Supplementary data for the manuscript on the Pyrenodesmia Syngameon. Part I

Fernando Fernández Mendoza et al.

Contents

1 Introduction **2**

2 Part I. Population and phylogenetic structure **2**

2.1 I.I Estimation of evolutionary populations in BAPS . . . 2
2.1.1 Supplementary Figure 0. Species distribution in the dataset . . . 5
2.1.2 Supplementary Figure 1. Differential contribution to admixture of each gene-pool. . . . 8
2.1.3 Supplementary Figure 2. Contribution of each genepool to the minoritary admixed fractions. . . . 9
2.1.4 Supplementary Table 1. Additive Contribution of each genepool to the admixed signal. 10
2.1.5 Supplementary Figure 3 (A-E). Contribution of each genepool to the minoritary admixed fractions. Added contribution of each component to the minor component vs fraction of the total contribution to ancestry of the whole dataset. . . . 11
2.1.6 Supplementary table 2. Contribution of each genepool to the minoritary admixed fractions. . . . 13
2.1.7 Supplementary Figure 4. Contribution of each genepool to the minoritary admixed fractions. . . . 13
2.1.8 Supplementary table 3. Contribution of each genepool to the minoritary admixed fractions. . . . 14
2.1.9 Supplementary table 4. Analisis of variance. . . . 14
2.1.10 Supplementary Figure 5. Contribution of each genepool to the minoritary admixed fractions considering at least 0.1 as a significant fraction. . . . 15
2.1.11 Supplementary table 5. Analisis of variance using average assignments . . . 16
2.1.12 Supplementary table 6. Analisis of variance using summed assignments. . . . 16
2.1.13 Supplementary table 7. Contribution of each gene-pool to the minor admixed component across the dataset. . . . 17
2.1.14 Supplementary Figure 6. . . . 17
2.1.15 Supplementary table 8. . . . 17
2.1.16 Supplementary table 9. . . . 18
2.1.17 Supplementary table 10. . . . 18
2.1.18 Geographic distribution of BAPS gene-pools . . . 19
2.1.19 Association between gene-pools and morphospecies . . . 28
2.1.20 How much of the species splitting is explained by the assignment to gene-pools? . . . 31
2.2 I.II Phylogenetic signal in BAPS clusters and species IDs . . . 33
2.2.1 Phylogenetic signal in BAPS clusters . . . 33
2.2.2 Migrate-n migration network between BAPS gene clusters . . . 42
2.2.3 Migrate-n migration network between BAPS gene clusters using mat genes . . . 46
2.2.4 Phylogenetic signal in Species IDs . . . 52
2.3 I.III Concordance between gene phylogenies . . . 54
2.3.1 ML . . . 54
2.3.2 Bayesian. Concordance within loci . . . 56

2.3.3 Within and between loci . . . 58
2.3.4 Between loci, random vs observed . . . 58
2.4 I.IV Multilocus summary of bGMYC delimitations . . . 59
2.4.1 Averaging coassignment matrices . . . 59
2.4.2 Extended range of K for kmedoids K=1:200 . . . 59
2.4.3 Reduced range of K values K=1:20 . . . 61
2.5 Single locus bGMYC . . . 66
2.5.1 ITS . . . 66
2.5.2 Beta Tubulin . . . 70
2.5.3 Elongation factor alpha . . . 74
2.5.4 MCM7 . . . 78
2.5.5 RPB1 . . . 83

3 Part II. Mating type coocurrence 87
3.1 II.I Read data and CD-HIT . . . 87
3.2 II.II Compartmentalization and modularity . . . 93
3.2.1 At haplotype level (Only compartments) . . . 93
3.2.2 Comps vs species . . . 95
3.2.3 At 99% similarity level . . . 95
3.3 II.III Networks at individual level . . . 104
3.3.1 Compartmentalization . . . 105
3.3.2 Test modularity algorythms . . . 105
3.3.3 Model selection based on BF . . . 115

1 Introduction

This document contains the suppelementary material necessary for the interpretation of the main manuscript regarding the taxonomically complex genus Pyrenodesmia. The supplement is organized in three parts.

The first part deals with population genetics and phylogenetic methods and aims at discussing the extent to which different methodological approaches can be used to identify species in this genus.

The second part uses sequences of the Mating type idiomorphs found in dykariotic samples to identify the extent to which there is a missmatch between postzigotic and prezygotic reproductive isolation.

Finally the third part surveys genomic signatures of hybridization encountered in the genome of Pyrenodesmia erodens.

The R code used for data processing, as untidy as it may be is shown together with the results. Some parts of the code have been commented out and substituted by importing a final environment to avoid repeating costly computations while compiling the supplement.

2 Part I. Population and phylogenetic structure

2.1 I.I Estimation of evolutionary populations in BAPS

```
knitr::opts_chunk$set(echo = TRUE)
load("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v15.Rdata")
#load("/Users/Fernando/Desktop/01_Sanger_paper/14_bGMYC_All_loci/03_Processed/workspace_step3.Rdata")

library(ggplot2)
library(reshape2)
library(fpc)
library(vcd)
```

```
## Loading required package: grid
```

```
library(ape)
library(phangorn)
library(bipartite)
```

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-2
##
## Attaching package: 'vegan'
## The following objects are masked from 'package:phangorn':
##
##     diversity, treedist
## Loading required package: sna
## Loading required package: statnet.common
##
## Attaching package: 'statnet.common'
## The following objects are masked from 'package:base':
##
##     attr, order
## Loading required package: network
##
## 'network' 1.17.1 (2021-06-12), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information
## sna: Tools for Social Network Analysis
## Version 2.6 created on 2020-10-5.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.
##
## Attaching package: 'sna'
## The following objects are masked from 'package:ape':
##
##     consensus, degree
##  This is bipartite 2.17.
##  For latest changes see versionlog in ?"bipartite-package". For citation see: citation("bipartite").
##  Have a nice time plotting and analysing two-mode networks.
##
## Attaching package: 'bipartite'
## The following object is masked from 'package:vegan':
##
##     nullmodel
```

```
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
colorinos.bipolar<-c("#CAB2D6",
                     "#A6CEE3",
                     "#999999",
                     "#6A3D9A",
                     "#339933",
                     "#E31A1C",
                     "#FF7F00",
                     "#1F78B4",
                     "#B2DF8A",
                     "#FB9A99",
                     "#720D0E")
#
#
# Take out samples 810,811 Americanas and correct 281 and 357
#
#baps.adm["281",c(3,4,5,6)]<-baps.adm["1111",c(3,4,5,6)]
#baps.adm["357",c(3,4,5,6)]<-baps.adm["355",c(3,4,5,6)]
#baps.adm["194",c(3,4,5,6)]<-baps.adm["192",c(3,4,5,6)]
#baps.adm["261",c(3,4,5,6)]<-baps.adm["259",c(3,4,5,6)]
#baps.adm["262",c(3,4,5,6)]<-baps.adm["259",c(3,4,5,6)]
#baps.adm["318",c(3,4,5,6)]<-baps.adm["317",c(3,4,5,6)]
#baps.adm["318",2]<-baps.adm["17",2]
#baps.adm["283",2]<-baps.adm["17",2]
#baps.adm["178",2]<-baps.adm["17",2]
#baps.adm["1181",2]<-baps.adm["17",2]
#baps.adm["5",c(3,4,5,6)]<-baps.adm["3",c(3,4,5,6)]
#baps.adm["5.1",c(3,4,5,6)]<-baps.adm["3",c(3,4,5,6)]
#baps.adm["439",c(3,4,5,6)]<-baps.adm["438",c(3,4,5,6)]
#baps.adm["753",c(3,4,5,6)]<-baps.adm["751",c(3,4,5,6)]
#baps.adm["753.1",c(3,4,5,6)]<-baps.adm["751",c(3,4,5,6)]
#baps.adm["792",c(3,4,5,6)]<-baps.adm["794",c(3,4,5,6)]
#baps.adm["792.1",c(3,4,5,6)]<-baps.adm["794",c(3,4,5,6)]
#baps.adm["364",c(3,4,5,6)]<-baps.adm["854",c(3,4,5,6)]
#save.image("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v12.Rdata")
#
# Did not trim
#
#baps.adm<-baps.adm[rownames(baps.adm)!="810",]
#baps.adm<-baps.adm[rownames(baps.adm)!="811",]
#
# Last edit, ID correction
```

```
#
#a<-read.table("/Users/Fernando/Desktop/01_paper_sanger_drive/reidentify.txt",header=FALSE,sep="\t")
#rownames(a)<-a[,1]
#baps.adm$Species<-a[as.character(sapply(strsplit(rownames(baps.adm),"\\."),`[`,1)),2]
#baps.adm["1049",3:6]<-c(439,6.465392,45.991996,1755)
#save.image("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v13.Rdata")

#

sample_names<-sapply(strsplit(rownames(baps.adm),"\\."),`[`,1)
condensed_baps_adm<-baps.adm[!duplicated(sample_names),]
foo<-aggregate(x=baps.adm[,13:22],by=list(sample_names),FUN=mean)
rownames(foo)<-foo$Group.1
condensed_baps_adm[,13:22]<-foo[rownames(condensed_baps_adm),-1]
condensed_baps_adm<-cbind(sample=rownames(condensed_baps_adm),condensed_baps_adm)
condensed_baps_adm$name<-apply(condensed_baps_adm[,14:23],1,FUN=which.max)

orden.foo<-rownames(condensed_baps_adm)[order(condensed_baps_adm$name,-condensed_baps_adm$X1,-condensed

foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo)
foo<-melt(foo[,c(1,15:24)])
```

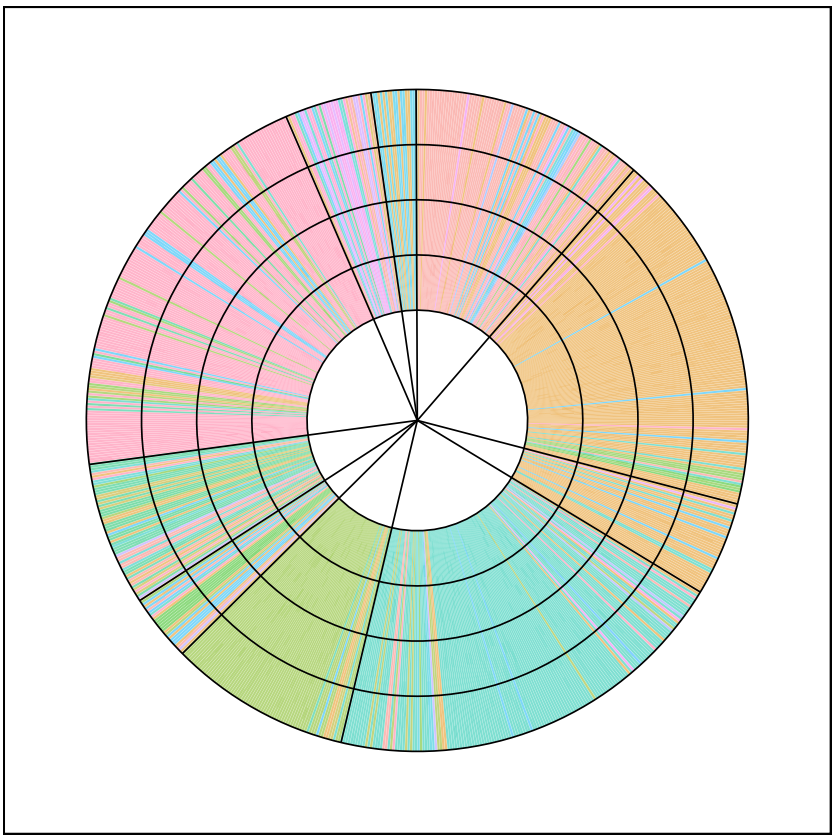
```
## Using orden as id variables
```

```
#### Rarefaction curves

draw_haplo<-function(x)
  {
require(future.apply)
plan(multicore,workers=100)
foo<-x
foo<-foo[row.names(foo)!="Xanpa",]
foo.list<-seq(from=5,to=dim(foo)[1],by=50)
salida<-future_lapply(foo.list,FUN=function(z){
  sapply(c(1:100),FUN=function(y){
    mean(
      dist(
        foo[sample(1:dim(foo)[1],size=z),]
  )
)
}
)
})
return(melt(salida))
}

#fer<-draw_haplo(seq.dataset[[1]])
```

2.1.1 Supplementary Figure 0. Species distribution in the dataset

```
species_foo<-condensed_baps_adm[orden.foo,]
species_foo<-cbind(orden=factor(c(1:dim(species_foo)[1])),species_foo)
species_foo<-melt(species_foo[,c(1,3)])
```

```
## Using orden, Species as id variables
p<-ggplot(species_foo) +
  geom_bar(aes(x=orden, y=1,fill=Species),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar) +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.25, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.5, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.75, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p+theme(legend.position = "none"))
```



```
legend <- cowplot::get_legend(p)
grid.newpage()
grid.draw(legend)
```

#CAB2D6

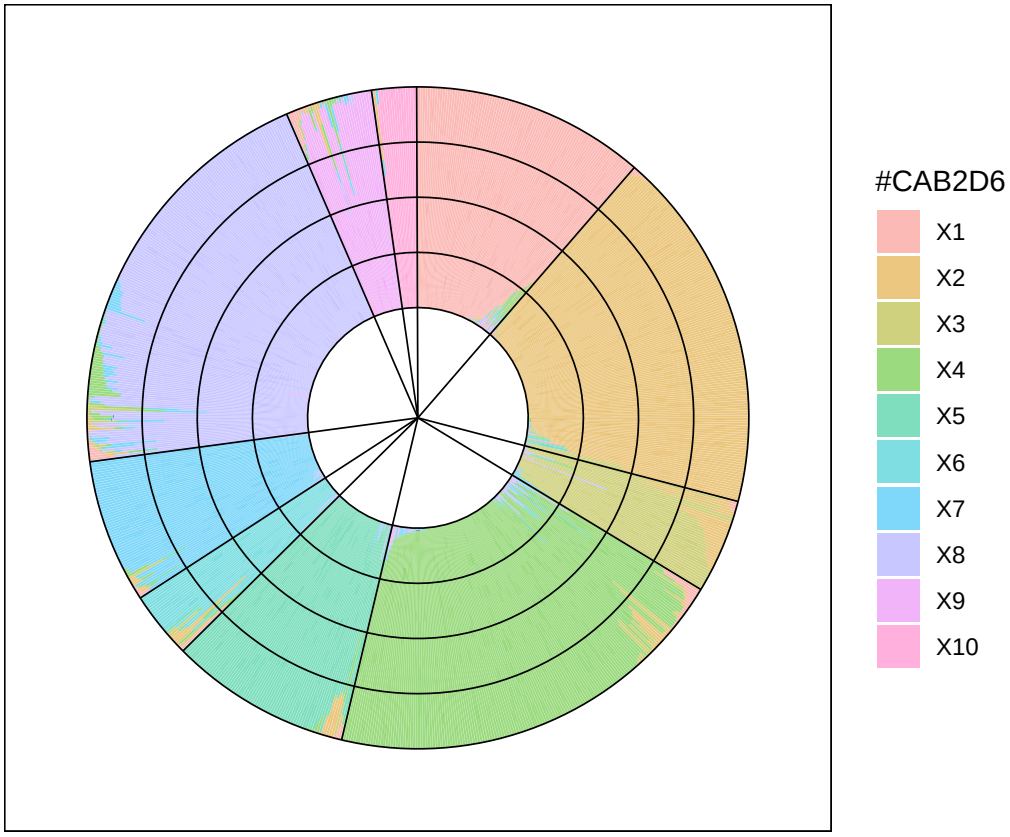
C. albopruinosa
C. albopustulata
C. albovariegata
C. alociza
C. alociza_non_oleose
C. badioreagens
C. blastidiate
C. burgos_7
C. cf albopruinosa
C. cf. circumalbata
C. cf. micromarina
C. chalybaea
C. circumalbata
C. circumalbata candida
C. concreticola
C. diphyodes
C. diphyodes_micro
C. diphyodes?
C. disjuncta
C. erodens
C. helygeoides?
C. medit
C. medit 3
C. medit1
C. medit2
C. micro
C. micro_endo
C. micromarina
C. micromontana
C. microsp.A
C. microsp.B
C. microstepossa
C. molariformis
C. sp teruel
C. sp.
C. sp. asia
C. sp. gr
C. sp. sorediate
C. transcaspica
C. transcaspica sorediate
C. variabilis
C. variabilis bullate

```
print(p+theme(legend.position = "none")+facet_wrap(vars(Species)))
```

2.1.2 Supplementary Figure 1. Differential contribution to admixture of each gene-pool.

```
p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=value,fill=variable),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar) +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.25, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.5, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 0.75, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
```

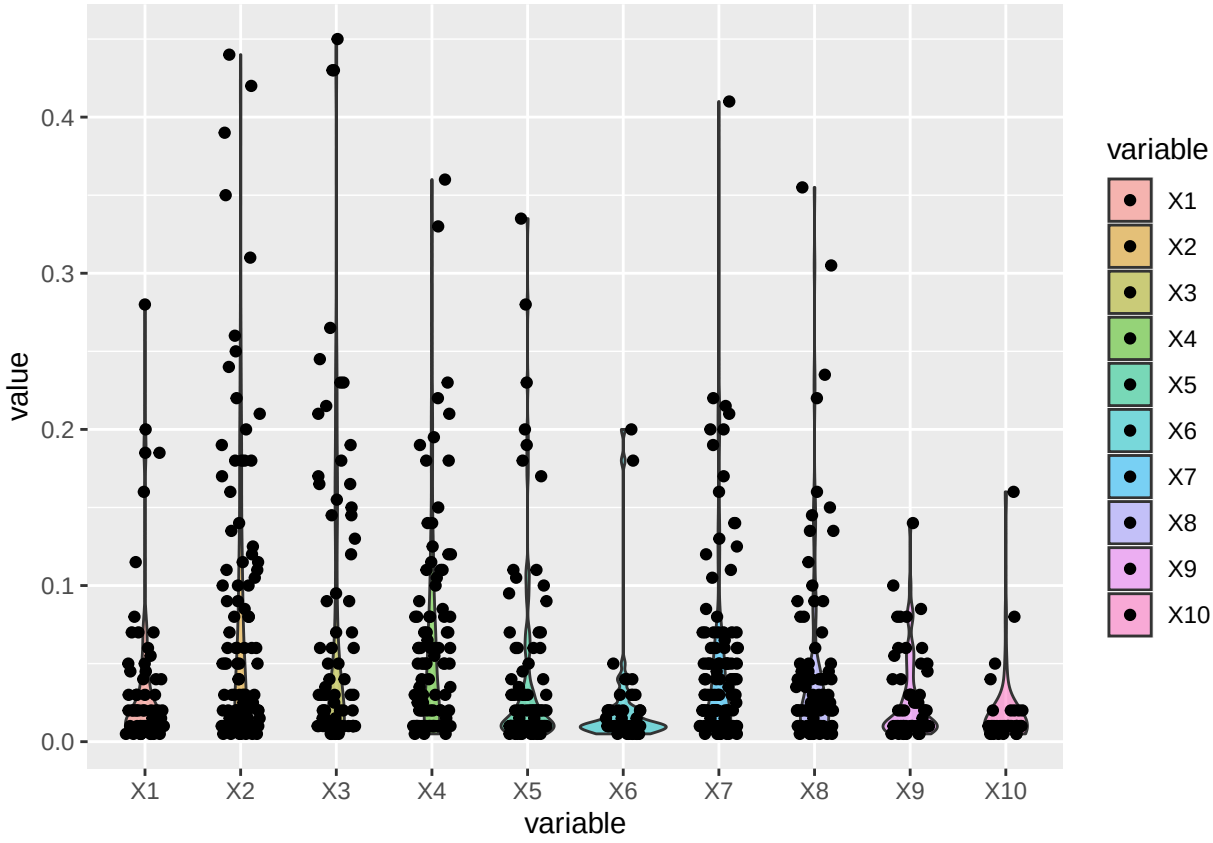
```
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



2.1.3 Supplementary Figure 2. Contribution of each genepool to the minoritary admixed fractions.

```
library(dplyr)
min_fract<-melt(condensed_baps_adm[,c(1,14:23)])

## Using sample as id variables

foo<-sum(min_fract$value)
min_fract %>% group_by(variable) %>% arrange(value,.by_group = TRUE) %>% summarize(Sum = sum(value, na.
```

```
min_fract %>% group_by(sample) %>% slice_max(.,order_by = value) -> max_fract
min_fract<-min_fract[min_fract$value!=0&min_fract$value!=1,]
#min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE)

min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE) %>% dplyr::slice(1:n()-1) %>% ggplot
```



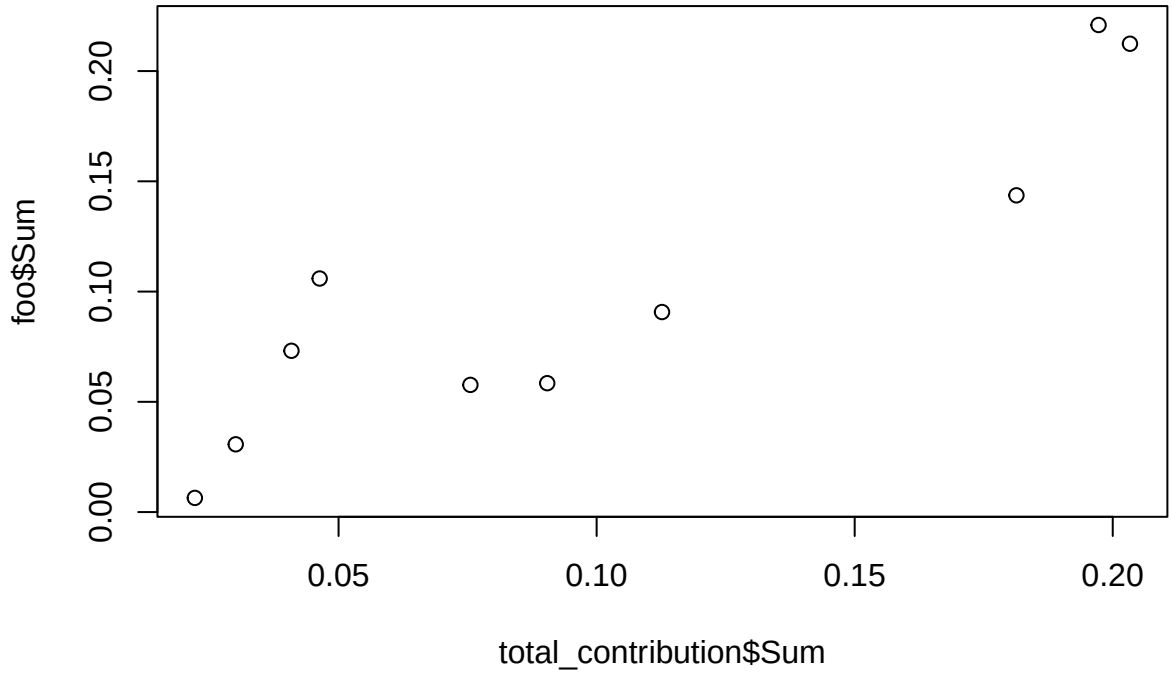
2.1.4 Supplementary Table 1. Additive Contribution of each genepool to the admixed signal.

```
foo<-sum(min_fract$value)
min_fract %>% group_by(variable) %>% arrange(value,.by_group = TRUE) %>% summarize(Sum = sum(value, na.
table_admixture_anovas<-cbind(total=total_contribution,admix=foo$Sum)
kable(table_admixture_anovas)
```

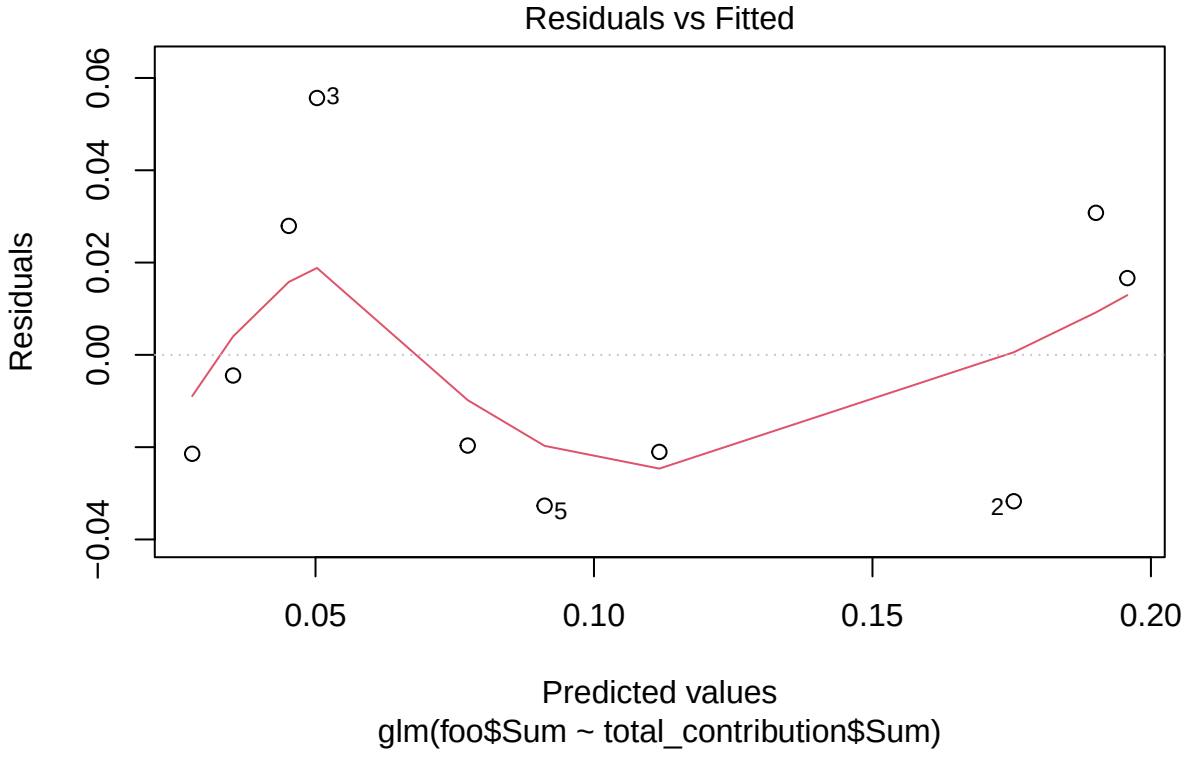
total.variable	total.Sum	admix
X1	0.1126699	0.0907123
X2	0.1813350	0.1436467
X3	0.0463350	0.1059259
X4	0.1972512	0.2208974
X5	0.0904308	0.0584473
X6	0.0300789	0.0307265
X7	0.0755400	0.0576781
X8	0.2033374	0.2123932
X9	0.0408677	0.0731481
X10	0.0221541	0.0064245

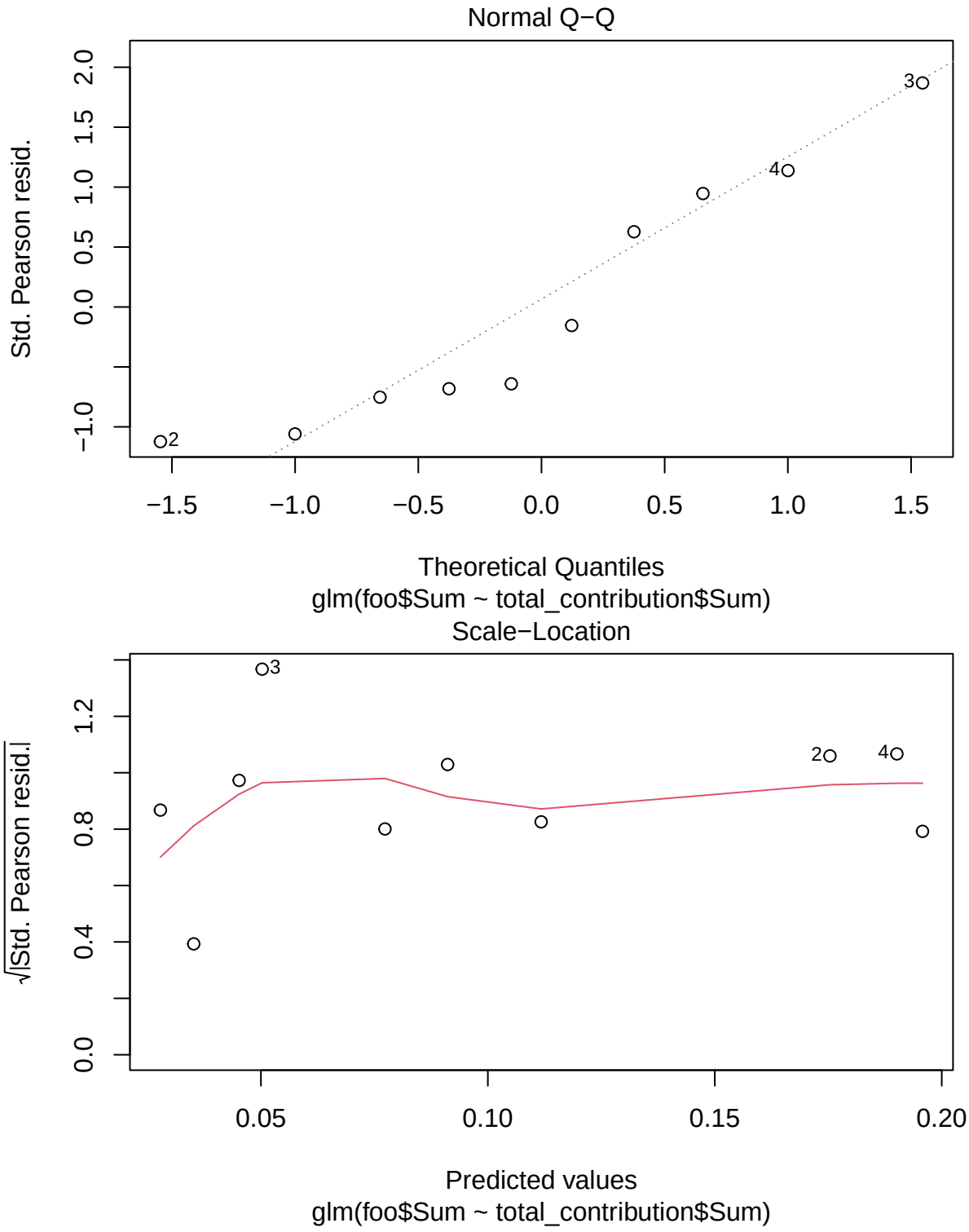
2.1.5 Supplementary Figure 3 (A-E). Contribution of each genepool to the minoritary admixed fractions. Added contribution of each component to the minor component vs fraction of the total contribution to ancestry of the whole dataset.

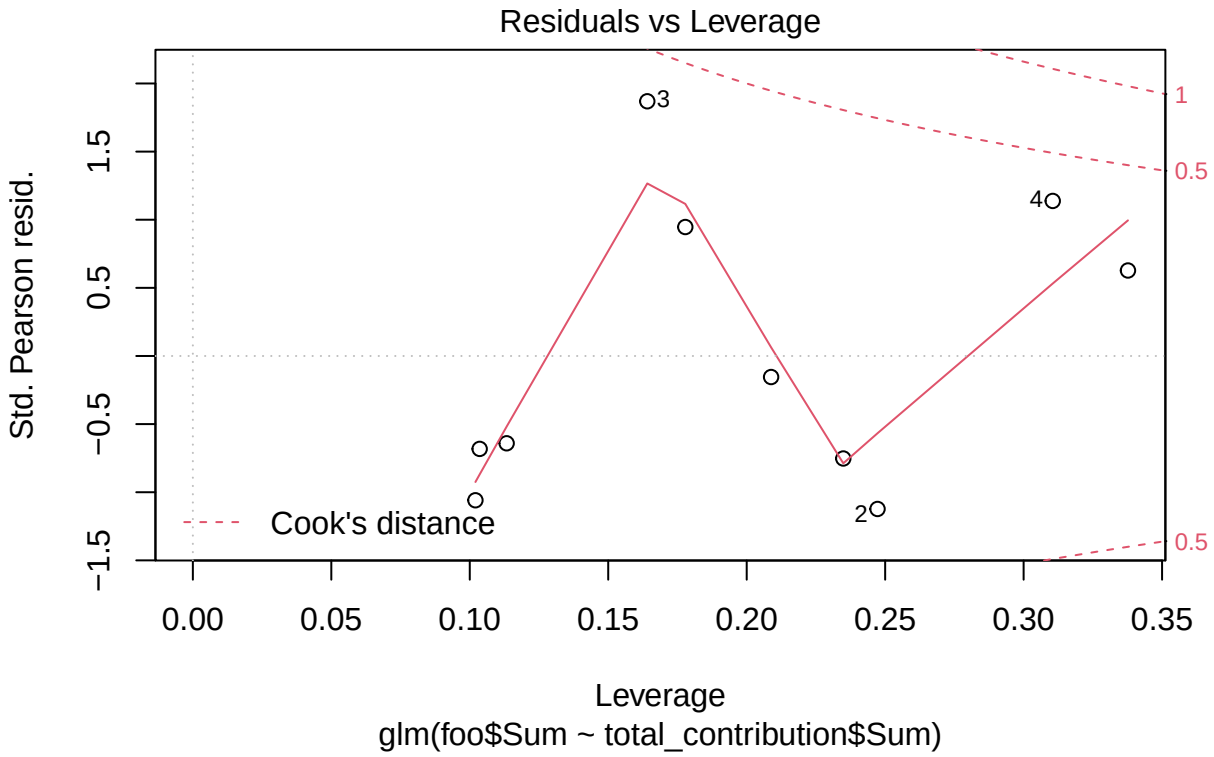
```
plot(foo$Sum~total_contribution$Sum)
```




```
linear_model<-glm(foo$Sum~total_contribution$Sum)
plot(linear_model)
```




Normal Q−Q
Std. Pearson resid.
−1.0
0.0
0.5
1.0
1.5
2.0
2
3
4
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
glm(foo$Sum ~ total_contribution$Sum)
Scale−Location
√|Std. Pearson resid.|
0.0
0.4
0.8
1.2
3
2
4
0.05
0.10
0.15
0.20
Predicted values
glm(foo$Sum ~ total_contribution$Sum)



2.1.6 Supplementary table 2. Contribution of each genepool to the minoritary admixed fractions.

```
kable(anova(linear_model))
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	0.0470688
total_contribution$Sum	1	0.0385835	8	0.0084853

2.1.7 Supplementary Figure 4. Contribution of each genepool to the minoritary admixed fractions.

```
min_fract<-melt(condensed_baps_adm[,c(1,14:23)])
```

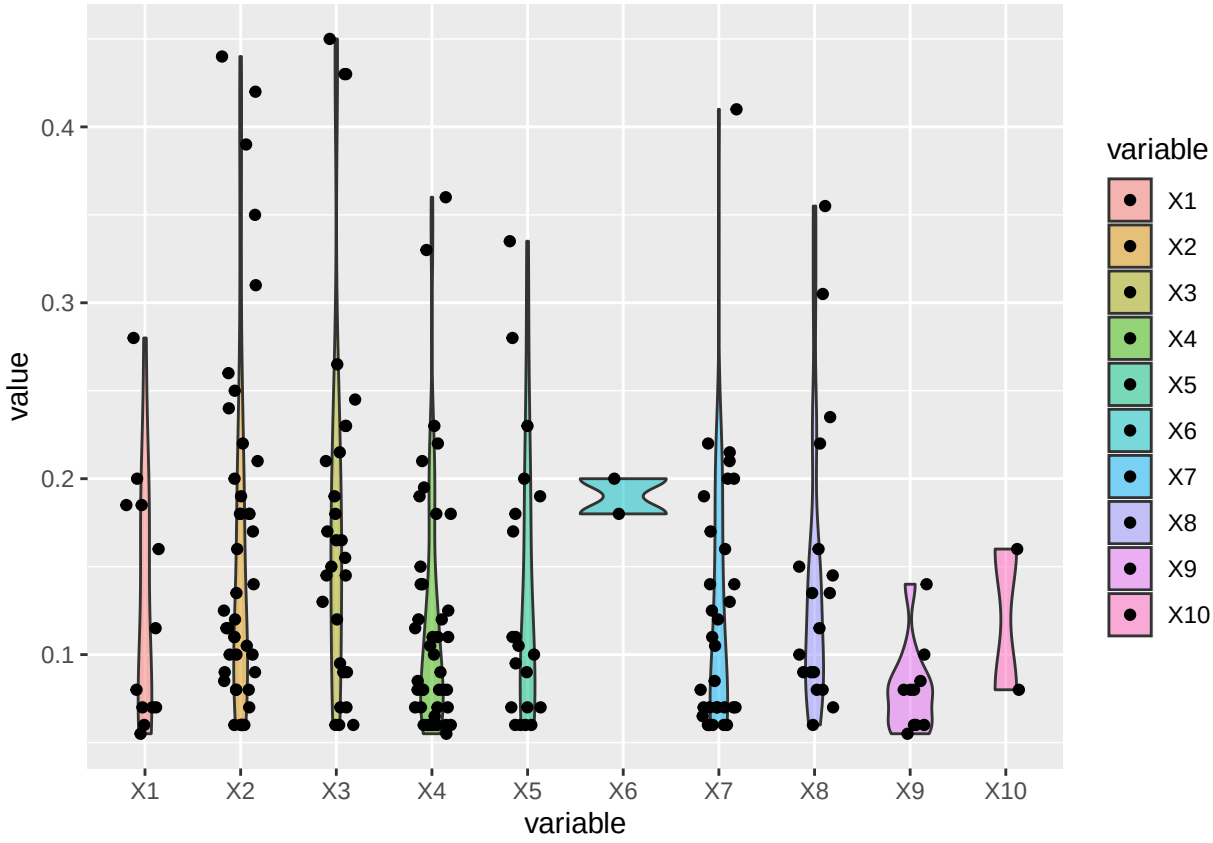
```
## Using sample as id variables
```

```
min_fract<-min_fract[min_fract$value>0.05&min_fract$value<0.95,]
min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE)
```

```
## # A tibble: 396 x 3
## # Groups:   sample [187]
##    sample variable value
##    <chr>  <fct>    <dbl>
##  1 1005   X5       0.105
##  2 1005   X3       0.15
##  3 1005   X7       0.675
##  4 1008   X7       0.9
##  5 1016   X4       0.065
##  6 1016   X8       0.235
```

```
##  7 1016   X1       0.28
##  8 1016   X6       0.335
##  9 1022   X8       0.135
## 10 1022   X1       0.86
## # ... with 386 more rows
```

```
min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE) %>% dplyr::slice(1:n()-1) %>% ggplot
```



2.1.8 Supplementary table 3. Contribution of each genepool to the minoritary admixed fractions.

```
min_fract %>% group_by(variable) %>% summarize(Mean = mean(value, na.rm=TRUE), Sum = sum(value, na.rm=TI

foo %>% glm(Mean~Max,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	0.1022319
Max	1	0.0007222	8	0.1015098

```
table_admixture_anovas<-cbind(table_admixture_anovas,Mean05=foo$Mean,Sum05=foo$Sum,Max05=foo$Max)
```

2.1.9 Supplementary table 4. Analisis of variance.

```
foo %>% lm(Sum~Max,data=.) %>% anova %>% kable()
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Max	1	791.2860	791.28595	11.49888	0.0094859
Residuals	8	550.5135	68.81418	NA	NA

2.1.10 Supplementary Figure 5. Contribution of each genepool to the minoritary admixed fractions considering at least 0.1 as a significant fraction.

```
min_fract<-melt(condensed_baps_adm[,c(1,14:23)])
```

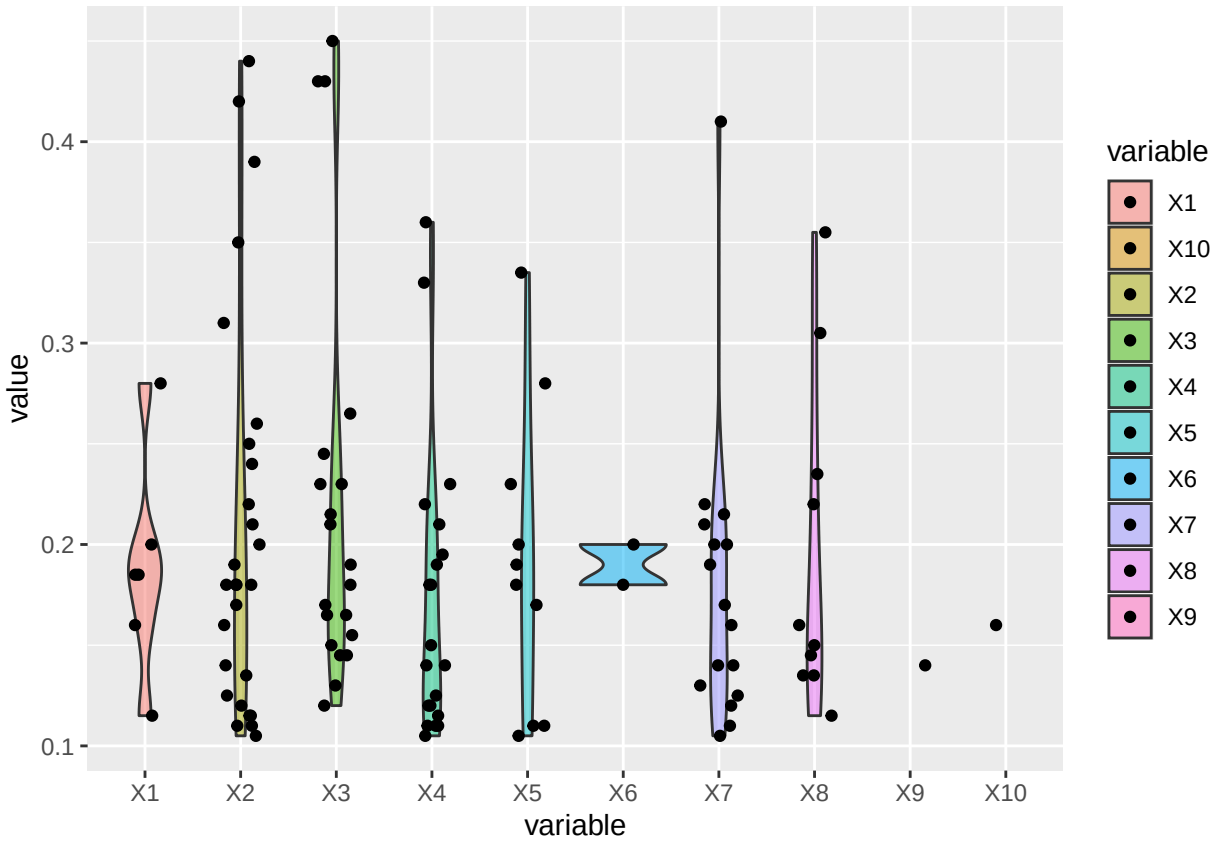
```
## Using sample as id variables
```

```
min_fract<-min_fract[min_fract$value>0.10&min_fract$value<0.90,]
min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE)
```

```
## # A tibble: 242 x 3
## # Groups:   sample [129]
##    sample variable value
##    <chr>  <fct>    <dbl>
##  1 1005   X5       0.105
##  2 1005   X3       0.15
##  3 1005   X7       0.675
##  4 1016   X8       0.235
##  5 1016   X1       0.28
##  6 1016   X6       0.335
##  7 1022   X8       0.135
##  8 1022   X1       0.86
##  9 1031   X1       0.895
## 10 1033   X2       0.105
## # ... with 232 more rows
```

```
min_fract %>% group_by(sample) %>% arrange(value,.by_group = TRUE) %>% dplyr::slice(1:n()-1) %>% ggplot
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



2.1.11 Supplementary table 5. Analisis of variance using average assignments

```
min_fract %>% group_by(variable) %>% summarize(Mean = mean(value, na.rm=TRUE), Sum = sum(value, na.rm=TR

foo %>% lm(Mean~Max,data=.) %>% anova
```

```
## Analysis of Variance Table
##
## Response: Mean
##           Df   Sum Sq   Mean Sq F value Pr(>F)
## Max        1 0.000626 0.0006264  0.0519 0.8254
## Residuals  8 0.096479 0.0120599
```

2.1.12 Supplementary table 6. Analisis of variance using summed assignments.

```
foo %>%  lm(Sum~Max,data=.) %>% anova
```

```
## Analysis of Variance Table
##
## Response: Sum
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Max        1 277.94 277.945   18.15 0.00276 **
## Residuals  8 122.51  15.314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

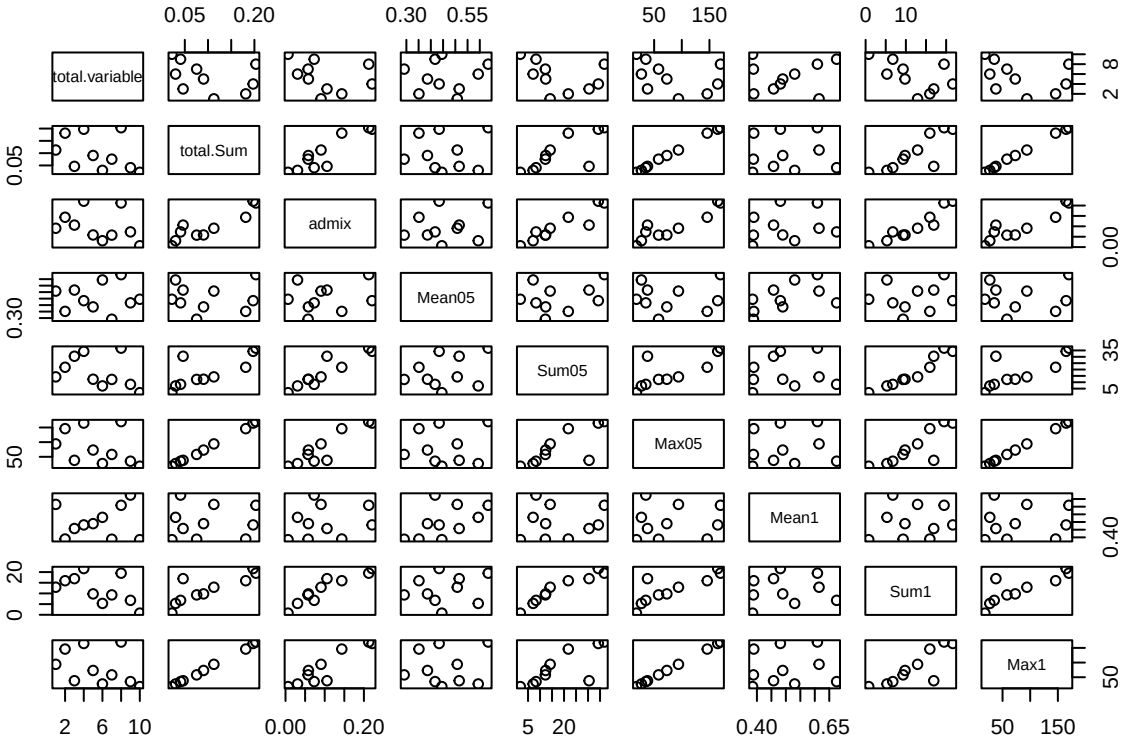
2.1.13 Supplementary table 7. Contribution of each gene-pool to the minor admixed component across the dataset.

```
table_admixture_anovas<-cbind(table_admixture_anovas,Mean1=foo$Mean,Sum1=foo$Sum,Max1=foo$Max)
kable(table_admixture_anovas)
```

total.variable	total.Sum	admix	Mean05	Sum05	Max05	Mean1	Sum1	Max1
X1	0.1126699	0.0907123	0.5080357	14.225	94	0.6147619	12.910	94
X2	0.1813350	0.1436467	0.3514516	21.790	146	0.3891463	15.955	146
X3	0.0463350	0.1059259	0.5158475	30.435	38	0.4575676	16.930	38
X4	0.1972512	0.2208974	0.4340506	34.290	165	0.4811111	21.650	165
X5	0.0904308	0.0584473	0.3862500	12.360	73	0.4892500	9.785	73
X6	0.0300789	0.0307265	0.5937500	7.125	27	0.5305000	5.305	27
X7	0.0755400	0.0576781	0.2909524	12.220	58	0.3879167	9.310	58
X8	0.2033374	0.2123932	0.6344828	36.800	170	0.6090625	19.490	170
X9	0.0408677	0.0731481	0.4172500	8.345	35	0.6755000	6.755	35
X10	0.0221541	0.0064245	0.4450000	1.780	18	0.3850000	0.770	18

2.1.14 Supplementary Figure 6.

```
plot(table_admixture_anovas)
```



2.1.15 Supplementary table 8.

```
table_admixture_anovas %>% glm(Max05~total.Sum,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	30794.40000

	Df	Deviance	Resid. Df	Resid. Dev
total.Sum	1	30742.67	8	51.73341

2.1.16 Supplementary table 9.

```
table_admixture_anovas %>% glm(Max1~total.Sum,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	30794.40000
total.Sum	1	30742.67	8	51.73341

2.1.17 Supplementary table 10.

