Hi-C contig processing

Alice Risely

23/10/2023

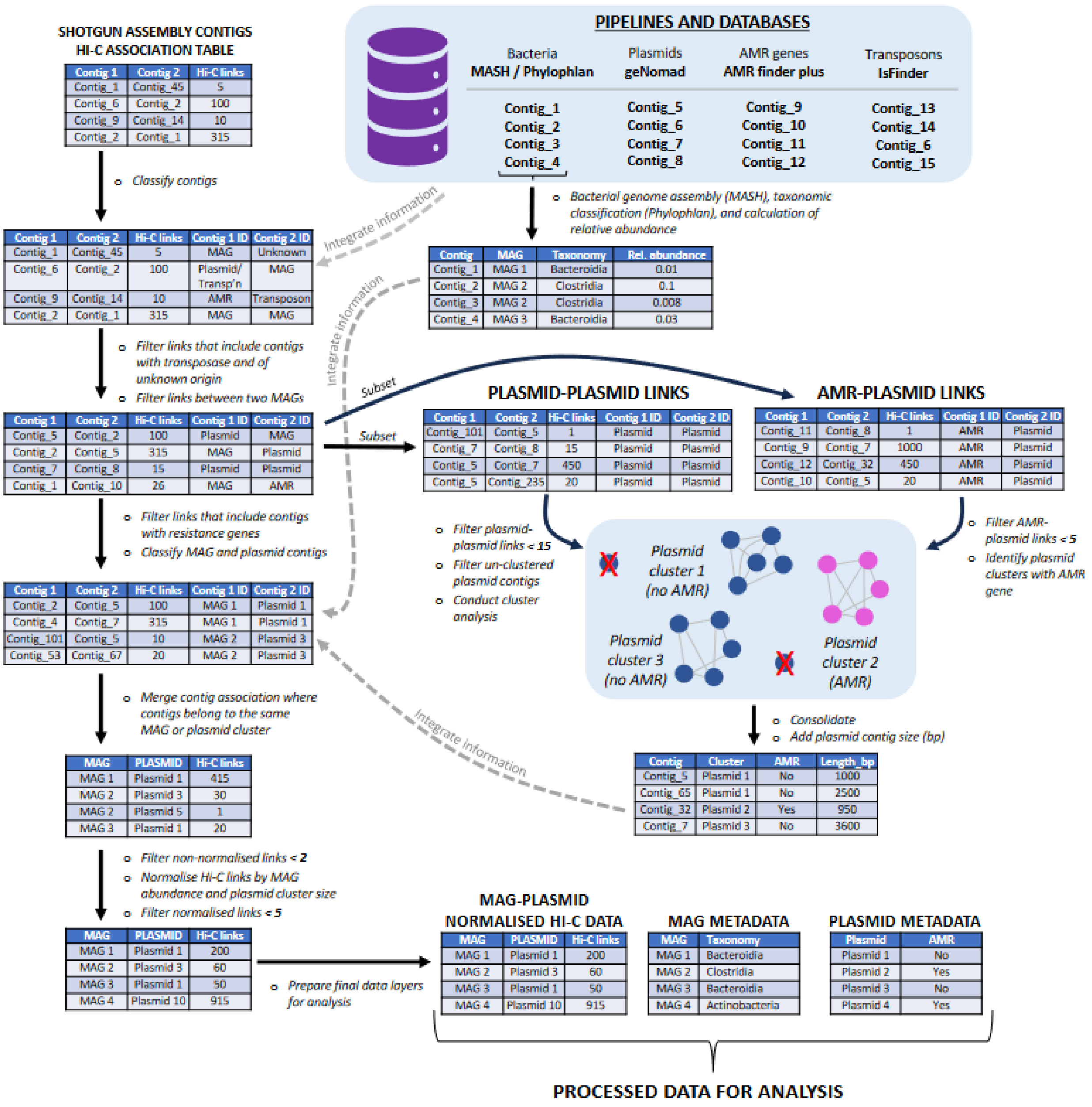
Contents

Info

Rmarkdown report for the analysis to go with the manuscript "AMR gene presence predicts plasmid network structure in wastewater" by Alice Risely, Thibault Stalder, Benno I. Simmons, Eva M. Top, Angus Buckling, and Dirk Sanders

Data generated by: Stalder et al. "Linking the resistome and plasmidome to the microbiome." The ISME journal 13.10 (2019): 2437-2446.

Contig processing steps performed under 'Contig processing'. See Fig S3 for flowchart:



SHOTGUN ASSEMBLY CONTIGS
HI-C ASSOCIATION TABLE

Contig 1	Contig 2	Hi-C links
Contig_1	Contig_45	5
Contig_6	Contig_2	100
Contig_9	Contig_14	10
Contig_2	Contig_1	315

- Classify contigs

Contig 1	Contig 2	Hi-C links	Contig 1 ID	Contig 2 ID
Contig_1	Contig_45	5	MAG	Unknown
Contig_6	Contig_2	100	Plasmid/ Transp'n	MAG
Contig_9	Contig_14	10	AMR	Transposon
Contig_2	Contig_1	315	MAG	MAG

- Filter links that include contigs with transposase and of unknown origin
- Filter links between two MAGs

Contig 1	Contig 2	Hi-C links	Contig 1 ID	Contig 2 ID
Contig_5	Contig_2	100	Plasmid	MAG
Contig_2	Contig_5	315	MAG	Plasmid
Contig_7	Contig_8	15	Plasmid	Plasmid
Contig_1	Contig_10	26	MAG	AMR

- Filter links that include contigs with resistance genes
- Classify MAG and plasmid contigs

Contig 1	Contig 2	Hi-C links	Contig 1 ID	Contig 2 ID
Contig_2	Contig_5	100	MAG 1	Plasmid 1
Contig_4	Contig_7	315	MAG 1	Plasmid 1
Contig_101	Contig_5	10	MAG 2	Plasmid 3
Contig_53	Contig_67	20	MAG 2	Plasmid 3

