
  theme(axis.text.y=element_blank())+
  theme(legend.position = "none")


grid.arrange(P2,P1, ncol = 2)
```



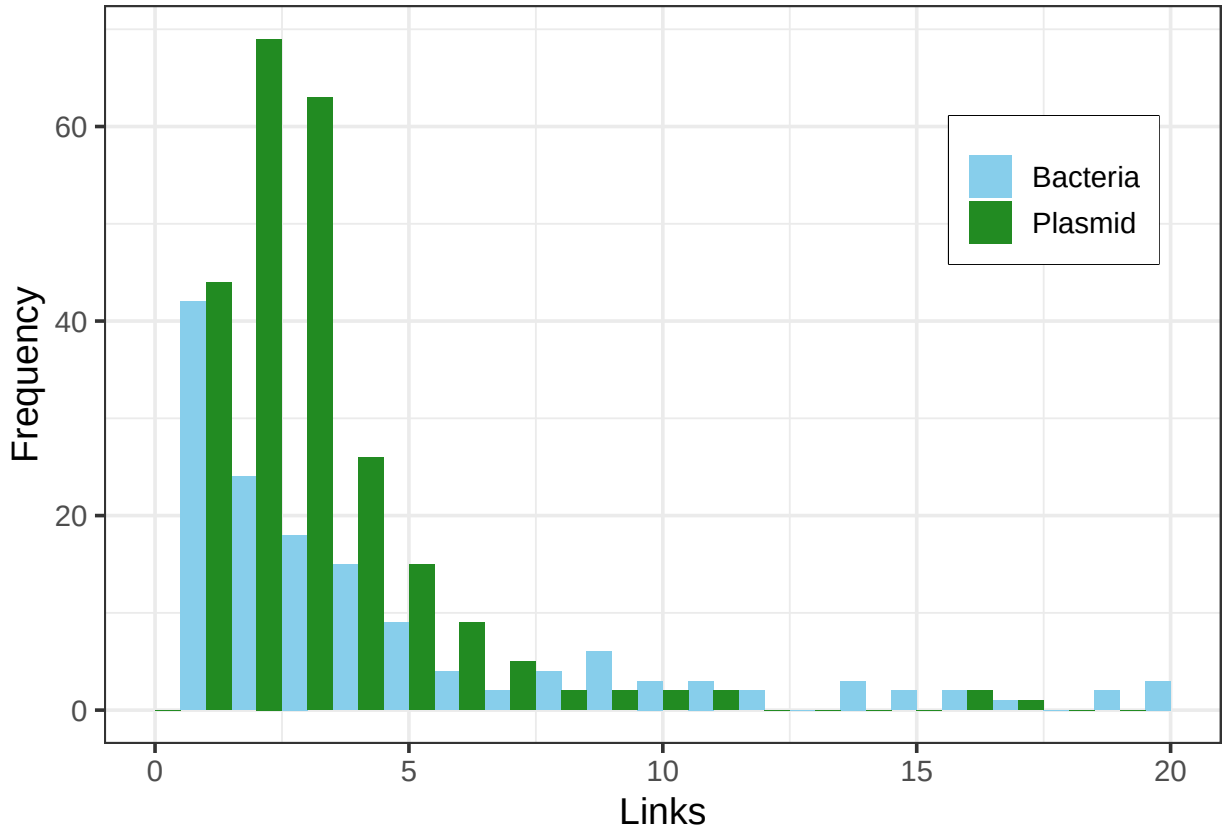

```
### comparison of linkage distribution with boxplots

P1<-ggplot(prevalence_distributions, aes(y = Prevalence, x = class, fill = class)) +
  geom_jitter( alpha = 0.2, width = 0.2)+
  geom_boxplot(alpha = 0.7)+
  scale_linetype_manual(values = linetypes)+
  theme_bw(base_size = 14)+
  ggtitle("Plasmid links")+
  scale_fill_manual(values = colors)+
  xlab("")+
  ylab("Frequency")+
 # theme(axis.text.y=element_blank())+
  theme(legend.position = "none")+
 theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))