```
table_admixture_anovas %>% glm(Mean05~total.Sum,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	0.1022319
total.Sum	1	0.0002686	8	0.1019633

```
table_admixture_anovas %>% glm(Mean1~total.Sum,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	0.0971052
total.Sum	1	0.0002626	8	0.0968427

```
table_admixture_anovas %>% glm(Sum05~total.Sum,data=.) %>% anova %>% kable ()
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	1341.7994
total.Sum	1	777.27	8	564.5295

```
table_admixture_anovas %>% glm(Sum1~total.Sum,data=.) %>% anova %>% kable ()
```

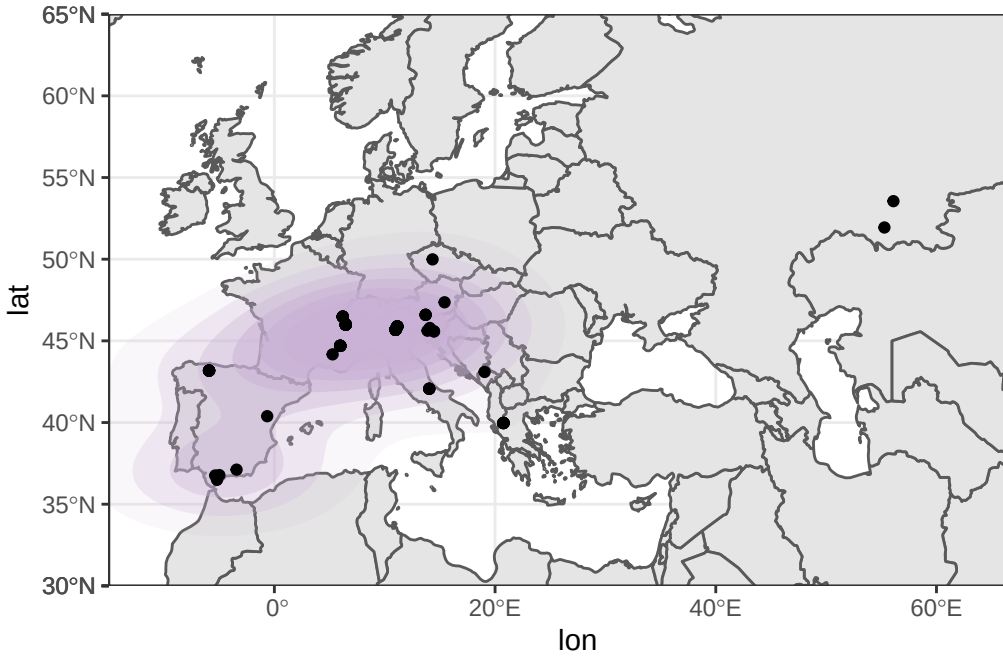
	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	400.4559
total.Sum	1	275.8771	8	124.5789

```
kable(anova(linear_model))
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	9	0.0470688
total_contribution$Sum	1	0.0385835	8	0.0084853

2.1.18 Geographic distribution of BAPS gene-pools

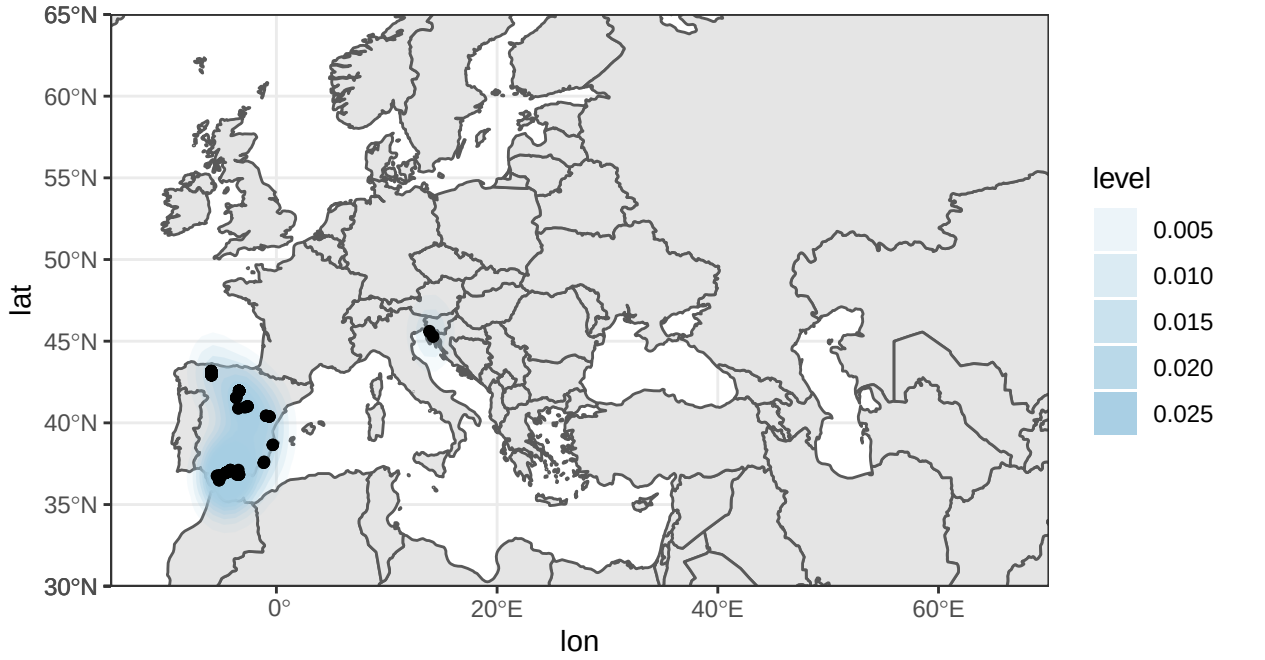
```
library("rnaturalearth")
library("rnaturalearthdata")

world <- ne_countries(scale = "medium", returnclass = "sf")
foo<-condensed_baps_adm[condensed_baps_adm$X1>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[1]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



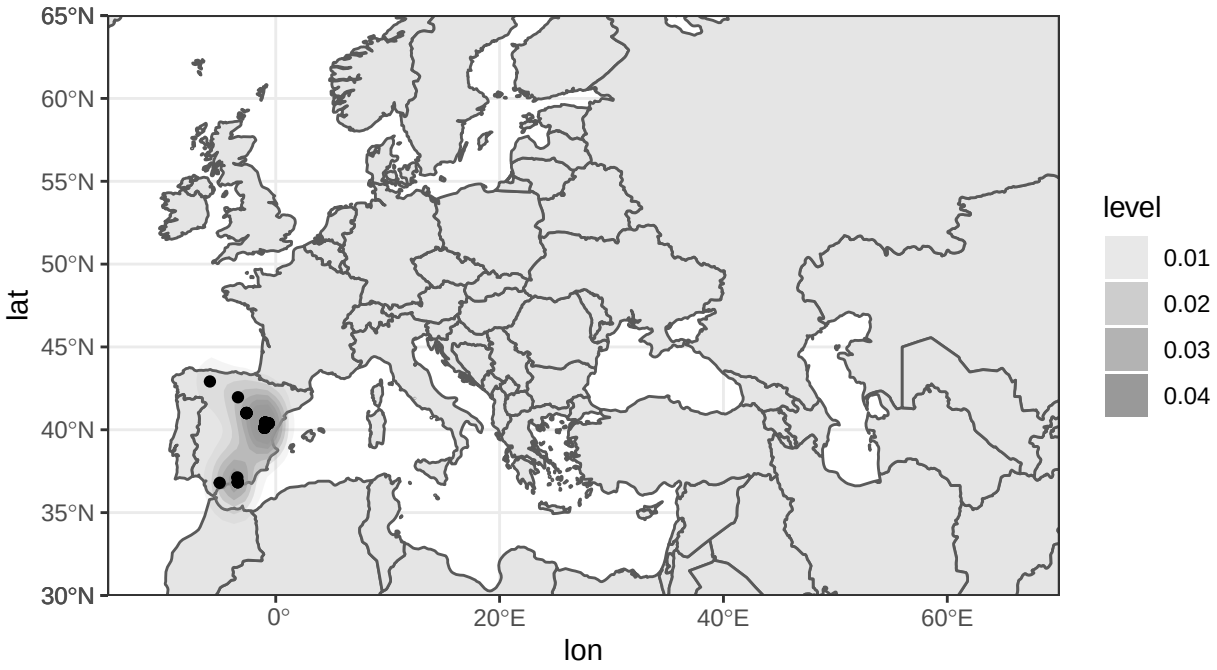
2.1.18.1 Supplementary figure

```
foo<-condensed_baps_adm[condensed_baps_adm$X2>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[2]) +
```

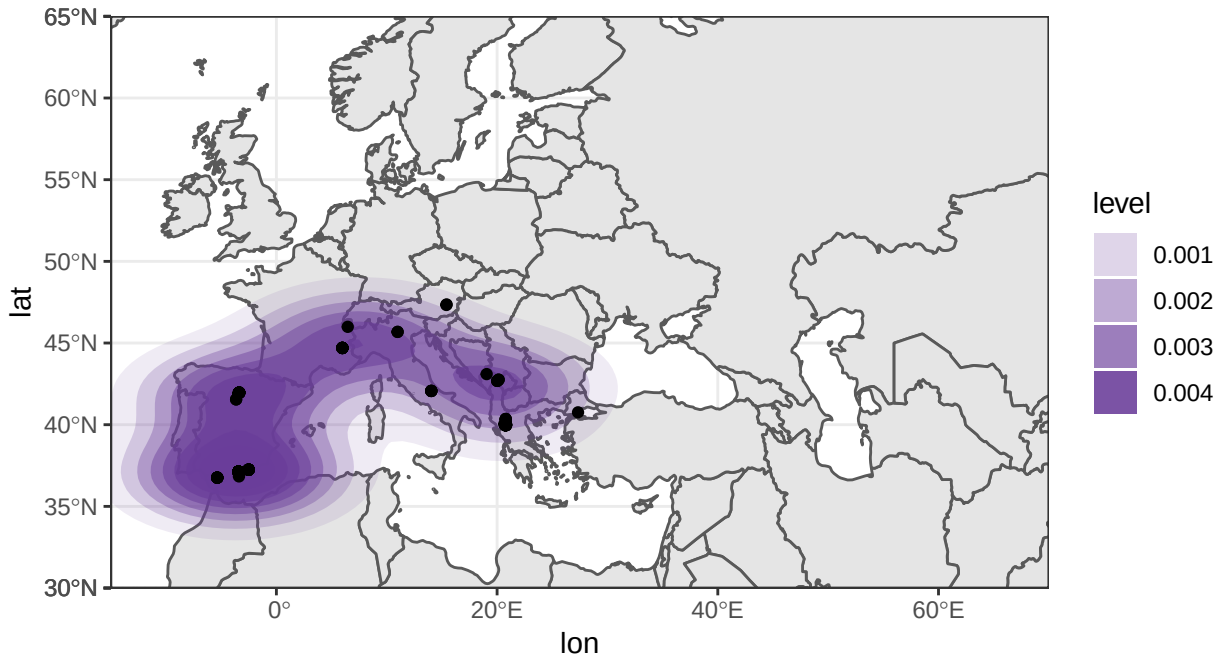
```
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



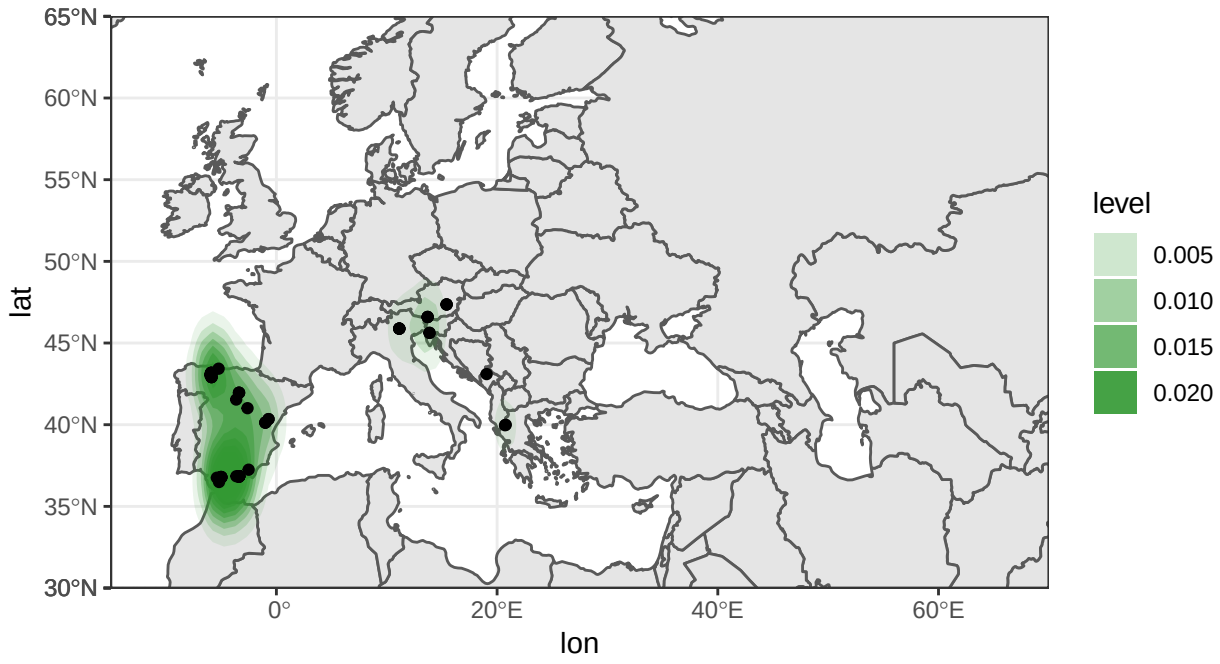

```
foo<-condensed_baps_adm[condensed_baps_adm$X3>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[3]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



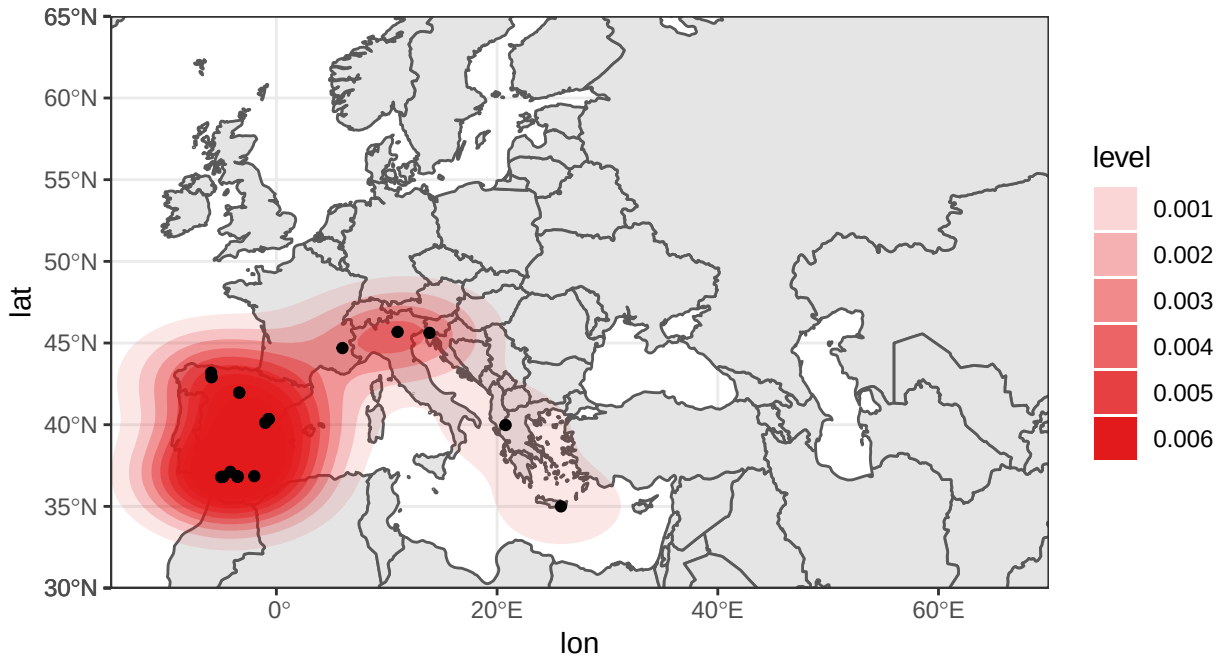

```
foo<-condensed_baps_adm[condensed_baps_adm$X4>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[4]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



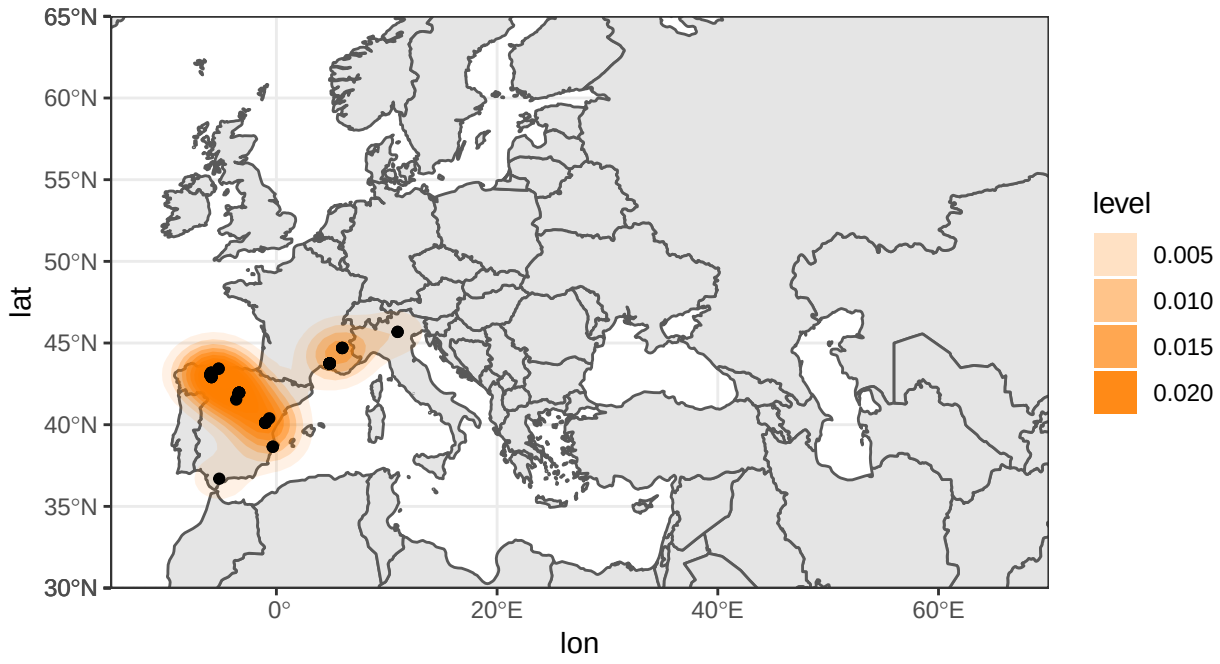

```
foo<-condensed_baps_adm[condensed_baps_adm$X5>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[5]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



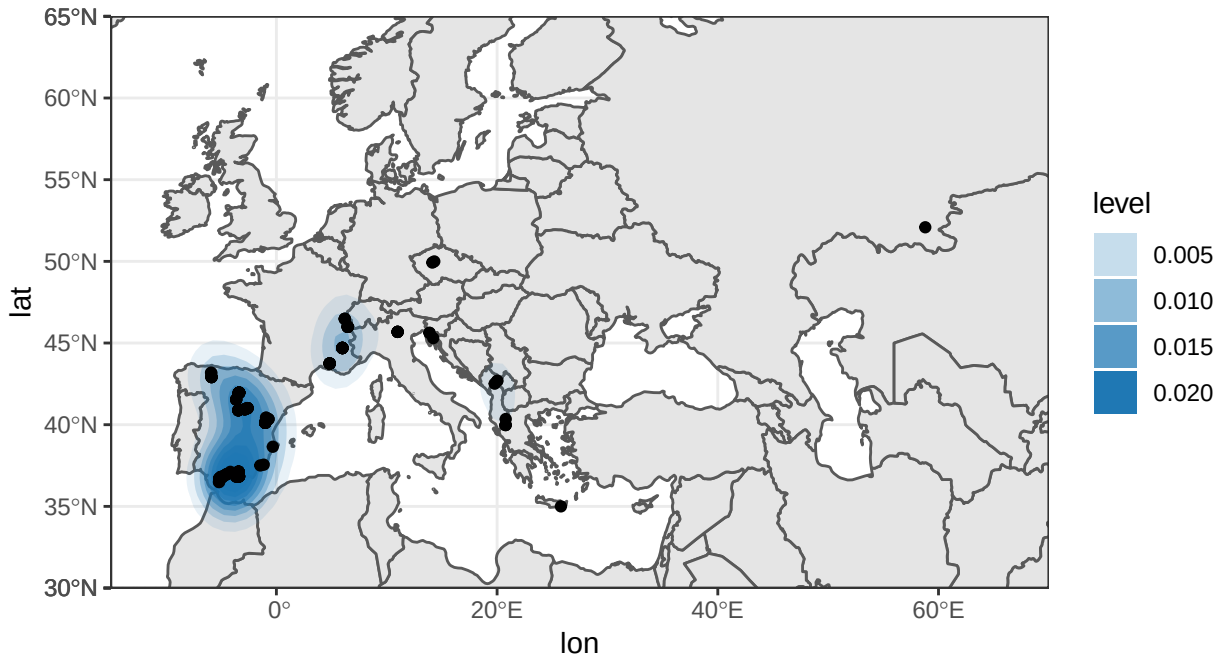

```
foo<-condensed_baps_adm[condensed_baps_adm$X6>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[6]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



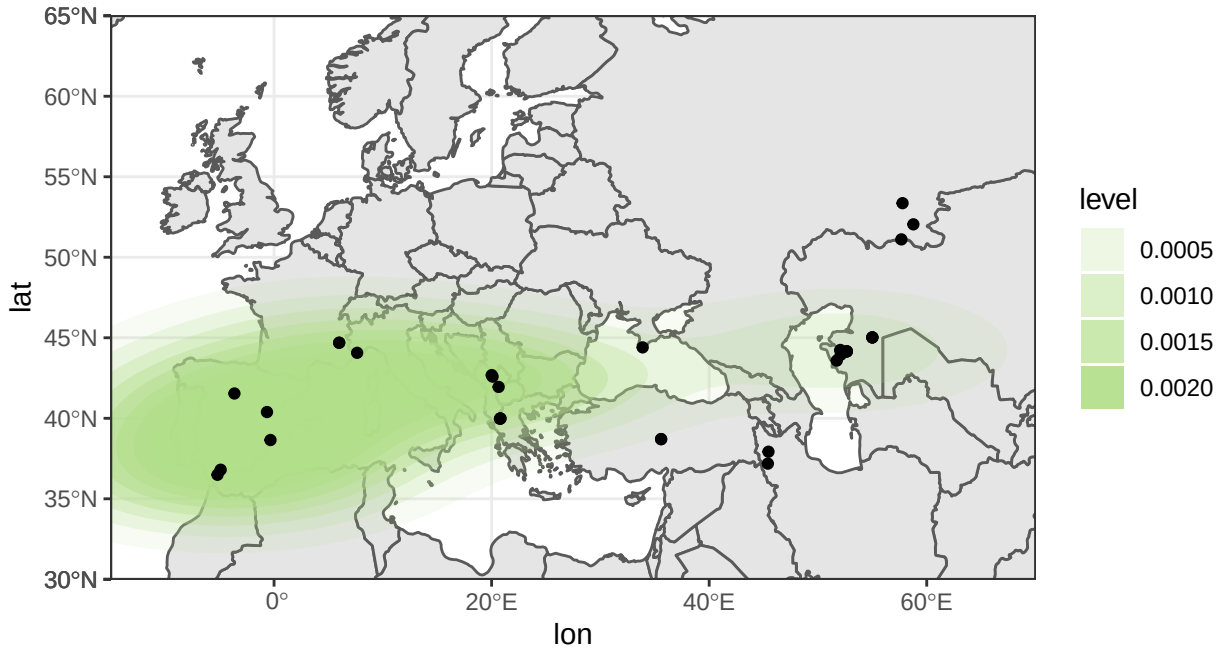

```
foo<-condensed_baps_adm[condensed_baps_adm$X7>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[7]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



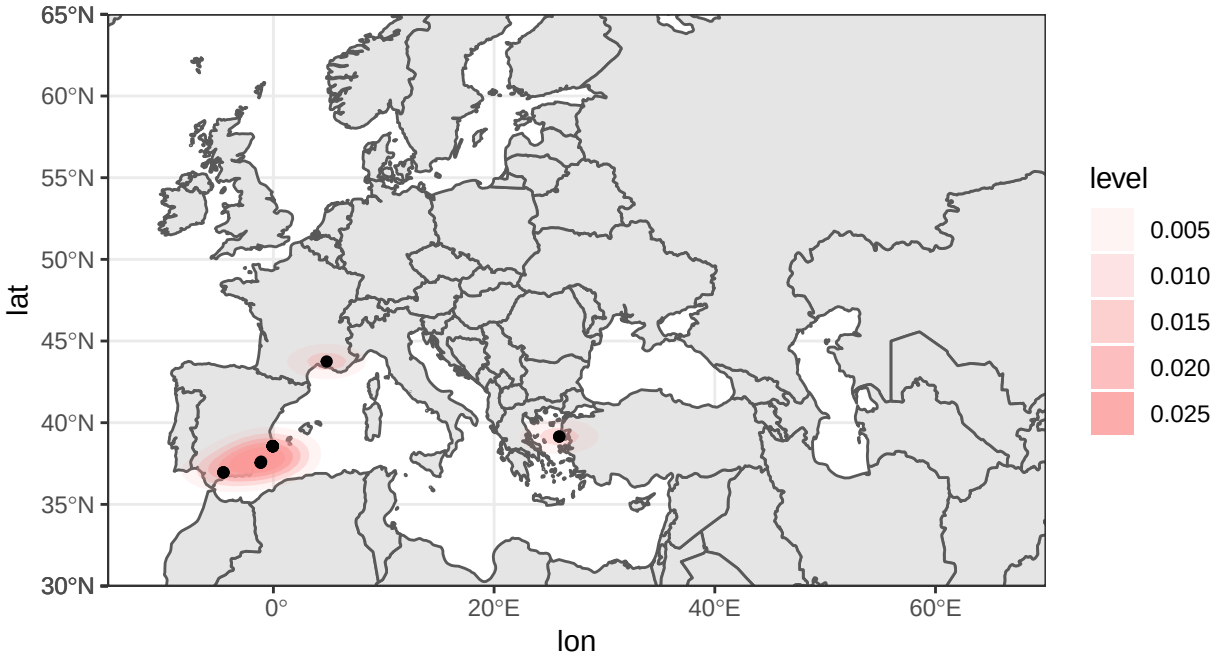

```
foo<-condensed_baps_adm[condensed_baps_adm$X8>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[8]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```




```
foo<-condensed_baps_adm[condensed_baps_adm$X9>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[9]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



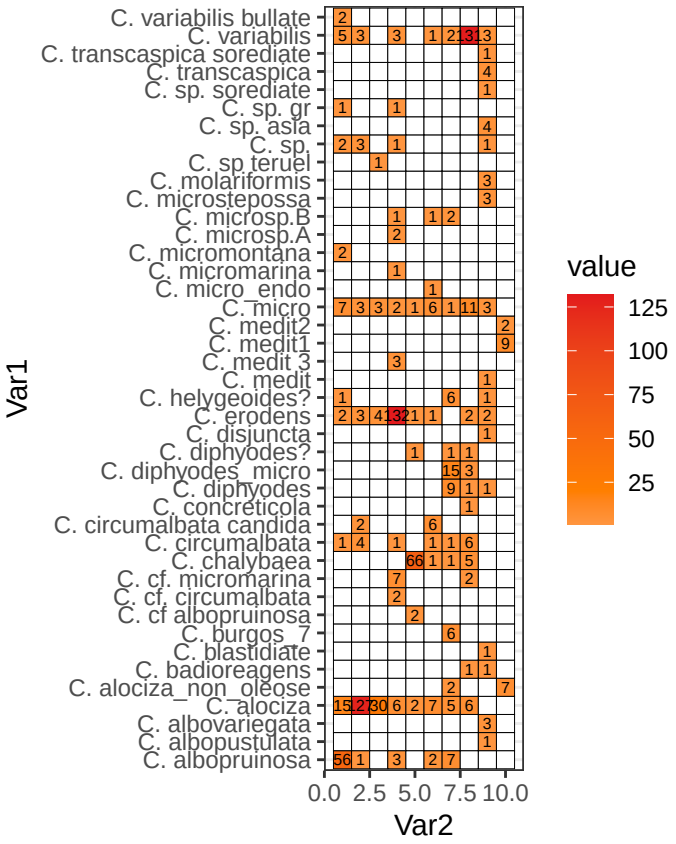

```
foo<-condensed_baps_adm[condensed_baps_adm$X10>0.5,]
ggplot(data = world) +
  geom_sf() +
  theme_bw() +
  coord_sf(xlim = c(-15, 70),
           ylim = c(30, 65),
           expand = FALSE) +
  stat_density_2d(geom = "polygon",
                  data=foo[!duplicated(foo$Locality),],
                  mapping = aes(x=lon,y=lat,alpha = ..level..),
                  fill = colorinos.bipolar[10]) +
  expand_limits(x= c(-30, 70),
                y = c(30, 65)) +
  geom_point(data=foo,
             aes(x=lon,y=lat))
```



2.1.19 Association between gene-pools and morphospecies

```
# Heatmap 1 Species vs BAPS

foo_tabla<-melt(table(condensed_baps_adm$Species,condensed_baps_adm$name))
foo_tabla$value[foo_tabla$value==0]<-NA


  ggplot(foo_tabla,aes(x=Var2,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = v
    scale_fill_gradient2(low = "#FFFFFF",
  mid = colorinos.bipolar[7],
  high = colorinos.bipolar[6],
  midpoint = 20,
  space = "Lab",
  na.value = "#FFFFFF",
  guide = "colourbar",
  aesthetics = "fill"
) +
  coord_fixed() + theme_bw()
```

```
## Warning: Removed 321 rows containing missing values (geom_text).
```




```
print(assoc(table(condensed_baps_adm$Species,condensed_baps_adm$name),shade=TRUE))
```


```
##                          B   1   2   3   4   5   6   7   8   9  10
## A
## C. albopruinosa             56   1   0   3   0   2   7   0   0   0
## C. albopustulata             0   0   0   0   0   0   0   0   1   0
## C. albovariegata             0   0   0   0   0   0   0   0   3   0
## C. alociza                  15 127  30   6   2   7   5   6   0   0
## C. alociza_non_oleose        0   0   0   0   0   0   2   0   0   7
## C. badioreagens              0   0   0   0   0   0   0   1   1   0
## C. blastidiate               0   0   0   0   0   0   0   0   1   0
## C. burgos_7                  0   0   0   0   0   0   6   0   0   0
## C. cf albopruinosa           0   0   0   0   2   0   0   0   0   0
## C. cf. circumalbata          0   0   0   2   0   0   0   0   0   0
## C. cf. micromarina           0   0   0   7   0   0   0   2   0   0
## C. chalybaea                 0   0   0   0  66   1   1   5   0   0
```

```
## C. circumalbata              1   4   0   1   0   1   1   6   0   0
## C. circumalbata candida      0   2   0   0   0   6   0   0   0   0
## C. concreticola              0   0   0   0   0   0   0   1   0   0
## C. diphyodes                 0   0   0   0   0   0   9   1   1   0
## C. diphyodes_micro           0   0   0   0   0   0  15   3   0   0
## C. diphyodes?                0   0   0   0   1   0   1   1   0   0
## C. disjuncta                 0   0   0   0   0   0   0   0   1   0
## C. erodens                   2   3   4 132   1   1   0   2   2   0
## C. helygeoides?              1   0   0   0   0   0   6   0   1   0
## C. medit                     0   0   0   0   0   0   0   0   1   0
## C. medit 3                   0   0   0   3   0   0   0   0   0   0
## C. medit1                    0   0   0   0   0   0   0   0   0   9
## C. medit2                    0   0   0   0   0   0   0   0   0   2
## C. micro                     7   3   3   2   1   6   1  11   3   0
## C. micro_endo                0   0   0   0   0   1   0   0   0   0
## C. micromarina               0   0   0   1   0   0   0   0   0   0
## C. micromontana              2   0   0   0   0   0   0   0   0   0
## C. microsp.A                 0   0   0   2   0   0   0   0   0   0
## C. microsp.B                 0   0   0   1   0   1   2   0   0   0
## C. microstepossa             0   0   0   0   0   0   0   0   3   0
## C. molariformis              0   0   0   0   0   0   0   0   3   0
## C. sp teruel                 0   0   1   0   0   0   0   0   0   0
## C. sp.                       2   3   0   1   0   0   0   0   1   0
## C. sp. asia                  0   0   0   0   0   0   0   0   4   0
## C. sp. gr                    1   0   0   1   0   0   0   0   0   0
## C. sp. sorediate             0   0   0   0   0   0   0   0   1   0
## C. transcaspica              0   0   0   0   0   0   0   0   4   0
## C. transcaspica sorediate    0   0   0   0   0   0   0   0   1   0
## C. variabilis                5   3   0   3   0   1   2 131   3   0
## C. variabilis bullate        2   0   0   0   0   0   0   0   0   0
```

```
assocstats(table(condensed_baps_adm$Species,condensed_baps_adm$name))
```

```
##                     X^2  df P(> X^2)
## Likelihood Ratio 2168.8 369        0
## Pearson          4234.2 369        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.915
## Cramer's V        : 0.756
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm$Species,condensed_baps_adm$name))
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm$Species, condensed_baps_adm$name)
## f(x) = 23.41, p-value < 2.2e-16
```

2.1.20 How much of the species splitting is explained by the assignment to gene-pools?

```
library(mda)
```

```
## Loading required package: class

## Loaded mda 0.5-3

##
## Attaching package: 'mda'

## The following object is masked from 'package:fpc':
##
##     confusion
```

```r
foo_fdas<-condensed_baps_adm[,c("Species","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10")]
disc_clusters<-mda::fda(Species~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10, foo_fdas)
print(disc_clusters)
```

```
## Call:
## mda::fda(formula = Species ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 +
##     X8 + X9 + X10, data = foo_fdas)
##
## Dimension: 26
##
## Percent Between-Group Variance Explained:
##     v1     v2     v3     v4     v5     v6     v7     v8     v9    v10    v11
##  36.82  61.53  73.39  82.90  90.00  94.98  98.32  99.96 100.00 100.00 100.00
##    v12    v13    v14    v15    v16    v17    v18    v19    v20    v21    v22
## 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00
##    v23    v24    v25    v26
## 100.00 100.00 100.00 100.00
##
## Degrees of Freedom (per dimension): 10
##
## Training Misclassification Error: 0.33374 ( N = 824 )
```

```r
kable(coef(disc_clusters,foo_fdas))
```

Intercept	21.89455	[illegible]	0e+00	[illegible]	2e-07	1e-07	1e-07	0e+00	[illegible]	-2e-07	-1e-07	0e+00	[illegible]	1e-07	-1e-07	0e+00	[illegible]	0e+00
X1	-22.17266	[illegible]	[illegible]	-1e-07	-2e-07	-3e-07	-1e-07	-1e-07	-2e-07	2e-07	1e-07	0e+00	-3e-07	0e+00	[illegible]	1e-07	-1e-07	0e+00
X2	-22.42849	[illegible]	[illegible]	-1e-07	-1e-07	-1e-07	-1e-07	0e+00	-2e-07	1e-07	1e-07	-1e-07	-3e-07	0e+00	[illegible]	0e+00	0e+00	-2e-07
X3	-22.51756	[illegible]	1e-07	1e-07	-5e-07	0e+00	-2e-07	-2e-07	0e+00	[illegible]	1e-07	2e-07	-2e-07	-2e-07	0e+00	0e+00	-2e-07	0e+00
X4	-22.74817	[illegible]	[illegible]	-1e-07	-2e-07	-1e-07	-1e-07	0e+00	-2e-07	1e-07	0e+00	0e+00	-4e-07	-1e-07	2e-07	0e+00	0e+00	[illegible]
X5	-23.12 [illegible]	[illegible]	[illegible]	-2e-07	-3e-07	-1e-07	-1e-07	0e+00	-2e-07	1e-07	1e-07	0e+00	-3e-07	-1e-07	2e-07	1e-07	-1e-07	-1e-07
X6	-22.42707	[illegible]	0e+00	[illegible]	-2e-07	-3e-07	-1e-07	0e+00	[illegible]	1e-07	0e+00	-3e-07	1e-07	-1e-07	-1e-07	1e-07	-1e-07	-3e-07

X7	-20.60123	1.265[illegible]	[illegible]	[illegible]	-0.4655[illegible]	-[illegible]	-[illegible]	-[illegible]	0.0260[illegible]	0e+00	-2e-07	-2e-07	-5e-07	-1e-07	-2e-07	-4e-07	4e-07	2e-07	1e-07	-3e-07	0e+00	1e-07	1e-07	-3e-07	0e+00	-3e-07
X8	-22.41035	0.801[illegible]	[illegible]	-0.7161290	2.8237446	-0.349[illegible]	-[illegible]	0.146[illegible]	[illegible]	0e+00	-1e-07	-2e-07	0e+00	-1e-07	0e+00	-1e-07	1e-07	2e-07	-1e-07	-4e-07	-2e-07	1e-07	0e+00	-1e-07	1e-07	-2e-07
X9	-22.51708	1.888[illegible]	[illegible]	-7.461[illegible]	-[illegible]	0.1440457	-0.2172969	0.127[illegible]	[illegible]	-1e-07	0e+00	-1e-07	0e+00	0e+00	0e+00	-1e-07	1e-07	0e+00	-1e-07	0e+00	0e+00	-1e-07	1e-07	0e+00	-2e-07	-1e-07
X10	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

2.2 I.II Phylogenetic signal in BAPS clusters and species IDs

2.2.1 Phylogenetic signal in BAPS clusters

```
library(picante)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## The following object is masked from 'package:sna':
##
##     gapply
```

```
library(ape)
library(adephylo)
```

```
## Loading required package: ade4
```

```
library(ade4)
library(phylobase)
```

```
##
## Attaching package: 'phylobase'

## The following object is masked from 'package:ape':
##
##     edges
```

```
library(geiger)
library(phytools)
```

```
## Loading required package: maps

##
## Attaching package: 'phytools'

## The following object is masked from 'package:phylobase':
##
##     readNexus

## The following object is masked from 'package:vegan':
```

```
##
##     scores
```

```
library(future.apply)
```

```
## Loading required package: future
```

```
foo_adm<-baps.adm
rownames(foo_adm)<-paste("X",gsub("\\.1","B",rownames(foo_adm)),sep="")
foo_adm<-rbind(foo_adm,XXanpa=0)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = 0): invalid factor level, NA
## generated
```

```
plan(multicore,workers=10)
```

```
## Warning in supportsMulticoreAndRStudio(...): [ONE-TIME WARNING] Forked
## processing ('multicore') is not supported when running R from RStudio
## because it is considered unstable. For more details, how to control forked
## processing or not, and how to silence this warning in future R sessions,
## see ?parallelly::supportsMulticore
```

```
#phylosig_final<-future_lapply(
#  c("X1","X2","X3","X4","X5","X6","X7","X8","X9","X10"),
#  FUN=function(GRUPO)
#  {
#  phylo_signal<-lapply(tree.dataset,
#       FUN=function(x){
#  groups_mix<-as.numeric(foo_adm[x$tip.label,GRUPO])
#  names(groups_mix)<-x$tip.label
#  foo_out<-phytools::phylosig(
#  root(
#    compute.brlen(x),
#    outgroup = "XXanpa",
#    node=NULL,
#    resolve.root = TRUE),
#  groups_mix, method="lambda", test=TRUE, nsim=999)
#  return(foo_out)
#})
#  return(phylo_signal)
#})
load("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v15.Rdata")
print(phylosig_final)
```

```
## [[1]]
## [[1]]$its
##
## Phylogenetic signal lambda : 0.985151
## logL(lambda) : 972.292
## LR(lambda=0) : 2371.59
## P-value (based on LR test) : 0
##
##
## [[1]]$bt
##
## Phylogenetic signal lambda : 0.976185
## logL(lambda) : 1306.81
```

```
## LR(lambda=0) : 3028.15
## P-value (based on LR test) : 0
##
##
## [[1]]$efa
##
## Phylogenetic signal lambda : 0.991727
## logL(lambda) : 1046.69
## LR(lambda=0) : 2524.15
## P-value (based on LR test) : 0
##
##
## [[1]]$mcm7
##
## Phylogenetic signal lambda : 0.949642
## logL(lambda) : 978.181
## LR(lambda=0) : 2384.65
## P-value (based on LR test) : 0
##
##
## [[1]]$rpb1
##
## Phylogenetic signal lambda : 0.980254
## logL(lambda) : 1242.56
## LR(lambda=0) : 2918.21
## P-value (based on LR test) : 0
##
##
##
## [[2]]
## [[2]]$its
##
## Phylogenetic signal lambda : 0.946358
## logL(lambda) : 306.393
## LR(lambda=0) : 1338.18
## P-value (based on LR test) : 5.71696e-293
##
##
## [[2]]$bt
##
## Phylogenetic signal lambda : 0.987747
## logL(lambda) : 632.235
## LR(lambda=0) : 1990.26
## P-value (based on LR test) : 0
##
##
## [[2]]$efa
##
## Phylogenetic signal lambda : 0.976179
## logL(lambda) : 479.532
## LR(lambda=0) : 1685.05
## P-value (based on LR test) : 0
##
##
```

```
## [[2]]$mcm7
##
## Phylogenetic signal lambda : 0.955581
## logL(lambda) : 340.943
## LR(lambda=0) : 1407.18
## P-value (based on LR test) : 5.78936e-308
##
##
## [[2]]$rpb1
##
## Phylogenetic signal lambda : 0.982278
## logL(lambda) : 1218.48
## LR(lambda=0) : 3158.82
## P-value (based on LR test) : 0
##
##
##
## [[3]]
## [[3]]$its
##
## Phylogenetic signal lambda : 0.980867
## logL(lambda) : 592.101
## LR(lambda=0) : 876.511
## P-value (based on LR test) : 1.25351e-192
##
##
## [[3]]$bt
##
## Phylogenetic signal lambda : 0.981899
## logL(lambda) : 471.122
## LR(lambda=0) : 614.289
## P-value (based on LR test) : 1.30562e-135
##
##
## [[3]]$efa
##
## Phylogenetic signal lambda : 0.995713
## logL(lambda) : 512.649
## LR(lambda=0) : 685.575
## P-value (based on LR test) : 4.09705e-151
##
##
## [[3]]$mcm7
##
## Phylogenetic signal lambda : 0.977399
## logL(lambda) : 424.968
## LR(lambda=0) : 556.075
## P-value (based on LR test) : 6.00321e-123
##
##
## [[3]]$rpb1
##
## Phylogenetic signal lambda : 0.976345
## logL(lambda) : 1145.33
```

```
## LR(lambda=0) : 1457.88
## P-value (based on LR test) : 5.55685513190567e-319
##
##
##
## [[4]]
## [[4]]$its
##
## Phylogenetic signal lambda : 0.998203
## logL(lambda) : 612.232
## LR(lambda=0) : 2032.77
## P-value (based on LR test) : 0
##
##
## [[4]]$bt
##
## Phylogenetic signal lambda : 0.985362
## logL(lambda) : 523.791
## LR(lambda=0) : 1843.61
## P-value (based on LR test) : 0
##
##
## [[4]]$efa
##
## Phylogenetic signal lambda : 0.99734
## logL(lambda) : 480.262
## LR(lambda=0) : 1761.6
## P-value (based on LR test) : 0
##
##
## [[4]]$mcm7
##
## Phylogenetic signal lambda : 0.982317
## logL(lambda) : 406.019
## LR(lambda=0) : 1606.21
## P-value (based on LR test) : 0
##
##
## [[4]]$rpb1
##
## Phylogenetic signal lambda : 0.987243
## logL(lambda) : 1315.6
## LR(lambda=0) : 3411.64
## P-value (based on LR test) : 0
##
##
##
## [[5]]
## [[5]]$its
##
## Phylogenetic signal lambda : 0.968204
## logL(lambda) : 863.889
## LR(lambda=0) : 1896.33
## P-value (based on LR test) : 0
```

```
##
##
## [[5]]$bt
##
## Phylogenetic signal lambda : 0.960392
## logL(lambda) : 1078.17
## LR(lambda=0) : 2329.35
## P-value (based on LR test) : 0
##
##
## [[5]]$efa
##
## Phylogenetic signal lambda : 0.983874
## logL(lambda) : 672.117
## LR(lambda=0) : 1519.24
## P-value (based on LR test) : 0
##
##
## [[5]]$mcm7
##
## Phylogenetic signal lambda : 0.967411
## logL(lambda) : 1160.41
## LR(lambda=0) : 2497.33
## P-value (based on LR test) : 0
##
##
## [[5]]$rpb1
##
## Phylogenetic signal lambda : 0.98799
## logL(lambda) : 1343.86
## LR(lambda=0) : 2906.63
## P-value (based on LR test) : 0
##
##
##
## [[6]]
## [[6]]$its
##
## Phylogenetic signal lambda : 0.929256
## logL(lambda) : 855.021
## LR(lambda=0) : 945.439
## P-value (based on LR test) : 1.30087e-207
##
##
## [[6]]$bt
##
## Phylogenetic signal lambda : 0.956926
## logL(lambda) : 1311.12
## LR(lambda=0) : 1873.37
## P-value (based on LR test) : 0
##
##
## [[6]]$efa
##
```

```
## Phylogenetic signal lambda : 0.996587
## logL(lambda) : 1207.97
## LR(lambda=0) : 1642.34
## P-value (based on LR test) : 0
##
##
## [[6]]$mcm7
##
## Phylogenetic signal lambda : 0.964438
## logL(lambda) : 976.087
## LR(lambda=0) : 1207.38
## P-value (based on LR test) : 1.51889e-264
##
##
## [[6]]$rpb1
##
## Phylogenetic signal lambda : 0.999718
## logL(lambda) : 1561.92
## LR(lambda=0) : 2385.7
## P-value (based on LR test) : 0
##
##
##
## [[7]]
## [[7]]$its
##
## Phylogenetic signal lambda : 0.978565
## logL(lambda) : 1334.59
## LR(lambda=0) : 2664.09
## P-value (based on LR test) : 0
##
##
## [[7]]$bt
##
## Phylogenetic signal lambda : 0.983128
## logL(lambda) : 679.032
## LR(lambda=0) : 1378.91
## P-value (based on LR test) : 8.0287e-302
##
##
## [[7]]$efa
##
## Phylogenetic signal lambda : 0.974724
## logL(lambda) : 688.562
## LR(lambda=0) : 1380.25
## P-value (based on LR test) : 4.11762e-302
##
##
## [[7]]$mcm7
##
## Phylogenetic signal lambda : 0.967518
## logL(lambda) : 970.301
## LR(lambda=0) : 1933.27
## P-value (based on LR test) : 0
```

```
##
##
## [[7]]$rpb1
##
## Phylogenetic signal lambda : 0.973388
## logL(lambda) : 1549.33
## LR(lambda=0) : 3097.74
## P-value (based on LR test) : 0
##
##
##
## [[8]]
## [[8]]$its
##
## Phylogenetic signal lambda : 0.996282
## logL(lambda) : 668.806
## LR(lambda=0) : 2153.55
## P-value (based on LR test) : 0
##
##
## [[8]]$bt
##
## Phylogenetic signal lambda : 0.990548
## logL(lambda) : 612.644
## LR(lambda=0) : 2035.68
## P-value (based on LR test) : 0
##
##
## [[8]]$efa
##
## Phylogenetic signal lambda : 0.996121
## logL(lambda) : 891.413
## LR(lambda=0) : 2595.75
## P-value (based on LR test) : 0
##
##
## [[8]]$mcm7
##
## Phylogenetic signal lambda : 0.975594
## logL(lambda) : 670.7
## LR(lambda=0) : 2146.01
## P-value (based on LR test) : 0
##
##
## [[8]]$rpb1
##
## Phylogenetic signal lambda : 0.967816
## logL(lambda) : 1064.16
## LR(lambda=0) : 2927.27
## P-value (based on LR test) : 0
##
##
##
## [[9]]
```

```
## [[9]]$its
##
## Phylogenetic signal lambda : 1.00089
## logL(lambda) : 1176.47
## LR(lambda=0) : 2018.47
## P-value (based on LR test) : 0
##
##
## [[9]]$bt
##
## Phylogenetic signal lambda : 1.00064
## logL(lambda) : 1227.75
## LR(lambda=0) : 2153.56
## P-value (based on LR test) : 0
##
##
## [[9]]$efa
##
## Phylogenetic signal lambda : 0.978528
## logL(lambda) : 1027.2
## LR(lambda=0) : 1731.78
## P-value (based on LR test) : 0
##
##
## [[9]]$mcm7
##
## Phylogenetic signal lambda : 0.942054
## logL(lambda) : 921.095
## LR(lambda=0) : 1499.77
## P-value (based on LR test) : 0
##
##
## [[9]]$rpb1
##
## Phylogenetic signal lambda : 0.998093
## logL(lambda) : 1411.81
## LR(lambda=0) : 2583.45
## P-value (based on LR test) : 0
##
##
##
## [[10]]
## [[10]]$its
##
## Phylogenetic signal lambda : 1.00103
## logL(lambda) : 2128.67
## LR(lambda=0) : 3254.84
## P-value (based on LR test) : 0
##
##
## [[10]]$bt
##
## Phylogenetic signal lambda : 0.986982
## logL(lambda) : 1342.61
```

```
## LR(lambda=0) : 1740.57
## P-value (based on LR test) : 0
##
##
## [[10]]$efa
##
## Phylogenetic signal lambda : 0.989386
## logL(lambda) : 2132.8
## LR(lambda=0) : 3281.78
## P-value (based on LR test) : 0
##
##
## [[10]]$mcm7
##
## Phylogenetic signal lambda : 1.00105
## logL(lambda) : 2050.02
## LR(lambda=0) : 3102.6
## P-value (based on LR test) : 0
##
##
## [[10]]$rpb1
##
## Phylogenetic signal lambda : 0.99608
## logL(lambda) : 1637.05
## LR(lambda=0) : 2428.47
## P-value (based on LR test) : 0
```

```
#save.image("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v14.Rdata")
```

2.2.2 Migrate-n migration network between BAPS gene clusters

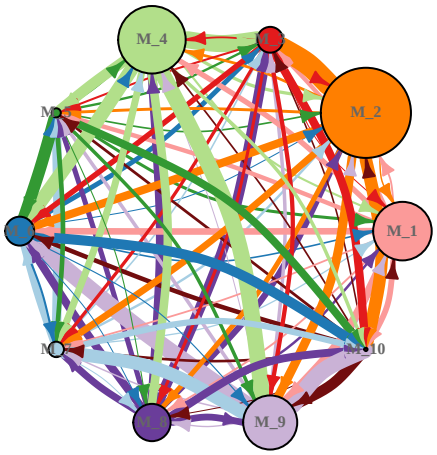
```
library(igraph)
```

```
##
## Attaching package: 'igraph'
## The following objects are masked from 'package:future':
##
##     %->%, %<-%
## The following object is masked from 'package:phylobase':
##
##     edges
## The following object is masked from 'package:class':
##
##     knn
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
## The following object is masked from 'package:bipartite':
##
##     strength
## The following objects are masked from 'package:sna':
```

```
##
##     betweenness, bonpow, closeness, components, degree, dyad.census,
##     evcent, hierarchy, is.connected, neighborhood, triad.census

## The following objects are masked from 'package:network':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##     get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##     is.directed, list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute

## The following object is masked from 'package:vegan':
##
##     diversity

## The following object is masked from 'package:permute':
##
##     permute

## The following object is masked from 'package:phangorn':
##
##     diversity

## The following objects are masked from 'package:ape':
##
##     degree, edges, mst, ring

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
connectors<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network.txt",header =
nodes<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network2.txt",header = FALS
nodes<-data.frame(nodes=as.character(nodes[,1]),theta=as.numeric(nodes[,2]))
connectors<-data.frame(from = as.character(connectors[,1]),to = as.character(connectors[,2]),weight = as
foo_col<-colorinos.bipolar[c(10,7,6,9,5,8,2,4,1,11)]
names(foo_col)<-nodes$nodes
rownames(nodes)<-nodes$nodes
migrate_M<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)
```

```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 500*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/20,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```

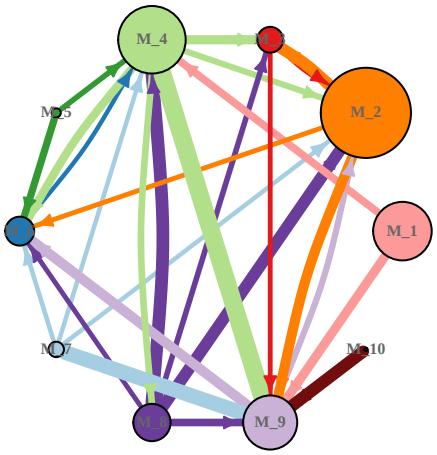