- Merge contig association where contigs belong to the same MAG or plasmid cluster

MAG	PLASMID	Hi-C links
MAG 1	Plasmid 1	415
MAG 2	Plasmid 3	30
MAG 2	Plasmid 5	1
MAG 3	Plasmid 1	20

- Filter non-normalised links < 2
- Normalise Hi-C links by MAG abundance and plasmid cluster size
- Filter normalised links < 5

MAG	PLASMID	Hi-C links
MAG 1	Plasmid 1	200
MAG 2	Plasmid 3	60
MAG 3	Plasmid 1	50
MAG 4	Plasmid 10	915

- Prepare final data layers for analysis

PIPELINES AND DATABASES

Bacteria MASH / Phylophlan	Plasmids geNomad	AMR genes AMR finder plus	Transposons IsFinder
Contig_1	Contig_5	Contig_9	Contig_13
Contig_2	Contig_6	Contig_10	Contig_14
Contig_3	Contig_7	Contig_11	Contig_6
Contig_4	Contig_8	Contig_12	Contig_15

- Bacterial genome assembly (MASH), taxonomic classification (Phylophlan), and calculation of relative abundance

Contig	MAG	Taxonomy	Rel. abundance
Contig_1	MAG 1	Bacteroidia	0.01
Contig_2	MAG 2	Clostridia	0.1
Contig_3	MAG 2	Clostridia	0.008
Contig_4	MAG 3	Bacteroidia	0.03

PLASMID-PLASMID LINKS

Contig 1	Contig 2	Hi-C links	Contig 1 ID	Contig 2 ID
Contig_101	Contig_5	1	Plasmid	Plasmid
Contig_7	Contig_8	15	Plasmid	Plasmid
Contig_5	Contig_7	450	Plasmid	Plasmid
Contig_5	Contig_235	20	Plasmid	Plasmid

AMR-PLASMID LINKS

Contig 1	Contig 2	Hi-C links	Contig 1 ID	Contig 2 ID
Contig_11	Contig_8	1	AMR	Plasmid
Contig_9	Contig_7	1000	AMR	Plasmid
Contig_12	Contig_32	450	AMR	Plasmid
Contig_10	Contig_5	20	AMR	Plasmid

- Filter plasmid-plasmid links < 15
- Filter un-clustered plasmid contigs
- Conduct cluster analysis

- Filter AMR-plasmid links < 5
- Identify plasmid clusters with AMR gene

- Consolidate
- Add plasmid contig size (bp)

Contig	Cluster	AMR	Length_bp
Contig_5	Plasmid 1	No	1000
Contig_65	Plasmid 1	No	2500
Contig_32	Plasmid 2	Yes	950
Contig_7	Plasmid 3	No	3600

MAG-PLASMID
NORMALISED HI-C DATA

MAG	PLASMID	Hi-C links
MAG 1	Plasmid 1	200
MAG 2	Plasmid 3	60
MAG 3	Plasmid 1	50
MAG 4	Plasmid 10	915

MAG METADATA

MAG	Taxonomy
MAG 1	Bacteroidia
MAG 2	Clostridia
MAG 3	Bacteroidia
MAG 4	Actinobacteria

PLASMID METADATA

Plasmid	AMR
Plasmid 1	No
Plasmid 2	Yes
Plasmid 3	No
Plasmid 4	Yes

PROCESSED DATA FOR ANALYSIS

Import packages

```
library(tidyverse)
library(data.table)
library(phyloseq)
library(igraph)
library(network)
library(expss)
library(here)
library(ggnetwork)
library(RColorBrewer)
library(ggvenn)
library(ggrepel)
library(ggpubr)

memory.limit(1000000)
```

```
## [1] 1e+06
```

Import data

Import HiC data

- NOT RUN
- First few basic filtering steps are skipped as they take a lot of RAM.
- These include filtering out links that are duplicates and between the same contigs.

```
## import association table

WW_links <- read.delim(here("DATA","WW_links.txt"), header=FALSE)
WW_links_DT<-as.data.table(WW_links) # convert to data table
rm(WW_links)
```

```


# remove contigs attached to themselves

WW_links_DT$self_link<-WW_links_DT$V1 == WW_links_DT$V2
WW_links_DT2<-subset(WW_links_DT, self_link == F)

## remove duplicates
WW_links_filt2<-WW_links_DT2 %>%
  group_by(grp = paste(pmax(V1, V2), pmin(V1, V2), sep = "_")) %>%
  slice(1) %>%
  ungroup() %>%
  select(-grp)
```

Remove duplicates

- Now import pre-filtered HiC links dataframe, called “WW_links_filt2”
- All steps from this point largely involve filtering this dataset of noise so that by the end, only meaningful links remain for analysis

```
# import dataset where reverse dulplicates have been excluded, as well as any contigs attached to thems

setwd("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Updated analysis2")
WW_links_filt2<-readRDS("DATA/WW_links_filt.RDS")

head(WW_links_filt2)
```

```
## # A tibble: 6 x 4
##   V1           V2              V3 self_link
##   <chr>        <chr>        <int> <lgl>
## 1 k141_1000386 k141_1000196     1 FALSE
## 2 k141_1000616 k141_1000543     1 FALSE
## 3 k141_1000630 k141_1000774     1 FALSE
## 4 k141_1000590 k141_1000833     3 FALSE
## 5 k141_1000728 k141_1001032     1 FALSE
## 6 k141_1001172 k141_1000706     1 FALSE
```

```
dim(WW_links_filt2) # 28 million rows
```

```
## [1] 28111659        4
```

```


setwd(here("DATA", "MAGS"))

myfiles<-list.files()
myfiles[1:5]
```