P2<-ggplot(alpha_df, aes(y = Observed, x = class, fill = class)) +
  geom_jitter(alpha = 0.2, width = 0.2)+
  geom_boxplot(alpha = 0.5)+
  scale_linetype_manual(values = linetypes)+
  theme_bw(base_size = 14)+
  ggtitle("Bacteria links")+
  scale_fill_manual(values = colors)+
  xlab("")+
  ylab("Frequency")+
 # theme(axis.text.y=element_blank())+
```

```
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))


grid.arrange(P2,P1, ncol = 2)
```




```
####

plasmids<-subset(prevalence_distributions, class == "All_classes")[,1:2]
hosts<-subset(alpha_df, class == "All_classes")[,3:4]

plasmids$TotalAbundance<-"Plasmid"
hosts$Shannon<-"Bacteria"
names(plasmids)<-c("Degree", "Type")
names(hosts)<-c("Degree", "Type")

degree_df<-rbind(plasmids, hosts)

# Figure 1e

ggplot(degree_df, aes(x = Degree, fill = Type))+geom_histogram(position = "dodge", binwidth = 1)+
  theme_bw(base_size = 14)+xlim(0,20)+
  scale_fill_manual(values = c("skyblue" ,"forestgreen"))+
  theme(legend.position=c(0.85,0.75))+
  theme(legend.title=element_blank(),
```

```
    legend.box.background = element_rect(colour = "black"))+
  ylab("Frequency")+
  xlab("Links")
```



Motif analysis

Calculate node positions for plasmids and bacterial hosts in the overall network

```
# network
data_pa <-phyloseq_standardize_otu_abundance(phylo_filtered, method = "pa")
otutable<-data.frame(otu_table(data_pa))
network <- as.matrix(otutable)

# calculate node positions for plasmids

plasmid_roles <- node_positions(M = network,
  level = "columns",
  weights_method = "none",
  weights_combine = "none",
  six_node = T,
  normalisation = "sizeclass")

plasmid_roles[1:5,1:5]
```

```
##              np1 np2       np3 np4        np5
## k141_1036310   1   0 0.8636364   0 0.13636364
## k141_1023016   1   0 0.6791444   0 0.32085561
## k141_2191729   1   0 0.9696970   0 0.03030303
## k141_1554928   1   0 0.9230769   0 0.07692308
## k141_1931562   1   0 0.9333333   0 0.06666667
```

```
row_position_numbers <- c(2,4,6,8,11,12,14,16,18,21,22,24,25,28,29,31,34,35,38,41,42,44,46,48,51,52,55,5
  61,63,64,67,68,70,73,74,77,78,81,82,86,87,88,92,93,94,98,99,102,103,106,107,108,111,112,114,117,118,1
  124,127,128,132,133,136,137,140,143,144,146,148) # all possible position numbers for row nodes

df <- plasmid_roles[,setdiff(colnames(plasmid_roles), paste0("np", row_position_numbers))] # remove col
dim(df)
```