```
connectors$weight<-connectors$weight*nodes[connectors$to,"theta"]
connectors_backup<-connectors
connectors<-connectors[connectors$weight>=2,]

#foo_col[connectors$to]
migrate_2Ne<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)

#set_vertex_attr(migrate, "theta", index = V(migrate), value = 100*as.numeric(nodes[,2]))
```

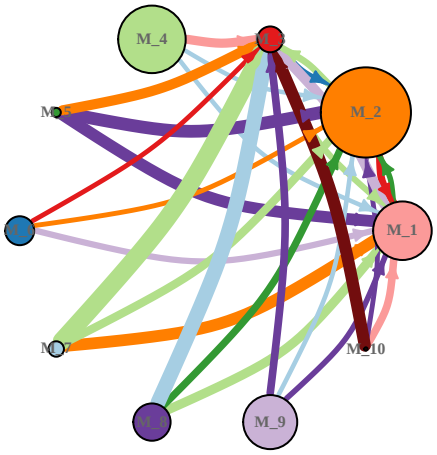
```
coords<-layout_(migrate_2Ne,
                in_circle())
curves<-mat.or.vec(ecount(migrate_2Ne),1)

paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep="")
```

```
##  [1]  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [13]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
## [25] FALSE FALSE FALSE
```

```
curves[gsub("M_","",connectors[,1])!=gsub("M_","",connectors[,2])]<--0.1
curves[!(paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep=""))]<-0
plot(migrate_2Ne,
     vertex.size = 500*vertex_attr(migrate_2Ne,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_2Ne)],
     edge.curved=curves,
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```




```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 500*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```




```
foo.table<-cbind(nodes,from=sapply(nodes[,1],FUN=function(x){sum(connectors_backup$weight[connectors_ba
kable(cbind(foo.table[,-1],balance=foo.table$to-foo.table$from))
```

	theta	from	to	balance
M_1	0.06120	14.903528	9.821988	-5.081540
M_2	0.09422	19.967251	23.771706	3.804455
M_3	0.02697	11.789365	18.226326	6.436961
M_4	0.07056	27.713918	22.219344	-5.494574
M_5	0.00956	13.920574	4.584307	-9.336267
M_6	0.03020	7.791225	25.014660	17.223435

	theta	from	to	balance
M_7	0.01612	16.856821	5.958758	-10.898062
M_8	0.03898	24.803882	9.014125	-15.789757
M_9	0.05651	12.572468	41.022869	28.450401
M_10	0.00346	12.025443	2.710391	-9.315052

```
#kable(
#  rbind(
#  c("M_2",
#      sum(connectors_backup$weight[connectors_backup$from=="M_2"&connectors_backup$to!="M_3"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_2"&connectors_backup$from!="M_3"])
#      ),
##       c("M_3",
#      sum(connectors_backup$weight[connectors_backup$from=="M_3"&connectors_backup$to!="M_2"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_3"&connectors_backup$from!="M_2"])
#      )
#  )
#)
```

2.2.3 Migrate-n migration network between BAPS gene clusters using mat genes

```
library(igraph)
connectors<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network_hmg2",header =
nodes<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network_hmg",header = FALSE
nodes<-data.frame(nodes=as.character(nodes[,1]),theta=as.numeric(nodes[,2]))
connectors<-data.frame(from = as.character(connectors[,1]),to = as.character(connectors[,2]),weight = as
foo_col<-colorinos.bipolar[c(10,7,6,9,5,8,2,4,1,11)]
names(foo_col)<-nodes$nodes
rownames(nodes)<-nodes$nodes
migrate_M<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)
```

2.2.3.1 HMG

```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 100*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/10,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```

2.2.3.2 Complete network edges proportional to M

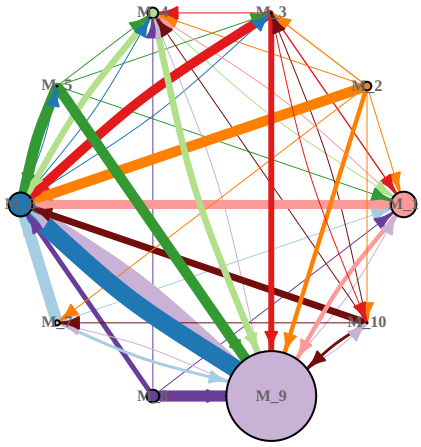
```
connectors$weight<-connectors$weight*nodes[connectors$to,"theta"]
connectors_backup<-connectors
connectors<-connectors[connectors$weight>=2,]

#foo_col[connectors$to]
migrate_2Ne<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)

#set_vertex_attr(migrate, "theta", index = V(migrate), value = 100*as.numeric(nodes[,2]))
```

```
coords<-layout_(migrate_2Ne,
                in_circle())
curves<-mat.or.vec(ecount(migrate_2Ne),1)

paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep="")
```

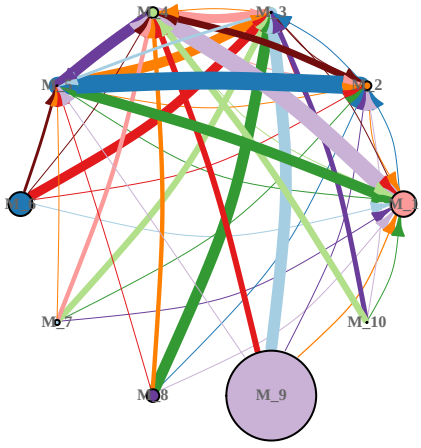
2.2.3.3 Reduced netwrok edges are 2Neu only connections above 2 migrants per generation are shown.

```
##  [1] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
## [13] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
## [25]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
## [37] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
```

```
curves[gsub("M_","",connectors[,1])!=gsub("M_","",connectors[,2])]<-0.1
curves[!(paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep=""))]<-0
plot(migrate_2Ne,
     vertex.size = 100*vertex_attr(migrate_2Ne,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/10,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_2Ne)],
     edge.curved=curves,
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```




```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 100*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/10,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```




```
foo.table<-cbind(nodes,from=sapply(nodes[,1],FUN=function(x){sum(connectors_backup$weight[connectors_ba
kable(cbind(foo.table[,-1],balance=foo.table$to-foo.table$from))
```

	theta	from	to	balance
M_1	0.13309	78.26837	29.448824	-48.81954
M_2	0.04521	96.42590	9.127899	-87.29800
M_3	0.00504	95.30452	19.258798	-76.04572
M_4	0.05593	74.11253	28.008066	-46.10446
M_5	0.00474	120.81621	11.834168	-108.98205
M_6	0.12475	104.67702	433.836838	329.15982

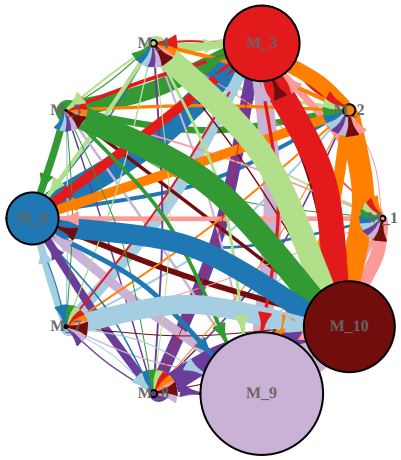
	theta	from	to	balance
M_7	0.02732	82.93010	15.611468	-67.31864
M_8	0.06655	91.77378	9.941904	-81.83187
M_9	0.46894	104.88441	334.077545	229.19313
M_10	0.00505	60.43188	18.479212	-41.95267

```
#kable(
#  rbind(
#  c("M_2",
#      sum(connectors_backup$weight[connectors_backup$from=="M_2"&connectors_backup$to!="M_3"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_2"&connectors_backup$from!="M_3"])
#      ),
##       c("M_3",
#      sum(connectors_backup$weight[connectors_backup$from=="M_3"&connectors_backup$to!="M_2"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_3"&connectors_backup$from!="M_2"])
#      )
#  )
#)
```

```
library(igraph)
connectors<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network_a2",header = 
nodes<-read.delim2("/Users/Fernando/Desktop/01_Sanger_paper/19_migrate-n/for_network_a",header = FALSE)
nodes<-data.frame(nodes=as.character(nodes[,1]),theta=as.numeric(nodes[,2]))
connectors<-data.frame(from = as.character(connectors[,1]),to = as.character(connectors[,2]),weight = as
foo_col<-colorinos.bipolar[c(10,7,6,9,5,8,2,4,1,11)]
names(foo_col)<-nodes$nodes
rownames(nodes)<-nodes$nodes
migrate_M<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)
```

2.2.3.4 Alpha

```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 50*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/50,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```



2.2.3.5 Complete network edges proportional to M

```
connectors$weight<-connectors$weight*nodes[connectors$to,"theta"]
connectors_backup<-connectors
connectors<-connectors[connectors$weight>=2,]

#foo_col[connectors$to]
migrate_2Ne<-graph_from_data_frame(connectors, directed = TRUE, vertices = nodes)

#set_vertex_attr(migrate, "theta", index = V(migrate), value = 100*as.numeric(nodes[,2]))
```

```
coords<-layout_(migrate_2Ne,
                in_circle())
curves<-mat.or.vec(ecount(migrate_2Ne),1)

paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep="")
```

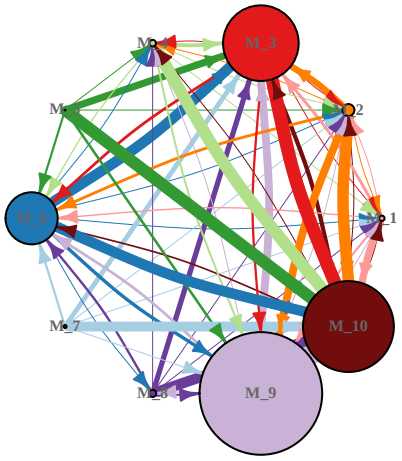
2.2.3.6 Reduced netwrok edges are 2Neu only connections above 2 migrants per generation are shown.

```
##  [1]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [13]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
## [25]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [49] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
## [61] FALSE  TRUE
```

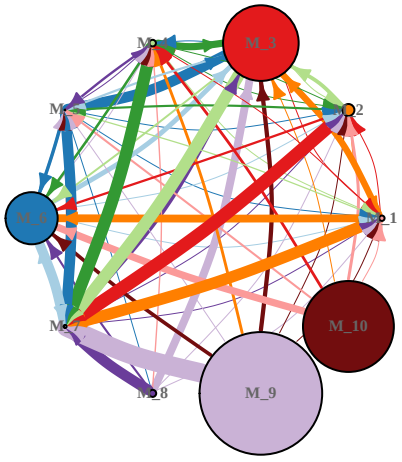
```
curves[gsub("M_","",connectors[,1])!=gsub("M_","",connectors[,2])]<--0.1
curves[!(paste(connectors[,1],connectors[,2],sep="")%in%paste(connectors[,2],connectors[,1],sep=""))]<-(
```

```
plot(migrate_2Ne,
     vertex.size = 50*vertex_attr(migrate_2Ne,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/100,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_2Ne)],
     edge.curved=curves,
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
```

```
      )
```




```
coords<-layout_(migrate_M,
                in_circle())
plot(migrate_M,
     vertex.size = 50*vertex_attr(migrate_M,"theta"),
     edge.arrow.size = 0.5,
     edge.width = connectors$weight/100,
     edge.color = foo_col[connectors$from],
     vertex.color = foo_col[V(migrate_M)],
     edge.curved=seq(-0.3, 0.3, length = ecount(migrate_M)),
     layout = coords,
     vertex.label.font=2,
     vertex.label.color="gray40",
     vertex.label.cex=.5
     )
```




```
foo.table<-cbind(nodes,from=sapply(nodes[,1],FUN=function(x){sum(connectors_backup$weight[connectors_ba
kable(cbind(foo.table[,-1],balance=foo.table$to-foo.table$from))
```

	theta	from	to	balance
M_1	0.06155	686.1425	25.343828	-660.7986
M_2	0.13727	1446.1728	101.846104	-1344.3267
M_3	0.86359	934.7295	2660.168092	1725.4386

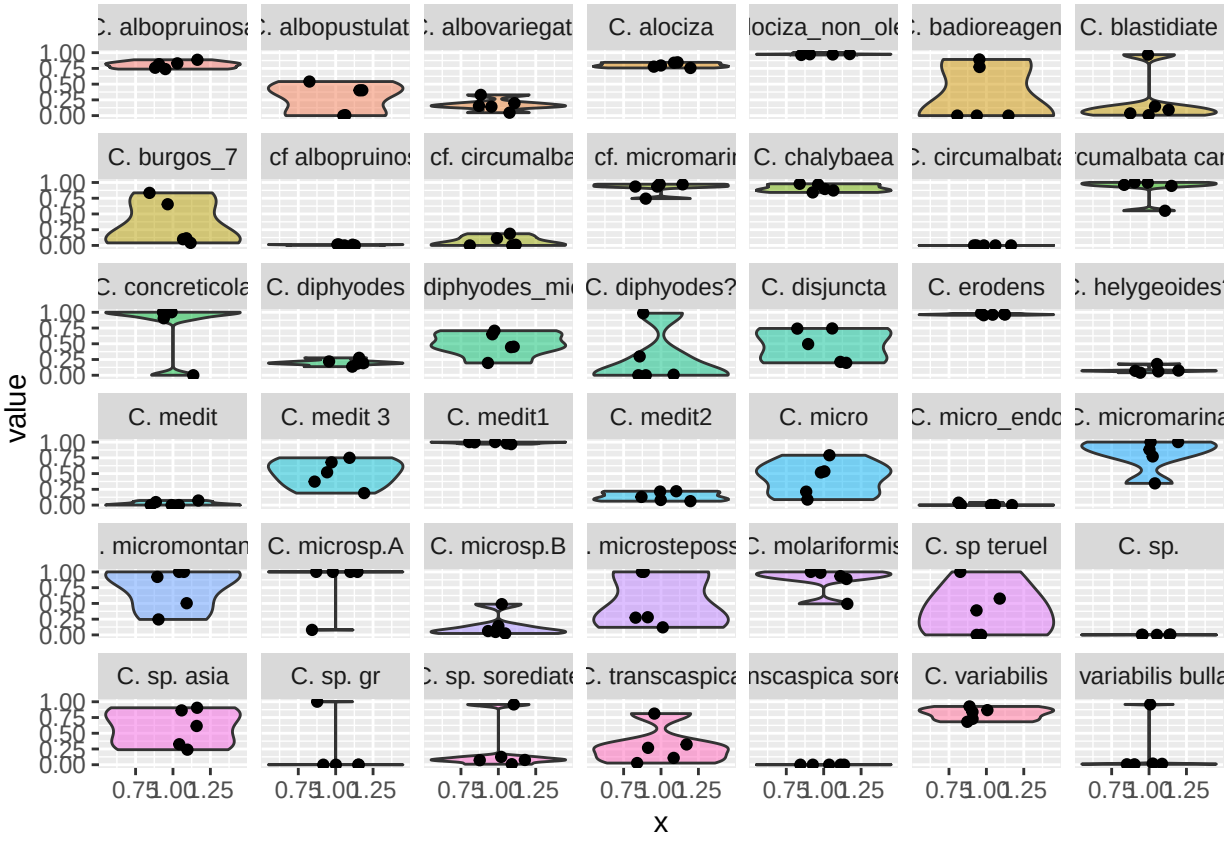
	theta	from	to	balance
M_4	0.07551	975.4596	28.823677	-946.6359
M_5	0.00995	1233.2686	5.462251	-1227.8064
M_6	0.59535	1254.0096	978.814935	-275.1947
M_7	0.03581	899.5700	10.061894	-889.5081
M_8	0.08169	1031.2539	16.370676	-1014.8832
M_9	1.40057	1528.0789	1339.056966	-189.0219
M_10	1.03815	494.2520	5316.989040	4822.7370

```
#kable(
#  rbind(
#  c("M_2",
#      sum(connectors_backup$weight[connectors_backup$from=="M_2"&connectors_backup$to!="M_3"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_2"&connectors_backup$from!="M_3"])
#      ),
##       c("M_3",
#      sum(connectors_backup$weight[connectors_backup$from=="M_3"&connectors_backup$to!="M_2"]),
#      sum(connectors_backup$weight[connectors_backup$to=="M_3"&connectors_backup$from!="M_2"])
#      )
#  )
#)
```

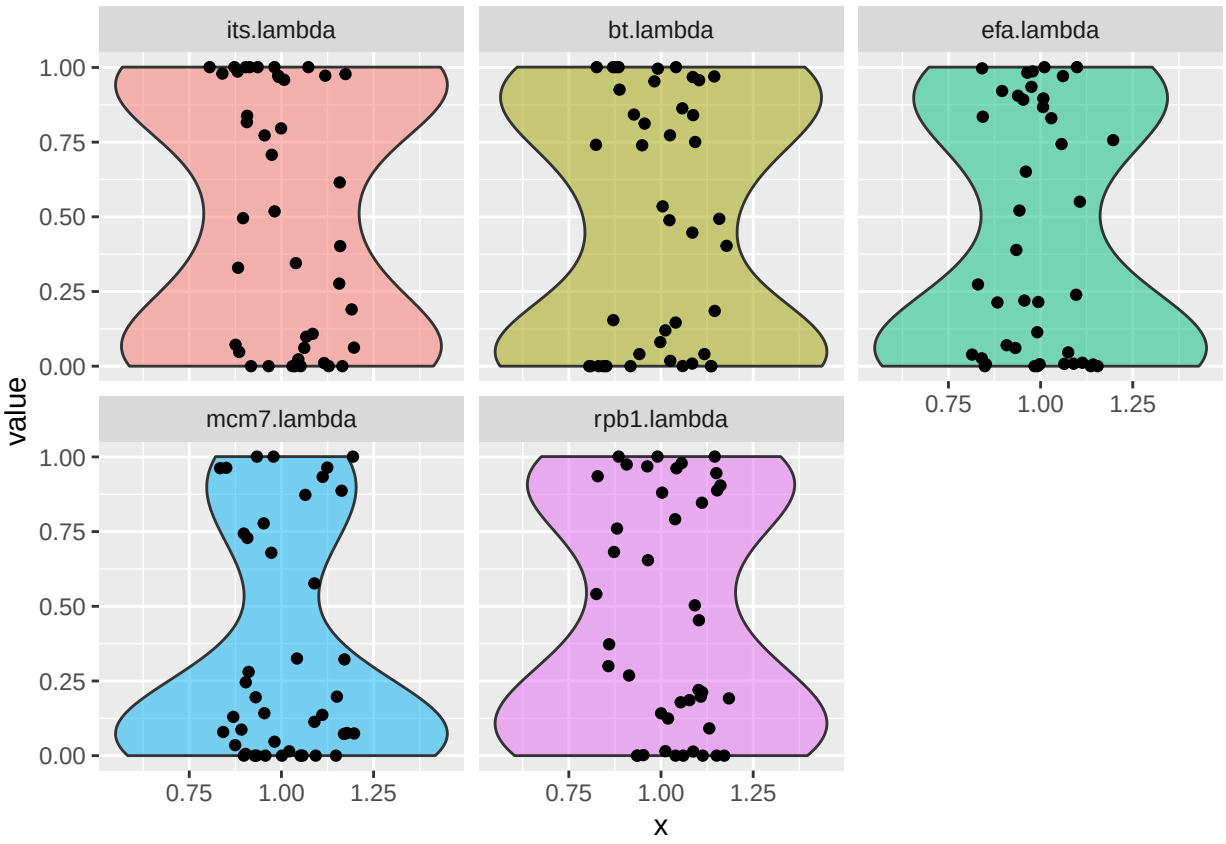
2.2.4 Phylogenetic signal in Species IDs

```
#load("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v14.Rdata")
#c(levels(factor(foo_adm$Species))[-1])
#plan(multicore,workers=12)
#phylosig_final2<-future_lapply(
#  levels(factor(foo_adm$Species))[-1],
#  FUN=function(GRUPO)
#  {
#  phylo_signal<-lapply(tree.dataset,
#       FUN=function(ARBOL)
#         {
#         foo_tree<-root(
#                   ARBOL,
#                   outgroup = "XXanpa",
#                   node=NULL,
#                   resolve.root = TRUE)
#        groups_mix<-as.numeric(factor(foo_adm[foo_tree$tip.label,"Species"])==GRUPO)
#        names(groups_mix)<-foo_tree$tip.label
   #     foo.out<-phytools::phylosig(
  #          compute.brlen(multi2di(foo_tree)),
 #           groups_mix, method="lambda", test=TRUE, nsim=100)
#        return(foo.out)
#})
#  return(phylo_signal)
#})
#names(phylosig_final2)<-levels(factor(foo_adm$Species))[-1]
#print(phylosig_final2)
#save.image("/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v15.Rdata")
```

```
fooeraseme<-melt(sapply(phylosig_final2,FUN=function(x){unlist(unlist(x)[c(1,6,11,16,21)])}))
colnames(fooeraseme)<-c("gene","sp","value")
ggplot(fooeraseme,aes(x=1,y=value,fill=sp))+geom_violin(alpha=0.5)+geom_point(position = position_jitte
```




```
ggplot(fooeraseme,aes(x=1,y=value,fill=gene))+geom_violin(alpha=0.5)+geom_point(position = position_jit
```



2.3 I.III Concordance between gene phylogenies

2.3.1 ML

The concordance between gene phylogenies was measured using Nye's similarity measurement (Nye 2000) implemented in r-package treedist. The trees were trimmed using function drop.tip to be made comparable and the number of shared The maximum concordance value was obtained as standarizing factor across loci, and trees with randomized tip assignments were used as random expectation.

```
tree_similarity_fer<-function(x,y,reps=100)
{
  require(TreeDist)
  require(ape)
  foo.x<-keep.tip(x,x$tip.label[x$tip.label%in%y$tip.label])
  foo.y<-keep.tip(y,y$tip.label[y$tip.label%in%x$tip.label])
  foo.max<-NyeSimilarity(foo.x,foo.x)
  foo.obs<-NyeSimilarity(foo.x,foo.y)
  foo.ran<-c(1:reps)
  foo.ran<-sapply(foo.ran,FUN=function(z){
  foo.y$tip.label<-sample(foo.y$tip.label)
   return(NyeSimilarity(foo.x,foo.y))
  })
  return(c(foo.max,foo.obs/foo.max,mean(foo.ran)/foo.max))
}

results_foo<-NULL
for (i in c(1:length(tree.dataset)))
{
```

```
  for (j in c(1:length(tree.dataset)))
{
    print(paste (i,j))
    if (i!=j)
    {
      results_foo<-rbind(results_foo,tree_similarity_fer(tree.dataset[[i]],tree.dataset[[j]],10))
    }
    }
}
```

```
## [1] "1 1"
## [1] "1 2"
```

```
## Loading required package: TreeDist
```

```
## [1] "1 3"
## [1] "1 4"
## [1] "1 5"
## [1] "2 1"
## [1] "2 2"
## [1] "2 3"
## [1] "2 4"
## [1] "2 5"
## [1] "3 1"
## [1] "3 2"
## [1] "3 3"
## [1] "3 4"
## [1] "3 5"
## [1] "4 1"
## [1] "4 2"
## [1] "4 3"
## [1] "4 4"
## [1] "4 5"
## [1] "5 1"
## [1] "5 2"
## [1] "5 3"
## [1] "5 4"
## [1] "5 5"
```

```
print(t.test(results_foo[,2],results_foo[,3],paired = TRUE,alternative = "greater"))
```

```
##
##  Paired t-test
##
## data:  results_foo[, 2] and results_foo[, 3]
## t = 35.01, df = 19, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2061996       Inf
## sample estimates:
## mean of the differences
##                0.216913
```

```
print(t.test(results_foo[,2],results_foo[,3],alternative = "greater"))
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  results_foo[, 2] and results_foo[, 3]
## t = 38.264, df = 20.015, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2071361       Inf
## sample estimates:
## mean of x mean of y
## 0.3767541 0.1598411
```

2.3.2 Bayesian. Concordance within loci

```
results_foo<-NULL
control_foo<-cbind(
  sample(c(1:length(all.trees)),1500,replace=TRUE),
  sample(c(1:length(all.trees)),1500,replace=TRUE),
  sample(c(1:50),1500,replace=TRUE),
  sample(c(1:50),1500,replace=TRUE)
)

control_foo<-control_foo[paste(control_foo[,1],control_foo[,3])!=paste(control_foo[,2],control_foo[,4])

control_foo<-control_foo[!(paste(control_foo[,1],control_foo[,3],control_foo[,2],control_foo[,4])%in%pas

control_foo<-control_foo[sample(c(1:dim(control_foo)[1]),1000),]

results_foo<-future_apply(control_foo,1,FUN=function(x){
  return(c(x[1],x[2],x[3],x[4],results_foo,tree_similarity_fer(all.trees[[x[1]]][[x[3]]],all.trees[[x[2
  })
```

```
## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-1') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".
```

```
## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-2') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".
```

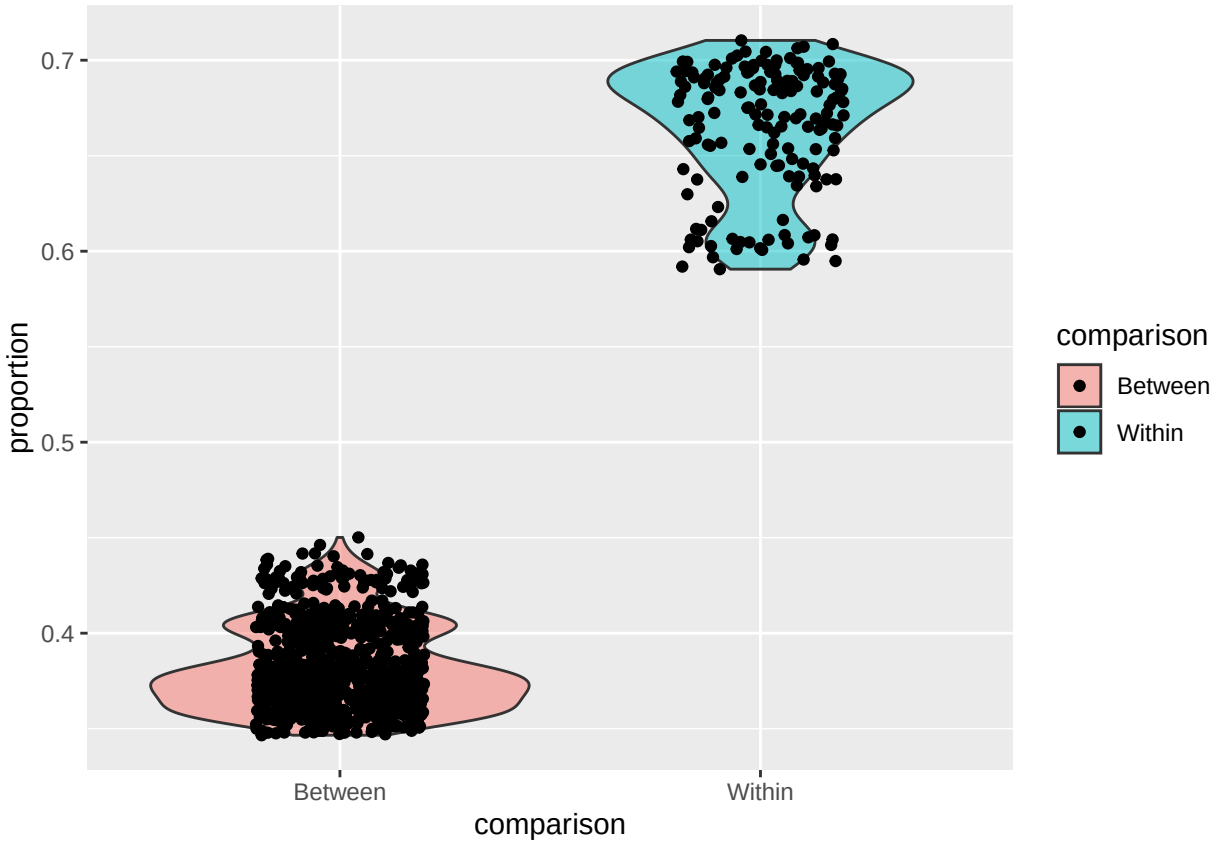
```
## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-3') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".
```

```
## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
```

```
## ('future_apply-4') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-5') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-6') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-7') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-8') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-9') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".

## Warning: UNRELIABLE VALUE: One of the 'future.apply' iterations
## ('future_apply-10') unexpectedly generated random numbers without declaring
## so. There is a risk that those random numbers are not statistically sound and
## the overall results might be invalid. To fix this, specify 'future.seed=TRUE'.
## This ensures that proper, parallel-safe random numbers are produced via the
## L'Ecuyer-CMRG method. To disable this check, use 'future.seed = NULL', or set
## option 'future.rng.onMisuse' to "ignore".
```

```
results_foo<-t(results_foo)
same_foo<-factor(results_foo[,1]==results_foo[,2])
```

```
levels(same_foo)<-c("Between","Within")
ggplot(data.frame(proportion=results_foo[,6],comparison=same_foo),aes(x=comparison,y=proportion,fill=co
```



2.3.3 Within and between loci

```
print(t.test(results_foo[same_foo=="Between",6],results_foo[same_foo=="Within",6],alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  results_foo[same_foo == "Between", 6] and results_foo[same_foo == "Within", 6]
## t = -101.38, df = 181.44, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.2772493
## sample estimates:
## mean of x mean of y
## 0.3826056 0.6644514
```

2.3.4 Between loci, random vs observed

```
print(t.test(results_foo[same_foo=="Between",6],results_foo[same_foo=="Between",7],alternative = "greate
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  results_foo[same_foo == "Between", 6] and results_foo[same_foo == "Between", 7]
## t = 242.4, df = 869.18, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.1899233       Inf
## sample estimates:
## mean of x mean of y
## 0.3826056 0.1913833
```

2.4 I.IV Multilocus summary of bGMYC delimitations

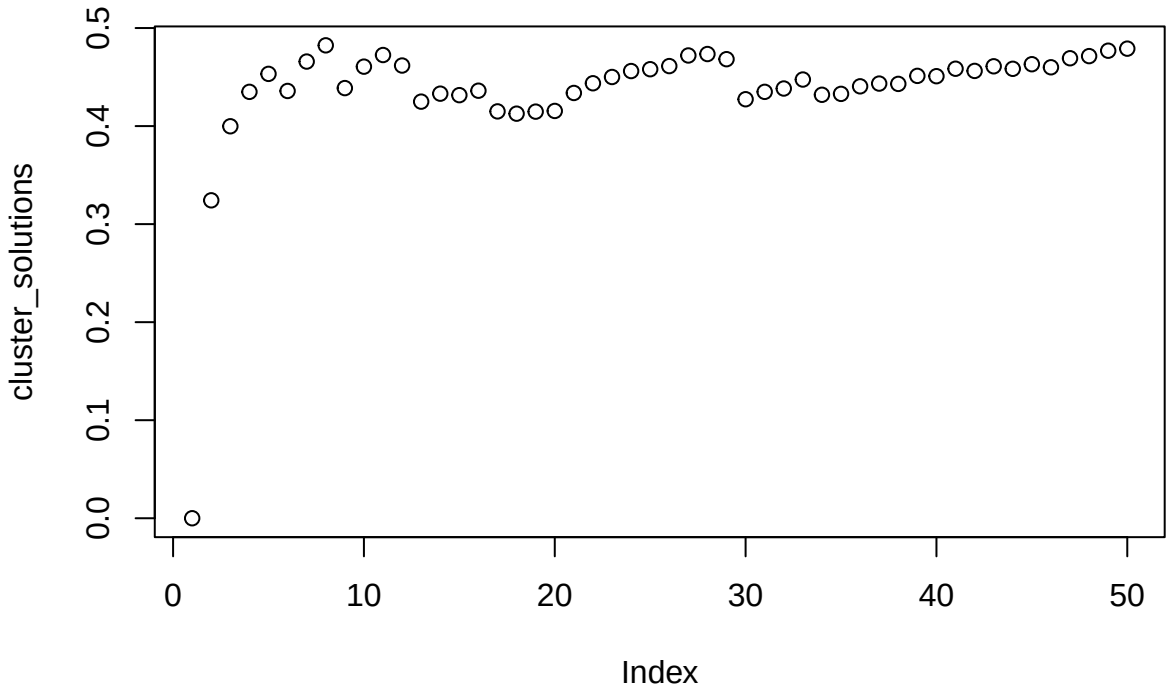
2.4.1 Averaging coassignment matrices

```
names.bGMYC<-c("its","bt","efa","mcm7","rpb1","alpha","hmg")
#
#
#
#
#
#
recast_probmat <- function (x)
  {
  class(x)<-NULL
  x<-x[rownames(x)!="Xanpa",colnames(x)!="Xanpa"]
  x<-melt(x)
  x$Var1<-gsub("[A-Z]","",x$Var1,perl=TRUE)
  x$Var2<-gsub("[A-Z]","",x$Var2,perl=TRUE)
  #x<-aggregate(value~Var1+Var2,data_bgmyc_all,FUN = combined_probability)
  return(acast(x,Var1~Var2,fun.aggregate = mean))
}


x1<-recast_probmat(output.bgmyc.1[[3]])
x2<-recast_probmat(output.bgmyc.2[[3]])
x3<-recast_probmat(output.bgmyc.3[[3]])
x4<-recast_probmat(output.bgmyc.4[[3]])
x5<-recast_probmat(output.bgmyc.5[[3]])

nombres_tabla<-c(rownames(x1), rownames(x2), rownames(x3), rownames(x4), rownames(x5))
nombres_tabla<-nombres_tabla[!duplicated(nombres_tabla)]
nombres_tabla<-nombres_tabla[order(as.numeric(nombres_tabla))]

data_bgmyc_all<-rbind(melt(x1),melt(x2),melt(x3),melt(x4),melt(x5))
data_bgmyc_all<-acast(data_bgmyc_all,Var1~Var2,fun.aggregate = mean)
```

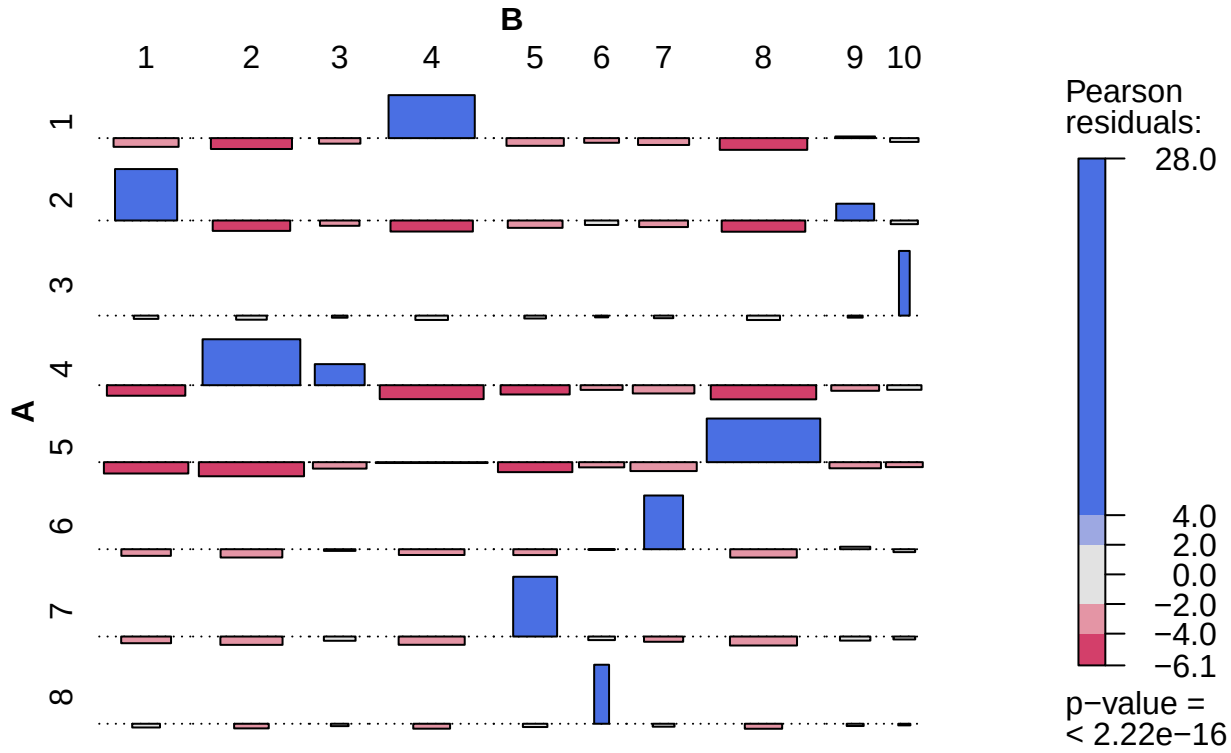
2.4.2 Extended range of K for kmedoids K=1:200

```
#coco<-pamk(1-data_bgmyc_all,1:200,diss=TRUE)
#cluster_solutions<-coco$crit
#plot(cluster_solutions)
```

```
coco<-pamk(1-data_bgmyc_all,1:50,diss=TRUE)
cluster_solutions<-coco$crit
plot(cluster_solutions)
```

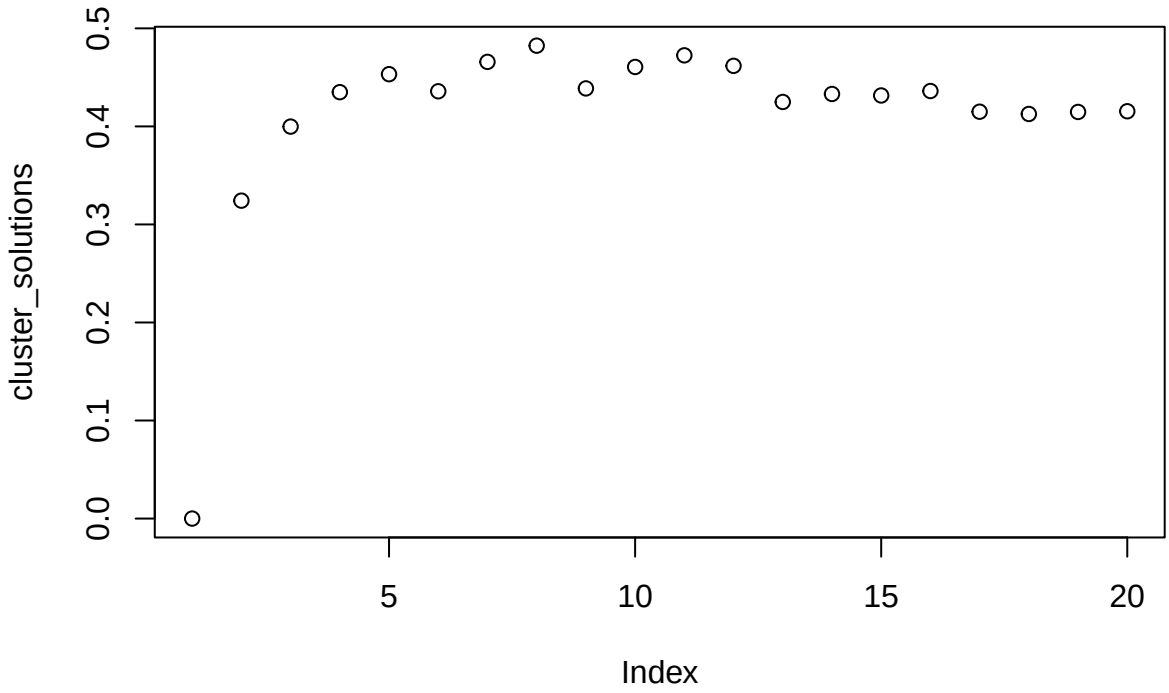



```
# Manually reorder
foo_bgmyc_clusters<-coco$pamobject$clustering

assoc(table(as.numeric(foo_bgmyc_clusters),condensed_baps_adm[names(foo_bgmyc_clusters),"name"]), shade
```



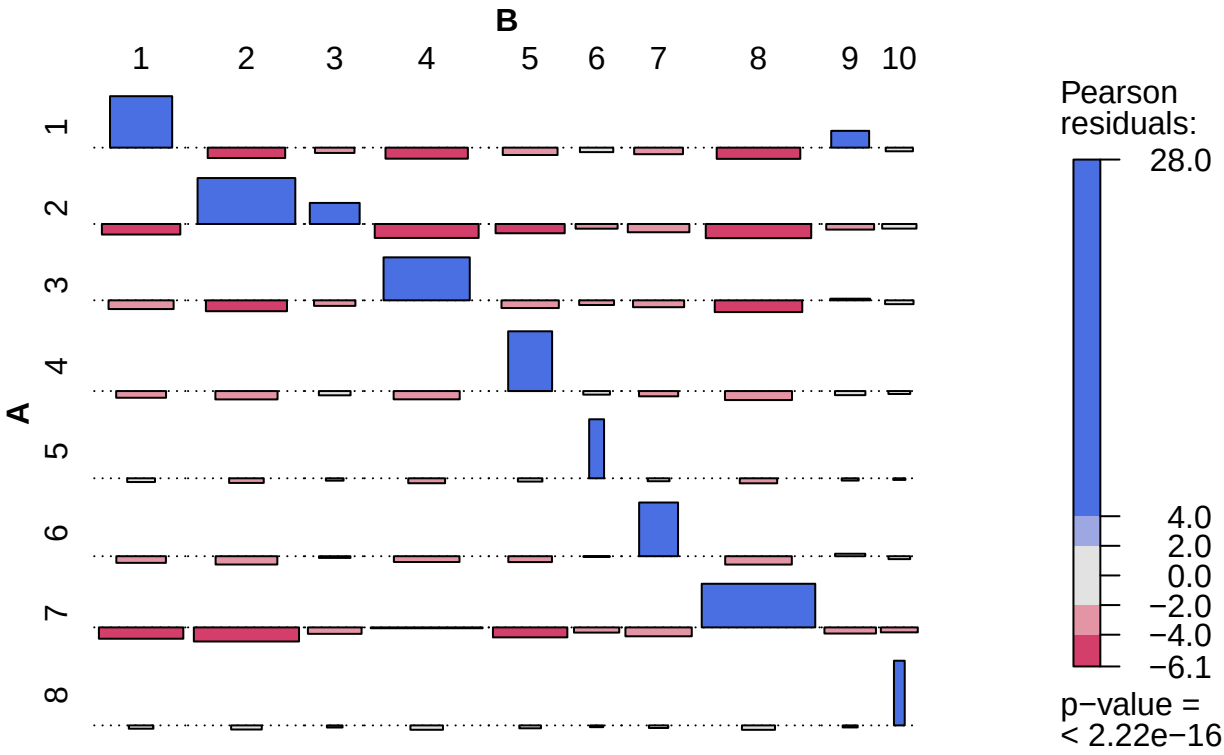
2.4.3 Reduced range of K values K=1:20

```
coco<-pamk(1-data_bgmyc_all,1:20,diss=TRUE)
cluster_solutions<-coco$crit
plot(cluster_solutions)
```




```
#save.image("/Users/Fernando/Desktop/01_Sanger_paper/14_bGMYC_All_loci/03_Processed/workspace_step5.Rda
```

```
# Manually reorder
foo_bgmyc_clusters<-coco$pamobject$clustering
foo_bgmyc_clusters<-factor(foo_bgmyc_clusters,levels = c(2,4,1,7,8,6,5,3))

assoc(table(as.numeric(foo_bgmyc_clusters),condensed_baps_adm[names(foo_bgmyc_clusters),"name"]), shade
```




```
assocstats(table(as.numeric(foo_bgmyc_clusters),condensed_baps_adm[names(foo_bgmyc_clusters),"name"]))
```

```
##                     X^2 df P(> X^2)
## Likelihood Ratio 2683.3 63        0
## Pearson          5010.4 63        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.927
## Cramer's V        : 0.932
```

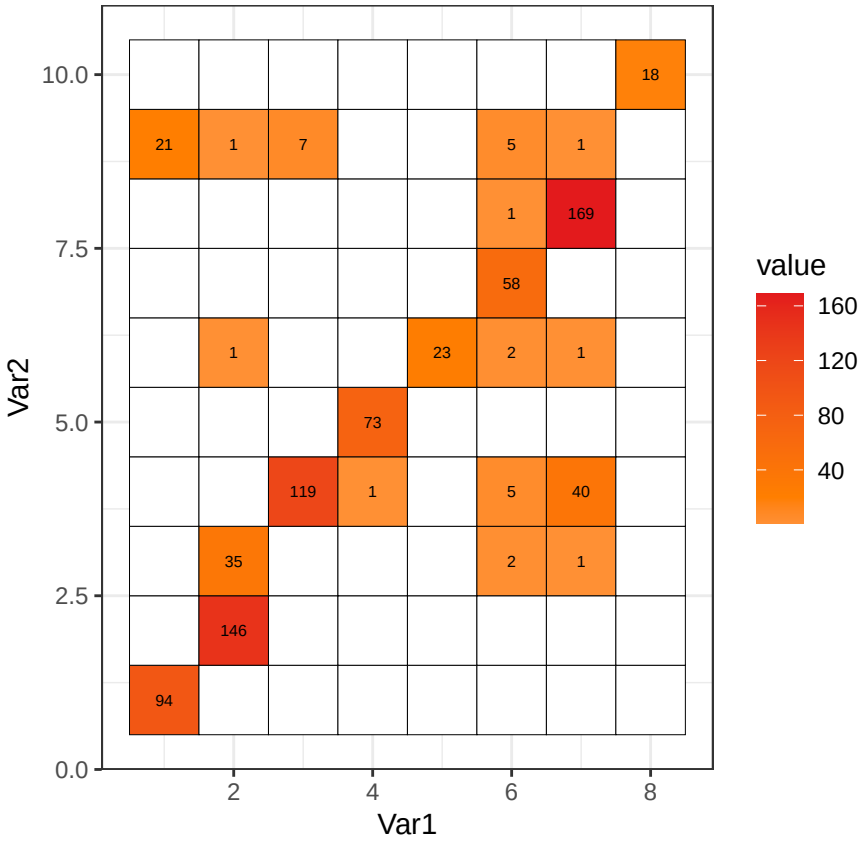
```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(as.numeric(foo_bgmyc_clusters),condensed_baps_adm[names(foo_bgmyc_clusters),"name"]))
```

```
##
##  Permutation test for conditional independence
##
## data:  table(as.numeric(foo_bgmyc_clusters), condensed_baps_adm[names(foo_bgmyc_clusters),     "name
## f(x) = 28.078, p-value < 2.2e-16
```

```
foo_tabla<-melt(table(as.numeric(foo_bgmyc_clusters),condensed_baps_adm[names(foo_bgmyc_clusters),"name
foo_tabla$value[foo_tabla$value==0]<-NA


  ggplot(foo_tabla,aes(x=Var1,y=Var2,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = 
    scale_fill_gradient2(low = "#FFFFFF",
  mid = colorinos.bipolar[7],
  high = colorinos.bipolar[6],
  midpoint = 20,
  space = "Lab",
  na.value = "#FFFFFF",
```

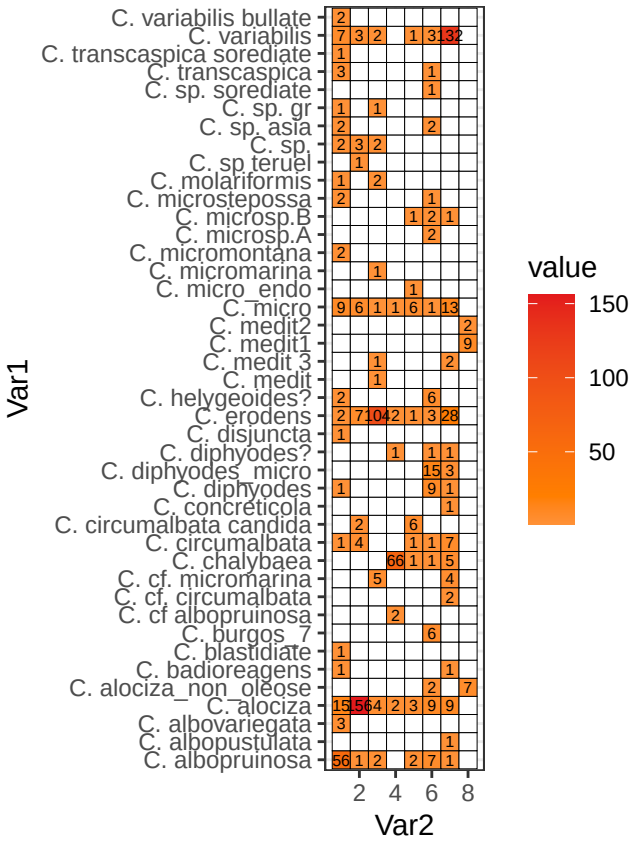
```
  guide = "colourbar",
  aesthetics = "fill"
) +
  coord_fixed() + theme_bw()
```

```
## Warning: Removed 57 rows containing missing values (geom_text).
```




```
# Heatmap 2 Species vs bGMYC

foo_tabla<-melt(table(condensed_baps_adm[names(foo_bgmyc_clusters),"Species"],as.numeric(foo_bgmyc_clust
foo_tabla$value[foo_tabla$value==0]<-NA


ggplot(foo_tabla,
       aes(x=Var2,y=Var1,fill=value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = value), color = "black", size = 2) +
  scale_fill_gradient2(low = "#FFFFFF",
  mid = colorinos.bipolar[7],
  high = colorinos.bipolar[6],
  midpoint = 20,
  space = "Lab",
  na.value = "#FFFFFF",
  guide = "colourbar",
  aesthetics = "fill") +
  coord_fixed() +
  theme_bw()
```

```
## Warning: Removed 239 rows containing missing values (geom_text).
```




```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters),"Species"],as.numeric(foo_bgmyc_clusters)))
```