Import MAG contigs

```
## [1] "bin_1.fasta"   "bin_10.fasta"  "bin_100.fasta" "bin_101.fasta"
## [5] "bin_102.fasta"
```

```
dataFiles <- lapply(myfiles, read.table)
myfiles<-str_remove(myfiles, ".fasta") # remove 'fasta' string

length(dataFiles) #379 MAGs
```

```
## [1] 379
```

```
#change names
names(dataFiles)<-myfiles

## loop to extract contigs per cluster data

mag_list<-list()

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

  cluster_x<-dataFiles[[i]]
  selectedRows <- cluster_x[grep(">k141_", cluster_x$V1), ]
  contigs<-substring(selectedRows, 2)
  df<-data.frame(contigs)
  df$cluster<-myfiles[i]
  mag_list[[i]]<-df
}

names(mag_list)<-myfiles
```

```r
mag_data<-do.call(rbind, mag_list)
mag_assignment<-mag_data
mag_list <- as.character(mag_assignment$contigs)

setwd("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Updated analysis2")
```

Import predited plasmid IDs

- Import results from Genomad

```r
genomad_plasmids <- read.delim("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnal

plasmid_list<- as.character(genomad_plasmids$seq_name)
```

```r


transposons <- read.delim(here("DATA","ISfinder_best_match_transposons.csv"), header=TRUE, sep = ",")
transposon_list<-as.character(transposons$contig)
```

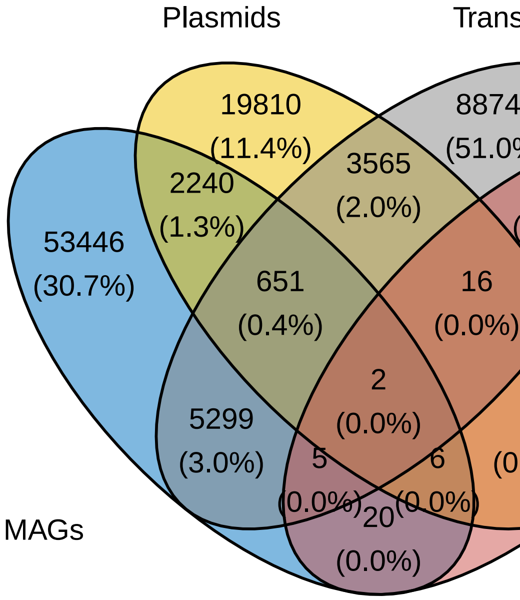
Import transposon list

```r


# import AMR gene data
amr_genes <- read.delim(here("DATA","amr_contigs_plassclass_and_plasflow.tsv"))
amr_genes <-amr_genes[,c("contig", "Gene.symbol", "Sequence.name", "Element.type", "Element.subtype")]
```

Import resistance gene list

```r


## generate list of all unique contigs

allContigs<-c(unique(WW_links_filt2$V1), unique(WW_links_filt2$V2))
allContigs<-unique(allContigs)

mag_list2<- mag_list[mag_list%in%allContigs]
plasmid_list2<- plasmid_list[plasmid_list%in%allContigs]
transposon_list2<- transposon_list[transposon_list%in%allContigs]
resistance_list2<- amr_genes$contig[amr_genes$contig%in%allContigs]


x <- list(
  MAGs = mag_list2,
  Plasmids = plasmid_list2,
  Transposons = transposon_list2,
  Resistance = resistance_list2
  )

```

```
ggvenn(
  x,
 fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 4
  )
```



Summarize contig assigments with Venn Diagram

Filter and assign identity

Remove links that involve an unidentified contigs

- Assign all contigs to whether they are a MAG, Plasmid, or resistance gene.

```
allContigs_df<-data.frame(allContigs)
names(allContigs_df)<-"Contig"

allContigs_df$MAG<- allContigs_df$Contig %in% mag_list
allContigs_df$Plasmid<- allContigs_df$Contig %in% plasmid_list
allContigs_df$Transposon<- allContigs_df$Contig %in% transposon_list
allContigs_df$Resistance<- allContigs_df$Contig %in% amr_genes$contig

allContigs_df$Remove<- allContigs_df$MAG==F & allContigs_df$Plasmid==F &  allContigs_df$Resistance==F

allContigs_filt<-subset(allContigs_df, Remove == F)
```

- Remove links from HiC links dataframe between contigs that are not identified as anything.
- Because we are not interested in these.

```
WW_links_filt2$Keep<- (WW_links_filt2$V1 %in% allContigs_filt$Contig) & (WW_links_filt2$V2 %in% allConti

WW_links_filt3<-subset(WW_links_filt2, Keep == T)
```

Remove transposons

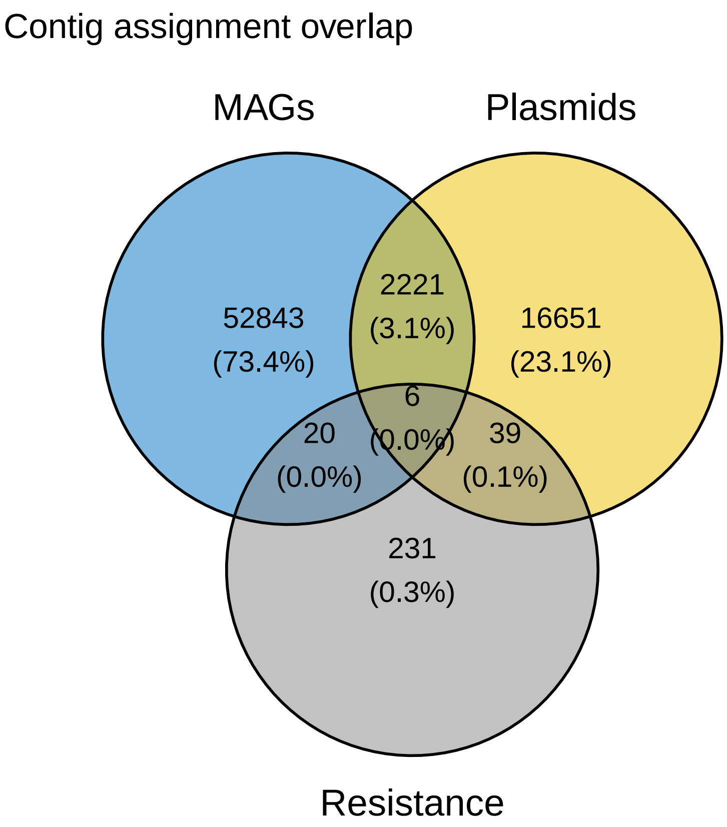
- Remove any links that involve a transposon

```
WW_links_filt3$Transposon <- WW_links_filt3$V1 %in% transposon_list | WW_links_filt3$V2 %in% transposon_
WW_links_filt4<- subset(WW_links_filt3, Transposon == F)
```