```
## [1] 249  74
```

```
### clean up

plasmid_roles<-df[,c(-1)]
plasmid_roles[1:5, 1:5]
```

```
##                    np3        np5         np7       np9       np10
## k141_1036310 0.8636364 0.13636364 0.005134788 0.5083440 0.12323492
## k141_1023016 0.6791444 0.32085561 0.062632815 0.2241360 0.34201991
## k141_2191729 0.9696970 0.03030303 0.000000000 0.3685446 0.01877934
## k141_1554928 0.9230769 0.07692308 0.000000000 0.4200000 0.12000000
## k141_1931562 0.9333333 0.06666667 0.001763668 0.3262787 0.12345679
```

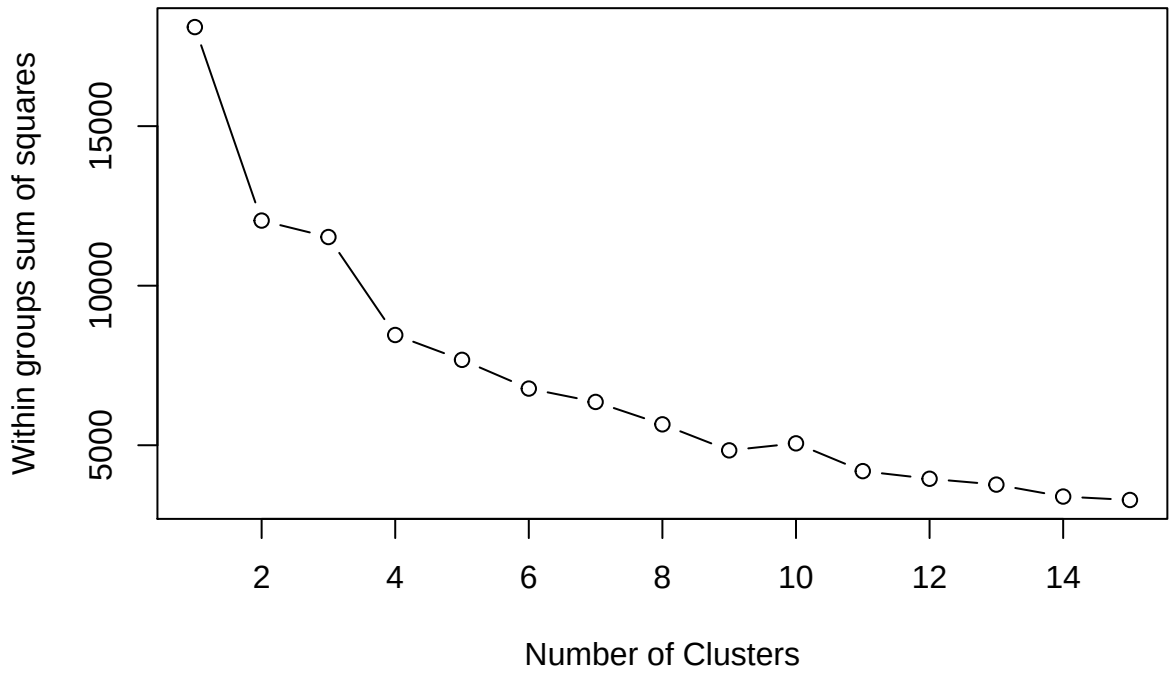
```
plasmid_roles<-na.omit(plasmid_roles)

### calculate node positions for bacterial hosts ############

# calculate node positions
host_roles <- node_positions(M = network,
  level = "rows",
  weights_method = "none",
  weights_combine = "none",
  six_node = T,
  normalisation = "sizeclass")

`%notin%` <- Negate(`%in%`)

df <- host_roles[,colnames(host_roles) %notin% colnames(plasmid_roles)] # remove column positions

dim(df) # 191 bacteria
```

```
## [1] 191  75
```

```
host_roles<-df[,c(-1, -2)] # noq 156 (since any with no links were deleted)