```
##                     X^2  df P(> X^2)
## Likelihood Ratio 1867.6 287        0
## Pearson          3428.6 287        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.898
## Cramer's V        : 0.771
```

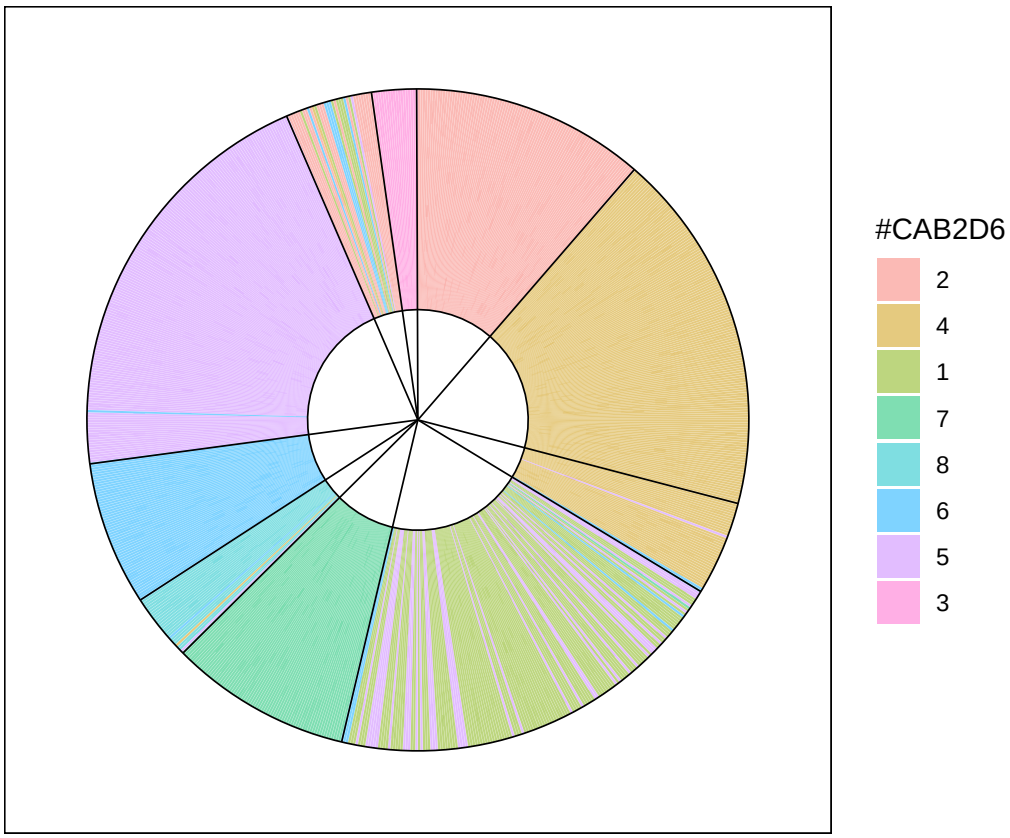
```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters),"Species"],as.numeric(foo_bgmyc_clusters
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters), "Species"],     as.numeric(foo_bgmyc_clust
## f(x) = 23.216, p-value < 2.2e-16
```

```
foo_bgmyc_clusters<-foo_bgmyc_clusters[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = foo_bgmyc_clusters[orden.foo])
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



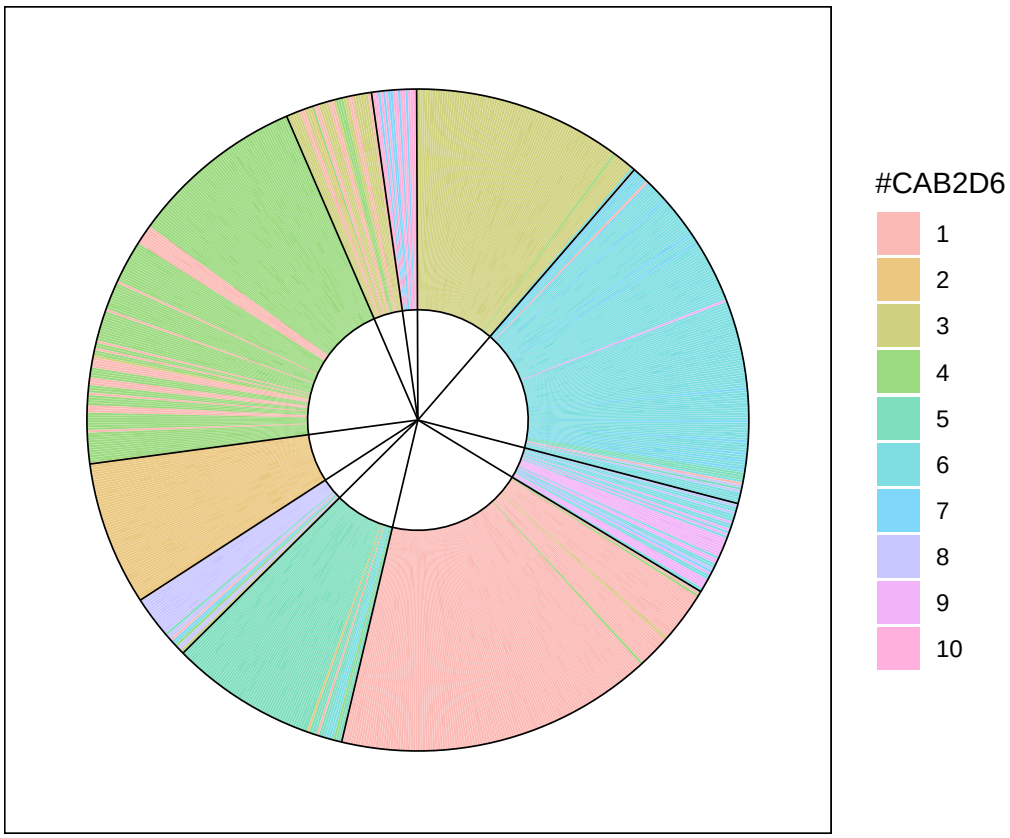
2.5 Single locus bGMYC

2.5.1 ITS

```
coco.its<-pamk(1-x1,2:10,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



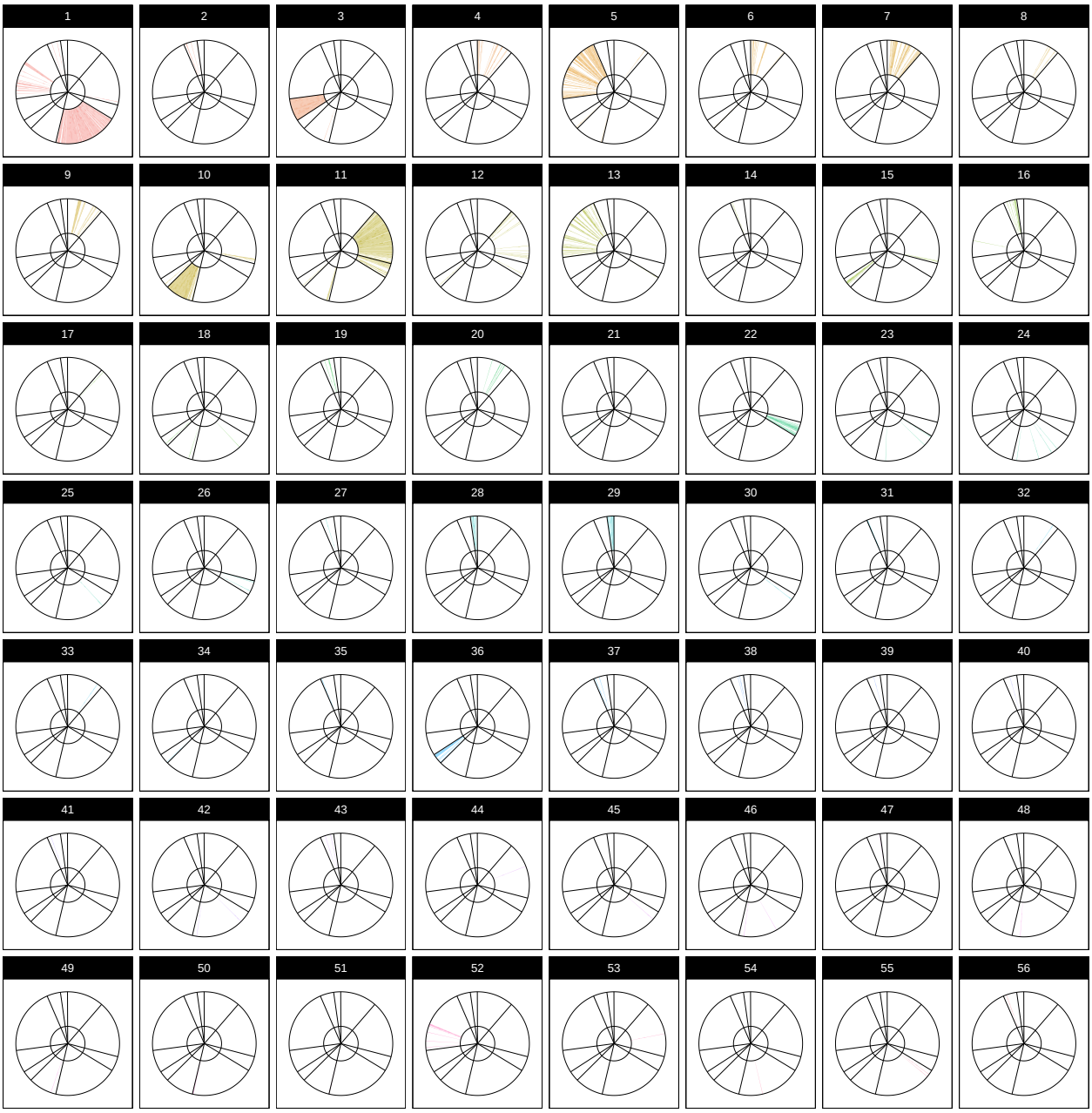

```
#+ facet_wrap(vars(clusters)))
```

```
coco.its<-pamk(1-x1,2:100,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white", guide = "none") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
```

```
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p + facet_wrap(vars(clusters)))
```




```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_clusters
```

2.5.1.1 Association with BAPS

```
##                     X^2  df P(> X^2)
```

```
## Likelihood Ratio 2940.8 495        0
## Pearson          6200.3 495        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.94
## Cramer's V        : 0.914
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_cluste
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "name"],     as.numeric(foo_bgmyc_clus
## f(x) = 26.168, p-value < 2.2e-16
```

```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_cluste
```

2.5.1.2 Association with species

```
##                     X^2   df  P(> X^2)
## Likelihood Ratio  2545.2 2255 1.603e-05
## Pearson          12928.6 2255 0.000e+00
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.97
## Cramer's V        : 0.619
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_clus
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "Species"],     as.numeric(foo_bgmyc_
## f(x) = 28.671, p-value = 0.224
```

```
assocstats(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

2.5.1.3 Association with multilocus bGMYC

```
##                    X^2  df P(> X^2)
## Likelihood Ratio 2581.2 385        0
## Pearson          4797.9 385        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.924
## Cramer's V        : 0.912
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)
```

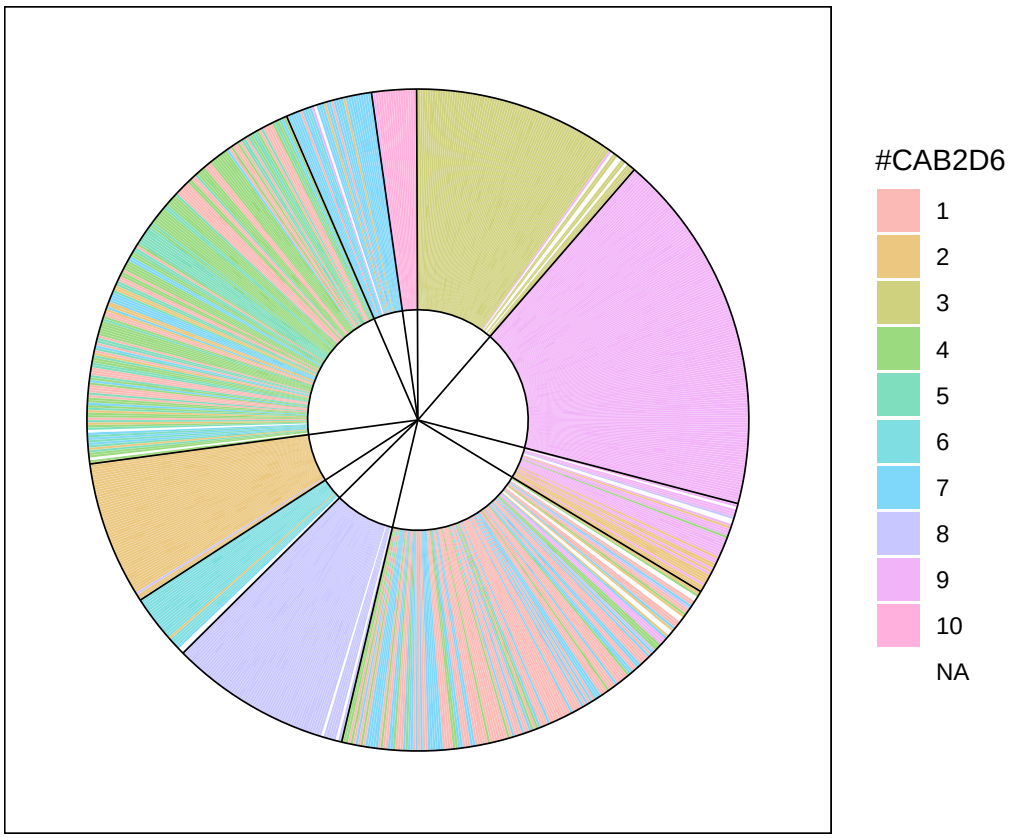
```
##
##  Permutation test for conditional independence
##
## data:  table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)], as.numeric(foo_bgmyc_clusters.its))
## f(x) = 23.417, p-value < 2.2e-16
```

2.5.2 Beta Tubulin

```
coco.its<-pamk(1-x2,2:10,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```




```
#+ facet_wrap(vars(clusters)))
```

```
coco.its<-pamk(1-x2,2:100,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white", guide = "none") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
```

```
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p + facet_wrap(vars(clusters)))
```


```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_clusters
```

2.5.2.1 Association with BAPS

```
##                      X^2  df P(> X^2)
## Likelihood Ratio 3005.0 765        0
## Pearson          6237.6 765        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.941
## Cramer's V        : 0.928
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_cluste
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "name"],     as.numeric(foo_bgmyc_clus
## f(x) = 25.218, p-value < 2.2e-16
```

```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_cluste
```

2.5.2.2 Association with species

```
##                      X^2   df P(> X^2)
## Likelihood Ratio  2665 3315        1
## Pearson          12847 3315        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.97
## Cramer's V        : 0.64
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_clus
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "Species"],     as.numeric(foo_bgmyc_
## f(x) = 28.32, p-value = 0.322
```

```
assocstats(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

2.5.2.3 Association with multilocus bGMYC

```
##                      X^2  df P(> X^2)
## Likelihood Ratio 2713.4 595        0
## Pearson          5065.3 595        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.929
## Cramer's V        : 0.949
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its))
```

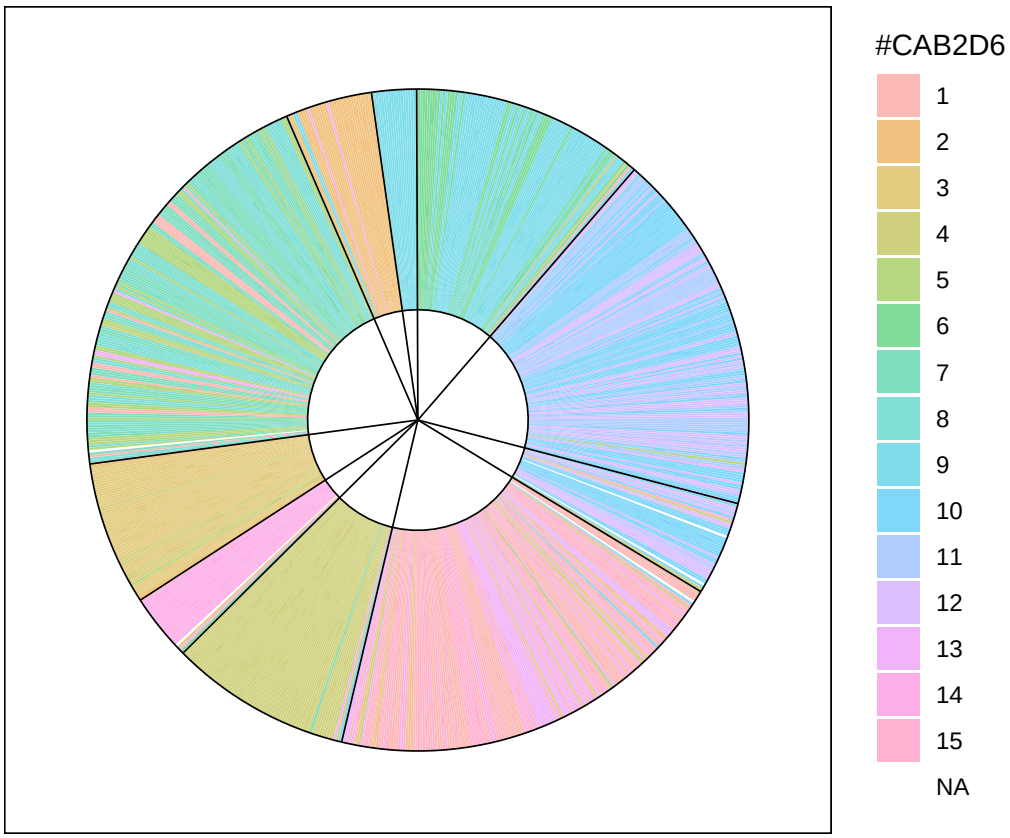
```
##
##  Permutation test for conditional independence
##
## data:  table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)], as.numeric(foo_bgmyc_clusters.its))
## f(x) = 25.395, p-value < 2.2e-16
```

2.5.3 Elongation factor alpha

```
coco.its<-pamk(1-x3,2:15,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



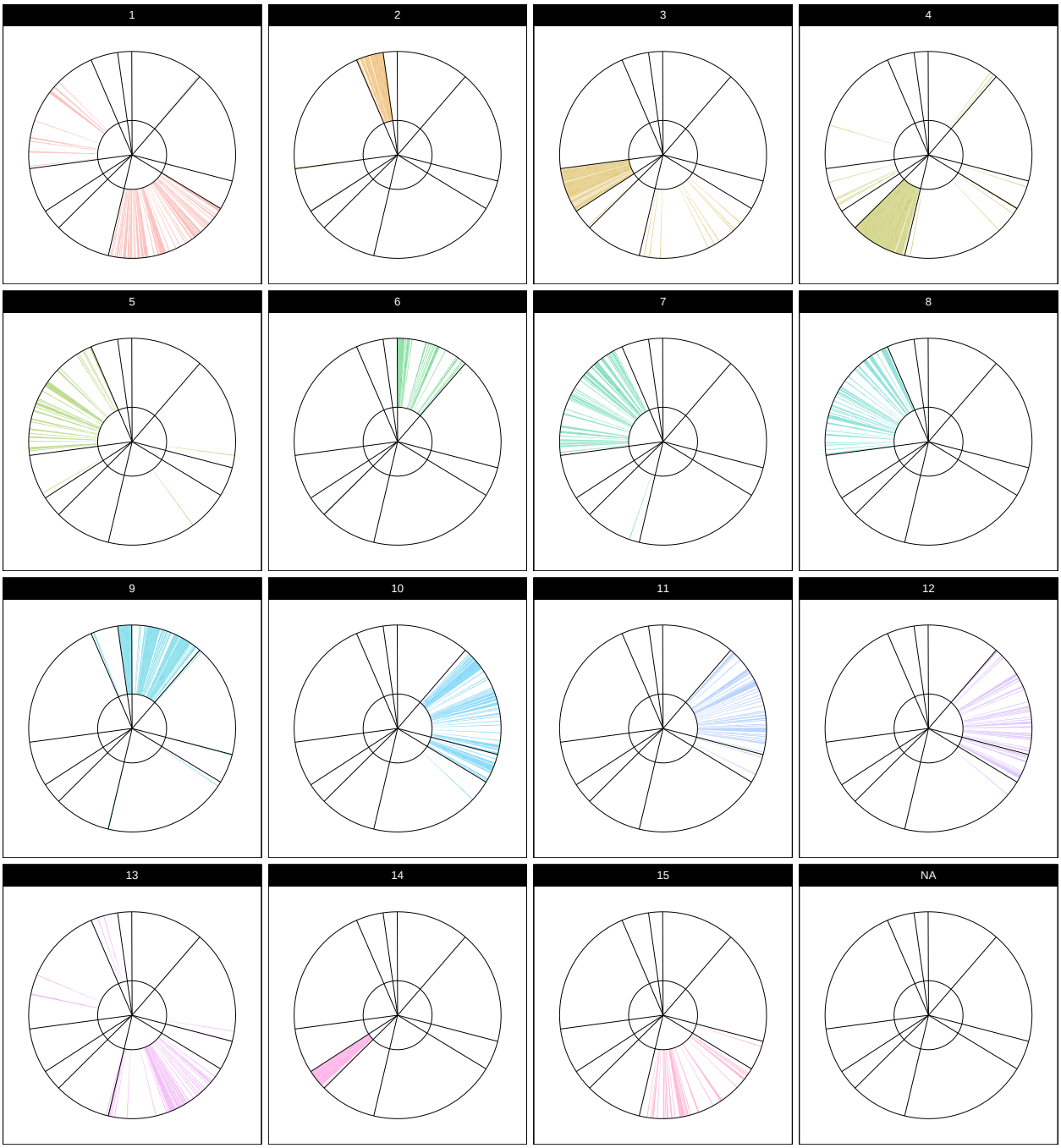

```
#+ facet_wrap(vars(clusters)))
```

```
coco.its<-pamk(1-x3,2:15,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white", guide = "none") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
```

```
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p + facet_wrap(vars(clusters)))
```




```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_clusters
```

2.5.3.1 Association with BAPS

```
##                      X^2  df P(> X^2)
## Likelihood Ratio 2658.7 126        0
## Pearson          4821.4 126        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.925
## Cramer's V        : 0.809
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_cluste
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "name"],     as.numeric(foo_bgmyc_clus
## f(x) = 26.062, p-value < 2.2e-16
```

```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_cluste
```

2.5.3.2 Association with species

```
##                      X^2  df P(> X^2)
## Likelihood Ratio 1916.6 574        0
## Pearson          3283.7 574        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.895
## Cramer's V        : 0.535
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_clus
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "Species"],     as.numeric(foo_bgmyc_
## f(x) = 20.653, p-value < 2.2e-16
```

```
assocstats(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

2.5.3.3 Association with multilocus bGMYC

```
##                      X^2 df P(> X^2)
## Likelihood Ratio 2318.6 98        0
## Pearson          3804.6 98        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.907
## Cramer's V        : 0.815
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its))
```

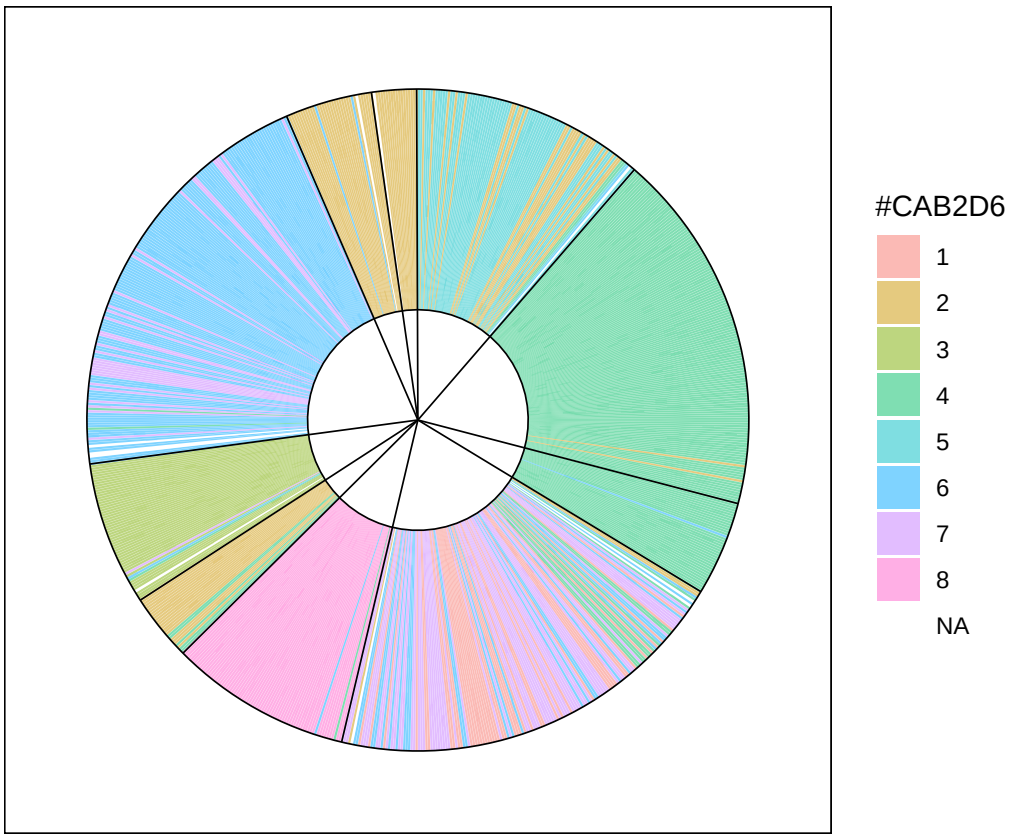
```
##
##  Permutation test for conditional independence
##
## data:  table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)], as.numeric(foo_bgmyc_clusters.its))
## f(x) = 26.57, p-value < 2.2e-16
```

2.5.4 MCM7

```
coco.its<-pamk(1-x4,2:10,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



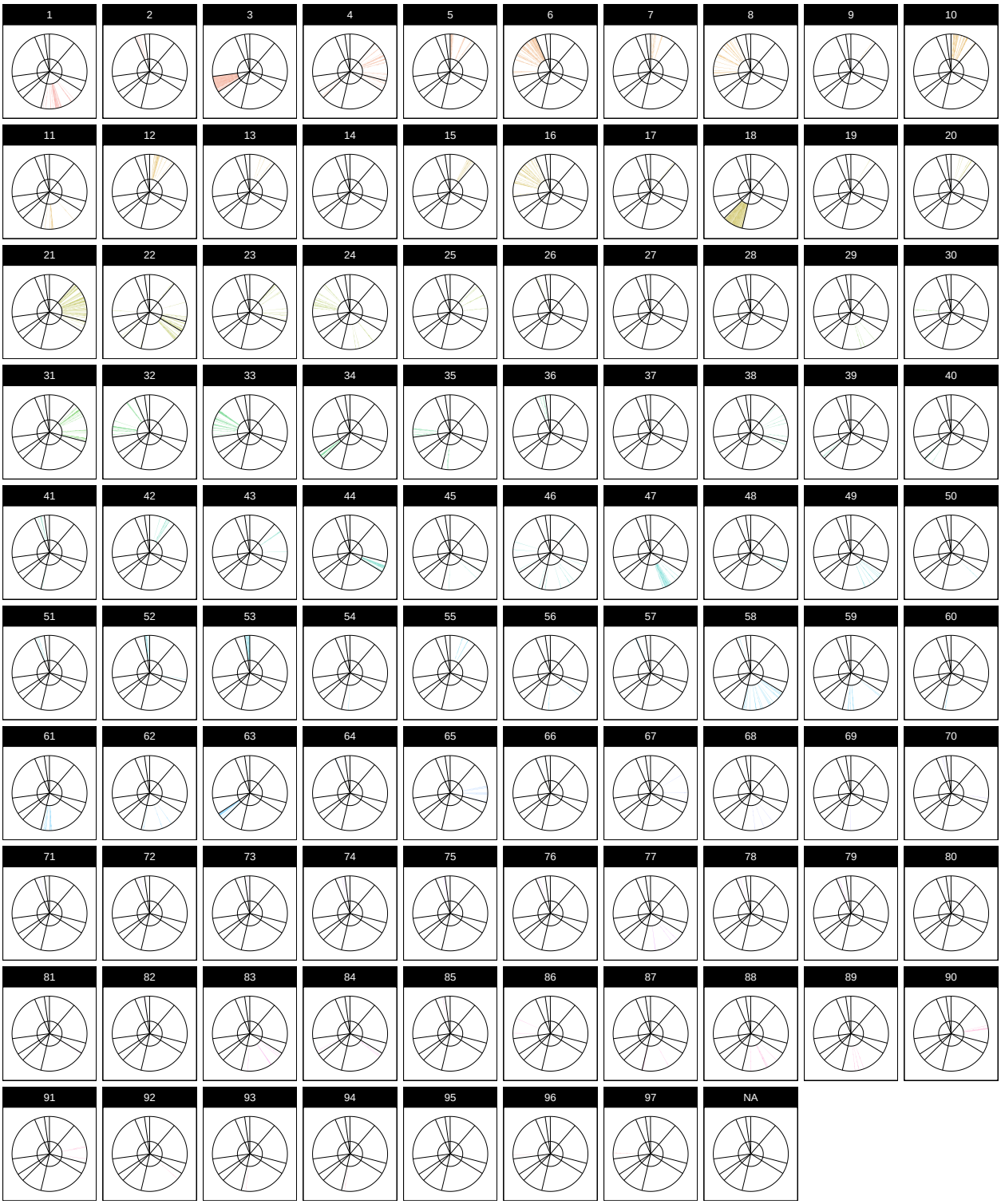

```
#+ facet_wrap(vars(clusters)))
```

```
coco.its<-pamk(1-x4,2:100,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white", guide= "none") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
```

```
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p + facet_wrap(vars(clusters)))
```




```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_clusters
```

2.5.4.1 Association with BAPS

```
##                     X^2  df P(> X^2)
```

```
## Likelihood Ratio 3091.4 864        0
## Pearson          6412.3 864        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.942
## Cramer's V        : 0.936
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_cluste
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "name"],     as.numeric(foo_bgmyc_clus
## f(x) = 25.342, p-value < 2.2e-16
```

```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_cluste
```

2.5.4.2 Association with species

```
##                      X^2   df P(> X^2)
## Likelihood Ratio  2806.5 3936        1
## Pearson          13734.2 3936        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.972
## Cramer's V        : 0.642
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_clus
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "Species"],     as.numeric(foo_bgmyc_
## f(x) = 28.496, p-value = 0.306
```

```
assocstats(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

2.5.4.3 Association with multilocus bGMYC

```
##                    X^2  df P(> X^2)
## Likelihood Ratio 2744.2 672        0
## Pearson          5085.6 672        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.928
## Cramer's V        : 0.945
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)
```

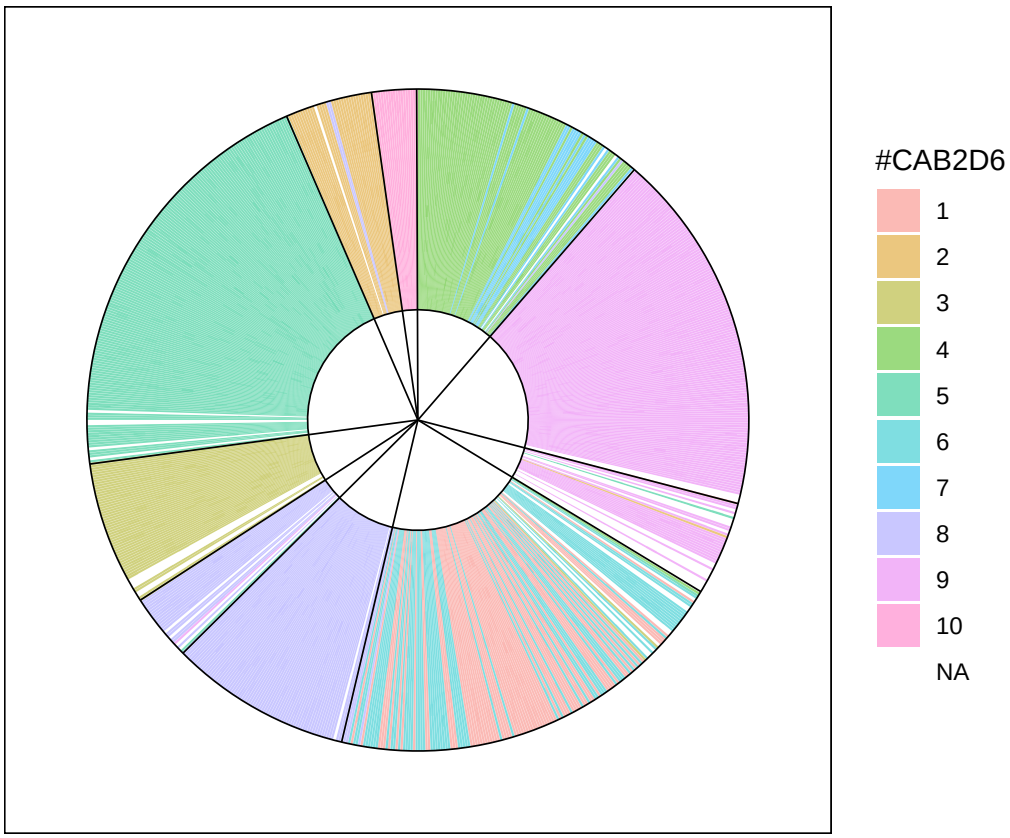
```
##
##  Permutation test for conditional independence
##
## data:  table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)], as.numeric(foo_bgmyc_clusters.its))
## f(x) = 24.68, p-value < 2.2e-16
```

2.5.5 RPB1

```
coco.its<-pamk(1-x5,2:10,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



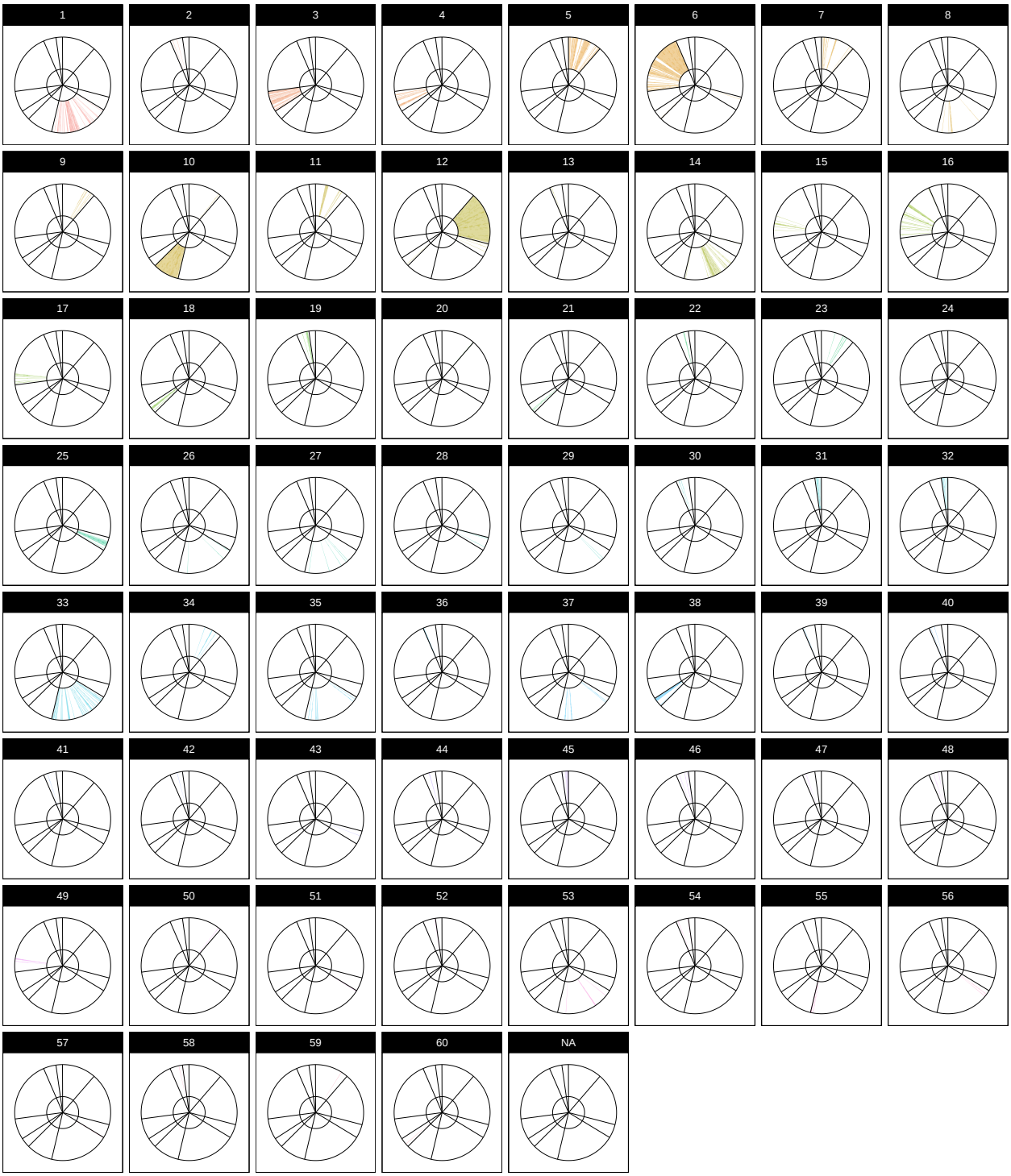

```
#+ facet_wrap(vars(clusters)))
```

```
coco.its<-pamk(1-x5,2:100,diss=TRUE)
foo_bgmyc_clusters.its<-coco.its$pamobject$clustering
foo_bgmyc_clusters.its<-foo_bgmyc_clusters.its[rownames(condensed_baps_adm)]
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo,clusters = as.factor(foo_bgmyc_clusters.its[orden.foo]))
foo<-melt(foo[,c(1,26)])
```

```
## Using orden, clusters as id variables
```

```
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=clusters),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos.bipolar, na.value="white", guide = "none") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
```

```
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p + facet_wrap(vars(clusters)))
```



```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_clusters
```

2.5.5.1 Association with BAPS

```
##                     X^2  df P(> X^2)
## Likelihood Ratio 3075.4 531        0
## Pearson          6656.4 531        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.946
## Cramer's V        : 0.977
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"name"],as.numeric(foo_bgmyc_cluste
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "name"],     as.numeric(foo_bgmyc_clus
## f(x) = 25.043, p-value < 2.2e-16
```

```
assocstats(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_cluste
```

2.5.5.2 Association with species

```
##                      X^2   df   P(> X^2)
## Likelihood Ratio  2654.8 2419 0.00049163
## Pearson          13451.3 2419 0.00000000
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.972
## Cramer's V        : 0.651
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[names(foo_bgmyc_clusters.its),"Species"],as.numeric(foo_bgmyc_clus
```

```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[names(foo_bgmyc_clusters.its), "Species"],     as.numeric(foo_bgmyc_
## f(x) = 27.803, p-value = 0.176
```

```
assocstats(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

2.5.5.3 Association with multilocus bGMYC

```
##                     X^2  df P(> X^2)
## Likelihood Ratio 2717.0 413        0
## Pearson          5119.2 413        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.932
```

```
## Cramer's V        : 0.971
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)],as.numeric(foo_bgmyc_clusters.its)))
```

```
##
##  Permutation test for conditional independence
##
## data:  table(foo_bgmyc_clusters[names(foo_bgmyc_clusters.its)], as.numeric(foo_bgmyc_clusters.its))
## f(x) = 25.043, p-value < 2.2e-16
```

3 Part II. Mating type coocurrence

3.1 II.I Read data and CD-HIT

```
i=9
for (file in c(
        "/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/proteins/alpha_v13_exons.fasta",
        "/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/proteins/hmg_v13_Only_exons.fasta"
        ))
        {
        sequences<-read.dna(file,format="fasta")
        seq.dataset[[i]]<-sequences
        names.dataset[[i]]<-strsplit(rownames(sequences),split="_")
        i<-i+1
        }
#   library(phangorn)
    i=11
    for (file in c(
        "/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/proteins/alpha_v13_translation.fasta",
        "/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/proteins/hmg_v13_shortened_translation.
        ))
        {
        sequences<-read.aa(file,format="fasta")
        seq.dataset[[i]]<-sequences
        names.dataset[[i]]<-strsplit(names(sequences),split="_")
        i<-i+1
        }
```

```
for ( i in 9:12)
    {
    for (j in 1:length(names.dataset[[i]]))
    {
    names.dataset[[i]][[j]]<-names.dataset[[i]][[j]][1]
    }
    names.dataset[[i]]<-unlist(names.dataset[[i]])
    }
#alpha
names.dataset[[9]][is.na(as.numeric(names.dataset[[9]]))]<-c("207","205")
```

```
## Warning in names.dataset[[9]][is.na(as.numeric(names.dataset[[9]]))] <-
## c("207", : NAs introducidos por coerción
```

```
names.dataset$alpha_tr<-gsub("X","",names.dataset$alpha_tr)
names.dataset[[11]][is.na(as.numeric(names.dataset[[11]]))]<-c("207","205")
```

```
## Warning in names.dataset[[11]][is.na(as.numeric(names.dataset[[11]]))] <-
## c("207", : NAs introducidos por coerción
```

```
#hmg_tr"
names.dataset$hmg_tr<-gsub("X","",names.dataset$hmg_tr)
#
#
#
rownames(seq.dataset$alpha)<-names.dataset$alpha
rownames(seq.dataset$hmg)<-names.dataset$hmg
names(seq.dataset$alpha_tr)<-names.dataset$alpha_tr
names(seq.dataset$hmg_tr)<-names.dataset$hmg_tr
#
names.dataset[[8]][names.dataset[[8]]=="00"]<-"0"
names.dataset[[9]][names.dataset[[9]]=="00"]<-"0"
#
# Renumber doubled from Vondrak
#
a<-as.character(seq(2,56,2))
b<-as.character(seq(1,55,2))
for (i in 9:length(names.dataset))
    {
        for (j in 1:28)
        {
            names.dataset[[i]][names.dataset[[i]]==a[j]]<-b[j]
        }
    }
rm(a,b)


reduced.dataset$alpha<-seq.dataset$alpha[names.dataset$alpha%in%rownames(dataset),]
reduced.dataset$hmg<-seq.dataset$hmg[names.dataset$hmg%in%rownames(dataset),]

reduced.dataset$alpha_tr<-subset(seq.dataset$alpha_tr,names(seq.dataset$alpha_tr)%in%names.dataset$alpha
reduced.dataset$hmg_tr<-subset(seq.dataset$hmg_tr,names(seq.dataset$hmg_tr)%in%names.dataset$hmg_tr[name
```

```
#
# 5. Export alignment files
#------------------------------
#for (i in 9:(length(names.dataset)-2))
#{
#if (is.list(names.dataset[[i]])==TRUE)#
#   {
#   names(seq.dataset[[i]])<-names.dataset[[i]]
#   write.dna(seq.dataset[[i]][names.dataset[[i]]%in%rownames(dataset)],paste("/Users/Fernando/Desktop/
#   } else
#       {
#       rownames(seq.dataset[[i]])<-names.dataset[[i]]
#       write.dna(seq.dataset[[i]][names.dataset[[i]]%in%rownames(dataset),],paste("/Users/Fernando/Des
#       }
#
```

```
#}
#
#
#
#
#
#
# Factor dataset for bipartite networks
#
alpha.haplo<-mat.or.vec(length(reduced.dataset$alpha_tr),2)
rownames(alpha.haplo)<-names(reduced.dataset$alpha_tr)
for (i in 1:dim(alpha.haplo)[1])
    {
    alpha.haplo[i,1]<-paste(as.vector(reduced.dataset$alpha_tr)[[i]],collapse="")
    }
alpha.haplo<-alpha.haplo[!(rownames(alpha.haplo)%in%c("377","1073","1170","1171")),]
alpha.haplo[,2]<-factor(alpha.haplo[,1])

hmg.haplo<-mat.or.vec(length(reduced.dataset$hmg_tr),2)
rownames(hmg.haplo)<-names(reduced.dataset$hmg_tr)
for (i in 1:dim(hmg.haplo)[1])
    {
    hmg.haplo[i,1]<-paste(as.character(reduced.dataset$hmg_tr)[[i]],collapse="")
    }
hmg.haplo<-hmg.haplo[!(rownames(hmg.haplo)%in%c("377","1073","1170","1171")),]
hmg.haplo[,2]<-factor(hmg.haplo[,1])
#
#
#
foo.alpha<-as.character(sort(as.numeric(rownames(alpha.haplo))))[duplicated(as.character(sort(as.numeri
#
foo.hmg<-as.character(sort(as.numeric(rownames(hmg.haplo))))[duplicated(as.character(sort(as.numeric(ro
foo.both<-c(foo.alpha,foo.hmg)
foo.both<-foo.both[!duplicated(foo.both)]
foo.both<-as.character(sort(as.numeric(foo.both)))
foo.names<-c(rownames(alpha.haplo),rownames(hmg.haplo))
foo.names<-as.character(sort(as.numeric(foo.names[!duplicated(foo.names)])))
#
#
out.haps<-NULL
for (i in foo.names)
    {
    if(i%in%rownames(alpha.haplo)&i%in%rownames(hmg.haplo))
      {
      a<-alpha.haplo[rownames(alpha.haplo)==i,2]
      h<-hmg.haplo[rownames(hmg.haplo)==i,2]
        for(j in a)
         {
          for (k in h)
          {
          out.haps<-rbind(out.haps,c(i,j,k))
          }
        }
```

```
    } else if (i%in%rownames(alpha.haplo)&!(i%in%rownames(hmg.haplo)))
      {
      a<-alpha.haplo[rownames(alpha.haplo)==i,2]
      out.haps<-rbind(out.haps,cbind(i,a,0))
      } else if (!(i%in%rownames(alpha.haplo))&i%in%rownames(hmg.haplo))
      {
        a<-hmg.haplo[rownames(hmg.haplo)==i,2]
        out.haps<-rbind(out.haps,cbind(i,0,a))
        }
    }
    rownames(out.haps)<-out.haps[,1]
#
#
# 4. External Esport for clustering in CD-Hit
#
#alpha.haplo.seq<-as.character(newnames.reduced.dataset$alpha_tr)
#alpha.haplo.seq<-subset(alpha.haplo.seq,!duplicated(alpha.haplo[,2]))
#rownames(alpha.haplo.seq)<-paste("SeqID",alpha.haplo[!duplicated(alpha.haplo[,2]),2],"_",table(alpha.h
#alpha.haplo.seq<-alpha.haplo.seq[order(table(alpha.haplo[,2])[as.numeric(alpha.haplo[!duplicated(alpha

#hmg.haplo.seq<-as.character(newnames.reduced.dataset$hmg_tr)
#hmg.haplo.seq<-subset(hmg.haplo.seq,!duplicated(hmg.haplo[,2]))
#rownames(hmg.haplo.seq)<-paste("SeqID",hmg.haplo[!duplicated(hmg.haplo[,2]),2],"_",table(hmg.haplo[,2]
#hmg.haplo.seq<-hmg.haplo.seq[order(table(hmg.haplo[,2])[as.numeric(hmg.haplo[!duplicated(hmg.haplo[,2]

#write.aa(alpha.haplo.seq,"/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/alpha_collapsed.fasta
#write.aa(hmg.haplo.seq,"/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/hmg_collapsed.fasta")
#
# Import CD HIT
#
import.cd.hit<-function(file,reference)
{
require(stringr)
foo.read<-readLines(file)
foo.alpha99<-list()
a<-grep(">Cluster ",foo.read);
a<-c(a,length(foo.read))
for (i in 1:(length(a)-1))
    {
            if((a[i]+1)==(a[i+1]-1))
            {
                foo.alpha99[[i]]<-foo.read[(a[i]+1)]
            }
            else
                {
                    foo.alpha99[[i]]<-foo.read[(a[i]+1):(a[i+1]-1)]
                }
            }
    names(foo.alpha99)<-foo.read[a[-length(a)]]
    for (i in 1:length(foo.alpha99)) foo.alpha99[[i]]<-str_extract(foo.alpha99[[i]], ">SeqID[0-9]+")
    for (i in 1:length(foo.alpha99)) foo.alpha99[[i]]<-gsub(">SeqID","",foo.alpha99[[i]])
    names(foo.alpha99)<-gsub(">Cluster ","",names(foo.alpha99))
    out<-mat.or.vec(length(reference[,2]),1)
```

```
    for (i in 1:length(foo.alpha99))
    {out[reference[,2]%in%foo.alpha99[[i]]]<-names(foo.alpha99)[i]}
    out<-as.numeric(out)+1
    return(out)
}

alpha.haplo<-cbind(alpha.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/pr
```

```
## Loading required package: stringr
```

```
alpha.haplo<-cbind(alpha.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/pr
alpha.haplo<-cbind(alpha.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/pr

hmg.haplo<-cbind(hmg.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/protei
hmg.haplo<-cbind(hmg.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/protei
hmg.haplo<-cbind(hmg.haplo,import.cd.hit("/Users/Fernando/Desktop/01_Sanger_paper/6_Mating_types/protei
```

```
#
# 99%
#
out.99<-NULL
out.97<-NULL
out.95<-NULL

for (i in foo.names)
    {
    if(i%in%rownames(alpha.haplo)&i%in%rownames(hmg.haplo))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,3]
    h<-hmg.haplo[rownames(hmg.haplo)==i,3]
    for(j in a)
    {
    for (k in h)
    {
    out.99<-rbind(out.99,c(i,j,k))
    }
    }
    }
    if (i%in%rownames(alpha.haplo)&!(i%in%rownames(hmg.haplo)))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,3]
    out.99<-rbind(out.99,cbind(i,a,0))
    }
    else if (!(i%in%rownames(alpha.haplo))&i%in%rownames(hmg.haplo))
    {
    a<-hmg.haplo[rownames(hmg.haplo)==i,3]
    out.99<-rbind(out.99,cbind(i,0,a))
    }
    }
    rownames(out.99)<-out.99[,1]

out.99<-out.99[!(rownames(out.99)%in%c("199","377","1073","1170","1171")),]
#
# 97%
```

```
#
for (i in foo.names)
    {
    if(i%in%rownames(alpha.haplo)&i%in%rownames(hmg.haplo))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,4]
    h<-hmg.haplo[rownames(hmg.haplo)==i,4]
    for(j in a)
    {
    for (k in h)
    {
    out.97<-rbind(out.97,c(i,j,k))
    }
    }
    }
    if (i%in%rownames(alpha.haplo)&!(i%in%rownames(hmg.haplo)))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,4]
    out.97<-rbind(out.97,cbind(i,a,0))
    }
    else if (!(i%in%rownames(alpha.haplo))&i%in%rownames(hmg.haplo))
    {
    a<-hmg.haplo[rownames(hmg.haplo)==i,4]
    out.97<-rbind(out.97,cbind(i,0,a))
    }
    }
    rownames(out.97)<-out.97[,1]

out.97<-out.97[!(rownames(out.97)%in%c("199","377","1073","1170","1171")),]
#
# 95% Similarity
#
for (i in foo.names)
    {
    if(i%in%rownames(alpha.haplo)&i%in%rownames(hmg.haplo))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,5]
    h<-hmg.haplo[rownames(hmg.haplo)==i,5]
    for(j in a)
    {
    for (k in h)
    {
    out.95<-rbind(out.95,c(i,j,k))
    }
    }
    }
    if (i%in%rownames(alpha.haplo)&!(i%in%rownames(hmg.haplo)))
    {
    a<-alpha.haplo[rownames(alpha.haplo)==i,5]
    out.95<-rbind(out.95,cbind(i,a,0))
    }
    else if (!(i%in%rownames(alpha.haplo))&i%in%rownames(hmg.haplo))
    {
```