- So far removed:links that are 1) duplicates, 2) contigs attached to themselves, 3) involve transposons, 4) involve any unidentifiable contigs (not mags/plasmids/AMR, etc)

Venn diagram after filtering transposons

```
## list of contigs left
allContigs<-c(unique(WW_links_filt4$V1), unique(WW_links_filt4$V2))
allContigs<-unique(allContigs)

## venn diagram

mag_list2<- mag_list[mag_list%in%allContigs]
plasmid_list2<- plasmid_list[plasmid_list%in%allContigs]
resistance_list2<- amr_genes$contig[amr_genes$contig%in%allContigs]


x <- list(
  MAGs = mag_list2,
  Plasmids = plasmid_list2,
  Resistance = resistance_list2
  )

ggvenn(
  x,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 5, text_size = 4
  )+ggtitle("Contig assignment overlap")
```



Assign remaining contigs to either MAG, plasmid, or resistance gene

```
allContigs_df<-data.frame(allContigs)
names(allContigs_df)<-"Contig"

allContigs_df$MAG<- allContigs_df$Contig %in% mag_list
allContigs_df$Plasmid<- allContigs_df$Contig %in% plasmid_list
allContigs_df$Resistance<- allContigs_df$Contig %in% amr_genes$contig

# final assignment
allContigs_df$Assignment <- ifelse(allContigs_df$Plasmid == TRUE, "PLASMID", "MAG")
allContigs_df$Assignment <- ifelse(allContigs_df$Resistance == TRUE, "AMR", allContigs_df$Assignment)
```

Add contig assignment to links DF

```
WW_links_filt4$contig1_assignment <- vlookup(WW_links_filt4$V1, allContigs_df, lookup_column = "Contig"

WW_links_filt4$contig2_assignment <- vlookup(WW_links_filt4$V2, allContigs_df, lookup_column = "Contig"

# remove extra columns
WW_links_filt4<-WW_links_filt4[,c(1,2,3,7,8)]
```

Remove MAG-MAG links

- After removing mag-mag links, we have a HiC links dataframe which includes Mag-plasmid links, plasmid-plasmid links, and plasmid-amr links, and mag-amr links.
- We will need this later for the plasmid clustering process

```
WW_links_filt5 <- WW_links_filt4[!(WW_links_filt4$contig1_assignment=="MAG" & WW_links_filt4$contig2_as
```

Cluster plasmids and link to AMR

Cluster plasmids

- Filter links just between putative plasmid contigs to cluster them

```
links_plasmids <- subset(WW_links_filt5, contig1_assignment == "PLASMID" & contig2_assignment == "PLASMI
links_plasmids <- links_plasmids %>% arrange(-V3)

# Links should be robust. Therefore require at least 15 Hi-C connections to assume real genetic connect

WW_links_plasmids_filtered<-subset(links_plasmids, V3 >14)

WW_links_plasmids_filtered<-WW_links_plasmids_filtered[,1:3] # keep only first 3 columns

unique_plasmids<-c(WW_links_plasmids_filtered$V1,WW_links_plasmids_filtered$V2)

# remove 33 contigs that are unlikely to be contigs (based on gene content)

gene_content <- read.csv("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/
unique(gene_content$remove)
```

```
## [1] ""    "yes"
```

```
remove_df<-subset(gene_content, remove == "yes")
dim(remove_df)
```

```
## [1] 34  9
```

```
contigs_remove<-as.character(remove_df$Contig)


WW_links_plasmids_filtered$Remove<- WW_links_plasmids_filtered$V1 %in% contigs_remove | WW_links_plasmi

table(WW_links_plasmids_filtered$Remove)
```

```
##
## FALSE  TRUE
##   366     2
```

```
WW_links_plasmids_filtered<-subset(WW_links_plasmids_filtered, Remove == F)

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

# convert into network object
# convert into network object
# convert into network object

igraph_net <- graph.data.frame(WW_links_plasmids_filtered[,c('V1','V2')])

# add number of links the the edge weight

E(igraph_net)$weight<-WW_links_plasmids_filtered$V3

igraph_net<-as.undirected(igraph_net) # make undirected

V(igraph_net)$degree<-igraph::degree(igraph_net) # add degree as a node characteristic
```

Indentify plasmid cluster memberships

- Use walktrap method to assign cluster membership

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

cluster.membership<-membership(wt)

# check how many contigs are assigned to each cluster
cluster_freq<- data.frame(table(cluster.membership))
cluster_freq<-cluster_freq[order(-cluster_freq$Freq),]

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


# plot network

layout_fr <- layout_with_fr(igraph_net, niter = 10000)
#layout_dh <- layout_with_dh(igraph_net, weight.edge.lengths = edge_density(igraph_net)/1)


n<-ggnetwork(igraph_net, layout = layout_fr)
#n<-ggnetwork(igraph_net)
n$ClusterMembership<-factor(n$ClusterMembership)


# make large colour vector
mypal1<-brewer.pal(12,"Paired")
mypal2<-brewer.pal(12,"Dark2")
```

```
mypal3<-c(mypal1, mypal2, mypal1, mypal2, mypal1, mypal2,mypal1, mypal2, mypal1, mypal2, mypal1, mypal2

mypal3<-c(mypal3, mypal3)
```

Connect plasmid clusters to resistance genes

- We've now clustered plasmid contigs, but which ones are also connected to resistance genes?
- Return to HiC links df with MAGs, plasmids, and resistance genes again
- This time keep only links between plasmid contigs and resistance genes