host_roles<-na.omit(host_roles)
```

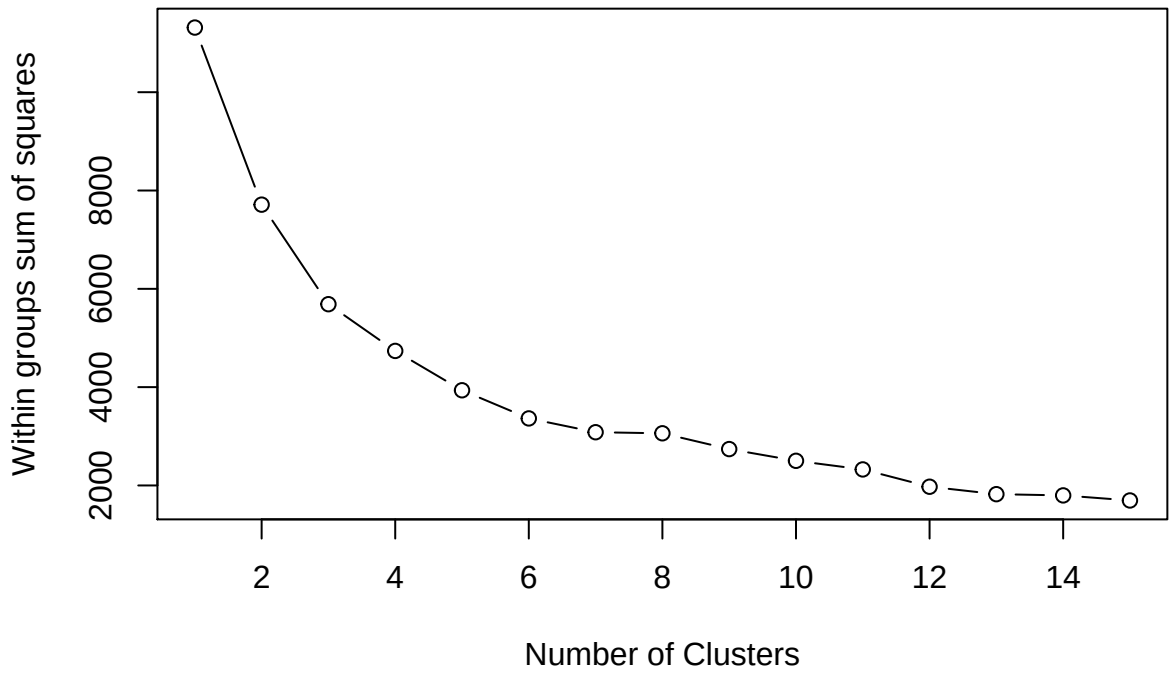
k-means Clustering of motif signature

```
######### PARTITIONING METHOD

## Cluster plasmids by their motif signature

## scale node positions

plasmid_roles_scaled <- na.omit(plasmid_roles) # listwise deletion of missing
plasmid_roles_scaled <- scale(plasmid_roles_scaled) # standardize variables

# Determine number of clusters
wss <- (nrow(plasmid_roles_scaled)-1)*sum(apply(plasmid_roles_scaled,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(plasmid_roles_scaled,
  centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```




```
# K-Means Cluster Analysis
fit <- kmeans(plasmid_roles_scaled, 6) # 5 cluster solution
# get cluster means
#aggregate(plasmid_roles_scaled,by=list(fit$cluster),FUN=mean)
# append cluster assignment
plasmid_roles_scaled <- data.frame(plasmid_roles_scaled, fit$cluster)
```

```
plasmid_roles_scaled$plasmidID<-row.names(plasmid_roles_scaled)


################## CLUSTERING HOSTS ##########

host_roles_scaled <- na.omit(host_roles) # listwise deletion of missing
host_roles_scaled <- scale(host_roles_scaled) # standardize variables

# Determine number of clusters
wss <- (nrow(host_roles_scaled)-1)*sum(apply(host_roles_scaled,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(host_roles_scaled,
  centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```




```
# K-Means Cluster Analysis
fit <- kmeans(host_roles_scaled, 6) # 5 cluster solution
# get cluster means
#aggregate(host_roles_scaled,by=list(fit$cluster),FUN=mean)
# append cluster assignment
host_roles_scaled <- data.frame(host_roles_scaled, fit$cluster)
host_roles_scaled$hostID<-row.names(host_roles_scaled)
```

Ordination

```
######### PLASMIDS ############

MDS_res=metaMDS(plasmid_roles, distance = "jaccard", k = 2, trymax = 100)
```

```
## Run 0 stress 0.08719081
## Run 1 stress 0.1687432
## Run 2 stress 0.13901
## Run 3 stress 0.08719089
## ... Procrustes: rmse 4.560166e-05  max resid 0.000524463
## ... Similar to previous best
## Run 4 stress 0.1371744
## Run 5 stress 0.1114424
## Run 6 stress 0.1131992
## Run 7 stress 0.1043136
## Run 8 stress 0.08719064
## ... New best solution
## ... Procrustes: rmse 5.700944e-05  max resid 0.0004712209
## ... Similar to previous best
## Run 9 stress 0.1144569
## Run 10 stress 0.1030988
## Run 11 stress 0.1097747
## Run 12 stress 0.1030994
## Run 13 stress 0.1524793
## Run 14 stress 0.1699626
## Run 15 stress 0.08719067
## ... Procrustes: rmse 2.589159e-05  max resid 0.0001204506
## ... Similar to previous best
## Run 16 stress 0.08719067
## ... Procrustes: rmse 1.99441e-05  max resid 0.0001999643
## ... Similar to previous best
## Run 17 stress 0.1178347
## Run 18 stress 0.1409001
## Run 19 stress 0.1391763
## Run 20 stress 0.087191
## ... Procrustes: rmse 8.177656e-05  max resid 0.00070457
## ... Similar to previous best
## *** Solution reached
```