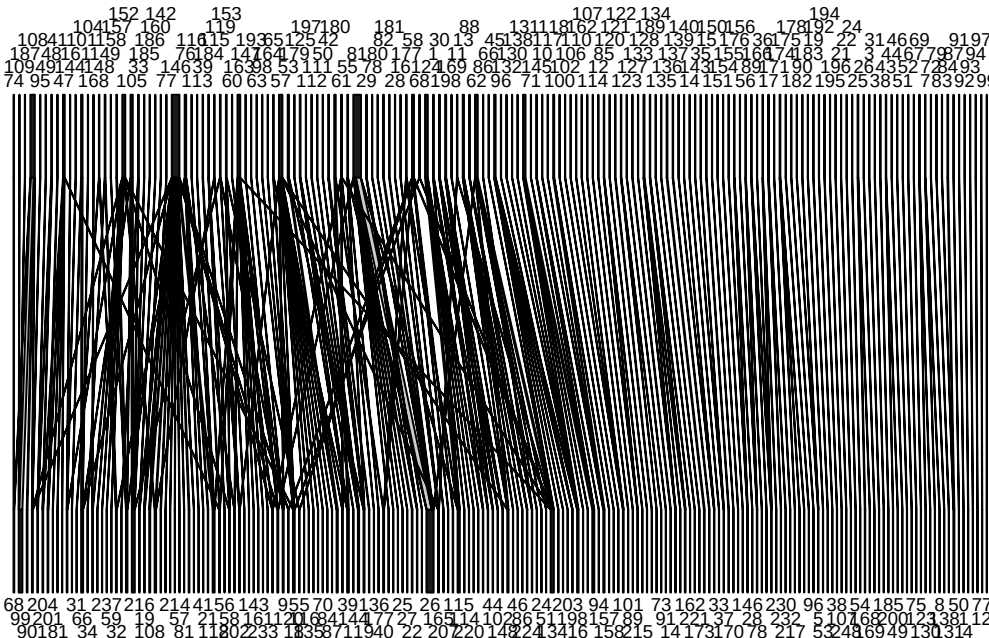
```
    a<-hmg.haplo[rownames(hmg.haplo)==i,5]
    out.95<-rbind(out.95,cbind(i,0,a))
    }
    }
    rownames(out.95)<-out.95[,1]

out.95<-out.95[!(rownames(out.95)%in%c("199","377","1073","1170","1171")),]
```

3.2 II.II Compartmentalization and modularity

3.2.1 At haplotype level (Only compartments)

```
table.mating.types<-out.haps[,-1]
table.mating.types<-table(table.mating.types[,1],table.mating.types[,2])
table.mating.types<-table.mating.types[-1,-1]
table.mating.types<-table.mating.types[rowSums(table.mating.types)>0,colSums(table.mating.types)>0]
comps<-compart(table.mating.types)
```

```
colorinos_compartments<-colorRampPalette(colorinos.bipolar)(comps$n.compart)
plotweb(table.mating.types)
```



3.2.1.1 Network compartments

```
kable(networklevel(table.mating.types))
```

```
## Error in (function (cl, name, valueClass)  :
##   assignment of an object of class "table" is not valid for @'originalWeb' in an object of class "mod
```

	x
connectance	0.0097337
web asymmetry	-0.0909091
links per species	0.8760331

	x
number of compartments	66.0000000
compartment diversity	12.4520631
cluster coefficient	0.0050505
modularity Q	NA
nestedness	1.3055647
NODF	1.4900002
weighted nestedness	NaN
weighted NODF	0.3293185
interaction strength asymmetry	-0.0333291
specialisation asymmetry	0.0505732
linkage density	3.2394124
weighted connectance	0.0089240
Fisher alpha	556.2818877
Shannon diversity	5.5192298
interaction evenness	0.5309907
Alatalo interaction evenness	0.5824975
H2	0.4479515
number.of.species.HL	165.0000000
number.of.species.LL	198.0000000
mean.number.of.shared.partners.HL	0.0171471
mean.number.of.shared.partners.LL	0.0299954
cluster.coefficient.HL	0.0256999
cluster.coefficient.LL	0.0171788
weighted.cluster.coefficient.HL	0.0187563
weighted.cluster.coefficient.LL	0.0422438
niche.overlap.HL	0.0071341
niche.overlap.LL	0.0162541
togetherness.HL	0.0009580
togetherness.LL	0.0038503
C.score.HL	0.9867321
C.score.LL	0.9738783
V.ratio.HL	2.7936078
V.ratio.LL	0.8737068
discrepancy.HL	283.0000000
discrepancy.LL	288.0000000
extinction.slope.HL	1.3140338
extinction.slope.LL	1.3864882
robustness.HL	0.5689320
robustness.LL	0.5819856
functional.complementarity.HL	341.8393154
functional.complementarity.LL	339.7511160
partner.diversity.HL	0.9994908
partner.diversity.LL	0.6602675
generality.HL	4.1900761
vulnerability.LL	2.2887487

```
kable(grouplevel(table.mating.types))
```

	x
number.of.species.HL	165.0000000
number.of.species.LL	198.0000000

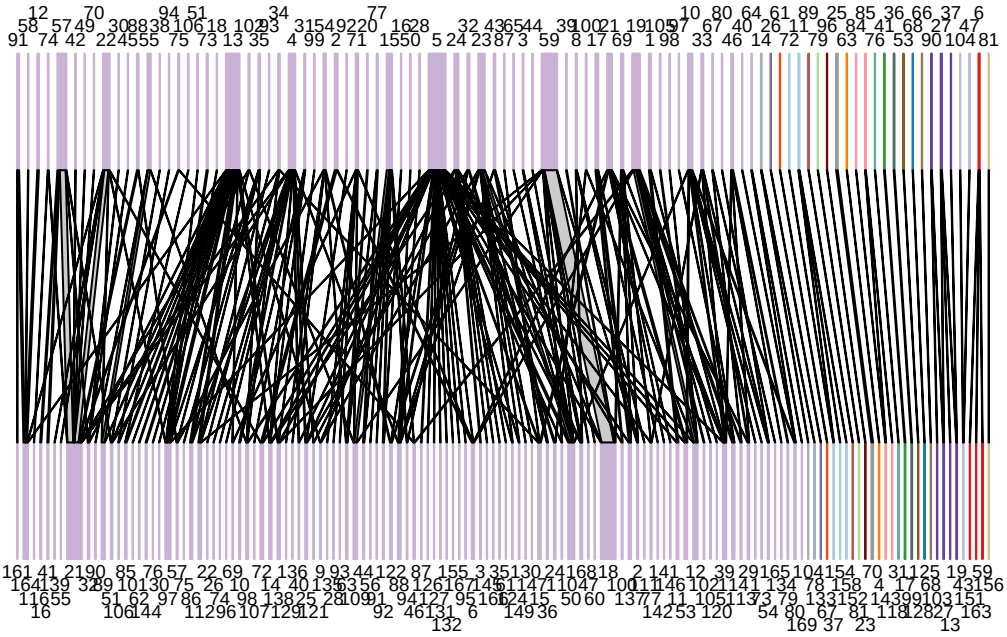
	x
mean.number.of.links.HL	5.0885781
mean.number.of.links.LL	2.8344988
mean.number.of.shared.partners.HL	0.0171471
mean.number.of.shared.partners.LL	0.0299954
cluster.coefficient.HL	0.0256999
cluster.coefficient.LL	0.0171788
weighted.cluster.coefficient.HL	0.0187563
weighted.cluster.coefficient.LL	0.0422438
niche.overlap.HL	0.0071341
niche.overlap.LL	0.0162541
togetherness.HL	0.0009580
togetherness.LL	0.0038503
C.score.HL	0.9867321
C.score.LL	0.9738783
V.ratio.HL	2.7936078
V.ratio.LL	0.8737068
discrepancy.HL	283.0000000
discrepancy.LL	288.0000000
extinction.slope.HL	1.3281663
extinction.slope.LL	1.3735409
robustness.HL	0.5705557
robustness.LL	0.5799338
functional.complementarity.HL	341.8393154
functional.complementarity.LL	339.7511160
partner.diversity.HL	0.9994908
partner.diversity.LL	0.6602675
generality.HL	4.1900761
vulnerability.LL	2.2887487

3.2.2 Comps vs species

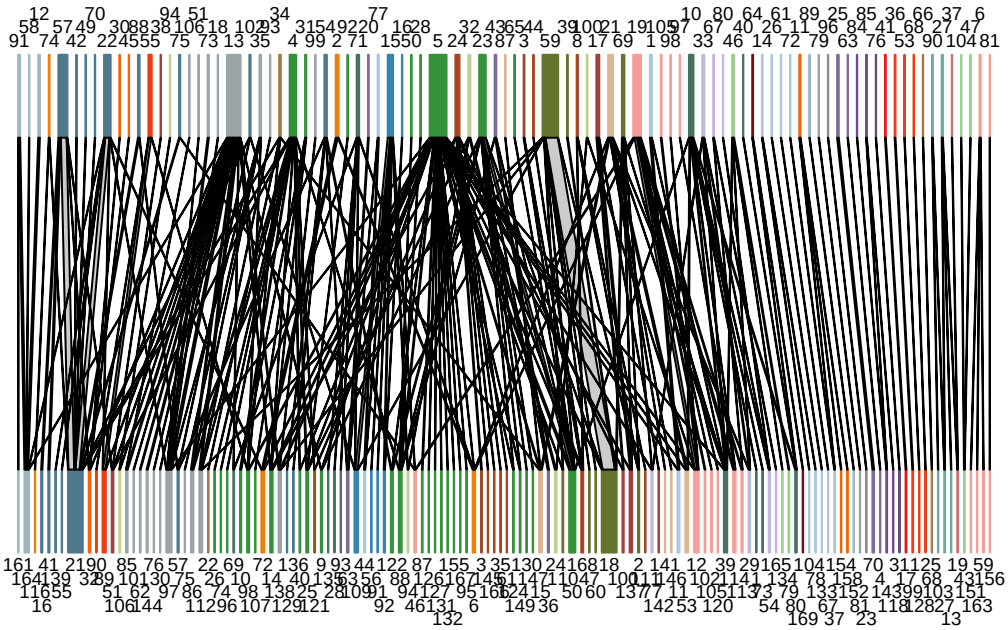
3.2.3 At 99% similarity level

```
table.mating.types<-out.99[,-1]
table.mating.types<-table(table.mating.types[,1],table.mating.types[,2])
table.mating.types<-table.mating.types[-1,-1]
table.mating.types<-table.mating.types[rowSums(table.mating.types)>0,colSums(table.mating.types)>0]
comps<-compart(table.mating.types)
class(table.mating.types)<-NULL
cubo<-as.matrix(table.mating.types)
class(cubo)<-"matrix"
modules99<-metaComputeModules(cubo,N=5, steps=1e+05)
```

```
colorinos_compartments<-colorRampPalette(colorinos.bipolar)(comps$n.compart)
plotweb(modules99@moduleWeb,
        col.high = colorinos_compartments[(apply(comps$cweb,2,min)*-1)[colnames(modules99@moduleWeb)]],
        col.low =  colorinos_compartments[(apply(comps$cweb,1,min)*-1)[rownames(modules99@moduleWeb)]],
        bor.col.high = colorinos_compartments[(apply(comps$cweb,2,min)*-1)[colnames(modules99@moduleWeb)
        bor.col.low =  colorinos_compartments[(apply(comps$cweb,1,min)*-1)[rownames(modules99@moduleWeb)
        bor.col.interaction = 1, y.width.low =0.15, y.width.high = 0.15)
```



```
foo1<-modules99@modules[-1,-c(1,2)]
foo2<-foo1[,(1+dim(modules99@originalWeb)[1]):dim(foo1)[2]]
foo1<-foo1[,1:dim(modules99@originalWeb)[1]]
foo1<-apply(foo1,2,FUN = function(x){which(x!=0)})
names(foo1)<-rownames(modules99@originalWeb)
foo2<-apply(foo2,2,FUN = function(x){which(x!=0)})
names(foo2)<-colnames(modules99@originalWeb)

colorinos_modules<-colorRampPalette(colorinos.bipolar)(dim(modules99@modules)[1]-1)

plotweb(modules99@moduleWeb,
        col.high = colorinos_modules[foo2[colnames(modules99@moduleWeb)]],
        col.low =  colorinos_modules[foo1[rownames(modules99@moduleWeb)]],
        bor.col.high = colorinos_modules[foo2[colnames(modules99@moduleWeb)]],
        bor.col.low =  colorinos_modules[foo1[rownames(modules99@moduleWeb)]],
        bor.col.interaction = 1, y.width.low =0.1, y.width.high = 0.1)
```



```
out.99.bckup<-out.99
out.99<-cbind(out.99,0,0,0)
out.99<-out.99[,-1]
colnames(out.99)<-c("alpha","hmg","compartments","modules1","modules2")

for (i in 1:length(rownames(out.99)))
    {
        if (out.99[i,1]!="0"&out.99[i,2]!="0")
        {
        out.99[i,3]<-comps$cweb[out.99[i,1],out.99[i,2]]
            }
        else if (out.99[i,1]!="0"&out.99[i,1]%in%rownames(comps$cweb))
        {
        out.99[i,3]<-min(as.numeric(comps$cweb[out.99[i,1],]))
        }
        else if (out.99[i,2]!="0"&out.99[i,2]%in%colnames(comps$cweb))
        {
        out.99[i,3]<-min(as.numeric(comps$cweb[,out.99[i,2]]))
        }
}
out.99[,4]<-foo1[as.character(out.99[,1])]
out.99[,5]<-foo1[as.character(out.99[,2])]



#out.99[is.na(out.99[,4]),4]<-foo2[as.character(out.99[,2])][is.na(out.99[,4])]
out.99[,3]<-as.numeric(out.99[,3])*-1
out.99[,4]<-as.numeric(out.99[,4])
out.99[,5]<-as.numeric(out.99[,5])
#
#
rownames(out.99)[duplicated(rownames(out.99))]<-paste(rownames(out.99)[duplicated(rownames(out.99))],".
```

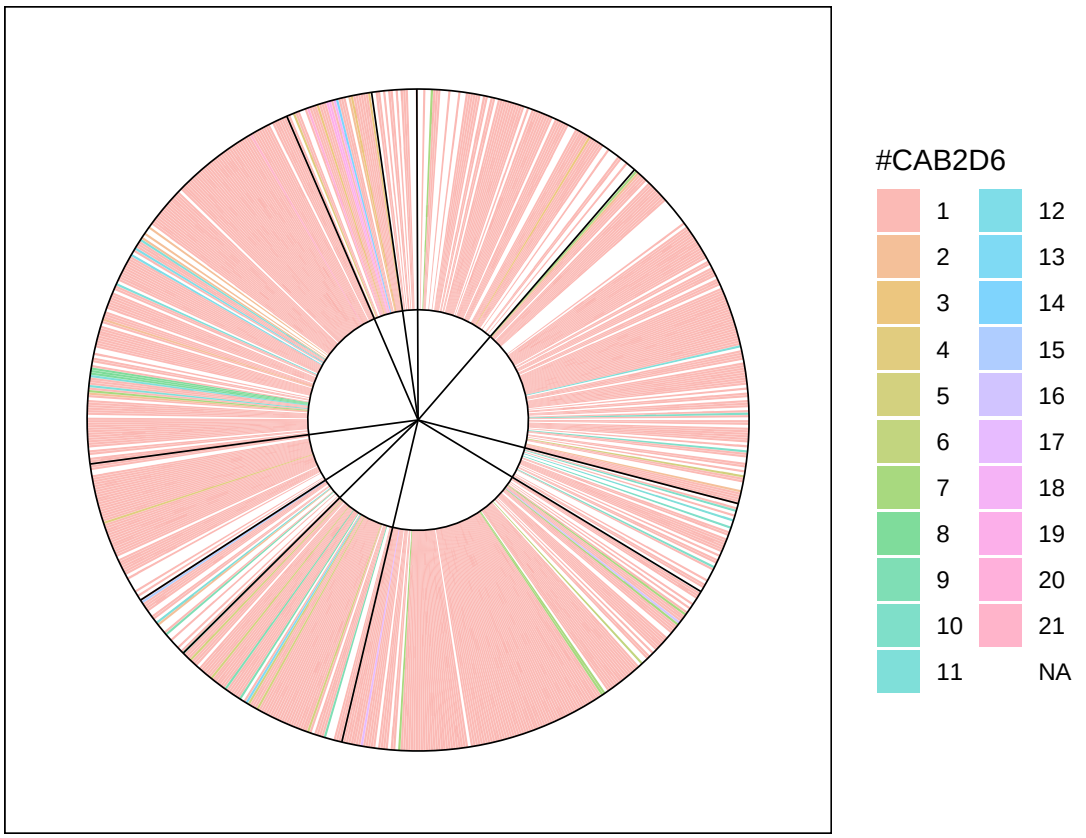
```
rownames(out.99)[duplicated(rownames(out.99))]<-gsub("\\.1","\\.2",rownames(out.99)[duplicated(rownames
rownames(out.99)[duplicated(rownames(out.99))]<-gsub("\\.2","\\.3",rownames(out.99)[duplicated(rownames
rownames(out.99)[duplicated(rownames(out.99))]<-gsub("\\.3","\\.4",rownames(out.99)[duplicated(rownames
#
```

```
foo_modules<-mat.or.vec(length(orden.foo),3)
rownames(foo_modules)<-orden.foo
colnames(foo_modules)<-colnames(out.99)[3:5]
foo_reduced<-out.99[rownames(out.99)%in%orden.foo,3:5]
foo_modules[rownames(foo_reduced),]<-foo_reduced
rm(foo_reduced)
#
#
#
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo, foo_modules[orden.foo,])
foo<-melt(foo[,c(1,26)])
```

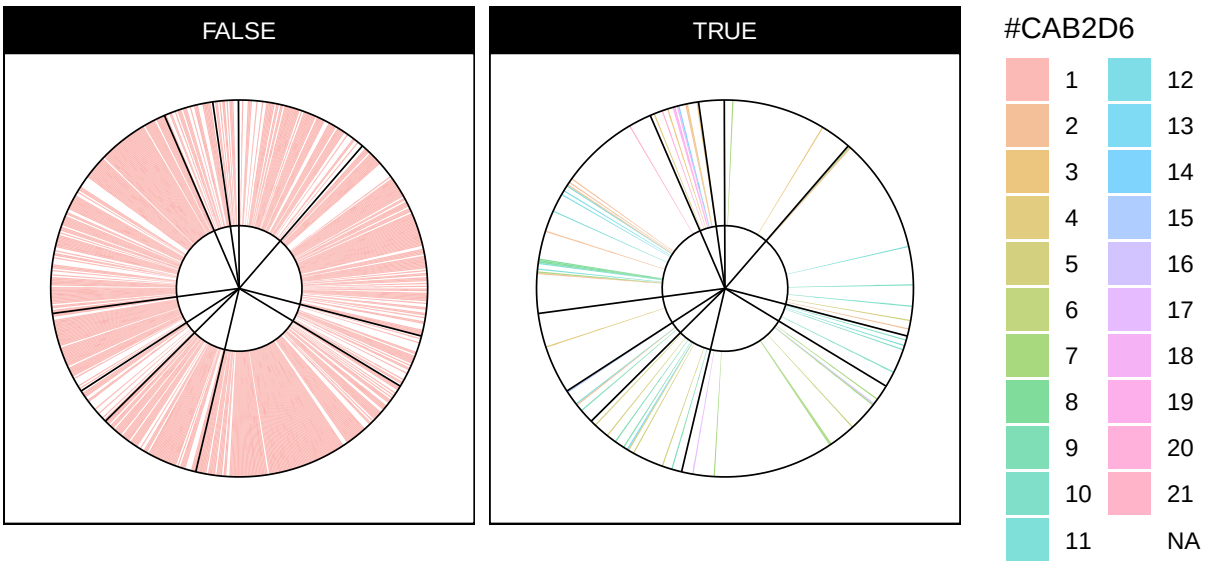
3.2.3.1 Donut plot compartments

```
## Using orden, compartments as id variables
```

```
foo$compartments[foo$compartments==0]<-NA
foo$compartments<-factor(foo$compartments,levels=1:comps$n.compart)
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=compartments),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos_compartments,na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```




```
print(p + facet_wrap(vars((compartments!=1|is.na(compartments)))))
```

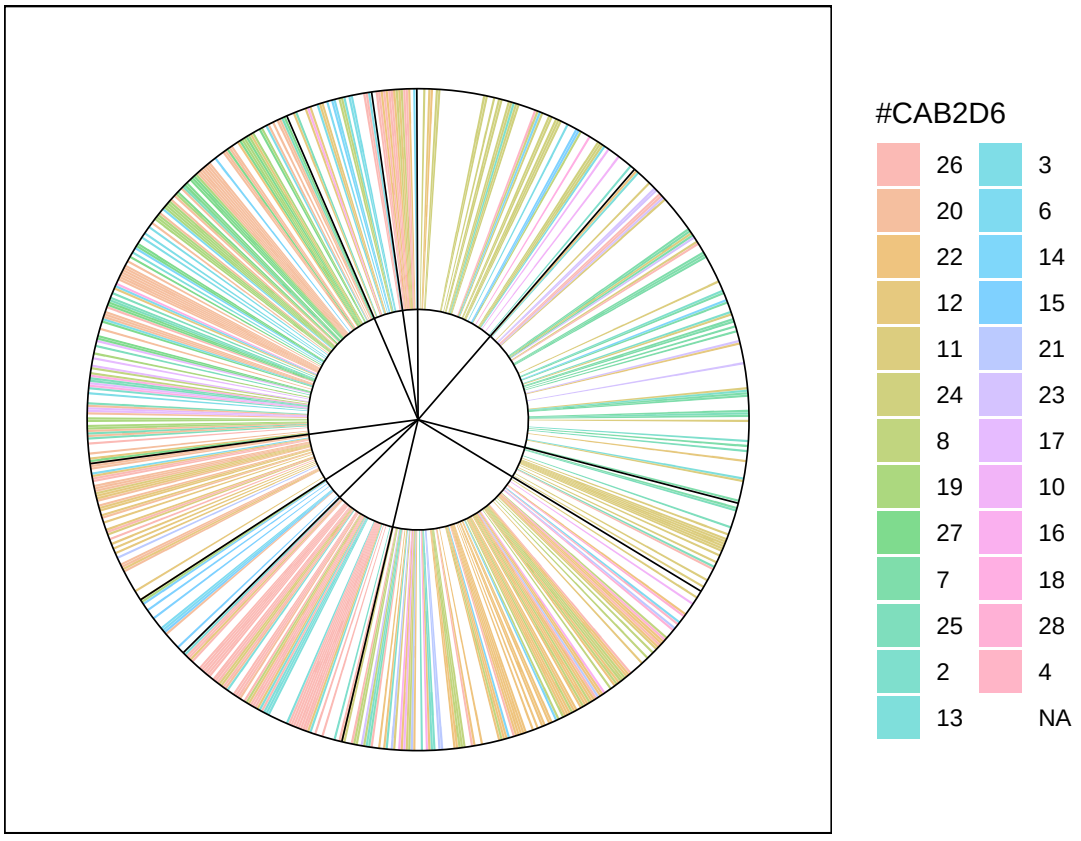



```
foo<-condensed_baps_adm[orden.foo,]
foo<-cbind(orden=factor(c(1:dim(foo)[1])),foo, foo_modules[orden.foo,])
foo<-melt(foo[,c(1,28)])
```

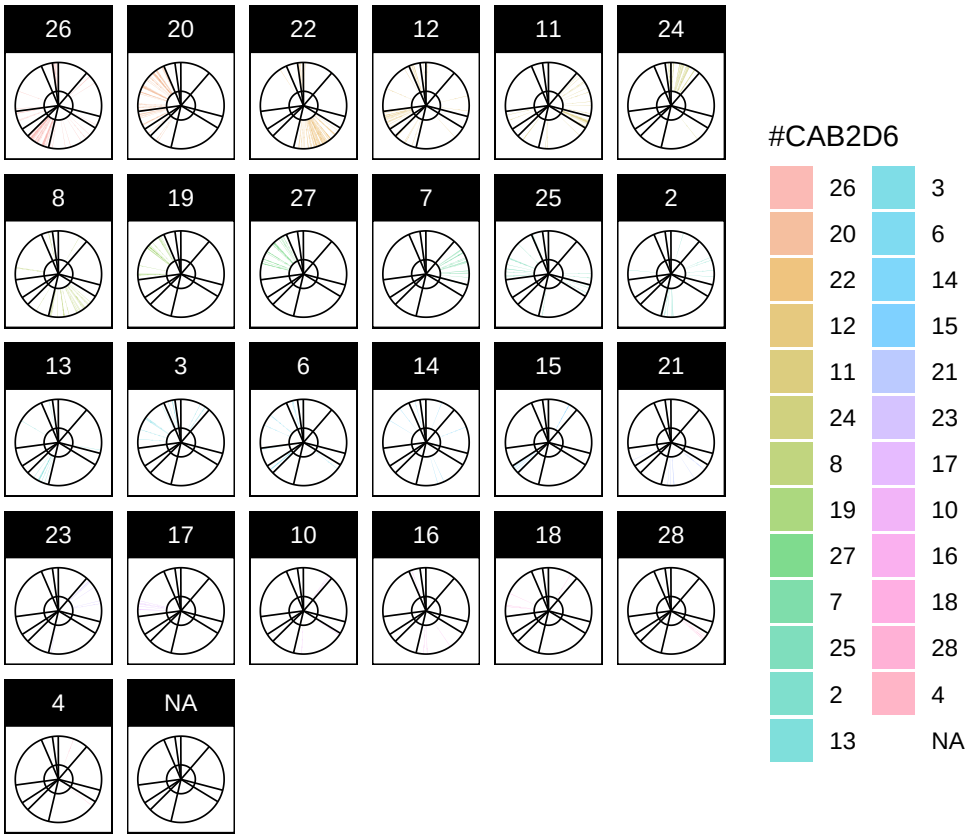
3.2.3.2 Donut plots Modules

```
## Using orden, modules2 as id variables
```

```
foo$modules2[foo$modules2==0]<-NA
foo$modules2<-factor(foo$modules2,levels=names(table(foo$modules2))[order(as.numeric(table(foo$modules2)
  p<-ggplot(foo) +
  geom_bar(aes(x=orden, y=1,fill=modules2),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos_modules,na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
   geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + coord_polar()
print(p)
```



```
print(p + facet_wrap(vars(modules2)))
```




```
kable(networklevel(table.mating.types))
```

	x
connectance	0.0218972
web asymmetry	-0.1982379
links per species	1.1938326
number of compartments	21.0000000
compartment diversity	3.4511332
cluster coefficient	0.0073529
modularity Q	0.7153134
nestedness	2.2983241
NODF	5.8353085
weighted nestedness	0.3378359
weighted NODF	1.6330296
interaction strength asymmetry	-0.0526979
specialisation asymmetry	0.0752745
linkage density	5.0572869
weighted connectance	0.0222788
Fisher alpha	315.9472254
Shannon diversity	5.2507149
interaction evenness	0.5571929
Alatalo interaction evenness	0.4894535
H2	0.4619052
number.of.species.HL	91.0000000
number.of.species.LL	136.0000000

	x
mean.number.of.shared.partners.HL	0.0759463
mean.number.of.shared.partners.LL	0.1172113
cluster.coefficient.HL	0.0648567
cluster.coefficient.LL	0.0432900
weighted.cluster.coefficient.HL	0.1161349
weighted.cluster.coefficient.LL	0.2020986
niche.overlap.HL	0.0191771
niche.overlap.LL	0.0431530
togetherness.HL	0.0027480
togetherness.LL	0.0114261
C.score.HL	0.9553825
C.score.LL	0.9141449
V.ratio.HL	6.1869741
V.ratio.LL	1.3942149
discrepancy.HL	210.0000000
discrepancy.LL	215.0000000
extinction.slope.HL	1.5362135
extinction.slope.LL	1.7249732
robustness.HL	0.6057715
robustness.LL	0.6313006
functional.complementarity.HL	285.5111560
functional.complementarity.LL	306.2279211
partner.diversity.HL	1.4568297
partner.diversity.LL	0.9011428
generality.HL	7.1765847
vulnerability.LL	2.9379890

```
kable(grouplevel(table.mating.types))
```

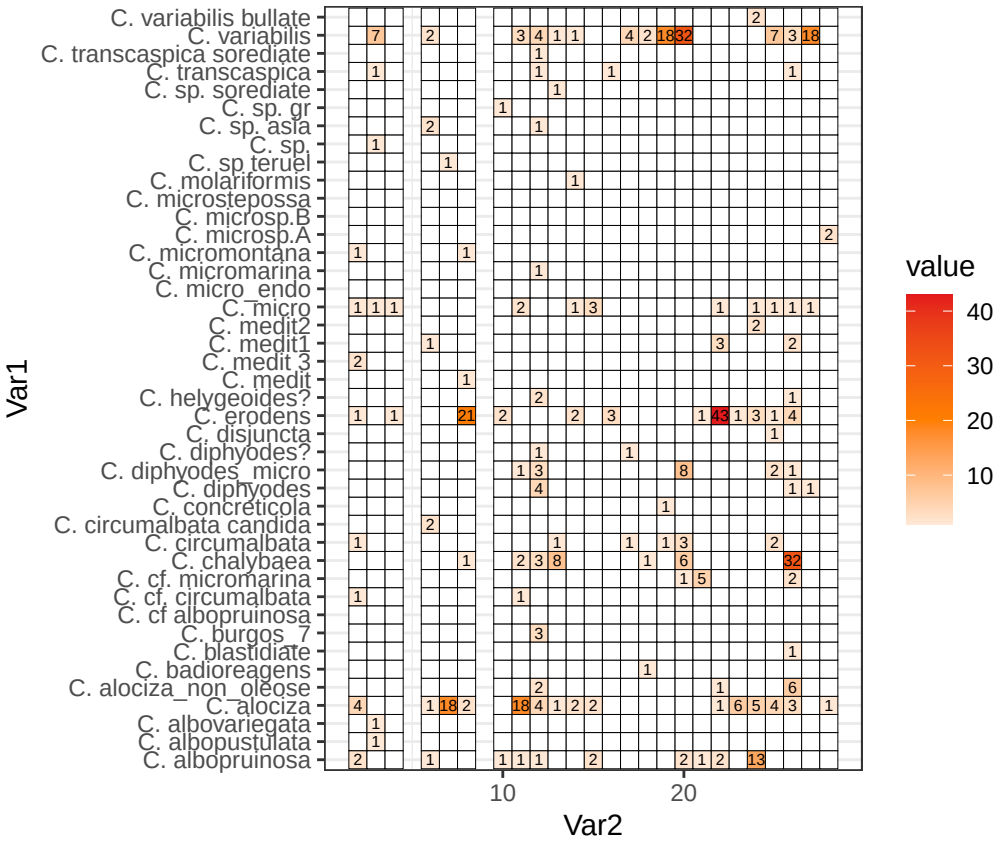
	x
number.of.species.HL	91.0000000
number.of.species.LL	136.0000000
mean.number.of.links.HL	8.8205128
mean.number.of.links.LL	3.9393939
mean.number.of.shared.partners.HL	0.0759463
mean.number.of.shared.partners.LL	0.1172113
cluster.coefficient.HL	0.0648567
cluster.coefficient.LL	0.0432900
weighted.cluster.coefficient.HL	0.1161349
weighted.cluster.coefficient.LL	0.2020986
niche.overlap.HL	0.0191771
niche.overlap.LL	0.0431530
togetherness.HL	0.0027480
togetherness.LL	0.0114261
C.score.HL	0.9553825
C.score.LL	0.9141449
V.ratio.HL	6.1869741
V.ratio.LL	1.3942149
discrepancy.HL	210.0000000
discrepancy.LL	215.0000000
extinction.slope.HL	1.5262411

	x
extinction.slope.LL	1.7328575
robustness.HL	0.6033901
robustness.LL	0.6329664
functional.complementarity.HL	285.5111560
functional.complementarity.LL	306.2279211
partner.diversity.HL	1.4568297
partner.diversity.LL	0.9011428
generality.HL	7.1765847
vulnerability.LL	2.9379890

```
#save.image('/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v16final.Rdata')
```

```
# Heatmap 3 Species vs modules

foo_tabla<-melt(table(condensed_baps_adm[as.character(rownames(foo)),2],foo$modules))
foo_tabla$value[foo_tabla$value==0]<-NA


  ggplot(foo_tabla,aes(x=Var2,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = v
    scale_fill_gradient2(low = "#FFFFFF",
  mid = colorinos.bipolar[7],
  high = colorinos.bipolar[6],
  midpoint = 20,
  space = "Lab",
  na.value = "#FFFFFF",
  guide = "colourbar",
  aesthetics = "fill"
) +
  coord_fixed() + theme_bw()
```

```
## Warning: Removed 925 rows containing missing values (geom_text).
```




```
write.table(cbind(condensed_baps_adm,foo_bgmyc_clusters[rownames(condensed_baps_adm)],out.99[rownames(co
```

3.3 II.III Networks at individual level

```
library(reshape2)
library(ggplot2)
library(fpc)
library(igraph)
library(robin)
library(scales)
```

```
##
## Attaching package: 'scales'
```

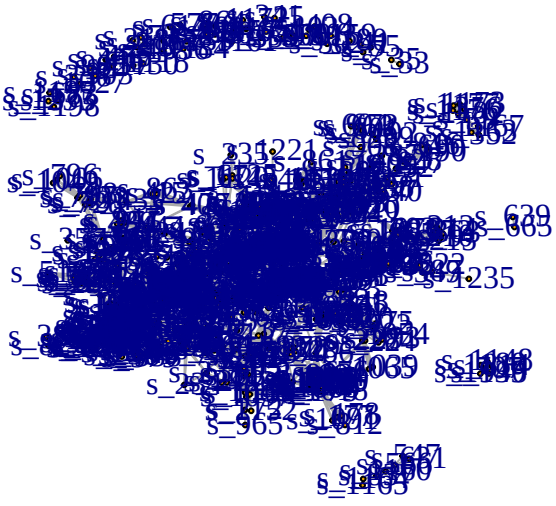
```
## The following object is masked from 'package:geiger':
##
##     rescale
```

```
#
# Create network of 99% similarity sequence clusters
#
network_indivs<-mat.or.vec(length(rownames(out.99)[!duplicated(rownames(out.99))]),length(rownames(out.9
rownames(network_indivs)<-rownames(out.99)[!duplicated(rownames(out.99))]
colnames(network_indivs)<-rownames(out.99)[!duplicated(rownames(out.99))]
names_foo_network<-sapply(strsplit(rownames(out.99),"\\."),`[`,1)
for (foo.levels in levels(factor(out.99[,1]))[-1])
{
  foo_select<-names_foo_network[out.99[,1]==foo.levels]
```

```
  network_indivs[foo_select,foo_select]<-1
}
for (foo.levels in levels(factor(out.99[,2]))[-1])
{
  foo_select<-names_foo_network[out.99[,2]==foo.levels]
  network_indivs[foo_select,foo_select]<-network_indivs[foo_select,foo_select]+1
}
rownames(network_indivs)<-paste("s",rownames(network_indivs),sep="_")
colnames(network_indivs)<-paste("s",colnames(network_indivs),sep="_")
network_indivs<-network_indivs[!(rowSums(network_indivs)==diag(network_indivs)),!(rowSums(network_indivs
network_indivs[network_indivs==2]<-1
network_indivs[network_indivs==3]<-1
uni_graph<-igraph::graph_from_adjacency_matrix(network_indivs, mode ="undirected",diag=FALSE)
```

3.3.1 Compartmentalization

```
#
# Components
#
components<-igraph::components(uni_graph)
#
# Communities
#
communities_mat<-igraph::communities(components)
#
# Plot Graph with BAPS assignments
#
plot.igraph(uni_graph,layout=layout_with_fr(uni_graph),vertex.size=2, edge.arrow.size=0.2)
```



3.3.2 Test modularity algorythms

```
uni_graph2<-prepGraph(file=uni_graph, file.format="igraph")
attributes<-vertex_attr(uni_graph2, index = V(uni_graph2))
set.seed(10)
VI_In<-list()
comp<-list()
```