```
links_resistance <- subset(WW_links_filt5, contig1_assignment != "MAG" & contig2_assignment != "MAG")
links_resistance <- subset(links_resistance, (contig1_assignment == "PLASMID" & contig2_assignment == "

# manipulate dataframe so all plasmids in left column and all resistance contigs in right column

plasmid_contig1<-subset(links_resistance, contig1_assignment == "PLASMID")
plasmid_contig2<-subset(links_resistance, contig2_assignment == "PLASMID")

plasmid_contig2<-plasmid_contig2[,c(2,1,3,5,4)]
names(plasmid_contig2)<-names(plasmid_contig1)

links_resistance2<-rbind(plasmid_contig1, plasmid_contig2)


links_resistance2<-subset(links_resistance2, V3>4) # only considered an AMR plasmid if found connected

head(links_resistance2)
```

```
## # A tibble: 6 x 5
##   V1           V2              V3 contig1_assignment contig2_assignment
##   <chr>        <chr>        <int> <chr>              <chr>
## 1 k141_1739103 k141_1474880     7 PLASMID            AMR
## 2 k141_1772484 k141_2087656     6 PLASMID            AMR
## 3 k141_2194155 k141_2245618     5 PLASMID            AMR
## 4 k141_2481168 k141_1333187     7 PLASMID            AMR
## 5 k141_2621312 k141_1809171     8 PLASMID            AMR
## 6 k141_2768091 k141_1665904     5 PLASMID            AMR
```

```
## what type of resistance is the AMR gene?

links_resistance2$Resistance<- vlookup(links_resistance2$V2, amr_genes, lookup_column = "contig", resul


links_resistance2$AMR_gene<- vlookup(links_resistance2$V2, amr_genes, lookup_column = "contig", result_

head(links_resistance2)
```

```
## # A tibble: 6 x 7
##   V1           V2              V3 contig1_assignment contig2_a~1 Resis~2 AMR_g~3
##   <chr>        <chr>        <int> <chr>              <chr>       <chr>   <chr>
## 1 k141_1739103 k141_1474880     7 PLASMID            AMR         AMR     tet(A)
## 2 k141_1772484 k141_2087656     6 PLASMID            AMR         AMR     blaAER
## 3 k141_2194155 k141_2245618     5 PLASMID            AMR         AMR     aph(6)~
## 4 k141_2481168 k141_1333187     7 PLASMID            AMR         AMR     tet(W)
## 5 k141_2621312 k141_1809171     8 PLASMID            AMR         AMR     icr-Mo
## 6 k141_2768091 k141_1665904     5 PLASMID            AMR         METAL   ncrA
## # ... with abbreviated variable names 1: contig2_assignment, 2: Resistance,
## #   3: AMR_gene
```

```
# make column indicating whether resistance gene in an AMR gene (rather than metal resistance, etc)

links_resistance2$Resistance2<- links_resistance2$V2 %in% subset(amr_genes, Element.type == "AMR")$conti

head(links_resistance2)
```

```
## # A tibble: 6 x 8
##   V1           V2              V3 contig1_assi~1 conti~2 Resis~3 AMR_g~4 Resis~5
##   <chr>        <chr>        <int> <chr>          <chr>   <chr>   <chr>   <lgl>
## 1 k141_1739103 k141_1474880     7 PLASMID        AMR     AMR     tet(A)  TRUE
## 2 k141_1772484 k141_2087656     6 PLASMID        AMR     AMR     blaAER  TRUE
## 3 k141_2194155 k141_2245618     5 PLASMID        AMR     AMR     aph(6)~ TRUE
## 4 k141_2481168 k141_1333187     7 PLASMID        AMR     AMR     tet(W)  TRUE
## 5 k141_2621312 k141_1809171     8 PLASMID        AMR     AMR     icr-Mo  TRUE
## 6 k141_2768091 k141_1665904     5 PLASMID        AMR     METAL   ncrA    FALSE
## # ... with abbreviated variable names 1: contig1_assignment,
## #   2: contig2_assignment, 3: Resistance, 4: AMR_gene, 5: Resistance2
```

```
# only keep links that involve AMR resistance
links_resistance3<-subset(links_resistance2, Resistance2 == TRUE)

head(links_resistance3)
```

```
## # A tibble: 6 x 8
##   V1           V2              V3 contig1_assi~1 conti~2 Resis~3 AMR_g~4 Resis~5
##   <chr>        <chr>        <int> <chr>          <chr>   <chr>   <chr>   <lgl>
## 1 k141_1739103 k141_1474880     7 PLASMID        AMR     AMR     tet(A)  TRUE
## 2 k141_1772484 k141_2087656     6 PLASMID        AMR     AMR     blaAER  TRUE
## 3 k141_2194155 k141_2245618     5 PLASMID        AMR     AMR     aph(6)~ TRUE
## 4 k141_2481168 k141_1333187     7 PLASMID        AMR     AMR     tet(W)  TRUE
## 5 k141_2621312 k141_1809171     8 PLASMID        AMR     AMR     icr-Mo  TRUE
## 6 k141_1905885 k141_307872     21 PLASMID        AMR     AMR     aadA27  TRUE
## # ... with abbreviated variable names 1: contig1_assignment,
## #   2: contig2_assignment, 3: Resistance, 4: AMR_gene, 5: Resistance2
```

```
############## now lets put this info back into plasmid link table

# make datafrmae with cluster and resistance into for each (clustered) plasmid contig

cluster_df<- n[,c(3,5)]
cluster_df<-unique(cluster_df)
```

```r
# add column with whether contig is connected to an AMR gene
cluster_df$Resistance<-cluster_df$name %in% links_resistance3$V1
cluster_df<-cluster_df%>% arrange(ClusterMembership)
cluster_df$Gene <- vlookup(cluster_df$name, links_resistance3, lookup_column = "V1", result_column = "A

head(cluster_df)
```

```
##             name ClusterMembership Resistance   Gene
## 372  k141_515203                 1       TRUE tet(Q)
## 381  k141_973004                 1      FALSE   <NA>
## 386  k141_190742                 1      FALSE   <NA>
## 390  k141_971894                 1      FALSE   <NA>
## 395 k141_1856898                 1      FALSE   <NA>
## 403 k141_1145966                 1      FALSE   <NA>
```