```
df<-data.frame(scores(MDS_res,display=c("sites")))

degree<-data.frame(colSums(network))
degree$plasmid<-row.names(degree)

df$degree<-as.numeric(vlookup(row.names(df), degree, lookup_column = "plasmid", result_column = "colSums
df$cluster<- as.factor(vlookup(row.names(df), plasmid_roles_scaled, lookup_column = "plasmidID", result_

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

bio.fit <- envfit(MDS_res, plasmid_roles, perm = 999)
```

```r
df_biofit<-data.frame(bio.fit$vectors$arrows)
df_biofit$r<-bio.fit$vectors$r
df_biofit$r2<-df_biofit$r^2
df_biofit$p.val<-bio.fit$vectors$pvals
df_biofit<-subset(df_biofit, r2 >0.3)

df_biofit$NMDS1<-as.numeric(scale(df_biofit$NMDS1))
df_biofit$NMDS2<-as.numeric(scale(df_biofit$NMDS2))
df_biofit$NodePosition<-row.names(df_biofit)


bmotif_node_positions <- read.csv("DATA/bmotif_node_positions.csv")

df_biofit$degree<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", result_c
df_biofit$complexity<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", resu
df_biofit$Indirect_deg<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", re
df_biofit$PathLength<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", resu
```

```r
ord_plasmids<- ggplot(data=df,aes(NMDS1,NMDS2))+
  stat_ellipse(geom = "polygon", alpha = 0.3, aes(fill = cluster)) +
  geom_point(aes(size = degree, fill = cluster), alpha = 0.9, pch = 21, col = "black")+
  scale_size(range = c(2,9))+
  theme_bw(base_size = 14)+
 theme(legend.position = "none")+
  scale_fill_manual(values = mypalx)+
  scale_color_gsea()+
  ggtitle("Plasmids")+
  geom_segment(data=df_biofit, aes(x = 0, y = 0, xend = NMDS1*0.5, yend = NMDS2*0.5,  col = Indirect_deg
    arrow = arrow(length = unit(0.1, "cm")),alpha=0.8, size = 1)
```

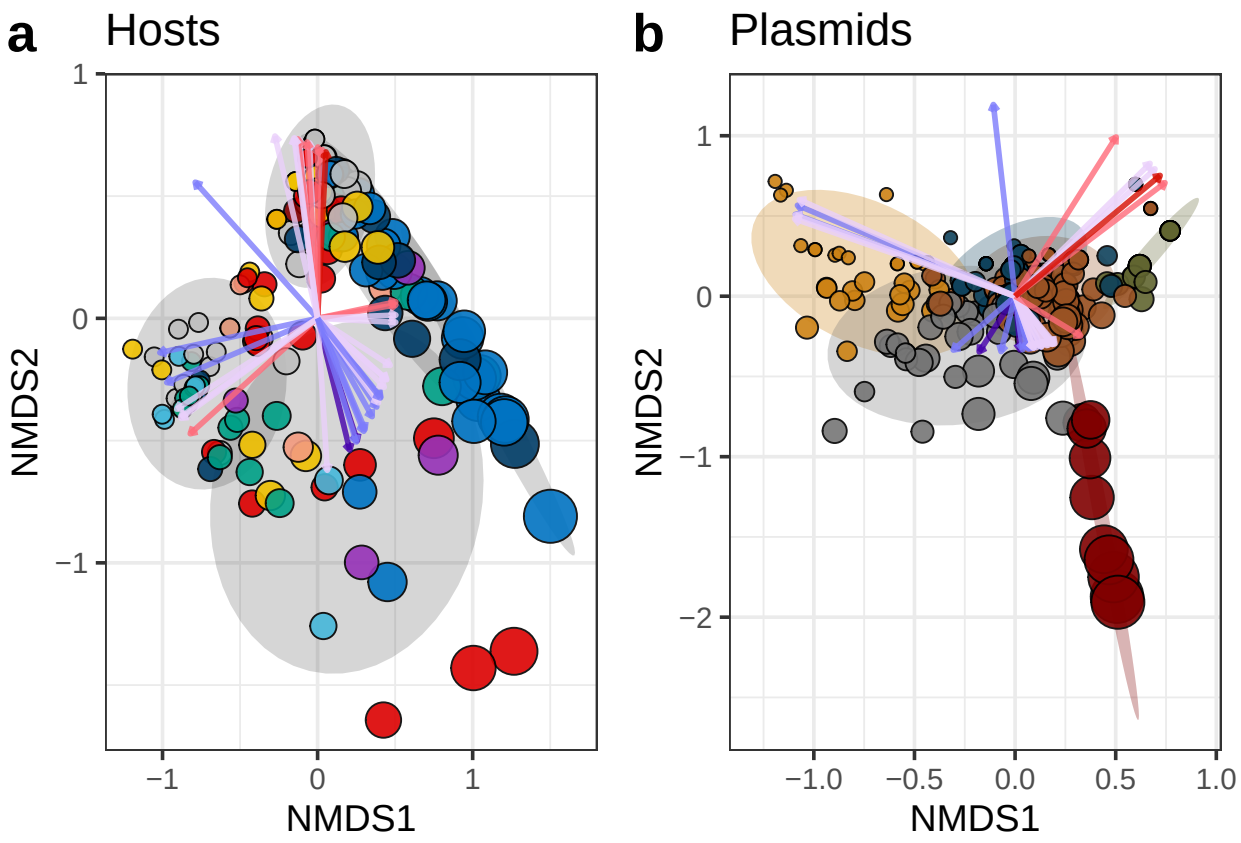
```r
############################## plot hosts ##############