```r
for (DA in c("walktrap", "louvain", "labelProp", "infomap"))
{
  members_In <- membershipCommunities(graph=uni_graph, method=DA)
  VI_In[[DA]] <- compare(components$membership, members_In, method="vi")
  for (DA1 in c("walktrap", "louvain", "labelProp", "infomap"))
  {
    if(DA!=DA1) {
      comp[[paste(DA,DA1,sep="_")]]<-robinCompare(graph=uni_graph2,method1 = DA, method2 = DA1, measure
    }
  }
}
```

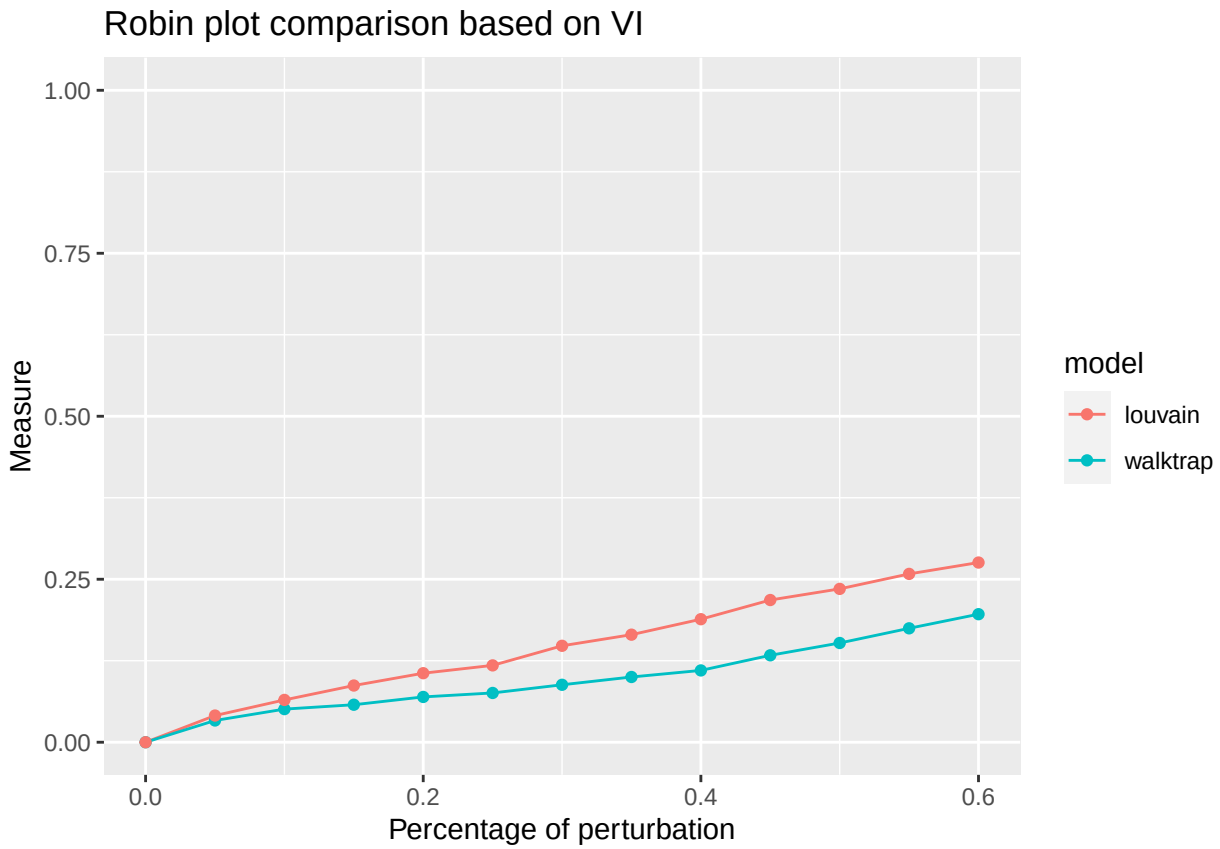
```
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
```

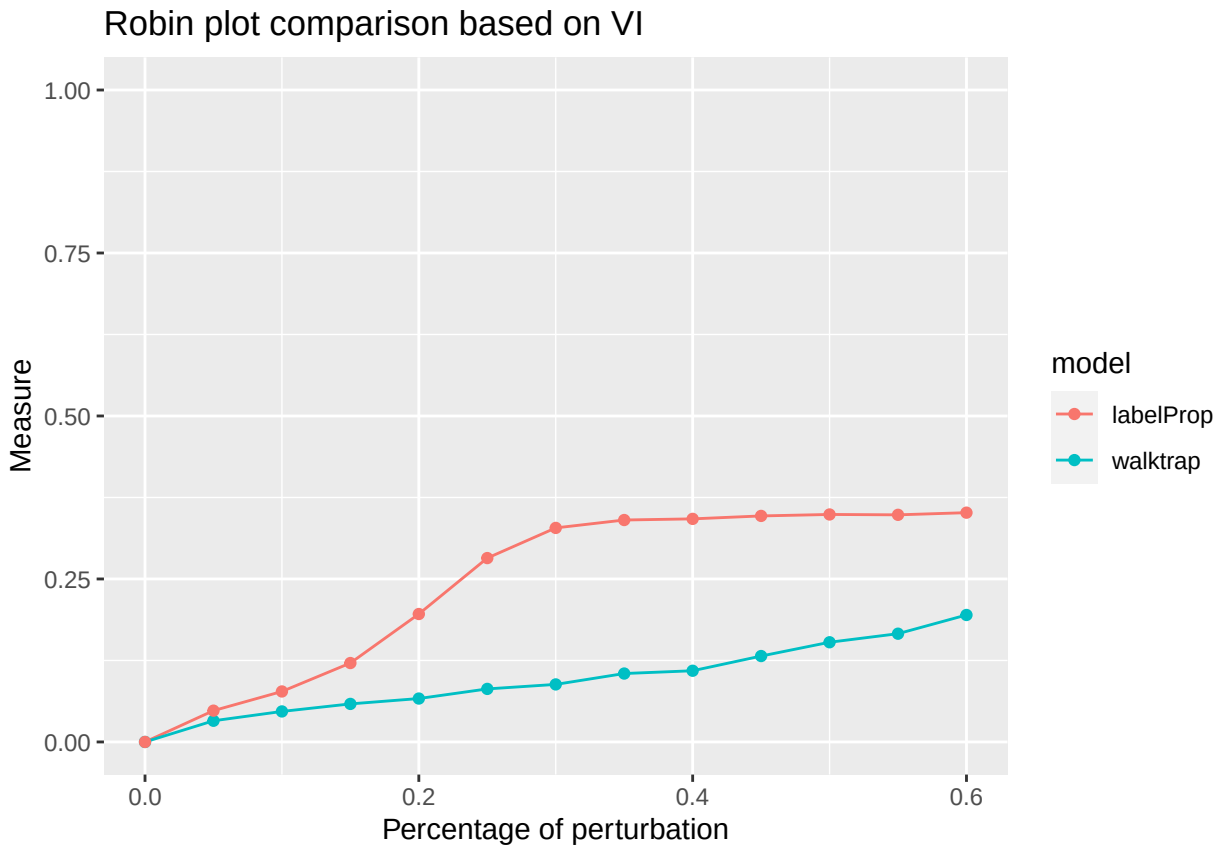
```
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
```

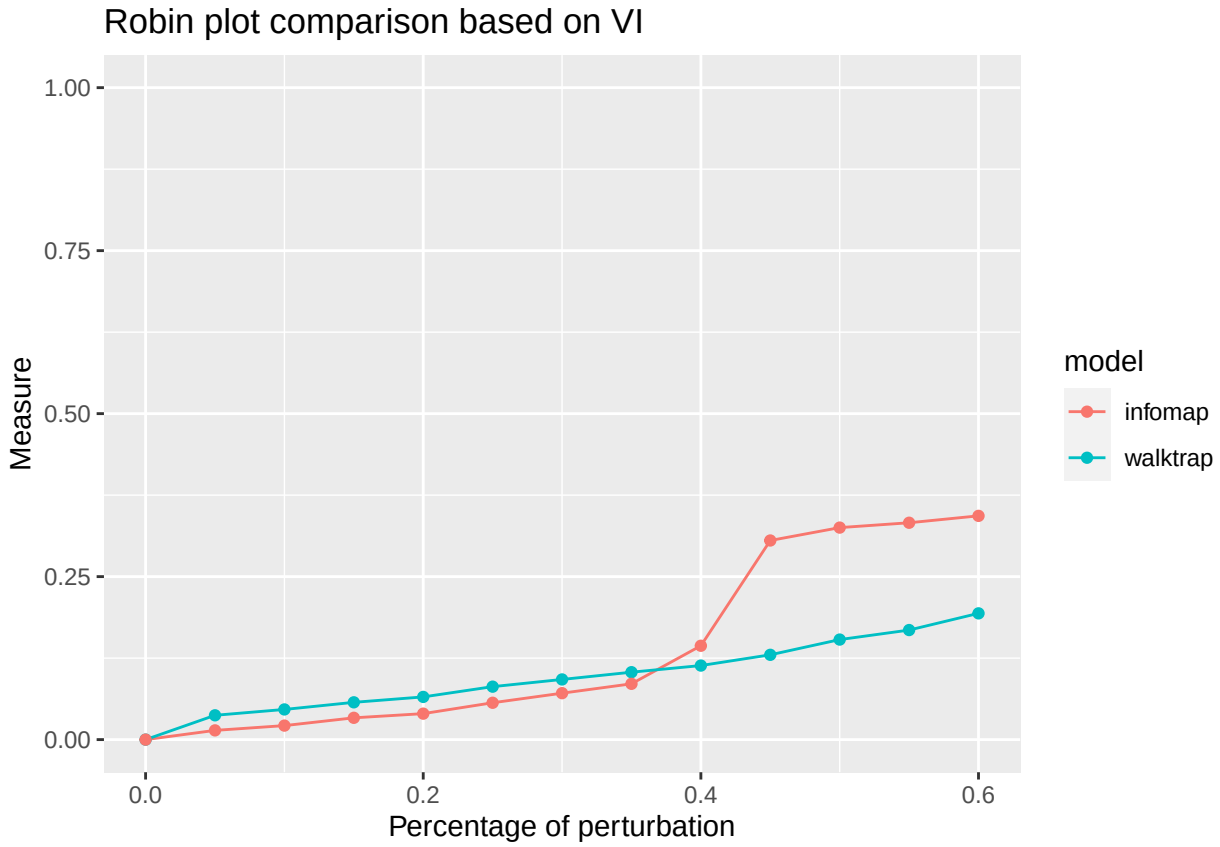
```
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
```

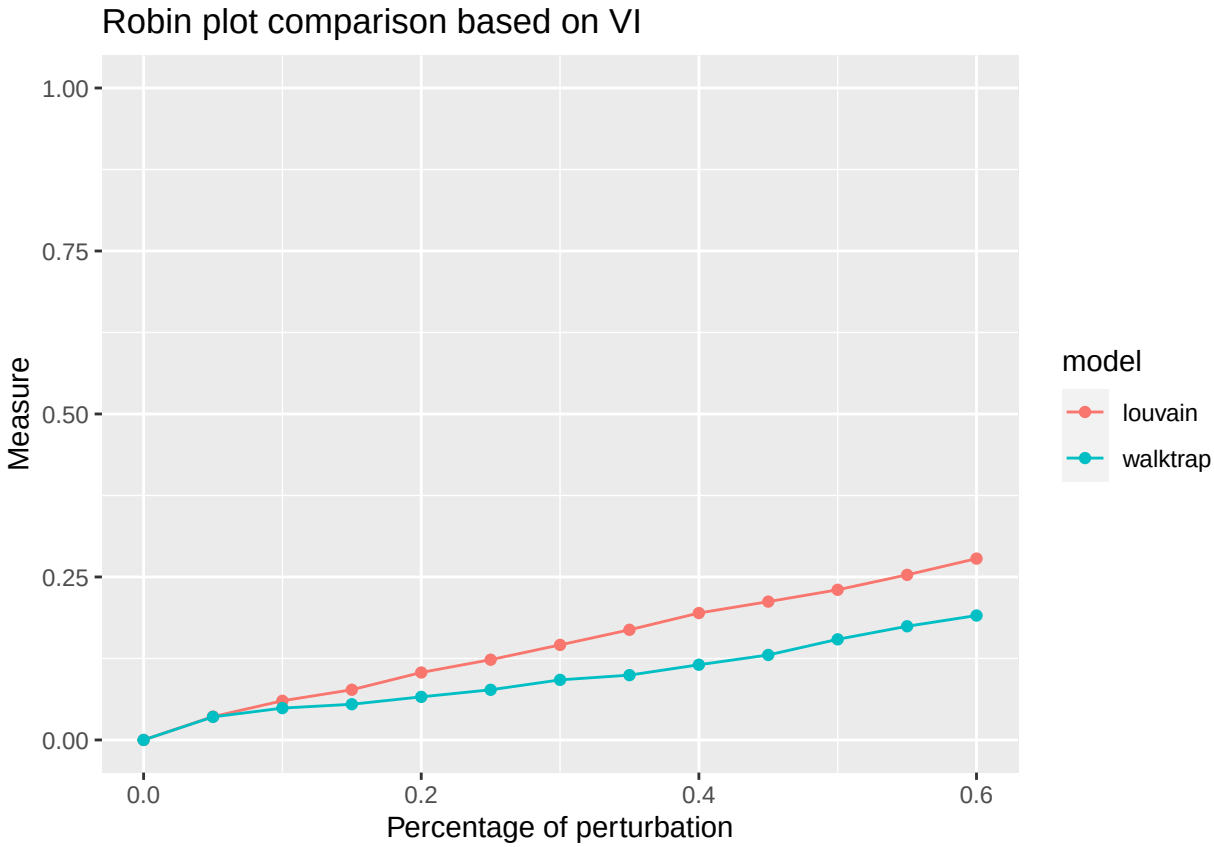
```
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
```

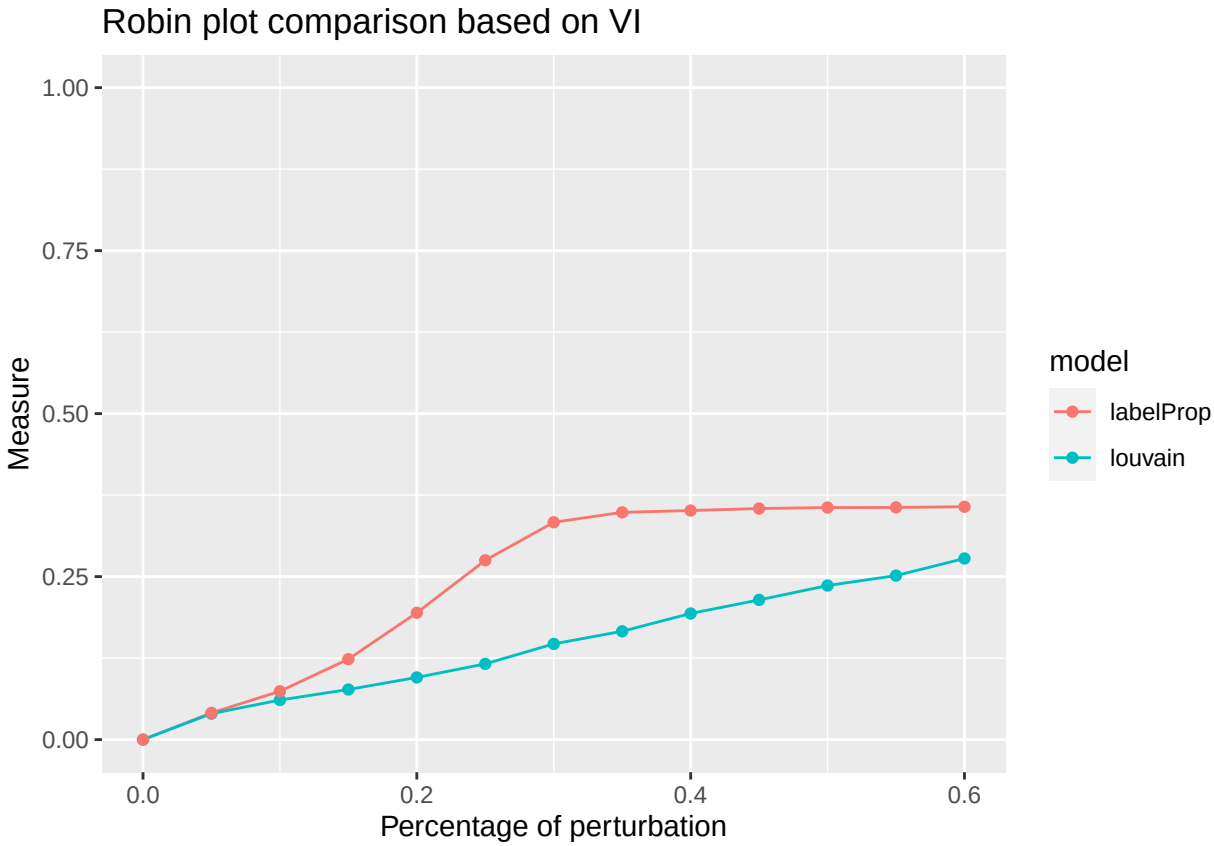
```r
for (i in 1:length(comp))
  {
  comp_foo<-comp[[i]]
  foo_bar1<-sapply(strsplit(names(comp)[i],"_"),`[`,1)
  foo_bar2<-sapply(strsplit(names(comp)[i],"_"),`[`,2)
  print(plotRobin(uni_graph2, model1=comp_foo$Mean1, model2=comp_foo$Mean2, legend = c(foo_bar1,foo_bar
                  title = "Robin plot comparison based on VI"))
}
```

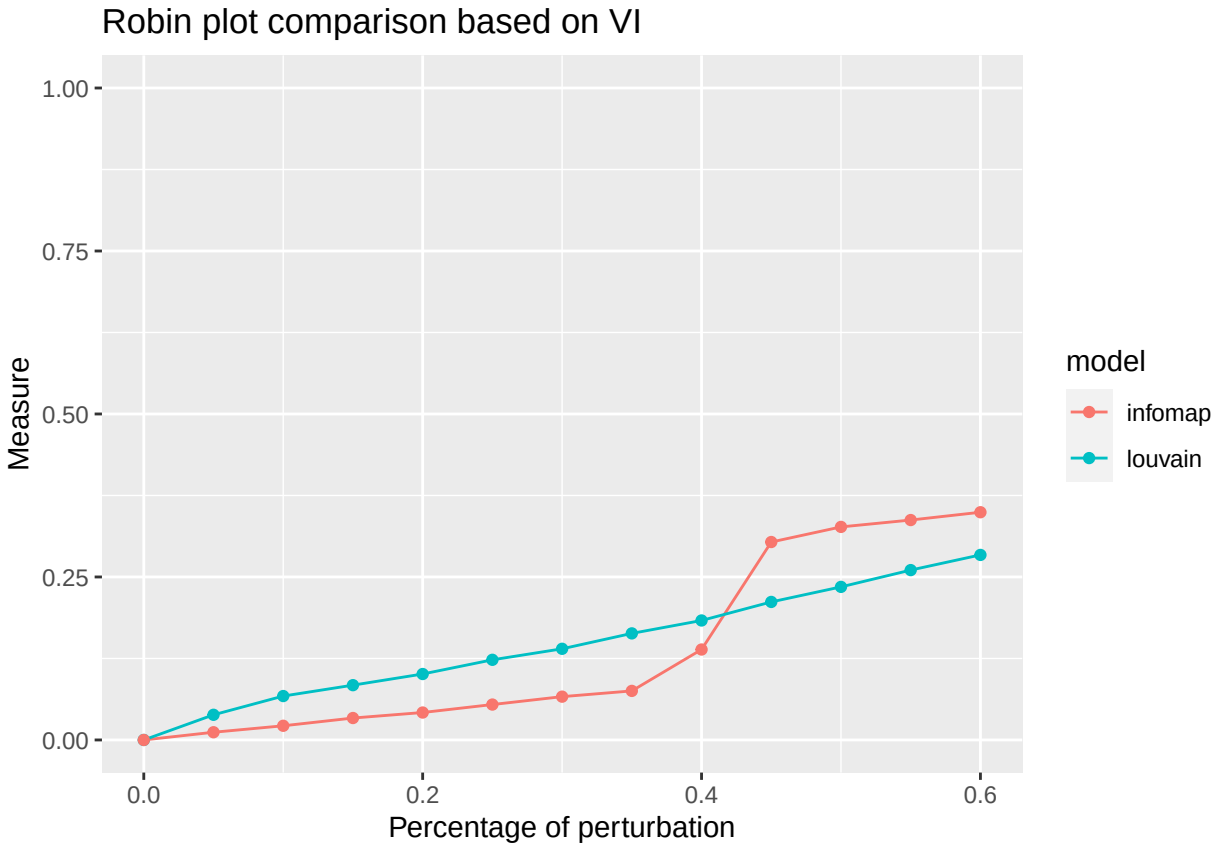


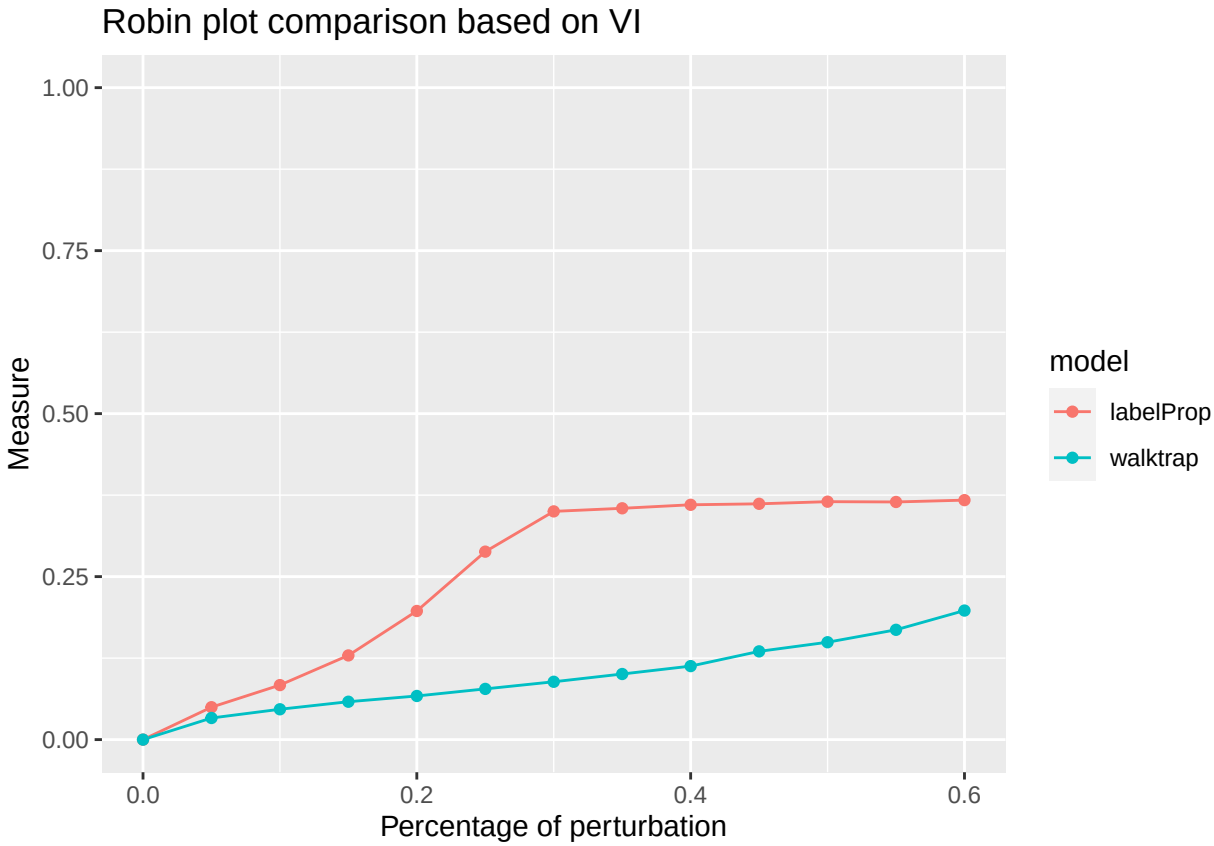

Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
walktrap

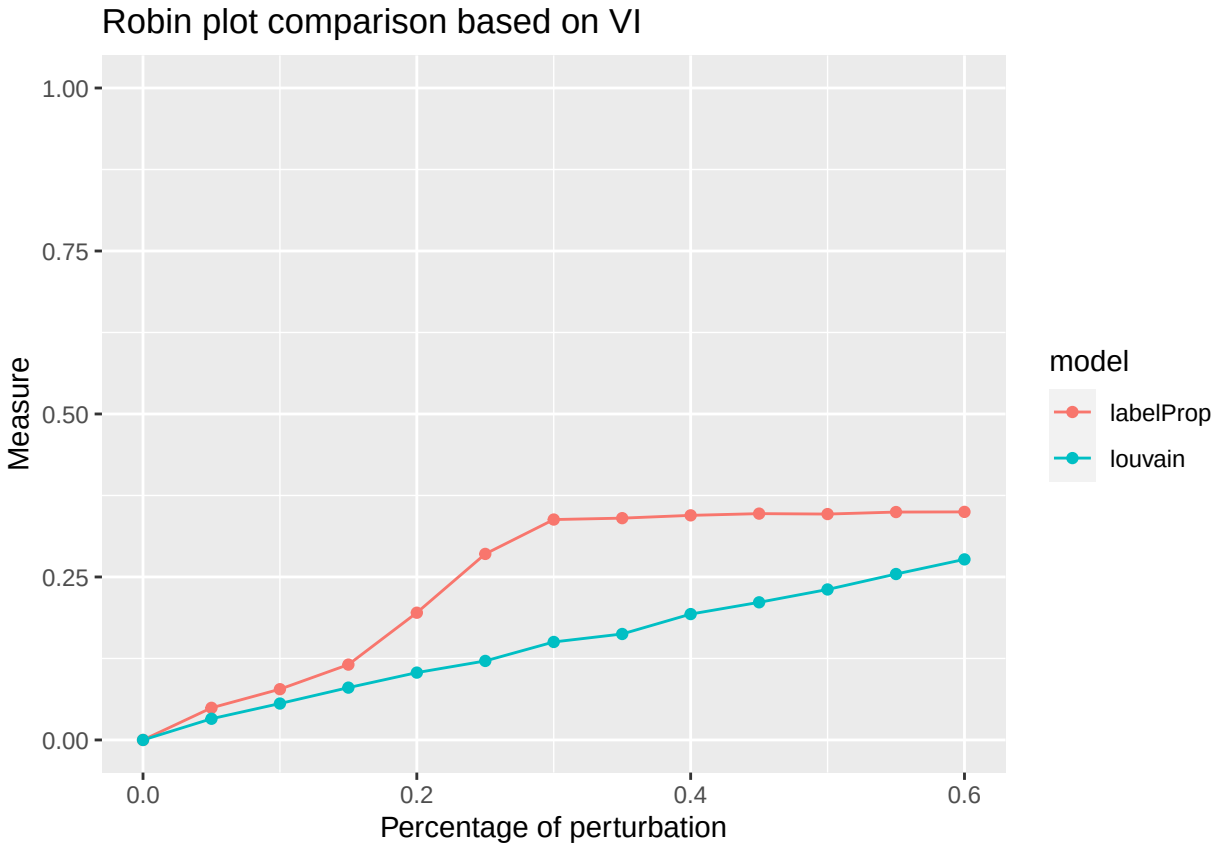

Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
walktrap


Robin plot comparison based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
louvain
walktrap

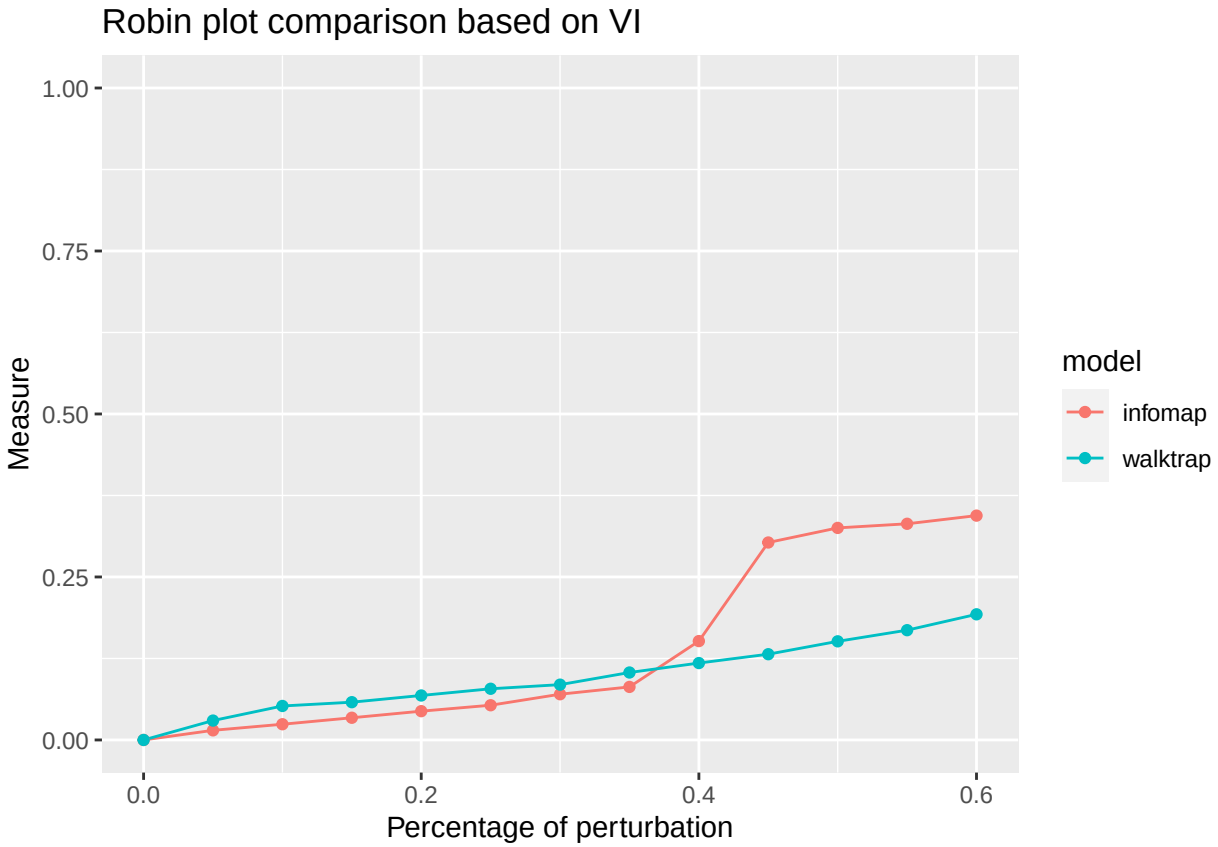

Robin plot comparison based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
louvain

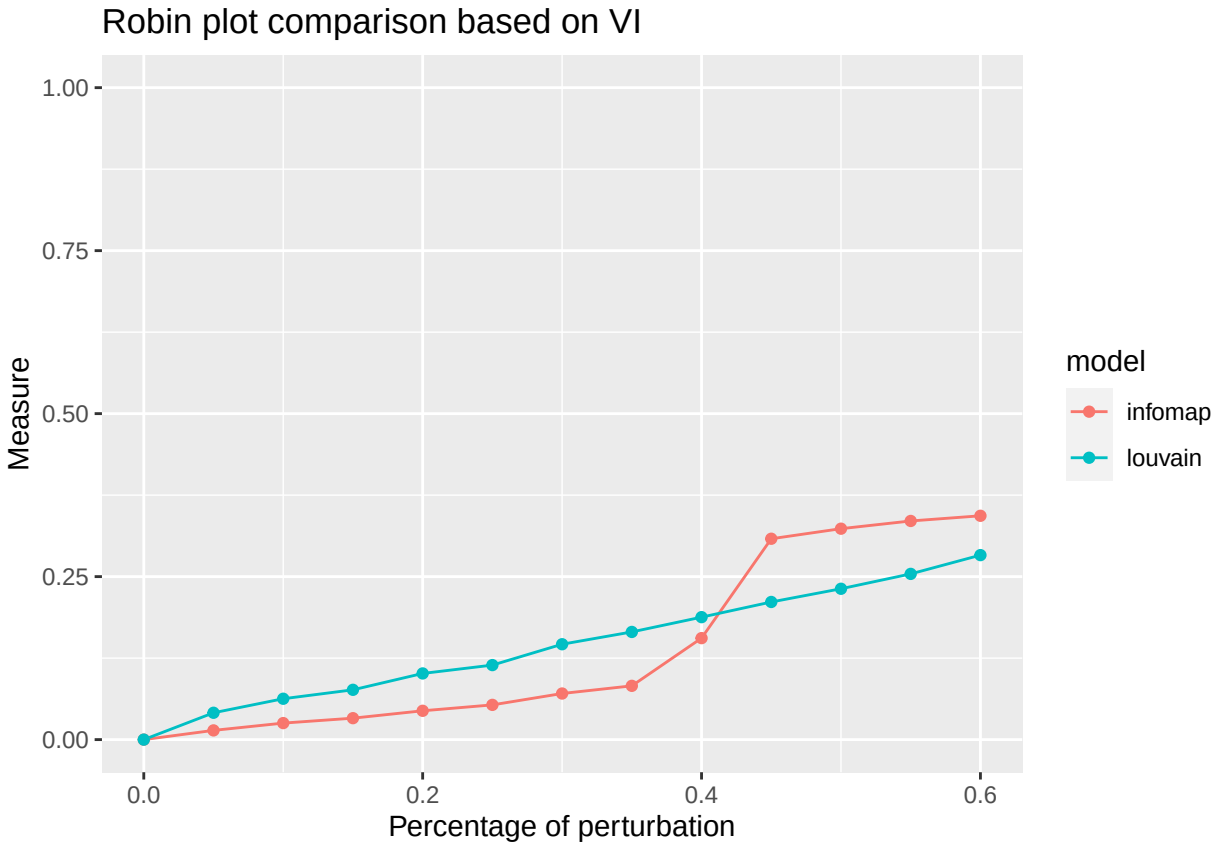

Robin plot comparison based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
louvain

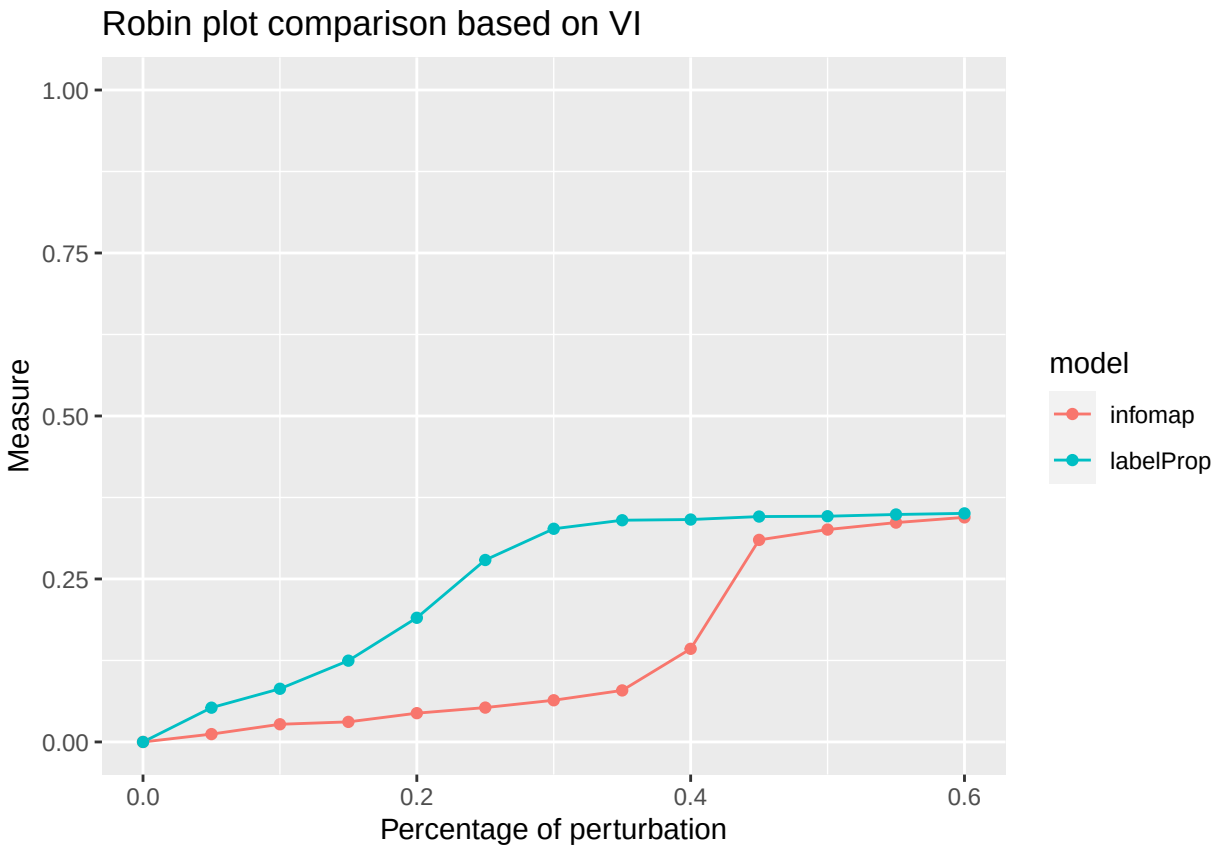

Robin plot comparison based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
walktrap


Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
louvain


Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
labelProp


Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
walktrap


Robin plot comparison based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
louvain



3.3.3 Model selection based on BF

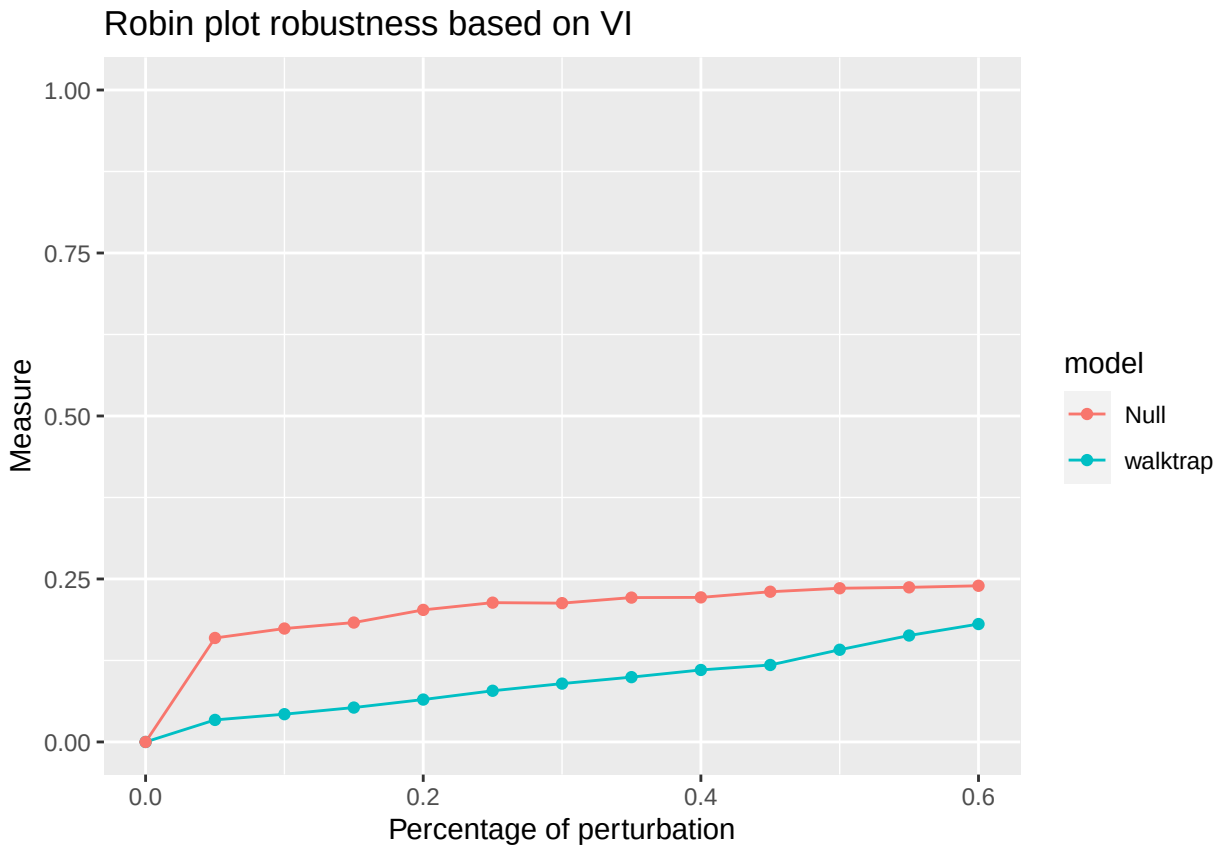
```
graphRandomCM <- random(graph=uni_graph2)
proc_CM<-list()
for (DA in c("walktrap", "louvain", "labelProp", "infomap"))
{
  proc_CM[[DA]] <- robinRobust(graph=uni_graph2, graphRandom=graphRandomCM,
                        measure="vi", method=DA, type="independent")
}
```

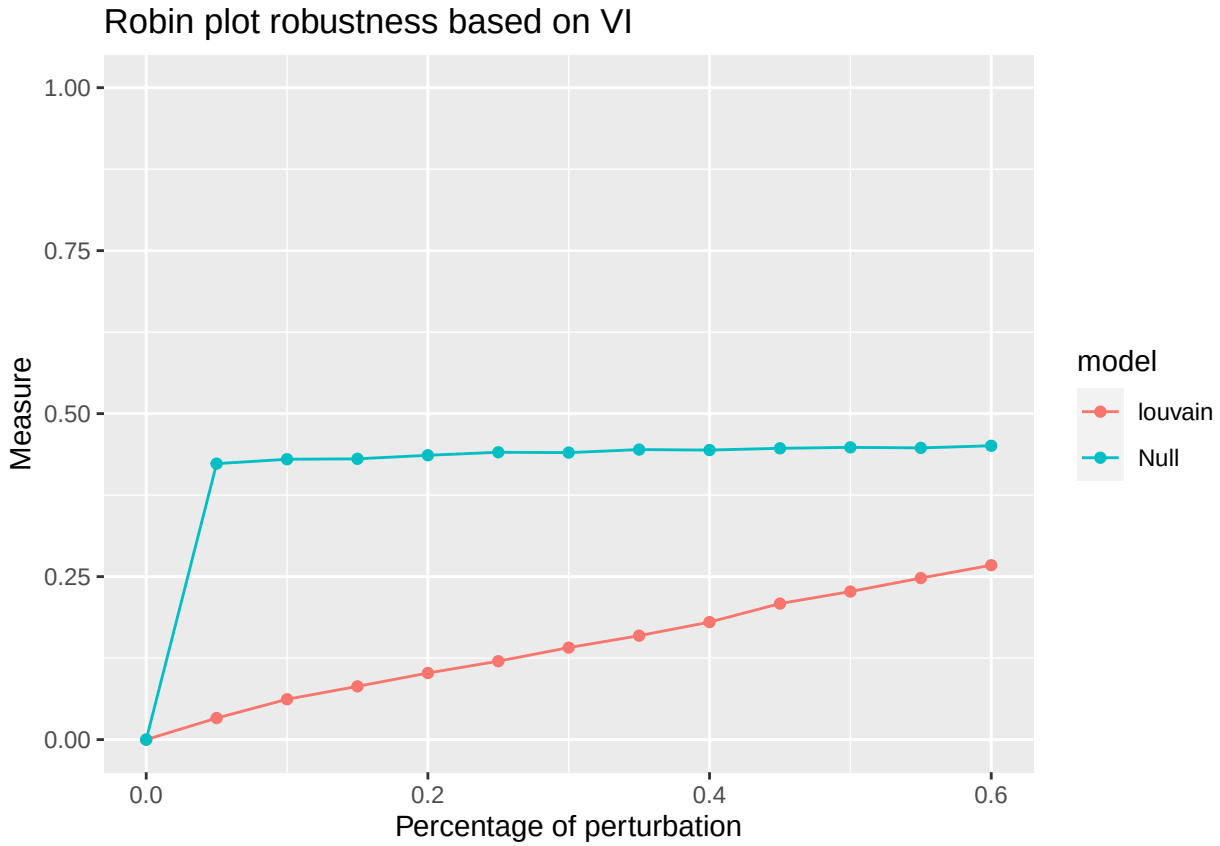
3.3.3.1 Simple random model

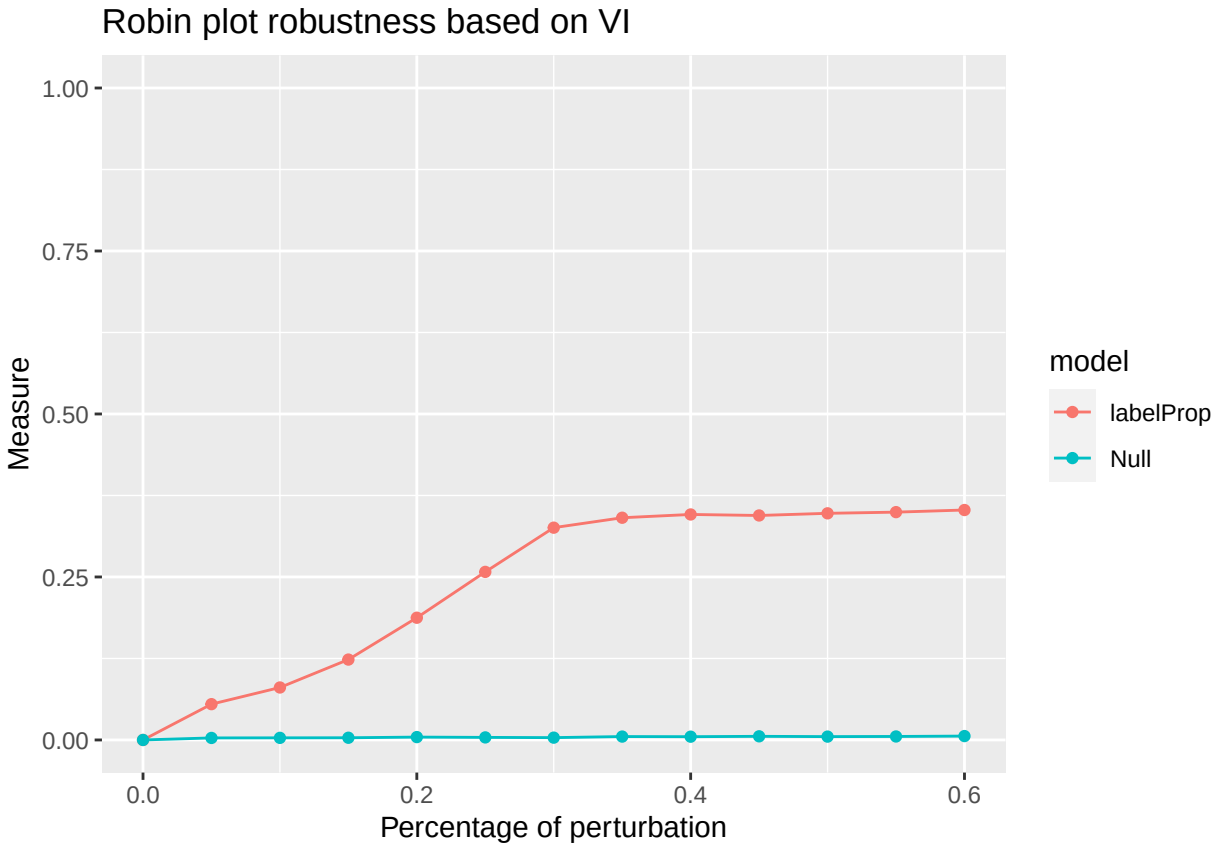
```
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
```

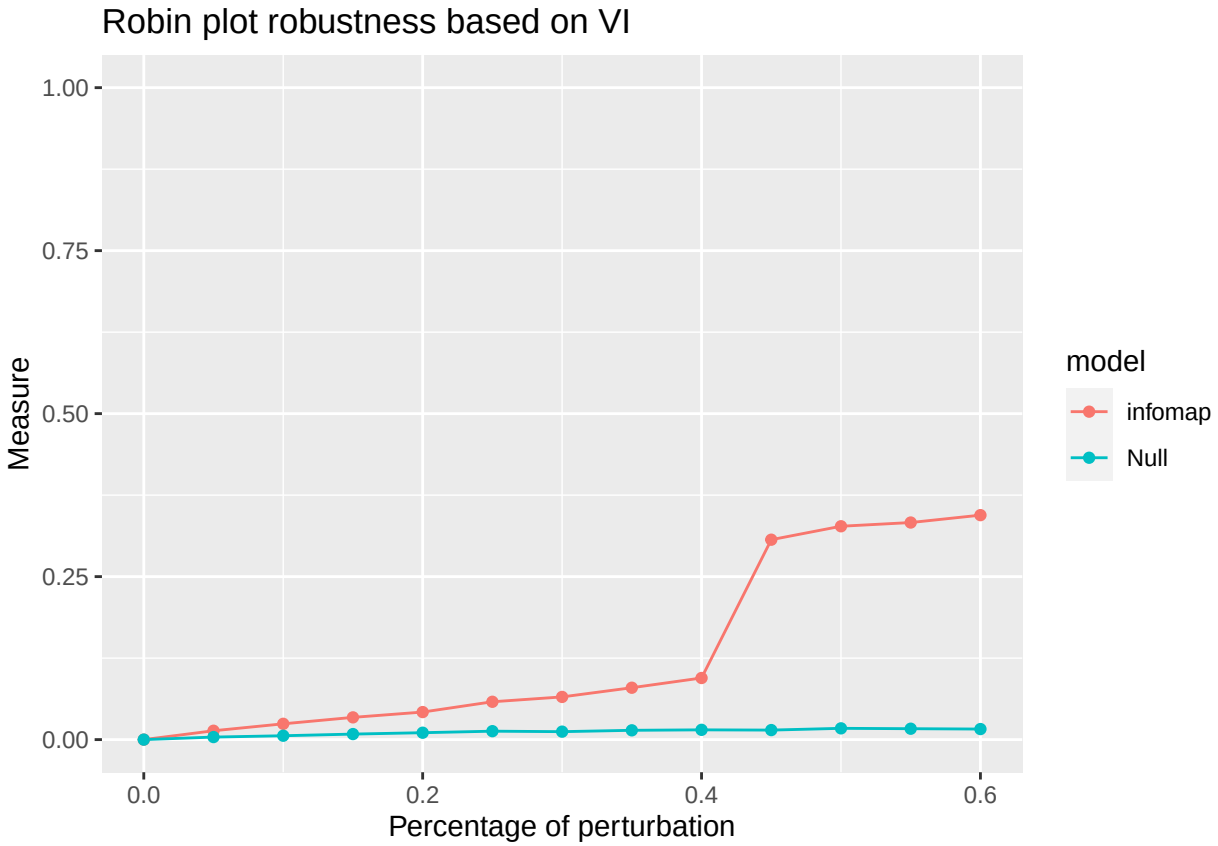
```
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
```

```
for (i in 1:length(proc_CM))
{
  comp_foo<-proc_CM[[i]]
  print(plotRobin(uni_graph2, model1=comp_foo$Mean, model2=comp_foo$MeanRandom, legend = c(names(proc_CM
                  title = "Robin plot robustness based on VI"))
}
```


Robin plot robustness based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
Null
walktrap

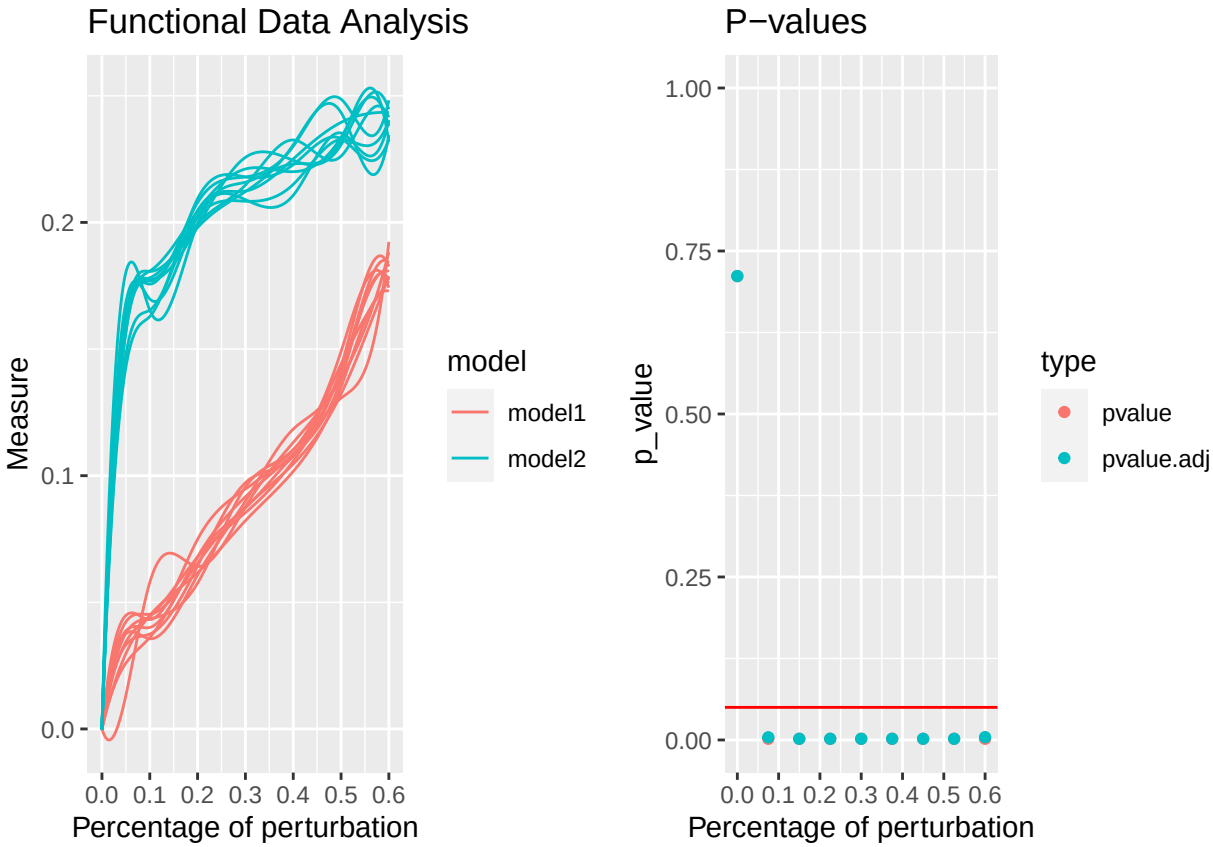

Robin plot robustness based on VI
1.00
0.75
0.50
0.25
0.00
Measure
0.0
0.2
0.4
0.6
Percentage of perturbation
model
louvain
Null


Robin plot robustness based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
Null

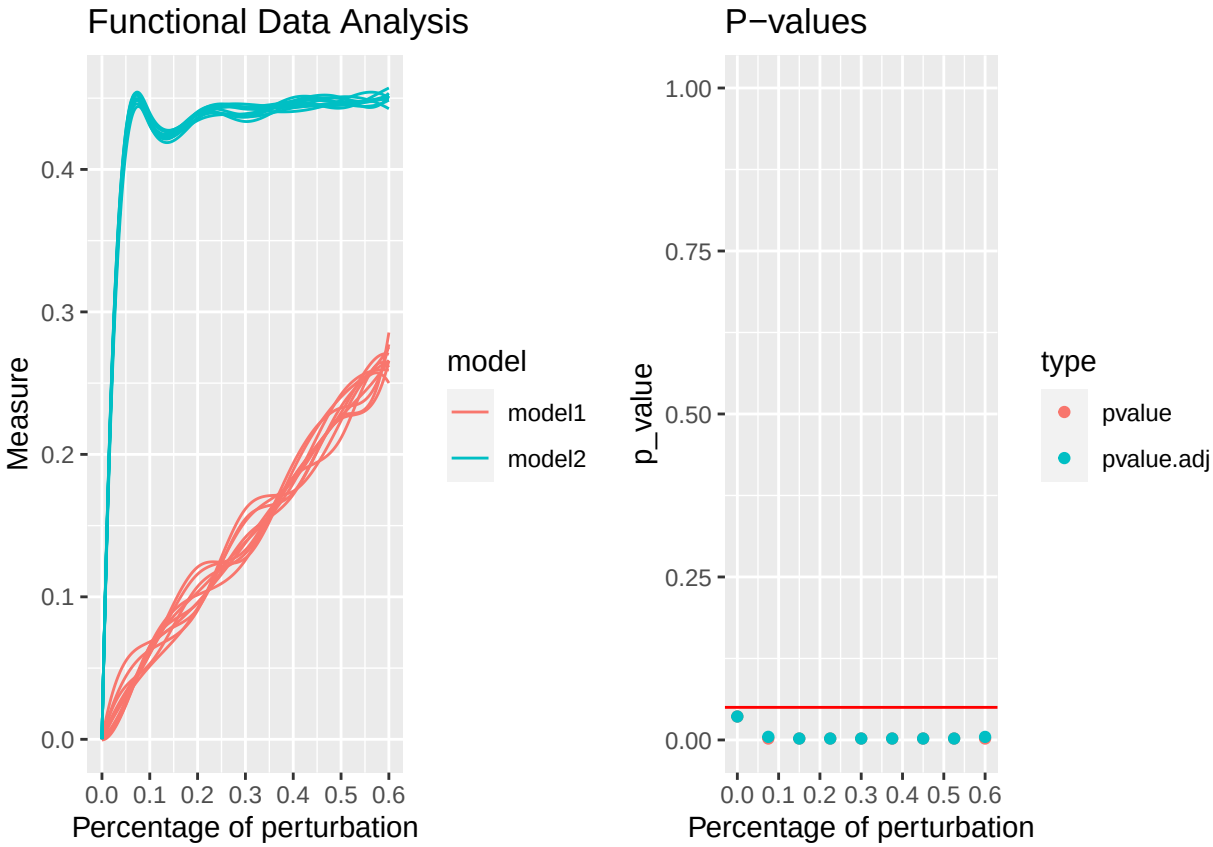

Robin plot robustness based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
infomap
Null

```
#
#
#
results_Robin<-mat.or.vec(length(proc_CM),4)
rownames(results_Robin)<-names(proc_CM)
colnames(results_Robin)<-c("bf_cm","auc_cm","bf_dk","auc_dk")
for (i in 1:length(proc_CM))
{
  results_Robin[i,1] <- robinGPTest(model1=proc_CM[[i]]$Mean, model2=proc_CM[[i]]$MeanRandom)
AUC <- robinAUC(graph=uni_graph2, model1=proc_CM[[i]]$Mean,
                model2=proc_CM[[i]]$MeanRandom)
results_Robin[i,2] <- AUC$area2/AUC$area1
robinFDATest(graph=uni_graph2, model1=proc_CM[[i]]$Mean, model2=proc_CM[[i]]$MeanRandom)
}
```

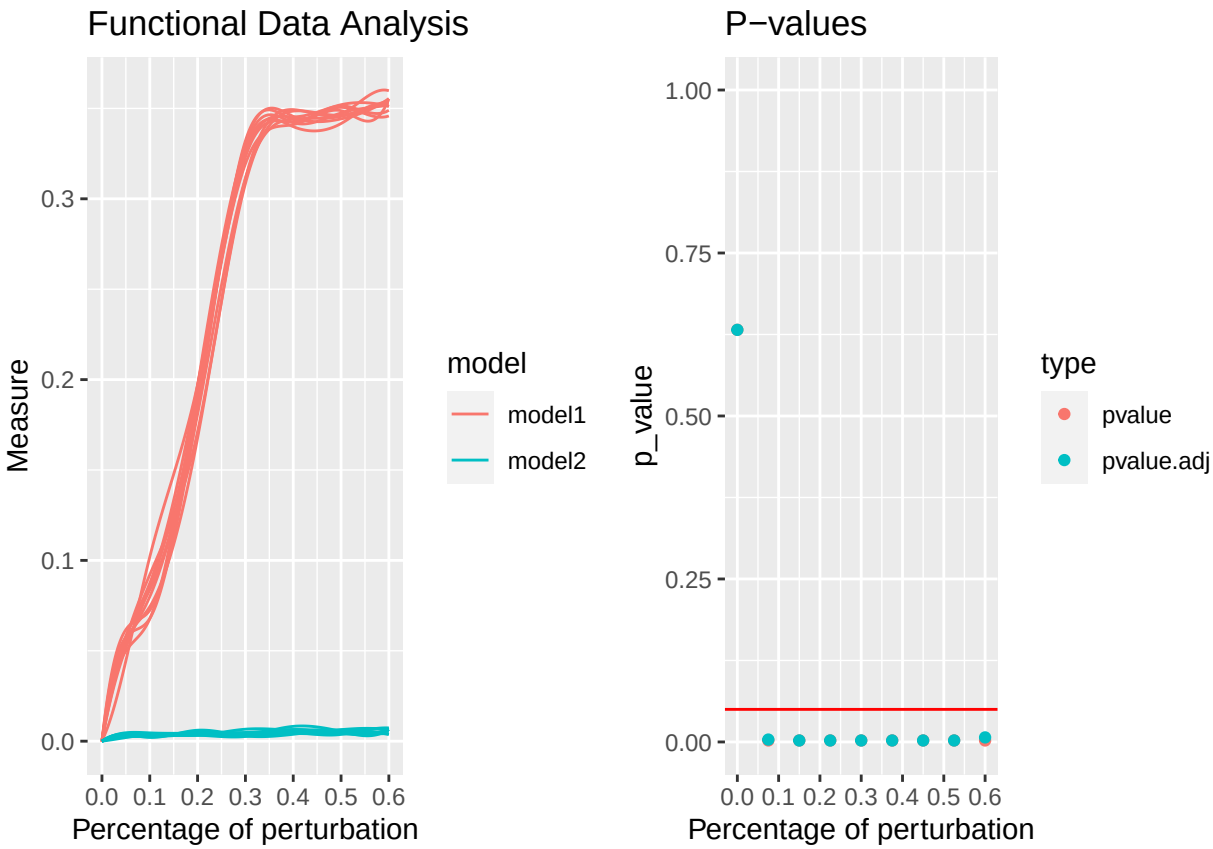
```
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```



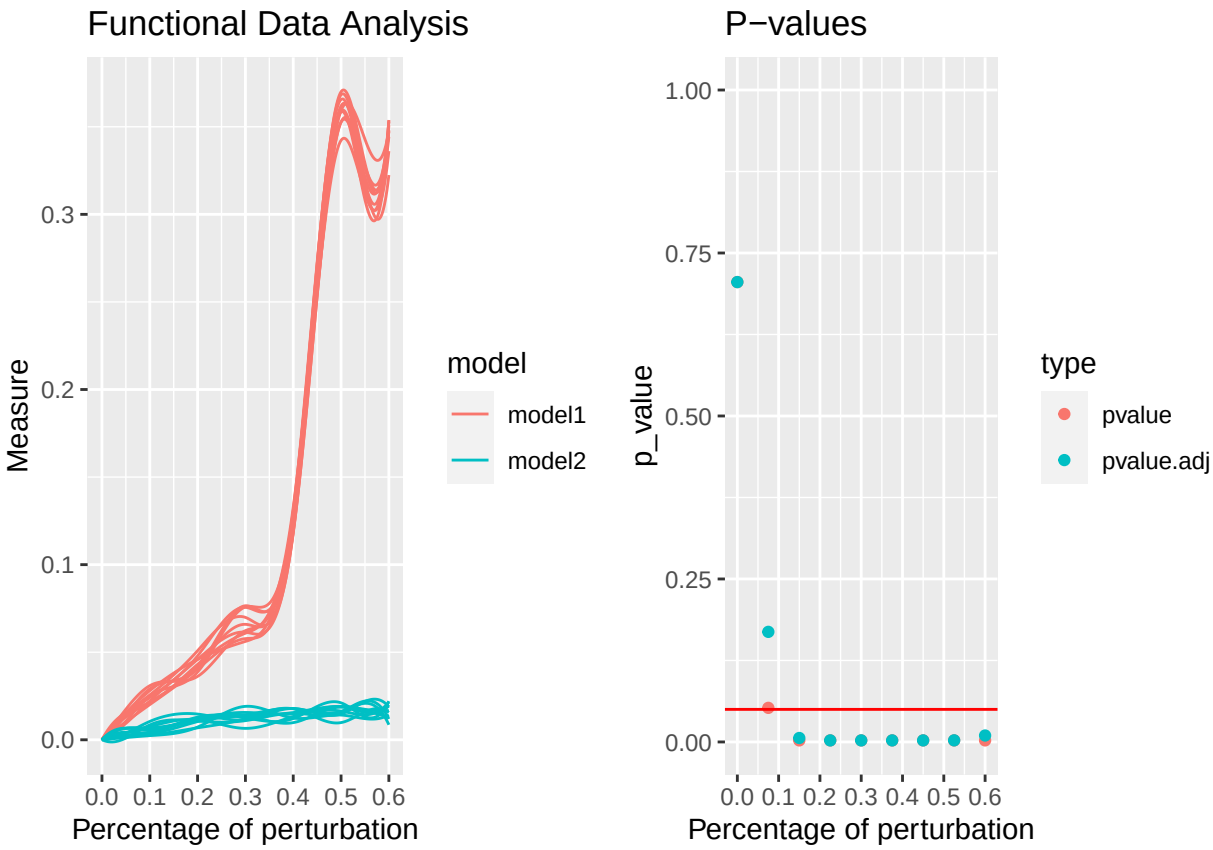

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

```
#
# Complex random model dk-series
#
#write_graph(
#  uni_graph2,
#  "/Users/Fernando/Desktop/01_paper_sanger_drive/export_graph",
#  format = "edgelist")
#save.image('/Users/Fernando/Desktop/01_Sanger_paper/improvement_dataset.Rdata')
save.image('/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v17final.Rdata')

graphRandomDK <- prepGraph(file="/Users/Fernando/Desktop/01_paper_sanger_drive/dk2.1_export_graph",
                           file.format = "edgelist")
proc_DK<-list()
for (DA in c("walktrap", "louvain", "labelProp", "infomap"))
{
  proc_DK[[DA]] <- robinRobust(graph=uni_graph2, graphRandom=graphRandomDK,
                               measure="vi", method=DA, type="independent")
}
```