```r
table(cluster_df$Gene)
```

```
##
##     aadA27       aadS  aph(6)-Id     blaAER     blaMCA      cmlA5     mph(A)
##          4          1          2          1          1          1          1
##     msr(E) qacEdelta1      qnrS2     tet(A)     tet(Q)
##         37          1          5          4          4
```

```r
subset(amr_genes, Gene.symbol == "msr(E)")
```

```
##          contig Gene.symbol                                  Sequence.name
## 364 k141_505817      msr(E) ABC-F type ribosomal protection protein Msr(E)
##     Element.type Element.subtype
## 364          AMR             AMR
```

```r
subset(amr_genes, Gene.symbol == "tet(Q)")
```

```
##          contig Gene.symbol
## 198 k141_238138      tet(Q)
## 410 k141_775292      tet(Q)
##                                                    Sequence.name Element.type
## 198 tetracycline resistance ribosomal protection protein Tet(Q)          AMR
## 410 tetracycline resistance ribosomal protection protein Tet(Q)          AMR
##     Element.subtype
## 198             AMR
## 410             AMR
```

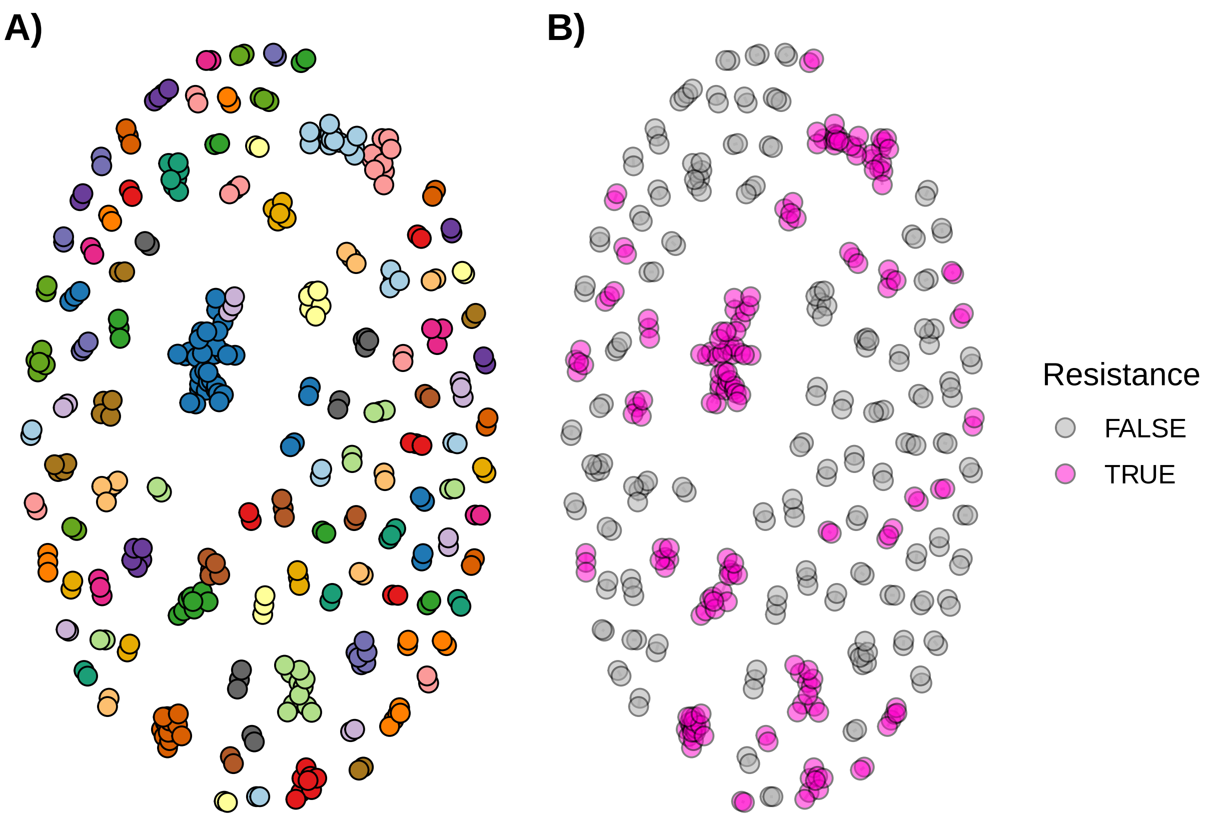
```r
# make list of plasmid clusters that are connected to a resistance gene
resistance_clusters2<-subset(cluster_df, Resistance==T)
resistance_clusters<-unique(resistance_clusters2$Cluster)

distinct(resistance_clusters2, ClusterMembership, Gene)
```

```
##     ClusterMembership      Gene
## 372                 1    tet(Q)
## 210                  2     blaAER
## 253                  2  aph(6)-Id
## 333                  3     msr(E)
## 189                  4     msr(E)
## 195                  4     aadA27
## 2051                 5       aadS
## 359                  6     msr(E)
## 485                  8     aadA27
## 487                  8     msr(E)
## 278                  9     msr(E)
## 235                 12     msr(E)
## 107                 14     msr(E)
## 4                   17     msr(E)
## 2171                17     aadA27
## 311                 18     msr(E)
## 56                  19     msr(E)
## 482                 21      qnrS2
## 34                  22     msr(E)
## 73                  24     msr(E)
## 410                 27 qacEdelta1
## 13                  28     msr(E)
## 89                  30     tet(A)
## 475                 33     msr(E)
## 274                 51     msr(E)
## 551                 54      qnrS2
## 343                 64     msr(E)
## 37                  70     msr(E)
## 44                  76     msr(E)
## 329                 79  aph(6)-Id
## 507                 82     aadA27
## 528                 83     blaMCA
## 540                 91     msr(E)
## 439                 99      qnrS2
## 2831               100     msr(E)
## 379                104     mph(A)
## 2421               104      cmlA5
```