MDS_res=metaMDS(host_roles, distance = "jaccard", k = 2, trymax = 100)
```

```
## Run 0 stress 0.05179776
## Run 1 stress 0.1035232
## Run 2 stress 0.1586378
## Run 3 stress 0.05179777
## ... Procrustes: rmse 7.337076e-06  max resid 5.210777e-05
## ... Similar to previous best
## Run 4 stress 0.05179776
## ... New best solution
## ... Procrustes: rmse 9.410121e-06  max resid 9.325193e-05
## ... Similar to previous best
## Run 5 stress 0.07667848
## Run 6 stress 0.05179783
## ... Procrustes: rmse 2.742644e-05  max resid 0.0002524593
## ... Similar to previous best
## Run 7 stress 0.0517975
## ... New best solution
## ... Procrustes: rmse 0.0007286793  max resid 0.007370044
## ... Similar to previous best
```

```
## Run 8 stress 0.05179777
## ... Procrustes: rmse 0.0007297684  max resid 0.007400149
## ... Similar to previous best
## Run 9 stress 0.05179776
## ... Procrustes: rmse 0.0007329878  max resid 0.007422432
## ... Similar to previous best
## Run 10 stress 0.133366
## Run 11 stress 0.07667755
## Run 12 stress 0.05179776
## ... Procrustes: rmse 0.0007333668  max resid 0.007434034
## ... Similar to previous best
## Run 13 stress 0.05179749
## ... New best solution
## ... Procrustes: rmse 6.556595e-06  max resid 6.460989e-05
## ... Similar to previous best
## Run 14 stress 0.07667749
## Run 15 stress 0.07667753
## Run 16 stress 0.05179777
## ... Procrustes: rmse 0.0007445455  max resid 0.007523419
## ... Similar to previous best
## Run 17 stress 0.05179754
## ... Procrustes: rmse 2.381967e-05  max resid 0.0002586541
## ... Similar to previous best
## Run 18 stress 0.0517975
## ... Procrustes: rmse 8.628809e-06  max resid 6.356589e-05
## ... Similar to previous best
## Run 19 stress 0.05179776
## ... Procrustes: rmse 0.0007312938  max resid 0.007426593
## ... Similar to previous best
## Run 20 stress 0.05179776
## ... Procrustes: rmse 0.0007408757  max resid 0.007511745
## ... Similar to previous best
## *** Solution reached
```