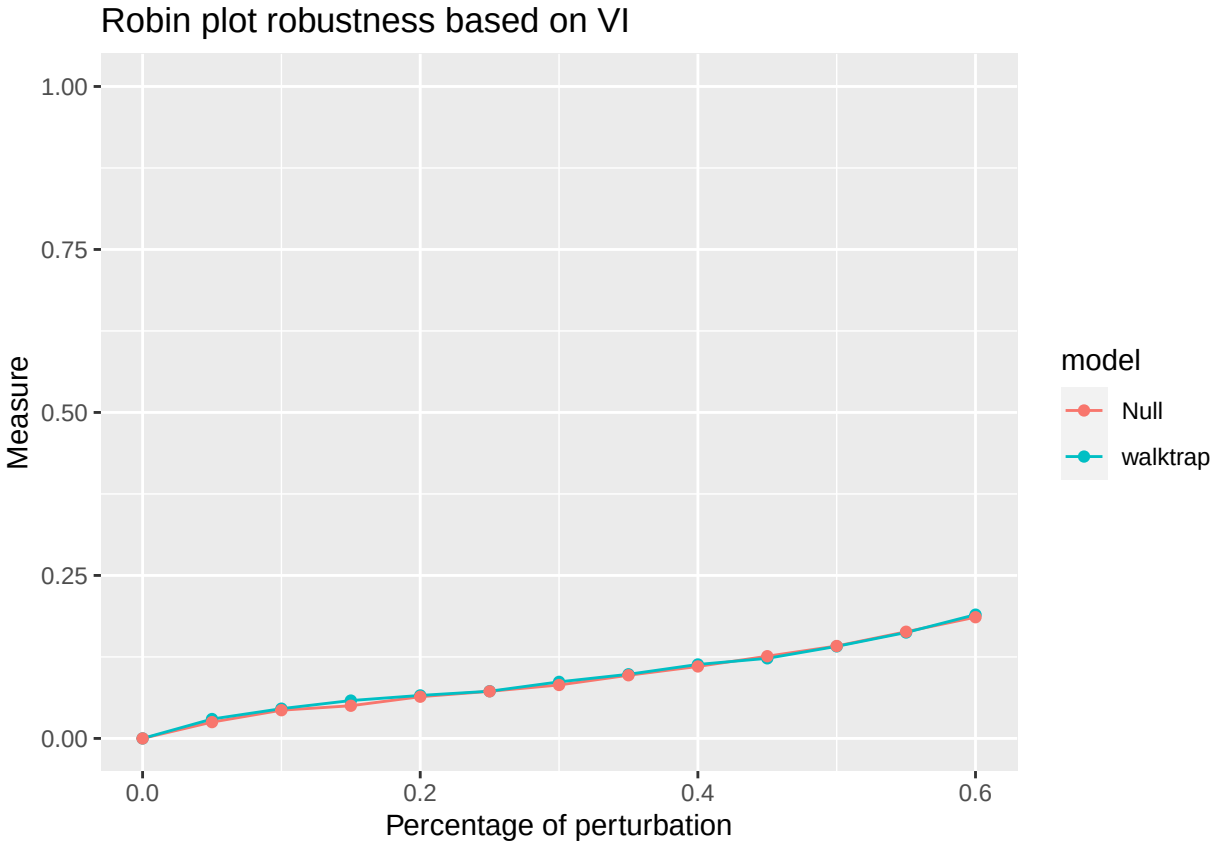
3.3.3.2 Random model based on dk-series

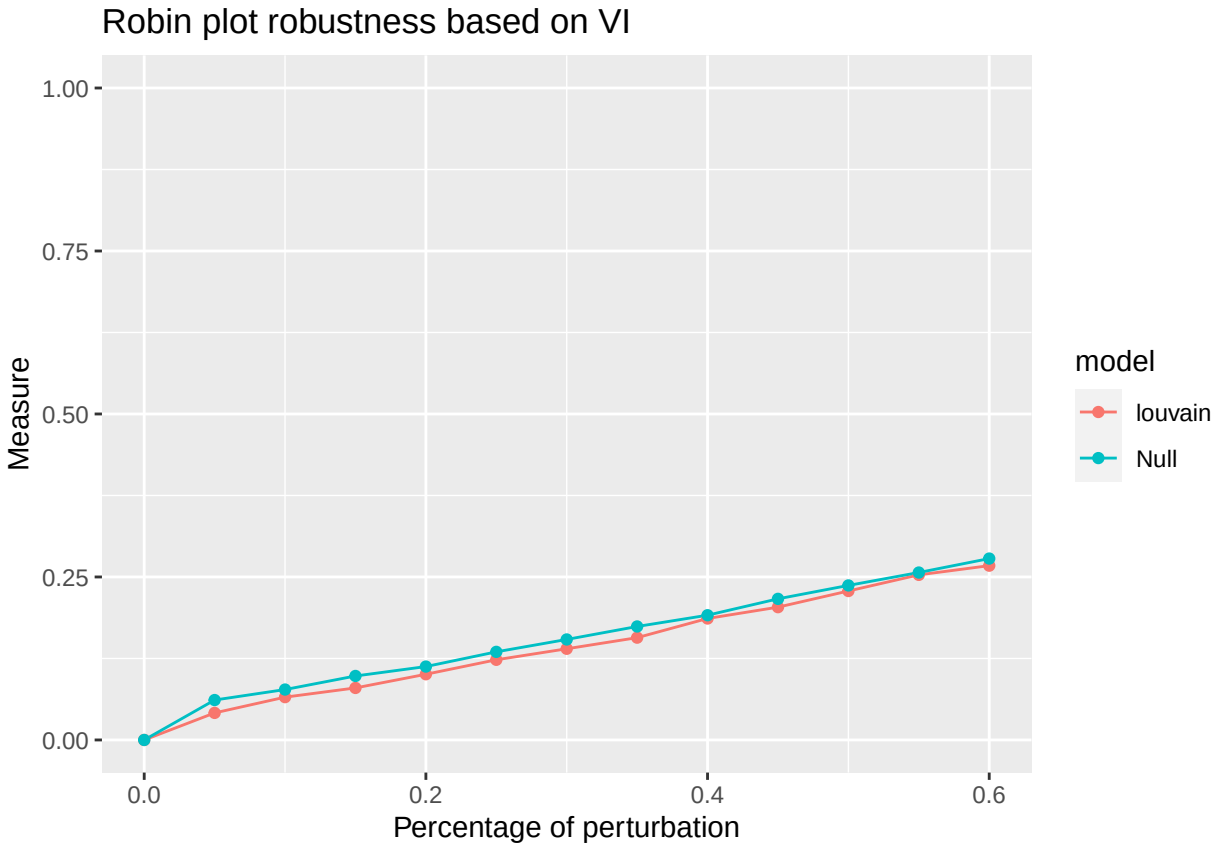
```
## Detected robin method  independent  type
```

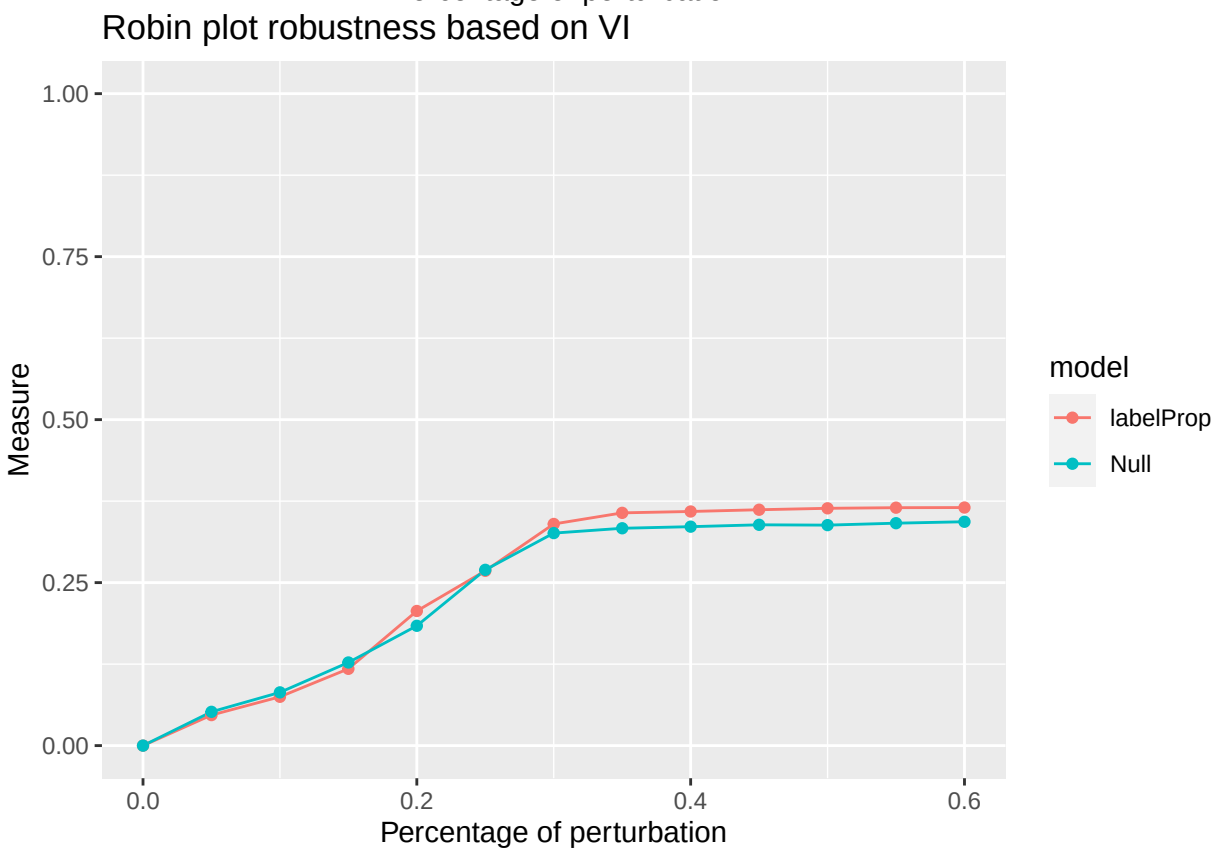
```
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
## Detected robin method  independent  type
## Perturbed  410  edges
## Perturbed  820  edges
## Perturbed  1230  edges
## Perturbed  1641  edges
## Perturbed  2051  edges
## Perturbed  2461  edges
## Perturbed  2871  edges
## Perturbed  3281  edges
## Perturbed  3691  edges
## Perturbed  4102  edges
## Perturbed  4512  edges
## Perturbed  4922  edges
```

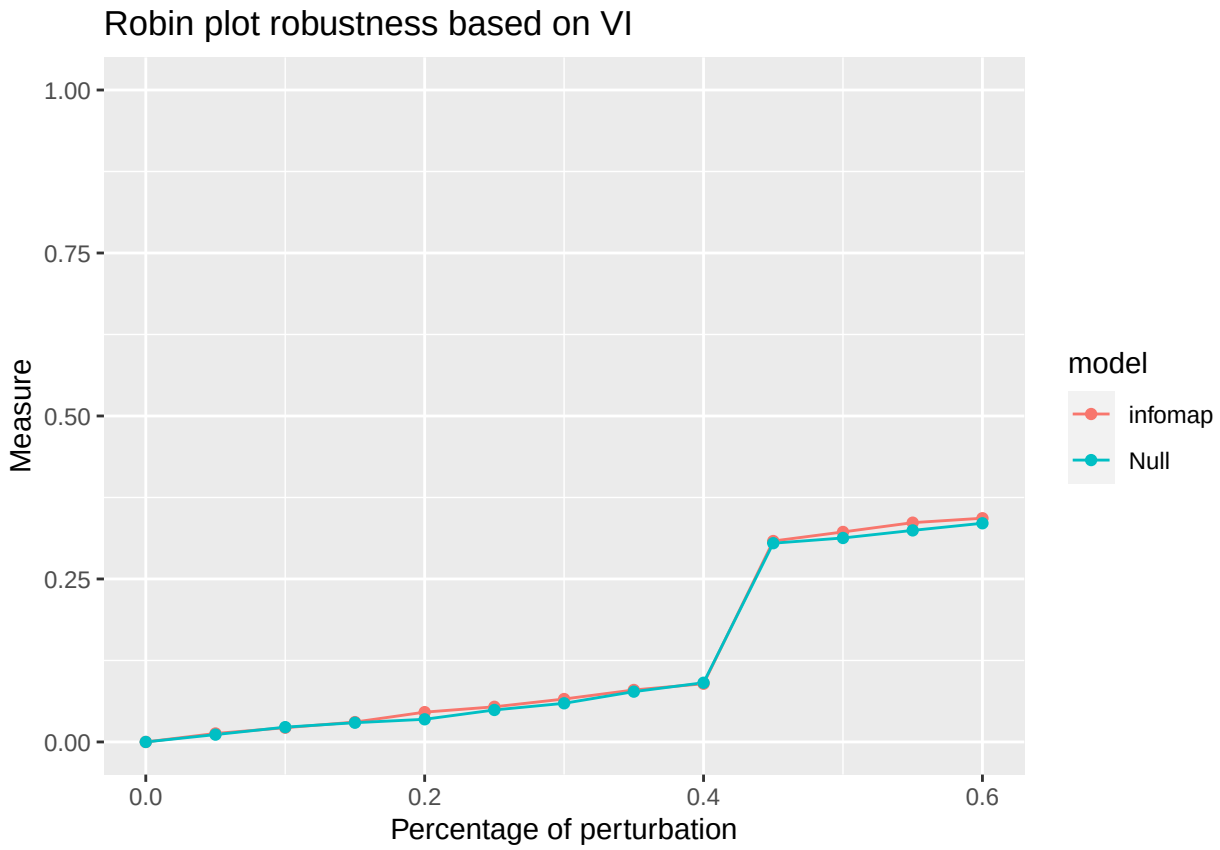
```
for (i in 1:length(proc_DK))
{
```

```
  comp_foo<-proc_DK[[i]]
  print(plotRobin(uni_graph2, model1=comp_foo$Mean, model2=comp_foo$MeanRandom, legend = c(names(proc_C
                  title = "Robin plot robustness based on VI"))
}
```




Robin plot robustness based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
louvain
Null

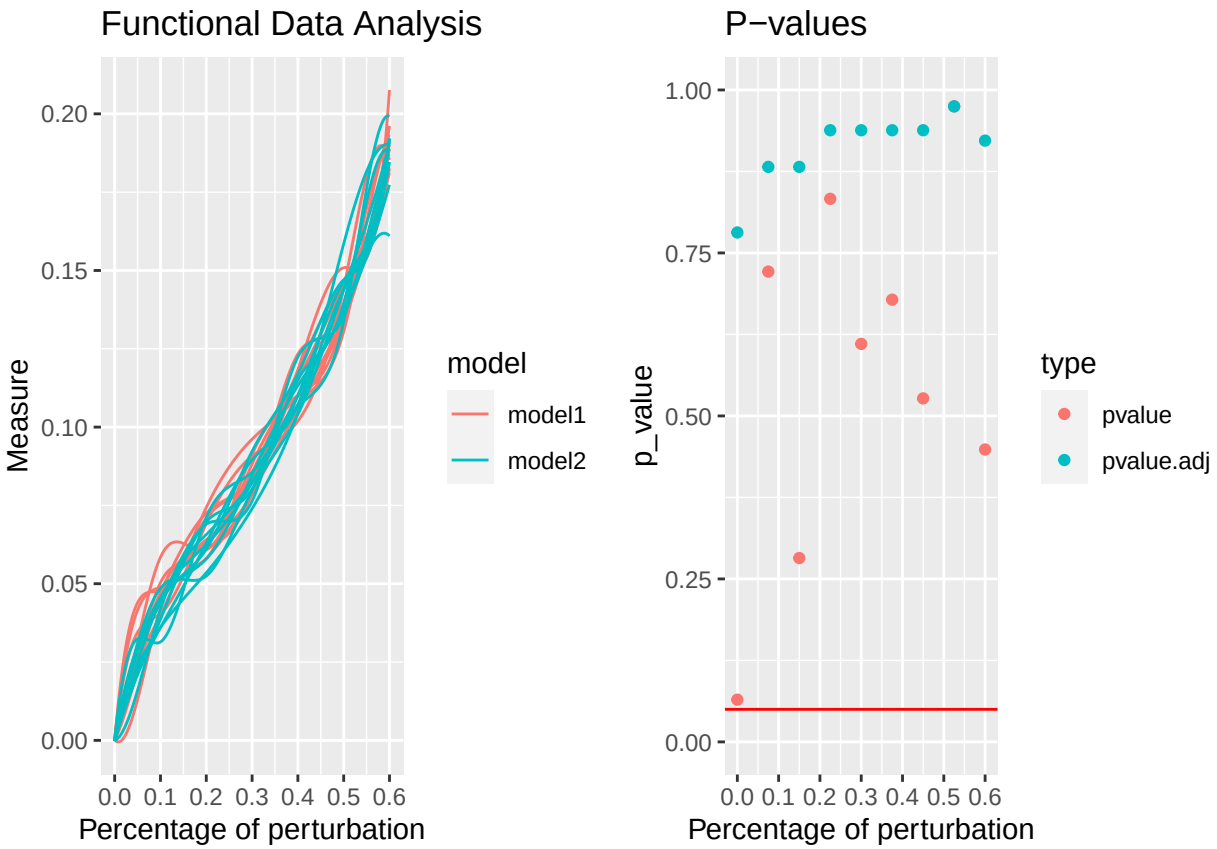

Robin plot robustness based on VI
Measure
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Percentage of perturbation
model
labelProp
Null



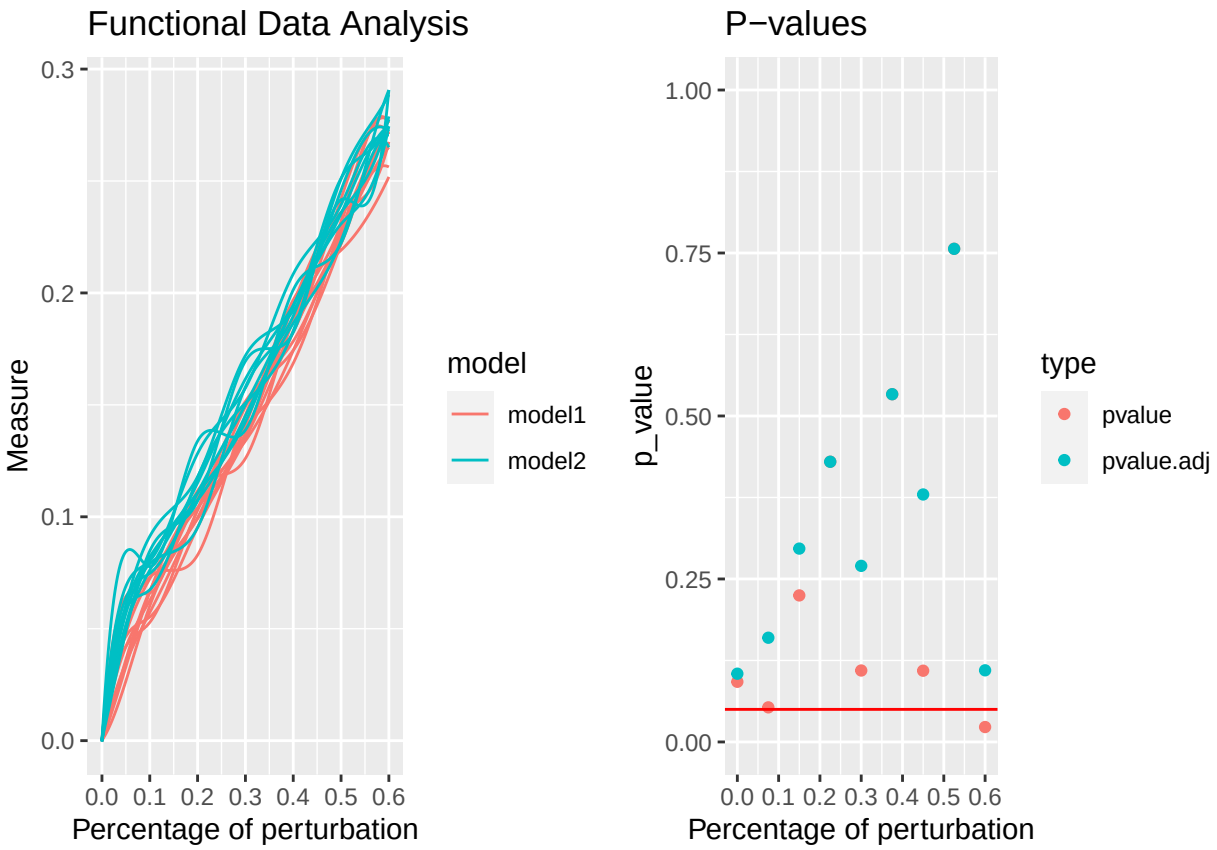

```
#
#
#
for (i in 1:length(proc_DK))
{
  results_Robin[i,3] <- robinGPTest(model1=proc_DK[[i]]$Mean, model2=proc_DK[[i]]$MeanRandom)
  AUC <- robinAUC(graph=uni_graph2,
                  model1=proc_DK[[i]]$Mean,
                  model2=proc_DK[[i]]$MeanRandom)
  results_Robin[i,4] <- AUC$area2/AUC$area1
  robinFDATest(graph=uni_graph2, model1=proc_DK[[i]]$Mean, model2=proc_DK[[i]]$MeanRandom)
}
```

```
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
```

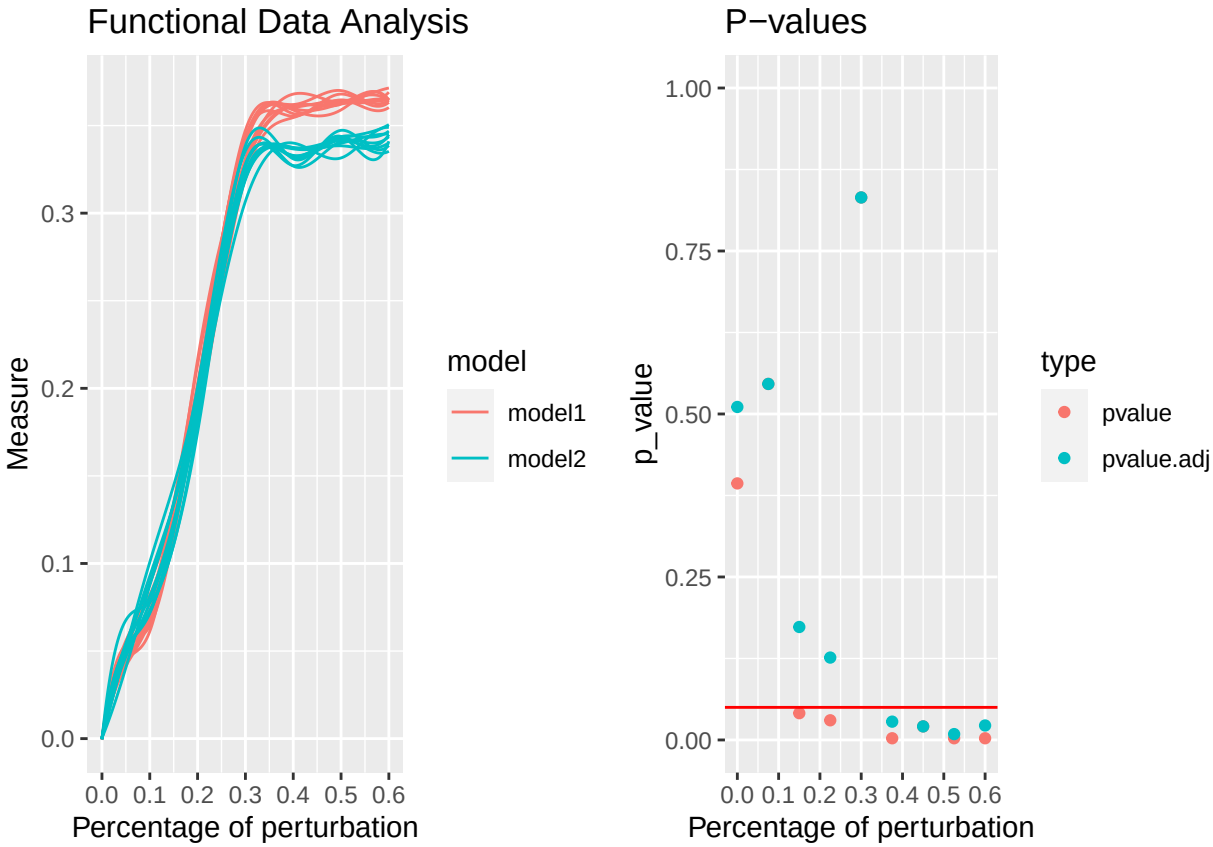
```
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```



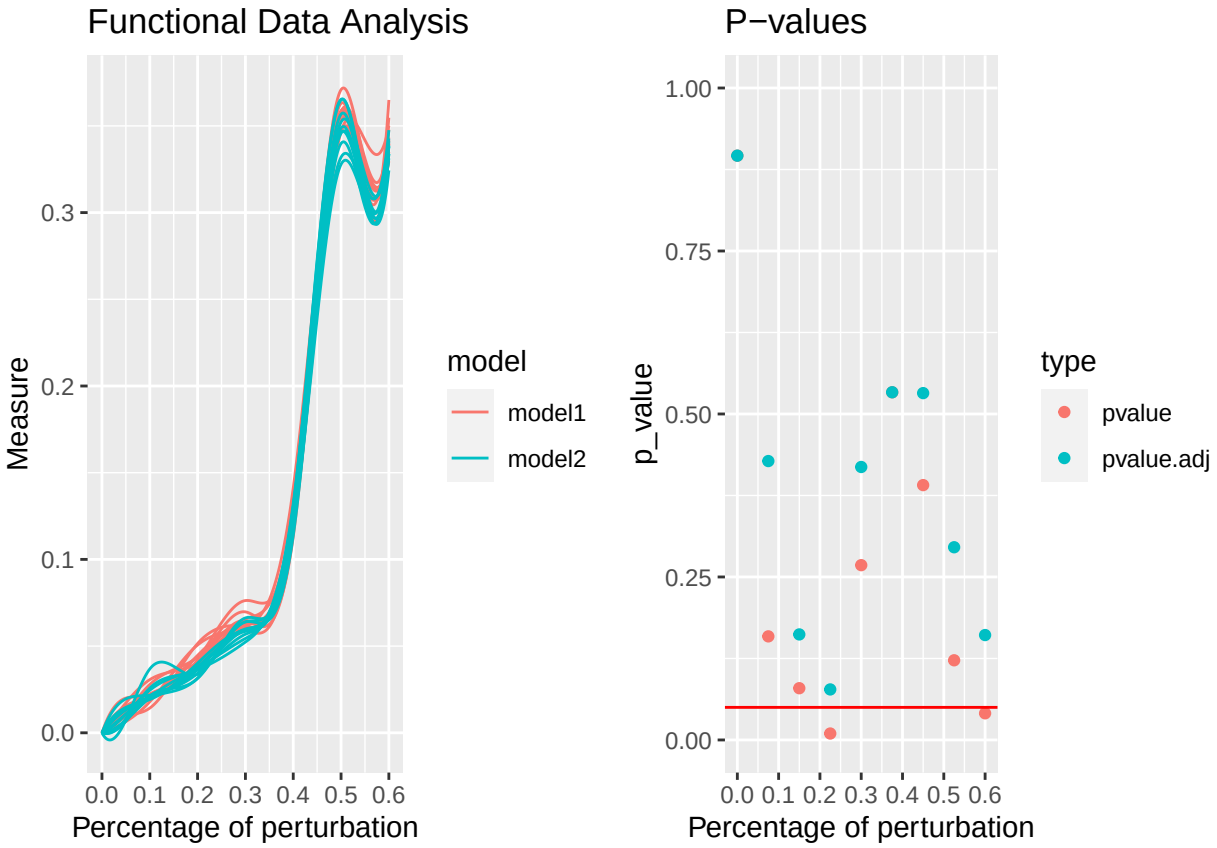

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
##  Profile  1
##  Profile  2
## [1] "First step: basis expansion"
## Swapping 'y' and 'argvals', because 'y' is  simpler,
##   and 'argvals' should be;  now  dim(argvals) =  13 ;  dim(y) =  13 x 20
## [1] "Second step: joint univariate tests"
## [1] "Third step: interval-wise combination and correction"
## [1] "creating the p-value matrix: end of row 2 out of 9"
## [1] "creating the p-value matrix: end of row 3 out of 9"
## [1] "creating the p-value matrix: end of row 4 out of 9"
## [1] "creating the p-value matrix: end of row 5 out of 9"
## [1] "creating the p-value matrix: end of row 6 out of 9"
## [1] "creating the p-value matrix: end of row 7 out of 9"
## [1] "creating the p-value matrix: end of row 8 out of 9"
## [1] "creating the p-value matrix: end of row 9 out of 9"
## [1] "Interval Testing Procedure completed"
```

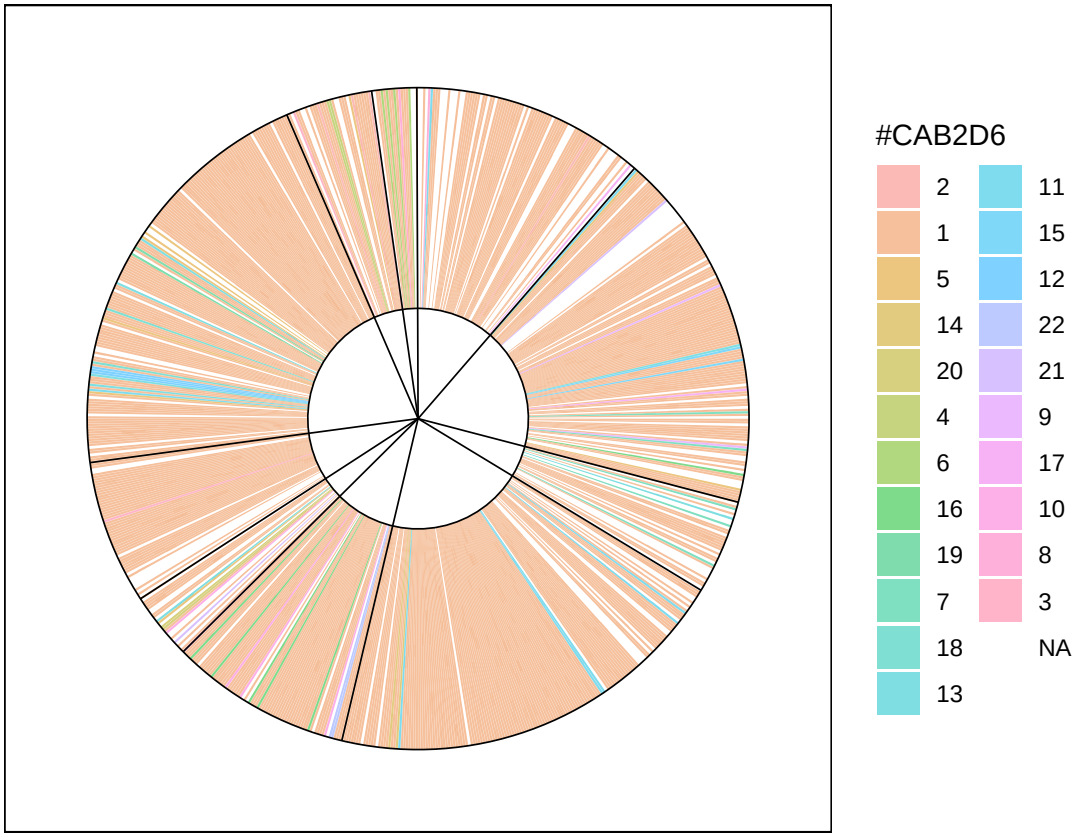



```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

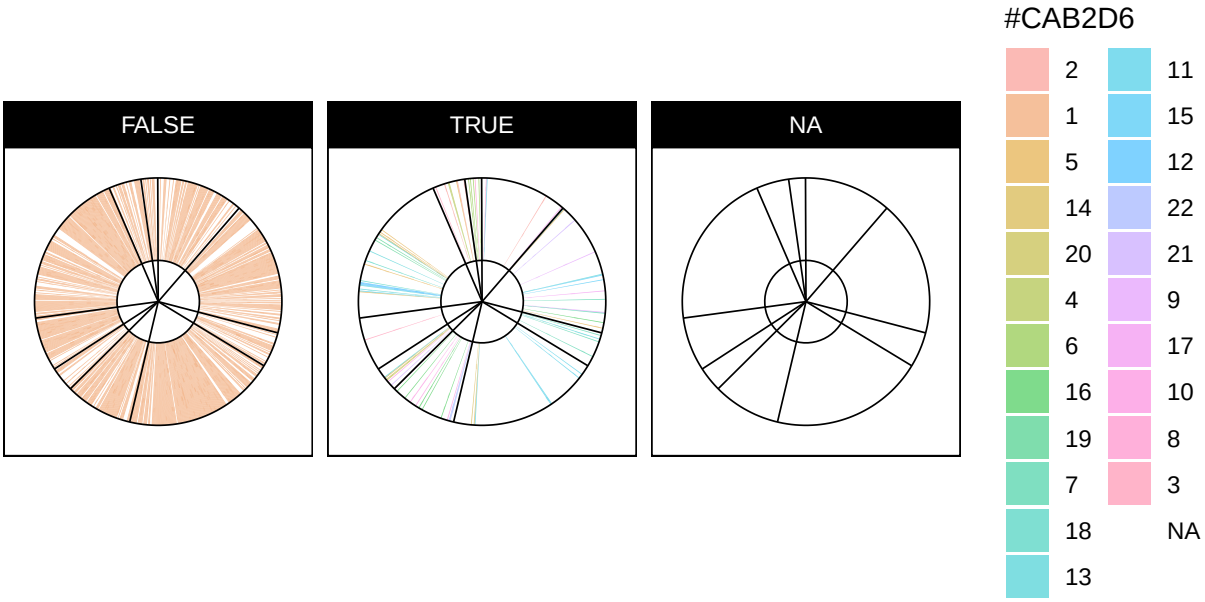
```
#save.image('/Users/Fernando/Desktop/01_Sanger_paper/improvement_dataset.Rdata')
#save.image('/Users/Fernando/Desktop/01_Sanger_paper/final_dataset_v18final.Rdata')
```

```
#
# Modules
#
foo_comps<-membership(components)
names(foo_comps)<-sapply(strsplit(names(foo_comps),"_"),`[`,2)

#
#
foo<-condensed_baps_adm[orden.foo,]
foo<-data.frame(orden=factor(c(1:dim(foo)[1])),modules=factor(foo_comps[orden.foo],levels = order(table

colorinos_modules<-colorRampPalette(colorinos.bipolar)(max(foo_comps))


p<-ggplot(data.frame(foo)) +
  geom_bar(aes(x=orden, y=1,fill=modules),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos_modules,na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
```

```
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),ax
                           axis.text.x=element_blank(),
                           axis.ticks.x=element_blank(),axis.title.y=element_blank(),
                           axis.text.y=element_blank(),
                           axis.ticks.y=element_blank()) + coord_polar()
print(p)
```

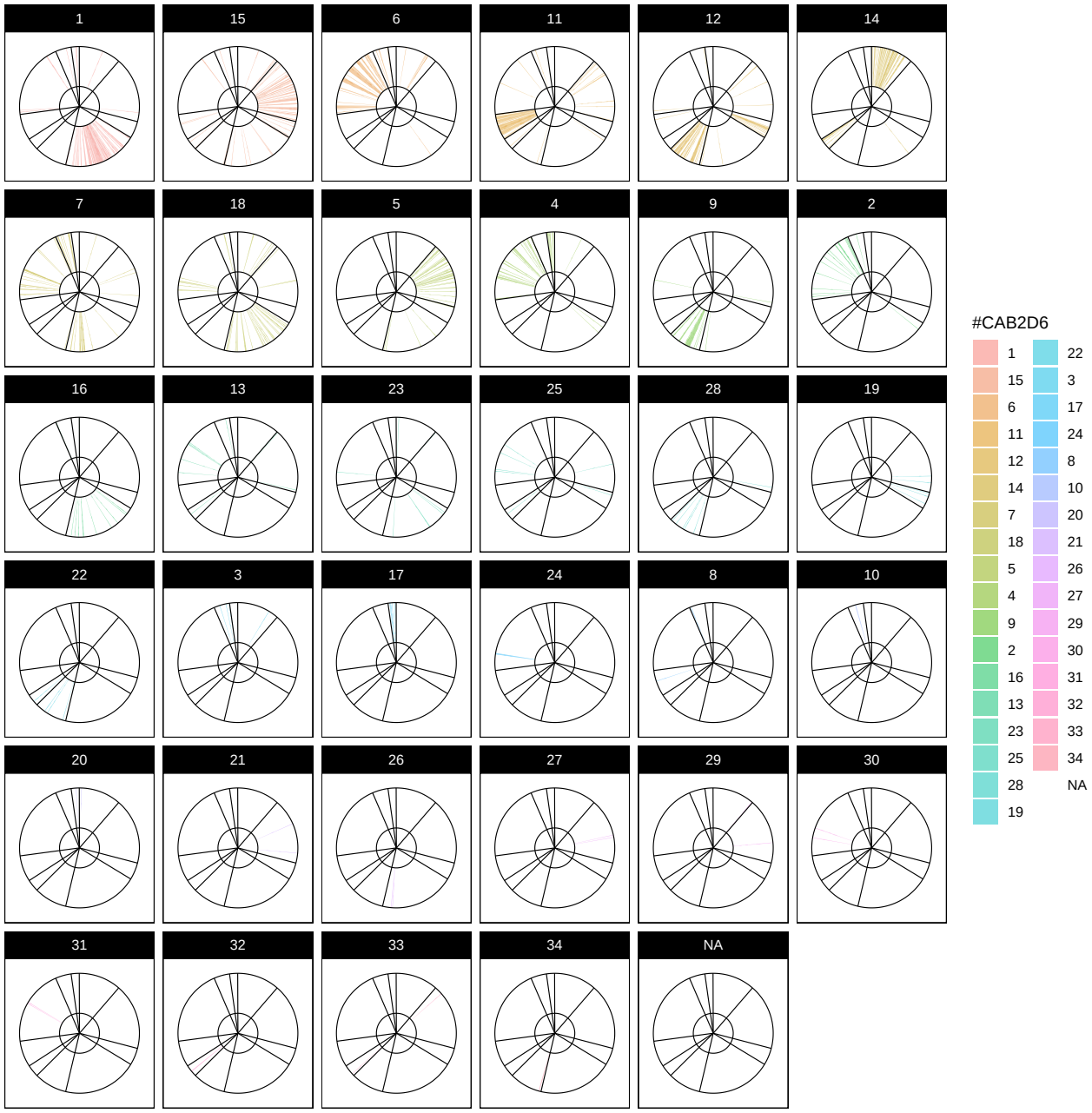



```
print(p + facet_wrap(vars(modules!=1)))
```



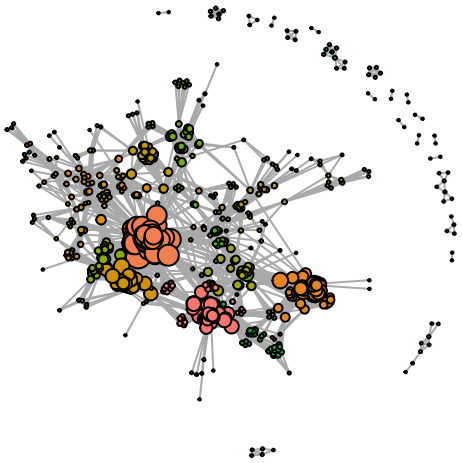

```
final_mods<-igraph::cluster_louvain(uni_graph2)
foo_modules<-membership(final_mods)
names(foo_modules)<-sapply(strsplit(names(foo_modules),"_"),`[`,2)

#
#
foo<-condensed_baps_adm[orden.foo,]
foo<-data.frame(orden=factor(c(1:dim(foo)[1])),modules=factor(foo_modules[orden.foo],levels = order(tabl

colorinos_modules<-colorRampPalette(colorinos.bipolar)(max(foo_modules))


p<-ggplot(data.frame(foo)) +
  geom_bar(aes(x=orden, y=1,fill=modules),
           stat="identity", alpha=0.5) +
  scale_fill_discrete(colorinos_modules,na.value="white") +
  geom_hline(yintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_hline(yintercept = 1, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 0, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 94, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 240, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 278, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 443, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 516, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 543, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 601, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 771, colour = 1, alpha=1, size=0.3) +
  geom_vline(xintercept = 806, colour = 1, alpha=1, size=0.3) +
  ylim(c(-0.5,1)) +
  theme_linedraw() + theme(panel.grid.major.x = element_blank(),panel.grid.major.y = element_blank(),axi
                           axis.text.x=element_blank(),
                           axis.ticks.x=element_blank(),axis.title.y=element_blank(),
                           axis.text.y=element_blank(),
                           axis.ticks.y=element_blank()) + coord_polar()
```

```
print(p + facet_wrap(vars(modules)))
```

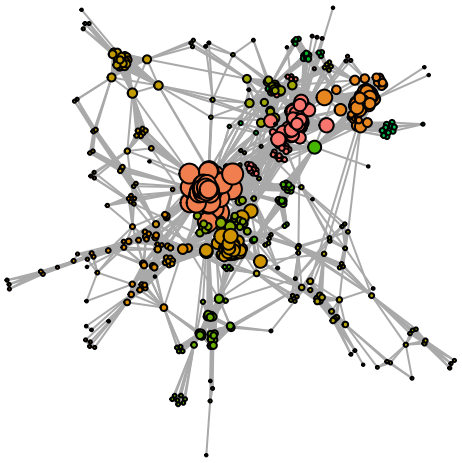