```
n$Resistance<-n$ClusterMembership %in% resistance_clusters

cluster_df$Resistance_cluster<-cluster_df$ClusterMembership %in% resistance_clusters

# saveRDS(cluster_df, "C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Upd
```

Supplementary figure of plasmid clusters

```
p1<-ggplot(n, aes(x = x, y = y, xend = xend, yend = yend, label = ClusterMembership)) +
  geom_edges( color = "black", alpha = 0.5) +
  theme_blank()+
  geom_nodes(aes( fill = ClusterMembership), pch = 21, size =3)+
  scale_size(range = c(2,6))+
 # ggtitle("Putative plasmid clusters")+
  scale_fill_manual(values = mypal3)+
  theme(legend.position = "none")

  p2<-ggplot(n, aes(x = x, y = y, xend = xend, yend = yend, label = name)) +
  geom_edges( color = "grey", alpha = 0.7) +
  theme_blank()+
  geom_nodes(aes(fill = Resistance), pch = 21, alpha = 0.5, size = 3)+
  scale_size(range = c(1.5,6))+
 # ggtitle("Plasmid-plasmid cluster network coloured by AMR presence")+
  scale_fill_manual(values = c("darkgrey", "#ff00cc"))

ggarrange(p1, p2, labels = c("A)", "B)"), widths = c(1,1.3))
```



```
#ggsave("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysis/Updated analysis2

#check k141_736244
```

Calculate plasmid length

```
## add number of contigs that make up each cluster

contig_no<-data.frame(table(cluster_df$ClusterMembership))
cluster_df$ContigNo<-expss::vlookup(cluster_df$ClusterMembership, contig_no, lookup_column = "Var1", re
## add contig length
contig_length <- read.table("C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAnalysi


cluster_df$LengthKB<- vlookup(cluster_df$name, contig_length, lookup_column = 1, result_column = 2)


summary(cluster_df$LengthKB)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     206    1984    3703    5570    6878   47938
```

```
plasmid_length<-cluster_df %>% group_by(ClusterMembership) %>% summarise(TotalLengthKB = sum(LengthKB))

summary(plasmid_length$TotalLengthKB)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1351    4845    8772   19368   21611  240077
```

Filter and merge

Keep only MAG-PLASMID associations

- Ultimately, we are interested in Mag-plasmid connections
- So at this point, we filter down our HiC links df to include only links between MAG contigs and plasmid contigs
- We couldnt do this before, because we didn't know which plasmid contigs to keep (based on clustering)

```
# keep only links that involve MAGs, and make sure they are all in one column
# at the moment they are spread out across the left and right columns

mag1<- subset(WW_links_filt5, contig1_assignment=="MAG")
mag2<- subset(WW_links_filt5, contig2_assignment=="MAG")

# change columns of mag2 around so they match mag1
mag2<-mag2[,c(2,1,3,5,4)]
names(mag2)<-names(mag1)

WW_links_filt6<-rbind(mag1, mag2)

# change column names
names(WW_links_filt6)[1:3]<-c("contigs_mags", "contigs_plasmids", "Count")

table(WW_links_filt6$contig2_assignment) # still AMR genes in there. Remove
```

```
##
##     AMR PLASMID
##    7333  243014
```

```
WW_links_filt6<-subset(WW_links_filt6, contig2_assignment == "PLASMID")
```

- Only keep plasmid contigs that are clustered
- This information is stored in the object “cluster_df”

```
WW_links_filt6$Cluster<- vlookup(WW_links_filt6$contigs_plasmids, cluster_df, lookup_column = "name", r

length(unique(WW_links_filt6$Cluster)) # how many clusters?
```

```
## [1] 110
```

```
WW_links_filt6$Cluster<-factor(WW_links_filt6$Cluster)

# remove any plasmid inks that are not assigned a cluster
WW_links_filt7<-na.omit(WW_links_filt6)
```

Add MAG bin assignment info to link df

```
WW_links_filt7$MAG_assigment<-vlookup(WW_links_filt7$contigs_mags, mag_assignment, lookup_column = "con
```

Merge by MAG AND plasmid cluster

```
WW_links_filt8<- WW_links_filt7 %>% group_by(MAG_assigment, Cluster) %>% summarise(Count = sum(Count))
```

Remove singletons

```
WW_links_filt9<-subset(WW_links_filt8, Count >1) # changed
```

Add resistance metadata

- Plasmid clusters that are associated with AMR are stored in the object “resistance_clusters”

```
WW_links_filt9$Resistance<-WW_links_filt9$Cluster %in% resistance_clusters

# how many unique plasmid clusters associate with AMR?
table(unique(WW_links_filt9[,c(2,4)])$Resistance)
```

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

Normalise data

Normalise by MAG abundance

- MAGs vary in their abundance, and this will affect the number of links
- We want the counts normalised in a way that assumes all MAGs were present at the same abundance
- To get this 'theoretical' same abundance, I use the mean abundance across MAGs
- If the mean abundance is 1%, but there is a MAG with 2% abundance, I normalise the count by multiplying it by (1/2 = 0.5).
- In this case, 0.5 is called the 'scaling factor' that I multiply counts by
- The scaling factor for each MAG is the mean MAG abundance divided by the specific MAG abundance
- However, I always add a psudocount of 1, to make sure when rounded, there are no zero counts (minimum count should always be 1)
- Same as just dividing by MAG abundance, but then you get <0 which is problematic for networks

```
MAG_metadata <- read.csv(here("DATA","MAG_metadata.csv"))

MAG_metadata<-MAG_metadata[,1:8]

WW_links_filt9$Abundance<-vlookup(WW_links_filt9$MAG_assigment, MAG_metadata, lookup_column = "cluster_

# generate scaling factor to normalise by

WW_links_filt9$Scaling_factor<-  mean(WW_links_filt9$Abundance)/WW_links_filt9$Abundance