```
df<-data.frame(scores(MDS_res,display=c("sites")))
degree<-data.frame(rowSums(network))
degree$host<-row.names(degree)

df$degree<-as.numeric(vlookup(row.names(df), degree, lookup_column = "host", result_column = "rowSums.ne
df$cluster<- as.factor(vlookup(row.names(df), host_roles_scaled, lookup_column = "hostID", result_colum

taxonomy <- read.csv("DATA/taxonomy_phylophlan.csv", sep=",") #taxonomy for genome clusters

df$class<- as.factor(vlookup(row.names(df), taxonomy, lookup_column = "cluster_id", result_column = "Cla

#classes to keep
classes<-c("c__Betaproteobacteria",  "c__Gammaproteobacteria", "c__Epsilonproteobacteria", "c__Flavobact
  "c__Bacteroidia",  "c__Actinobacteria", "c__Clostridia", "c__Bacilli",  "c__Fusobacteriia", "c__Negati

df$class_plot<-ifelse(df$class %in% classes, as.character(df$class), "Other")

### biofit

bio.fit <- envfit(MDS_res, host_roles, perm = 999)
```

```r
df_biofit<-data.frame(bio.fit$vectors$arrows)
df_biofit$r<-bio.fit$vectors$r
df_biofit$r2<-df_biofit$r^2
df_biofit$p.val<-bio.fit$vectors$pvals
#hist(df_biofit$r2)

df_biofit<-subset(df_biofit, r2 >0.3)

df_biofit$NMDS1<-as.numeric(scale(df_biofit$NMDS1))
df_biofit$NMDS2<-as.numeric(scale(df_biofit$NMDS2))
df_biofit$NodePosition<-row.names(df_biofit)


#bmotif_node_positions <- read.csv("DATA/bmotif_node_positions.csv")

df_biofit$degree<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", result_c
df_biofit$complexity<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", resul
df_biofit$Indirect_deg<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", res
df_biofit$PathLength<-vlookup(df_biofit$NodePosition, bmotif_node_positions, lookup_column = "NP", resul

df$class_plot<-factor(df$class_plot, levels = c("c__Betaproteobacteria", "c__Gammaproteobacteria" ,  "c_
  "c__Bacteroidia"  , "c__Clostridia" , "c__Bacilli", "c__Negativicutes" , "c__Actinobacteria",  "c__Fus

mypal =     pal_jco("default", alpha = 1)(8)
mypal1 =    pal_locuszoom("default", alpha = 1)(8)
mypal2 =    pal_npg("nrc", alpha = 1)(8)
mypal3 =    pal_uchicago("dark", alpha = 1)(8)

cols<- c(mypal[1], mypal[6], mypal1[4], mypal2[3], mypal2[8], mypal2[5], mypal3[1], mypal[2], mypal1[6]

ord_hosts<- ggplot(data=df,aes(NMDS1,NMDS2))+
  stat_ellipse(geom = "polygon", alpha = 0.2, aes(group = cluster)) +
  geom_point(aes(size = degree, fill = class_plot), alpha = 0.9, pch = 21, col = "black")+
  scale_size(range = c(3,9))+
  theme_bw(base_size = 14)+
 theme(legend.position = "none")+
  scale_fill_manual(values = cols)+
  scale_color_gsea()+
  ggtitle("Hosts")  +
  geom_segment(data=df_biofit, aes(x = 0, y = 0, xend = NMDS1*0.5, yend = NMDS2*0.5,  col = Indirect_deg
    arrow = arrow(length = unit(0.1, "cm")),alpha=0.8, size = 1)+
  guides(fill = guide_legend(override.aes = list(size = 7)))


ggarrange(ord_hosts, ord_plasmids, labels = c("a","b"), font.label = list(size = 20))
```


a
Hosts
b
Plasmids
NMDS1
NMDS2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0