WW_links_filt9$Count_normalised<-WW_links_filt9$Count*WW_links_filt9$Scaling_factor
WW_links_filt9$Count_normalised<-WW_links_filt9$Count_normalised+1 # add psudocount of one so we can ro

WW_links_filt9$Count_normalised<-round(WW_links_filt9$Count_normalised,0)
```

Normalise by plasmid size

```
WW_links_filt9$PlasmidLength<- vlookup(WW_links_filt9$Cluster, plasmid_length, lookup_column = 1, resul


WW_links_filt9$Scaling_factor_length<-  mean(WW_links_filt9$PlasmidLength)/WW_links_filt9$PlasmidLength

WW_links_filt9$Count_normalised1<-WW_links_filt9$Count_normalised*WW_links_filt9$Scaling_factor_length
WW_links_filt9$Count_normalised1<-WW_links_filt9$Count_normalised1+1 # add psudocount of one so we can

WW_links_filt9$Count_normalised1<-round(WW_links_filt9$Count_normalised1,0)
WW_links_filt9$Count_normalised<-WW_links_filt9$Count_normalised1
```

Process data for final analysis

Convert count table into adjacency matrix

We have finished filtering, but at this point we need to generate a few bits of final information so that we can put it all together into a phyloseq object. Phyloseq is an R package that is used to handle 16S microbiome data, but it can also be used to handle other sorts of data like bacteria-plasmid associations.

We need: 1) A MAG x Plasmid count table/adjacency matrix 2) A dataframe with metadata for each MAG (eg Taxonomy) 3) a dataframe with metadata for each plasmid (ie Whether it has a resistance gene or not)

```
# first we make the MAG x plasmic count table

WW_links_clustered2<-WW_links_filt9[,c("MAG_assigment", "Cluster", "Count_normalised")]

count_table_clustered<- pivot_wider( WW_links_clustered2, names_from = MAG_assigment, values_from = Cou


##turn NAs to zeros
count_table_clustered_dt<-as.data.table(count_table_clustered)
count_table_clustered_dt[is.na(count_table_clustered_dt)] <- 0
count_table_clustered_dt[1:5, 1:5]
```

```
##    Cluster bin_1 bin_10 bin_103 bin_104
## 1:       2     4      8       7       0
## 2:       5    17      0       0     418
## 3:      30     6   1059       0       0
## 4:       1     0      2       0     241
## 5:       3     0      3      49       0
```

```
count_table_clustered_df<-data.frame(count_table_clustered_dt)
#count_table_clustered_df[1:5,1:5]

#make first col row names
row.names(count_table_clustered_df)<-count_table_clustered_df$Cluster

# remove first column
count_table_clustered_df<-count_table_clustered_df[,-1]
count_table_clustered_df[1:5,1:5]
```

```
##    bin_1 bin_10 bin_103 bin_104 bin_105
## 2      4      8       7       0      15
## 5     17      0       0     418       0
## 30     6   1059       0       0      52
## 1      0      2       0     241       0
## 3      0      3      49       0       0
```

```
# plasmid clusters are rows, MAGs are columns


# remove normalised links under 5
count_table_clustered_df[count_table_clustered_df < 5] <- 0
```

Re-add MAGs that don't associate with any plasmids

- MAGs which don't associate with any plasmids were filtered out, therefore we need to add these back in

```
# add column for bacterial species that are not associated with any plasmids
# these got removed as they had zero counts
mag_IDs<-unique(mag_assignment$cluster)
mags_to_add<-data.frame(mag_IDs)
mags_to_add$Included<-mags_to_add$mag_IDs %in% names(count_table_clustered_df)
mags_to_add<-subset(mags_to_add, Included==FALSE)
df_to_add<-data.frame(matrix(0, nrow = nrow(count_table_clustered_df), ncol = nrow(mags_to_add)))
names(df_to_add)<-mags_to_add$mag_IDs
row.names(df_to_add)<-row.names(count_table_clustered_df)
new_count_table_clustered<-cbind(count_table_clustered_df, df_to_add)
```

Generate MAG metadata

```
phylophlan_taxonomy <- read.csv(here("DATA","phylophlan_taxonomy_updated.csv"))

MAG_metadata<-MAG_metadata[,1:8]

new_mag_metadata<-merge(MAG_metadata, phylophlan_taxonomy, by.x = "cluster_id", by.y = "Bin")
row.names(new_mag_metadata)<-new_mag_metadata$cluster_id
```

Generate plasmid metadata

```
plasmid_metadata_clustered<-data.frame(unique(WW_links_filt9$Cluster))
names(plasmid_metadata_clustered)<-"contigs_plasmids"

plasmid_metadata_clustered$Resistance<- plasmid_metadata_clustered$contigs_plasmids %in% resistance_clus

row.names(plasmid_metadata_clustered)<-plasmid_metadata_clustered$contigs_plasmids

# turn it into a tax_table object
plasmid_metadata_clustered<-tax_table(plasmid_metadata_clustered)
taxa_names(plasmid_metadata_clustered)<-data.frame(plasmid_metadata_clustered)$ta1
```

Convert all layers into phyloseq object

```
OTU = otu_table(new_count_table_clustered, taxa_are_rows = TRUE)
dim(new_count_table_clustered)
```

```
## [1] 109 379
```

```
plasmid_ps_clustered<-merge_phyloseq(OTU, sample_data(new_mag_metadata), plasmid_metadata_clustered)

plasmid_ps_clustered
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 109 taxa and 379 samples ]
## sample_data() Sample Data:       [ 379 samples by 15 sample variables ]
## tax_table()   Taxonomy Table:    [ 109 taxa by 2 taxonomic ranks ]
```

```
## saveRDS(plasmid_ps_clustered, "plasmid_ps_clustered_genomad_final.RDS")
# saveRDS(plasmid_ps_clustered, "C:/Users/risel/Dropbox/Sommer postdoc/Plasmid project/PlasmidProjectAn
```