```
#
# Plot beyond to illustrate the networks
#
colorinos_neo<-hue_pal()(34)[c(1,12,20,10,9,3,7,23,11,24,4,5,14,6,2,13,21,8,18,25,26,19,15,22,16,27,28,
color_clusters<-foo_modules
names(color_clusters)<-paste("s_",names(color_clusters),sep="")
color_clusters<-color_clusters[names(V(uni_graph2))]
plot(
  uni_graph2,
  vertex.size      = degree(uni_graph2)/10 +1,
  vertex.label     = NA,
```

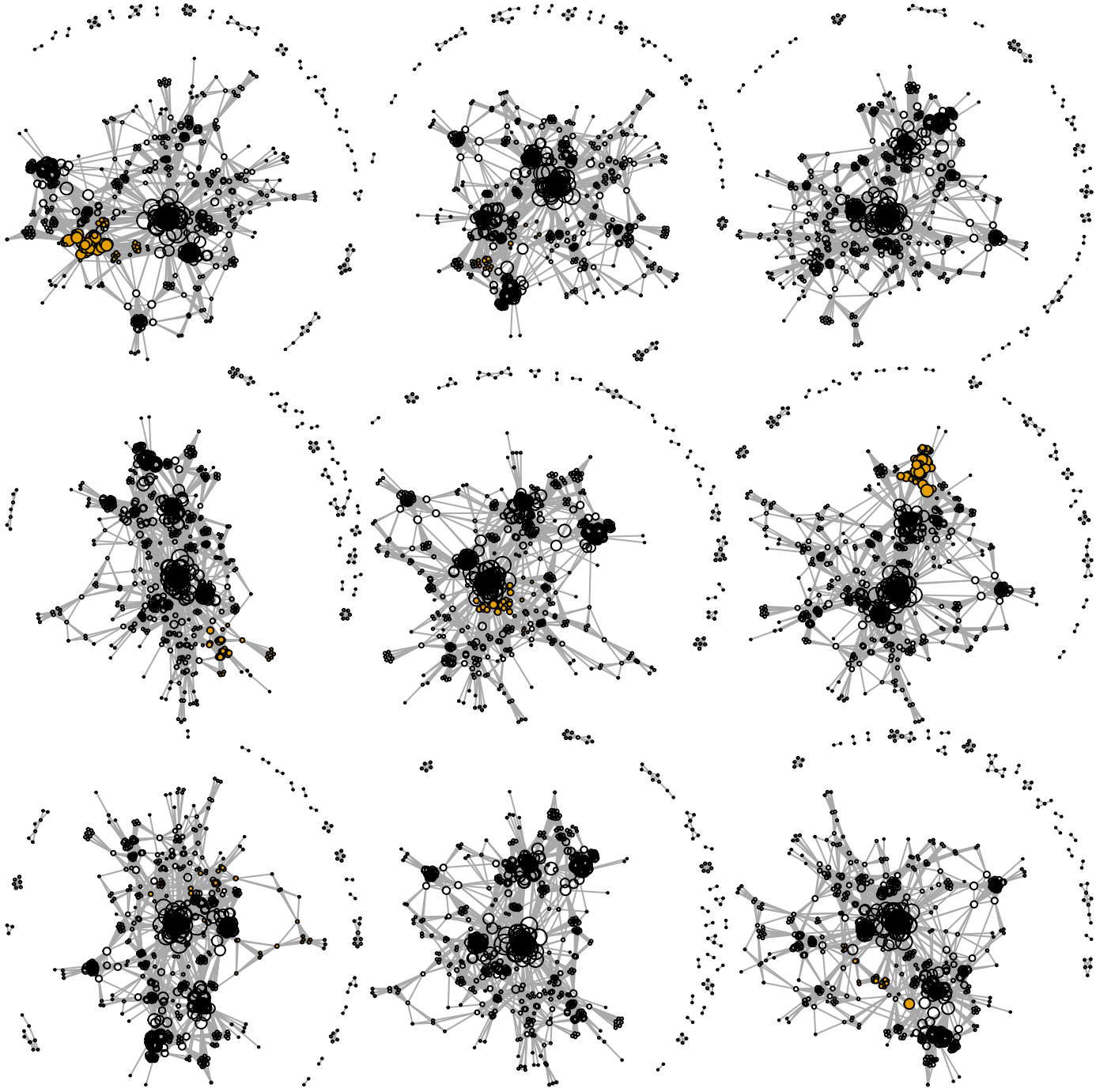
```
  edge.arrow.size = .25,
  vertex.color    = colorinos_neo[color_clusters]
)
```

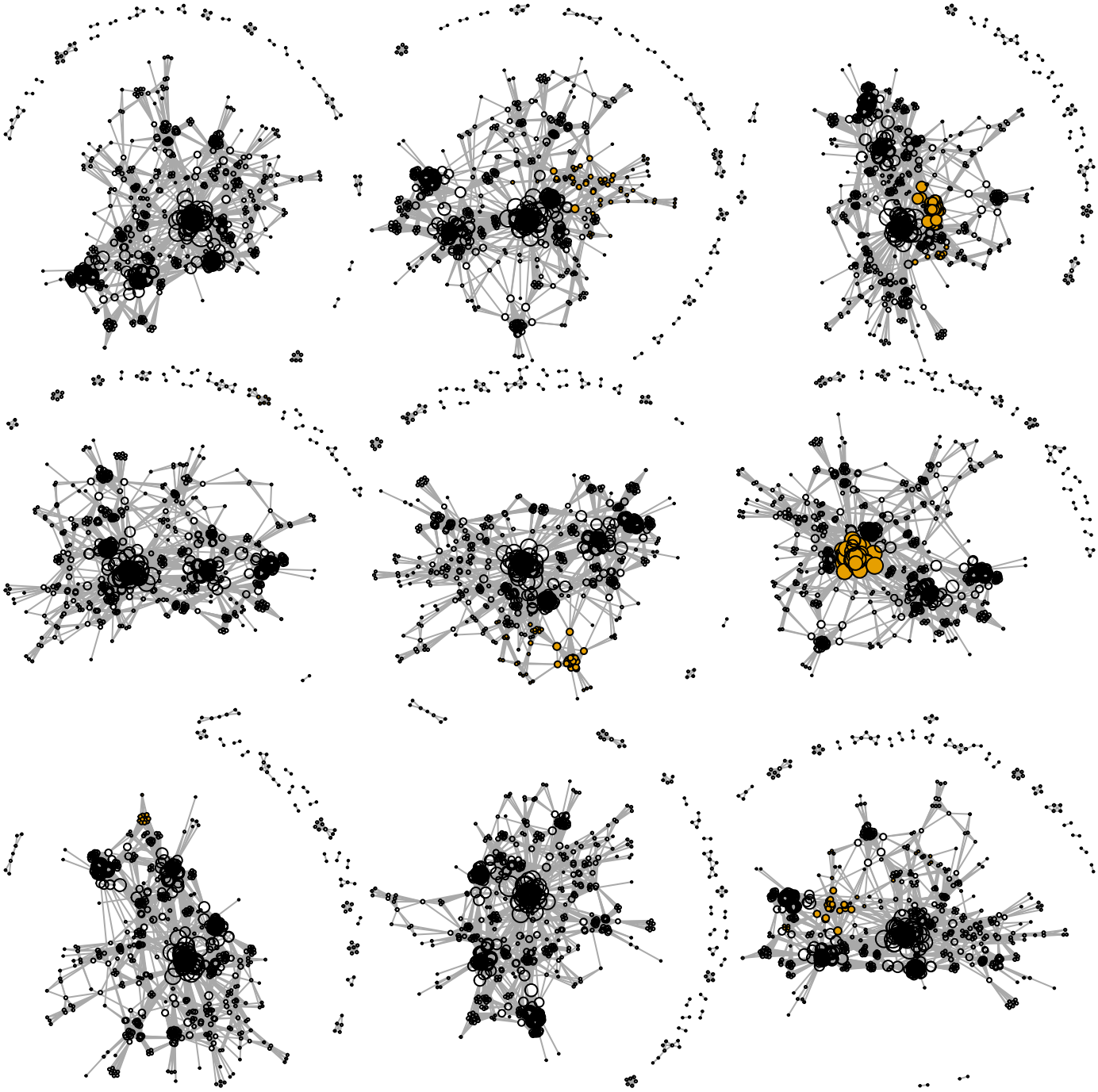


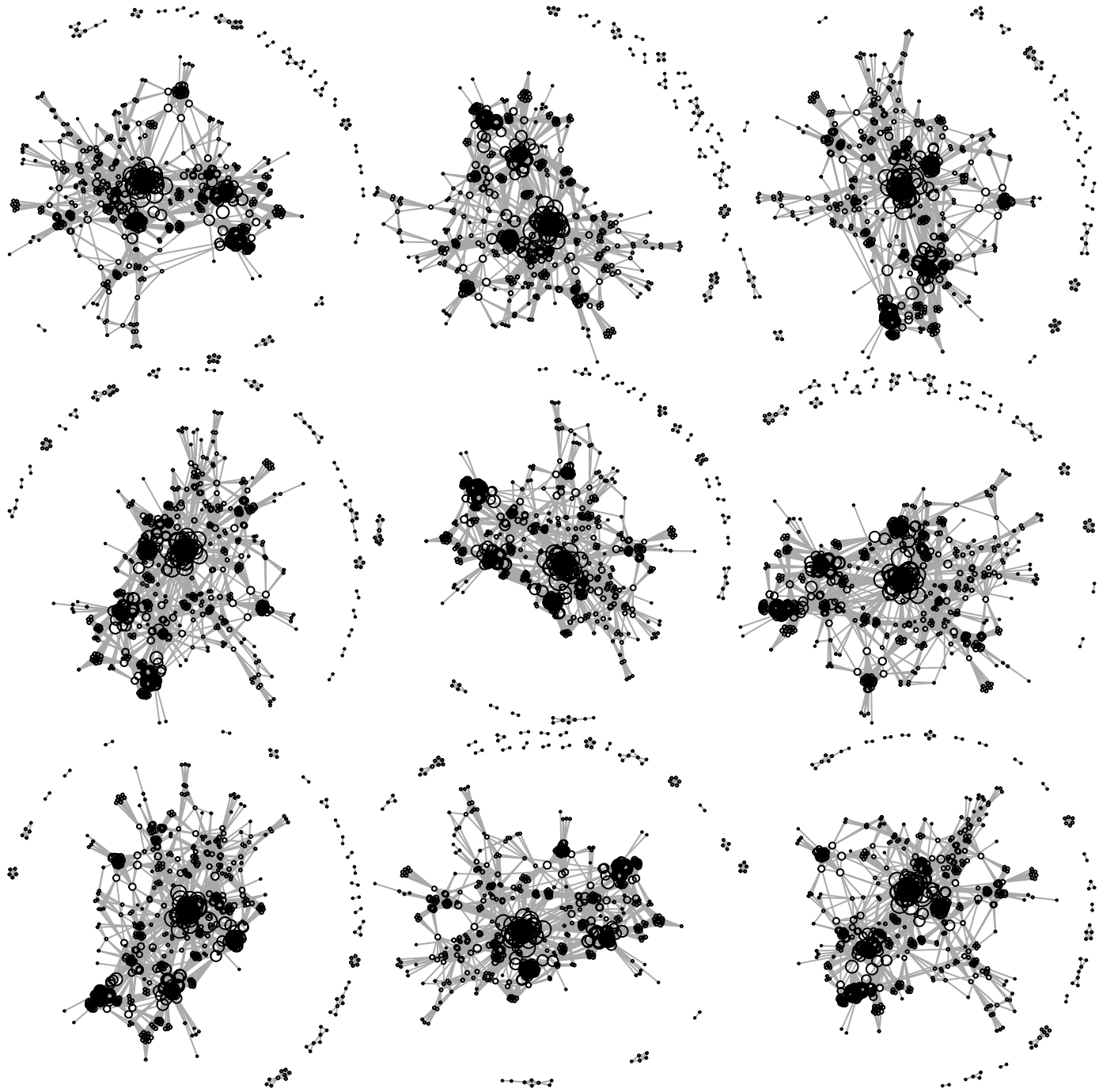
```
foo<-induced_subgraph(uni_graph2,which(color_clusters%in%c(1,2,4,5,6,7,9,11,12,14,15,16,18)))
plot(
  foo,
  vertex.size     = degree(foo)/10 +1,
  vertex.label    = NA,
  edge.arrow.size = .25,
  vertex.color    = colorinos_neo[color_clusters[which(color_clusters%in%c(1,2,4,5,6,7,9,11,12,14,15,16
)
```

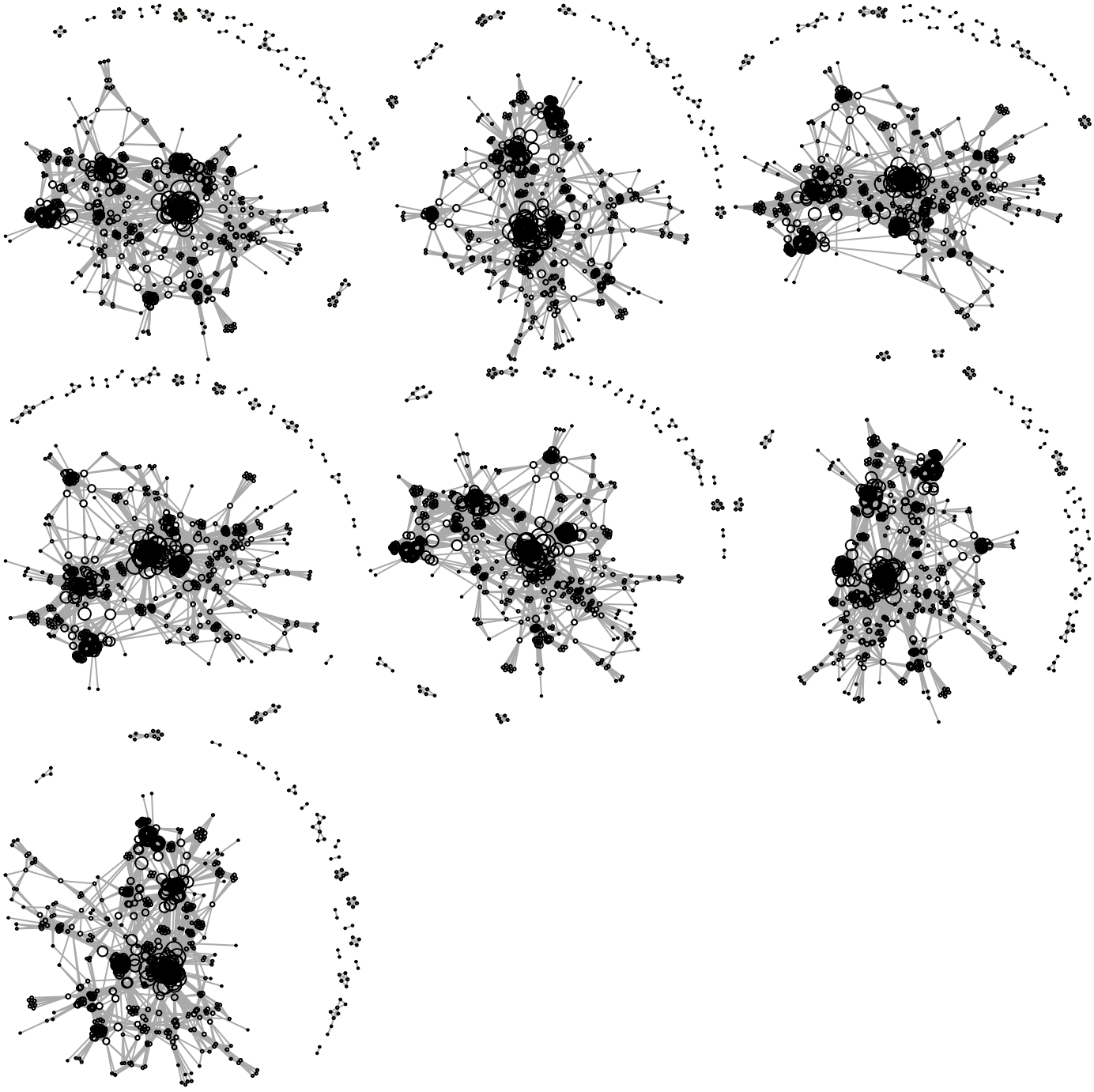


```
for (i in 1:34)
{
  plot(
    uni_graph2,
    vertex.size     = degree(uni_graph2)/10 +1,
    vertex.label    = NA,
    edge.arrow.size = .25,
    vertex.color    = color_clusters==i
  )
}
```









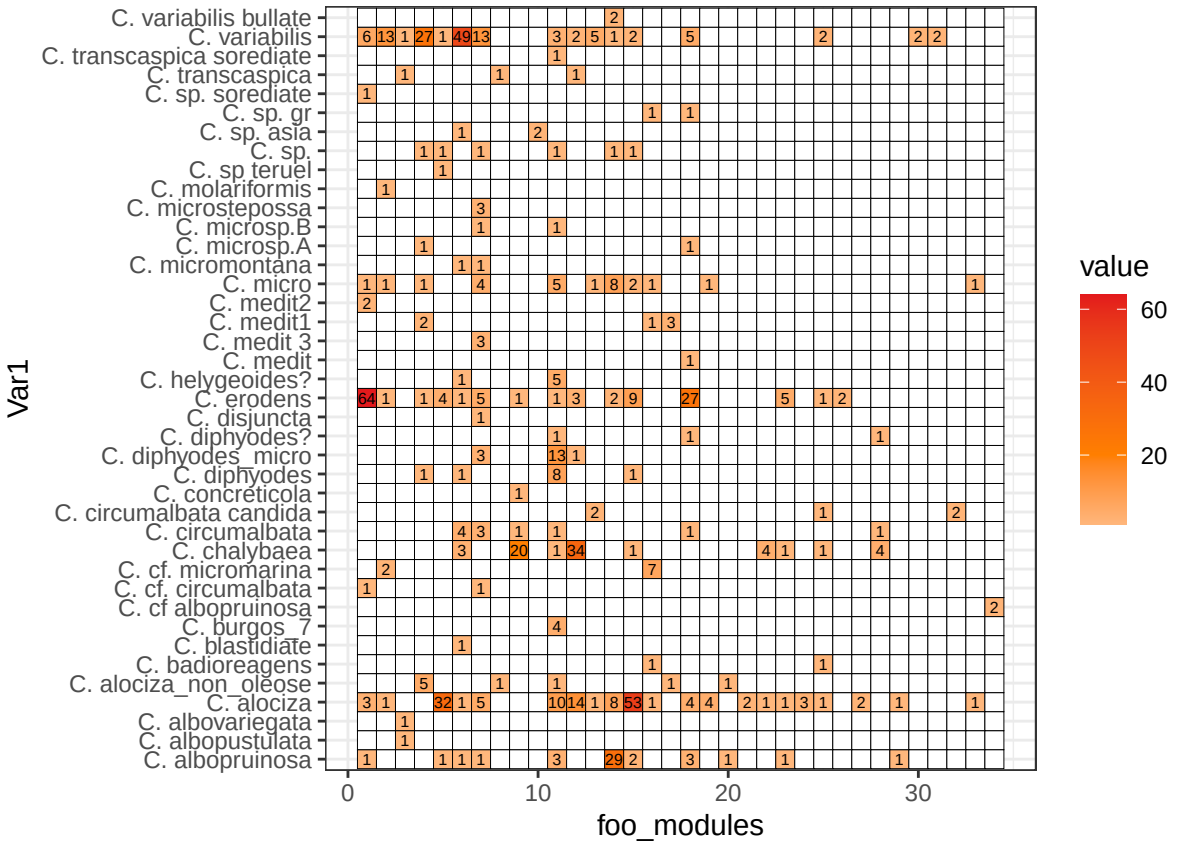
```
stats_degree<-list()
stats_desc<-list()
stats_networks<-list()
for (i in 1:34)
  {
  foo<-induced_subgraph(uni_graph2,which(color_clusters==i))
  # Computing centrality measures for each vertex
  V(foo)$outdegree  <- degree(foo, mode = "out")
  V(foo)$closeness  <- closeness(foo, mode = "total", normalized = TRUE)
  V(foo)$betweeness <- betweenness(foo, normalized = TRUE)
```

```
  # Extracting each vectex features as a data.frame
  stats <- as_data_frame(foo, what = "vertices")
  stats_networks[[i]]<-stats
  # Computing quantiles for each variable
  stats_degree[[i]] <- with(stats, {
  cbind(
    outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
    closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
    betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
  )
})
  stats_desc[[i]] <- cbind(
    size    = vcount(foo),
    nedges  = ecount(foo),
    density = edge_density(foo),
    recip   = reciprocity(foo),
    centr   = centr_betw(foo)$centralization,
    pathLen = mean_distance(foo)
  )
}

foo<-induced_subgraph(uni_graph2,which(color_clusters%in%c(1,2,4,5,6,7,9,11,12,14,15,16,18)))
V(foo)$outdegree  <- degree(foo, mode = "out")
V(foo)$closeness  <- closeness(foo, mode = "total", normalized = TRUE)
V(foo)$betweeness <- betweenness(foo, normalized = TRUE)
stats <- as_data_frame(foo, what = "vertices")
stats_networks[[35]]<-stats
stats_degree[[35]] <- with(stats, {
  cbind(
    outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
    closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
    betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
  )
})
stats_desc[[35]] <- cbind(
  size    = vcount(foo),
  nedges  = ecount(foo),
  density = edge_density(foo),
  recip   = reciprocity(foo),
  centr   = centr_betw(foo)$centralization,
  pathLen = mean_distance(foo)
)

foo<-uni_graph2
V(foo)$outdegree  <- degree(foo, mode = "out")
V(foo)$closeness  <- closeness(foo, mode = "total", normalized = TRUE)
V(foo)$betweeness <- betweenness(foo, normalized = TRUE)
stats <- as_data_frame(foo, what = "vertices")
stats_networks[[36]]<-stats
stats_degree[[36]] <- with(stats, {
  cbind(
    outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
    closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
```

```
    betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
  )
})
stats_desc[[36]] <- cbind(
  size    = vcount(foo),
  nedges  = ecount(foo),
  density = edge_density(foo),
  recip   = reciprocity(foo),
  centr   = centr_betw(foo)$centralization,
  pathLen = mean_distance(foo)
)
stats_degree_trim<-stats_degree
stats_desc_trim<-stats_desc
```

```
# Heatmap 3 Species vs modules
foo_tabla<-melt(table(condensed_baps_adm[as.character(names(foo_modules)),2],foo_modules))
foo_tabla$value[foo_tabla$value==0]<-NA


ggplot(foo_tabla,aes(x=foo_modules,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(labe
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 20,
                       space = "Lab",
                       na.value = "#FFFFFF",
                       guide = "colourbar",
                       aesthetics = "fill"
  ) +
  coord_fixed() + theme_bw()
```

```
## Warning: Removed 1207 rows containing missing values (geom_text).
```




```
#
#
#
```

```
assocstats(table(condensed_baps_adm[as.character(names(foo_modules)),2],foo_modules))
```

```
##                    X^2   df   P(> X^2)
## Likelihood Ratio 1644 1287 4.3692e-11
## Pearson          4868 1287 0.0000e+00
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.937
## Cramer's V        : 0.467
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(condensed_baps_adm[as.character(names(foo_modules)),2],foo_modules))
```

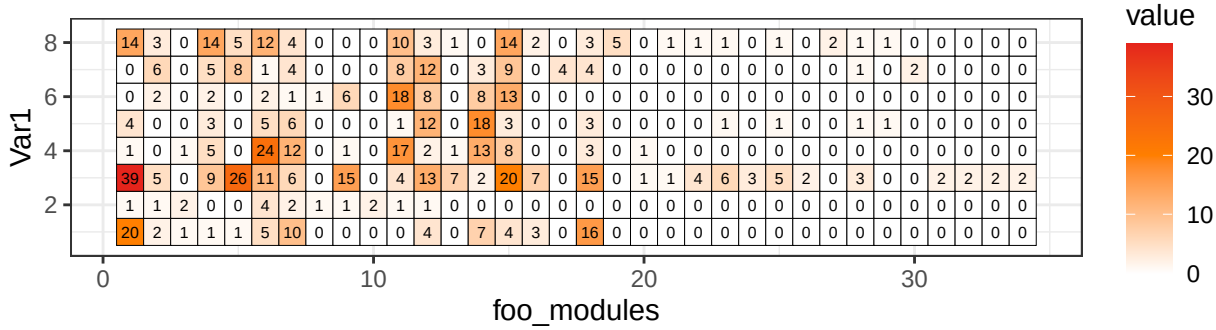
```
##
##  Permutation test for conditional independence
##
## data:  table(condensed_baps_adm[as.character(names(foo_modules)), 2],     foo_modules)
## f(x) = 25.923, p-value = 0.001
```

```
geo_distances<-mat.or.vec(length(condensed_baps_adm[as.character(names(foo_modules)),2]),length(condens

for (X in 1:length(condensed_baps_adm[as.character(names(foo_modules)),2]))
{
  geo_distances[,X]<-geosphere::distGeo(cbind(as.numeric(condensed_baps_adm[as.character(names(foo_modul
```

```
}

regions<-pamk(geo_distances,krange=2:10,diss=TRUE)

regions$pamobject$clustering
```

```
##   [1] 1 2 2 3 3 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 5 5 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [38] 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 6 6 6 6 7 4 4 4 4 4 4 1 1 1 8
##  [75] 8 8 8 3 3 3 3 3 3 3 3 3 3 3 3 7 7 7 7 7 7 7 7 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 6 6 7 6 6 6 6 6 6 6 7 7 7 7 7 7 8 8 7 7 7 8 8 7 3 7 7
## [149] 7 7 7 7 7 7 3 3 5 5 8 3 3 3 3 8 8 8 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [186] 8 8 8 8 8 3 3 3 3 3 3 3 3 3 8 7 7 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 7
## [223] 7 7 7 7 7 7 8 8 8 8 8 8 3 3 3 3 3 3 7 8 8 8 8 8 8 7 1 1 1 1 1 3 3 3 3 3
## [260] 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3
## [297] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [334] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 8 8 8 8
## [371] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [408] 8 8 8 8 8 8 3 3 3 8 8 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [445] 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 5 5 5 1 1 3 3
## [482] 6 6 6 5 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [519] 7 7 7 7 7 7 6 6 6 6 6 6 6 6 6 6 6 6 6 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5
## [556] 5 5 5 4 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [593] 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [630] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [667] 3 3 3 3 3 3 3 3 3 3
```

```
foo_tabla<-melt(table(regions$pamobject$clustering,foo_modules))
#foo_tabla$value[foo_tabla$value==0]<-NA


ggplot(foo_tabla,aes(x=foo_modules,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(labe
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 20,
                       space = "Lab",
                       na.value = "#FFFFFF",
                       guide = "colourbar",
                       aesthetics = "fill"
  ) +
  coord_fixed() + theme_bw()
```



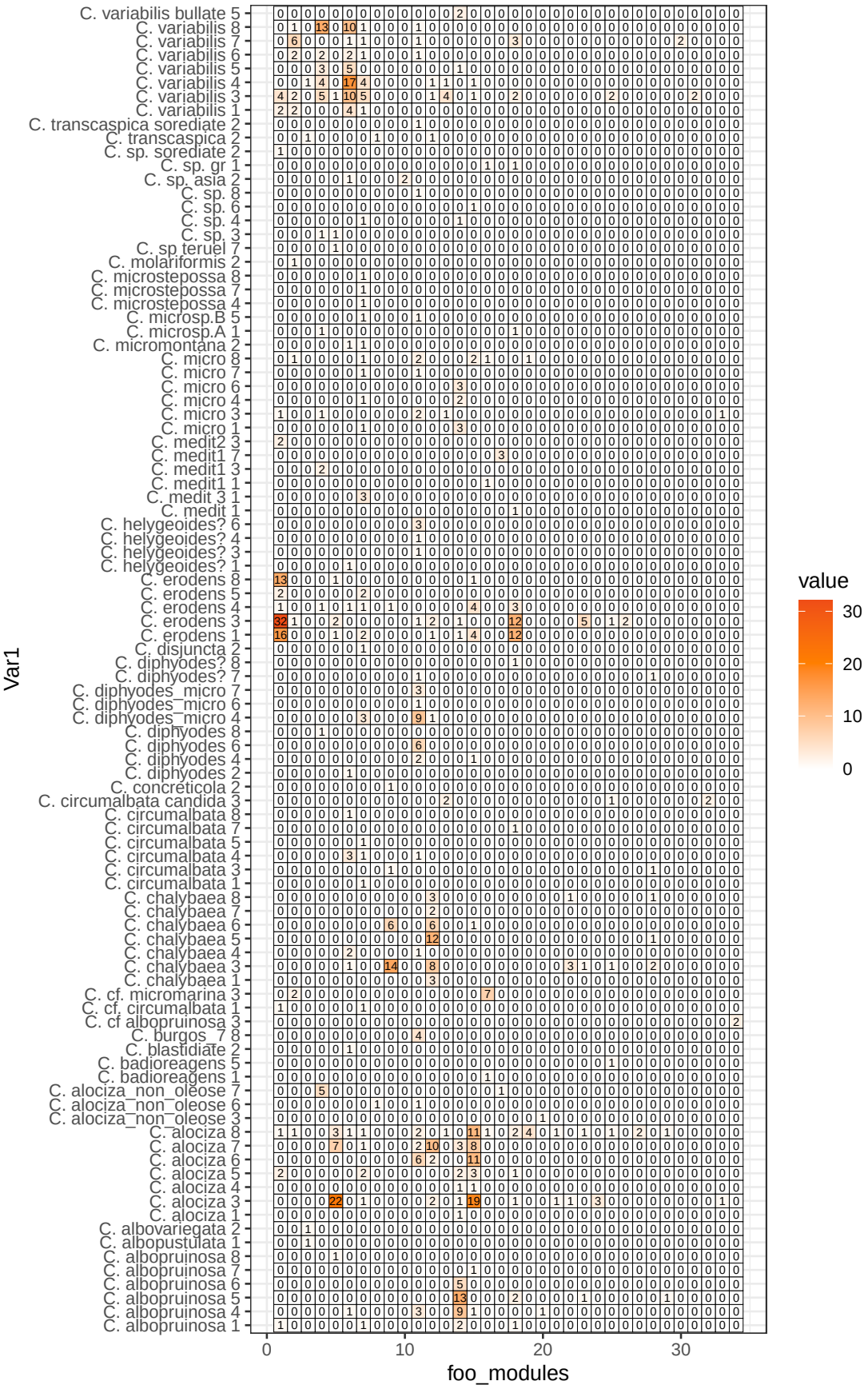

```
assocstats(table(regions$pamobject$clustering,foo_modules))
```

```
##                      X^2  df P(> X^2)
## Likelihood Ratio 632.37 231        0
## Pearson          755.86 231        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.727
## Cramer's V        : 0.4
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(regions$pamobject$clustering,foo_modules))
```

```
##
##  Permutation test for conditional independence
##
## data:  table(regions$pamobject$clustering, foo_modules)
## f(x) = 8.9748, p-value = 0.008
```

```
foo_tabla<-melt(table(paste(condensed_baps_adm[as.character(names(foo_modules)),2],regions$pamobject$clu

ggplot(foo_tabla,aes(x=foo_modules,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(labe
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 20,
                       space = "Lab",
                       na.value = "#FFFFFF",
                       guide = "colourbar",
                       aesthetics = "fill"
  ) + coord_fixed() + theme_bw()
```


C. variabilis bullate 5
C. variabilis 8
C. variabilis 7
C. variabilis 6
C. variabilis 5
C. variabilis 4
C. variabilis 3
C. variabilis 1
C. transcaspica sorediate 2
C. transcaspica 2
C. sp. sorediate 2
C. sp. gr 1
C. sp. asia 2
C. sp. 8
C. sp. 6
C. sp. 4
C. sp. 3
C. sp teruel 7
C. molariformis 2
C. microstepossa 8
C. microstepossa 7
C. microstepossa 4
C. microsp.B 5
C. microsp.A 1
C. micromontana 2
C. micro 8
C. micro 7
C. micro 6
C. micro 4
C. micro 3
C. micro 1
C. medit2 3
C. medit1 7
C. medit1 3
C. medit1 1
C. medit 3 1
C. medit 1
C. helygeoides? 6
C. helygeoides? 4
C. helygeoides? 3
C. helygeoides? 1
C. erodens 8
C. erodens 5
C. erodens 4
C. erodens 3
C. erodens 1
C. disjuncta 2
C. diphyodes? 8
C. diphyodes? 7
C. diphyodes_micro 7
C. diphyodes_micro 6
C. diphyodes_micro 4
C. diphyodes 8
C. diphyodes 6
C. diphyodes 4
C. diphyodes 2
C. concreticola 2
C. circumalbata candida 3
C. circumalbata 8
C. circumalbata 7
C. circumalbata 5
C. circumalbata 4
C. circumalbata 3
C. circumalbata 1
C. chalybaea 8
C. chalybaea 7
C. chalybaea 6
C. chalybaea 5
C. chalybaea 4
C. chalybaea 3
C. chalybaea 1
C. cf. micromarina 3
C. cf. circumalbata 1
C. cf albopruinosa 3
C. burgos_7 8
C. blastidiate 2
C. badioreagens 5
C. badioreagens 1
C. alociza_non_oleose 7
C. alociza_non_oleose 6
C. alociza_non_oleose 3
C. alociza 8
C. alociza 7
C. alociza 6
C. alociza 5
C. alociza 4
C. alociza 3
C. alociza 1
C. albovariegata 2
C. albopustulata 1
C. albopruinosa 8
C. albopruinosa 7
C. albopruinosa 6
C. albopruinosa 5
C. albopruinosa 4
C. albopruinosa 1
Var1
0
10
20
30
foo_modules
value
30
20
10
0

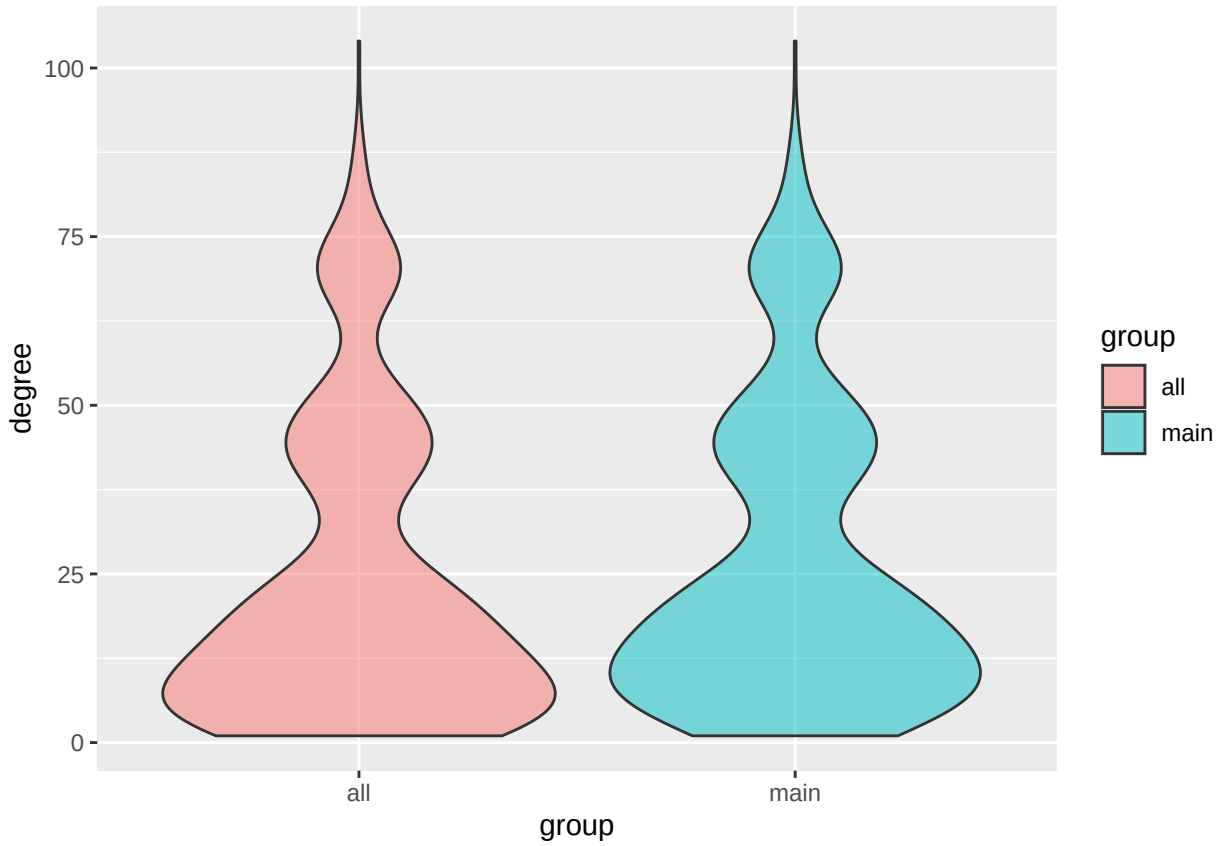
```
assocstats(table(paste(condensed_baps_adm[as.character(names(foo_modules)),2],regions$pamobject$clusteri
```

```
##                     X^2   df P(> X^2)
## Likelihood Ratio 2270.6 3135        1
## Pearson          7258.0 3135        0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.956
## Cramer's V        : 0.57
```

```
#kable(chisq.residuals(table(condensed_baps_adm$Species,condensed_baps_adm$name)))

coindep_test(table(paste(condensed_baps_adm[as.character(names(foo_modules)),2],regions$pamobject$cluste
```

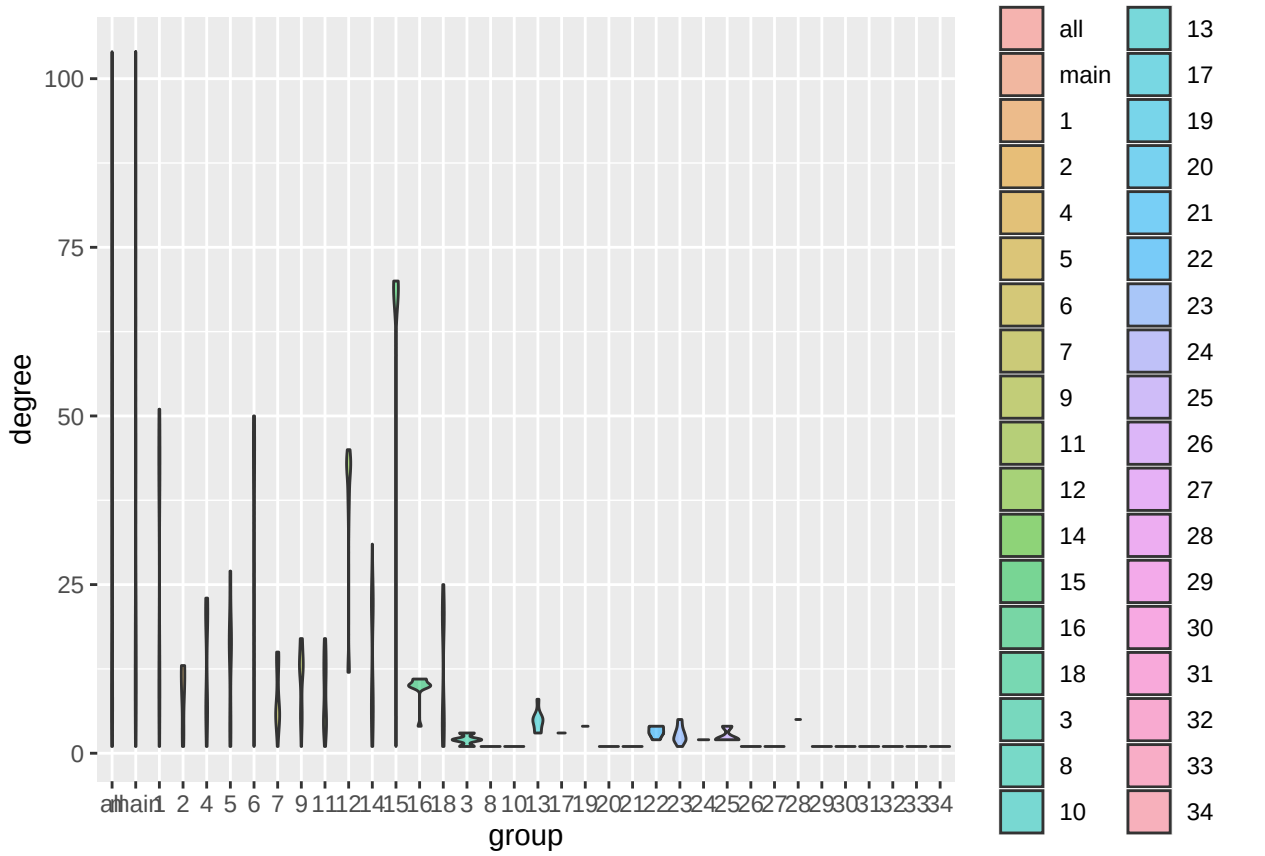
```
##
##  Permutation test for conditional independence
##
## data:  table(paste(condensed_baps_adm[as.character(names(foo_modules)),     2], regions$pamobject$clu
## f(x) = 25.923, p-value < 2.2e-16
```

```
#
# Boxplots
#
i="all"
values<-cbind(i,degree(uni_graph2, mode = "out"),closeness(uni_graph2, mode = "total", normalized = TRU
i="main"
foo<-induced_subgraph(uni_graph2,which(color_clusters%in%c(1,2,4,5,6,7,9,11,12,14,15,16,18)))
values<-rbind(values,cbind(i,degree(foo, mode = "out"),closeness(foo, mode = "total", normalized = TRUE)
for (i in c(1,2,4,5,6,7,9,11,12,14,15,16,18,3,8,10,13,17,19:34))
{
  foo<-induced_subgraph(uni_graph2,which(color_clusters==i))
  # Computing centrality measures for each vertex
values<-rbind(values,cbind(i,degree(foo, mode = "out"),closeness(foo, mode = "total", normalized = TRUE)
}
#colnames(values)<-c("group","degree","close","between")
values<-
  data.frame(group=factor(values[,1],
                          levels=c("all","main",1,2,4,5,6,7,9,11,12,14,15,16,18,3,8,10,13,17,19:34)),
             degree=as.numeric(values[,2]),close=as.numeric(values[,3]),between=as.numeric(values[,4]))
ggplot(values[values$group%in%c("all","main"),],aes(x=group,y=degree,fill=group))+geom_violin(alpha=0.5)
```



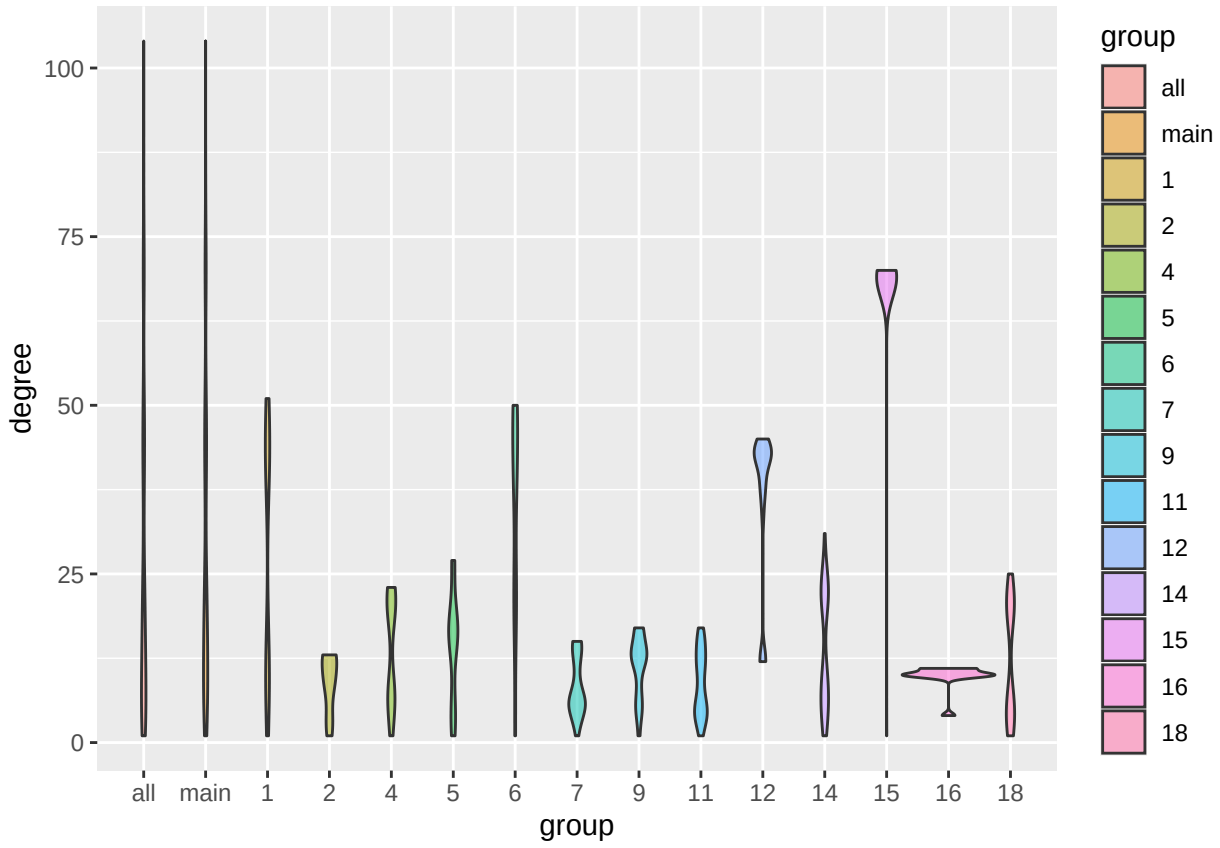

```
ggplot(values,aes(x=group,y=degree,fill=group))+geom_violin(alpha=0.5,width=1.25)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

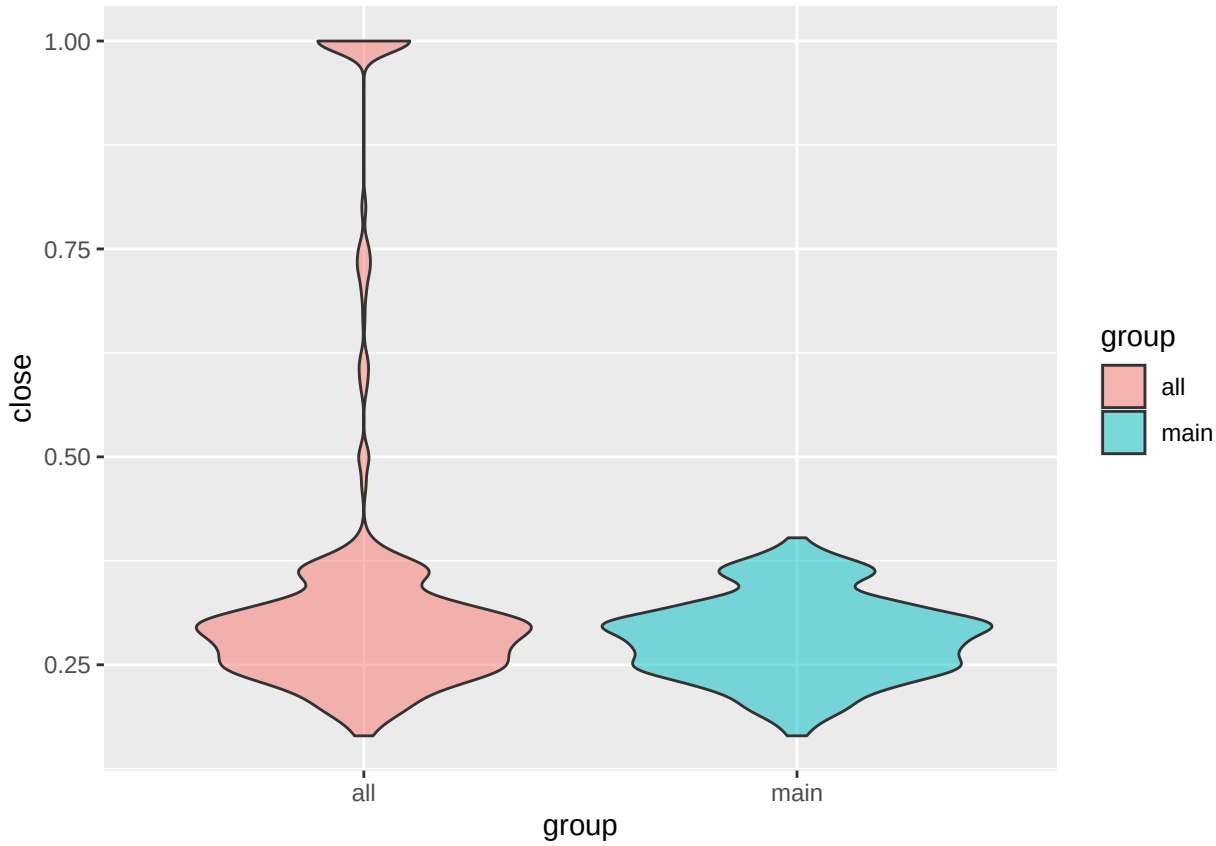



```
ggplot(values[values$group%in%c("all","main",1,2,4,5,6,7,9,11,12,14,15,16,18),],
       aes(x=group,y=degree,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

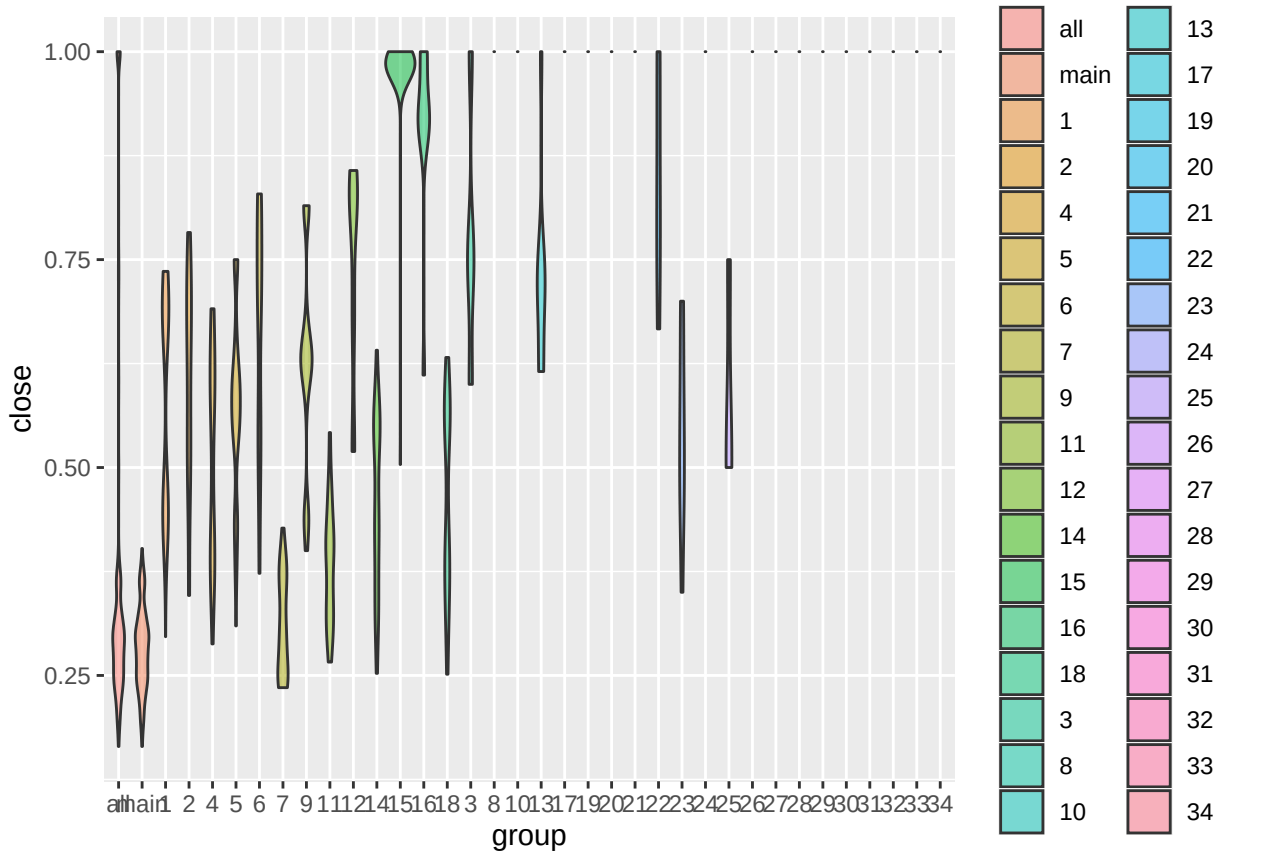



```
#
ggplot(values[values$group%in%c("all","main"),],aes(x=group,y=close,fill=group))+geom_violin(alpha=0.5)
```



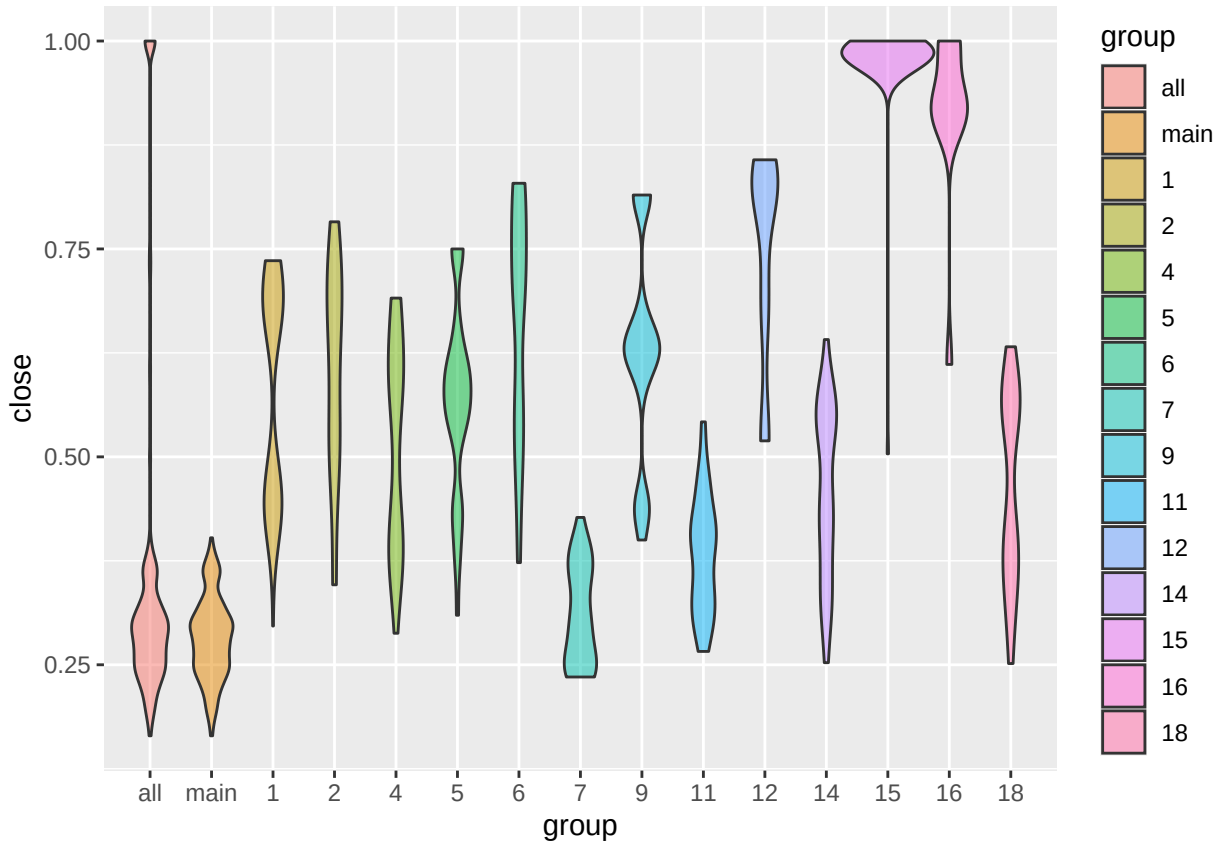

```
ggplot(values,aes(x=group,y=close,fill=group))+geom_violin(alpha=0.5,width=1.25)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```




```
ggplot(values[values$group%in%c("all","main",1,2,4,5,6,7,9,11,12,14,15,16,18),],
       aes(x=group,y=close,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```




```
values$close
```

```
##    [1] 0.3084833 0.2474227 0.7500000 0.2193784 0.2730997 0.2670227 0.2437043
##    [8] 0.2835539 0.2671416 1.0000000 0.2638522 0.3387916 1.0000000 1.0000000
##   [15] 0.3134796 0.2827521 1.0000000 0.2961500 0.2770083 0.2831524 0.2613240
##   [22] 0.2613240 0.7500000 0.1973035 1.0000000 0.2970297 0.2970297 0.2721088
##   [29] 0.2568493 0.2965892 0.2568493 0.2922552 0.2970297 0.2877698 0.2425222
##   [36] 0.2492730 0.2426203 0.3025719 0.2568493 0.2568493 0.2922552 0.2439024
##   [43] 0.2568493 0.3255562 0.2884615 0.3255562 0.3255562 0.3033367 0.1969150
##   [50] 0.2710027 0.2828854 0.2732240 0.3799873 0.3001501 0.3612282 0.2732240
##   [57] 0.2478315 0.2302379 0.2481390 0.2702703 0.3255562 0.2477291 0.2732240
##   [64] 0.3007519 0.2539145 0.2556455 0.2027712 0.2027712 0.2670227 0.6000000
##   [71] 0.3134796 0.3134796 0.2408671 0.2714932 0.2436054 0.2894356 0.1816530
##   [78] 0.3255562 0.3255562 0.3255562 0.2976190 0.2954210 0.3033367 0.3816794
##   [85] 0.2954210 0.2714932 0.3095975 0.3612282 0.2670227 0.2193784 1.0000000
##   [92] 0.2193784 1.0000000 0.2193784 0.2193784 1.0000000 0.3201708 0.3225806
##   [99] 0.2439024 0.3215434 0.2278769 0.2436054 0.2346500 0.2346500 0.3500583
##  [106] 0.2902758 0.2436054 0.3293085 0.2902758 0.2254791 0.2154399 0.2897151
##  [113] 0.2604167 0.1932990 0.2897151 0.2954210 0.2979146 0.2439024 0.2752294
##  [120] 0.2408671 0.2588438 0.2824859 0.2898551 0.2955665 0.2309469 0.2214022
##  [127] 0.2827521 0.3733665 0.3612282 0.3836317 0.2732240 0.2652520 0.2752294
##  [134] 0.2400960 0.3007519 0.2400960 0.3224073 0.2962963 1.0000000 0.1837110
##  [141] 0.3224073 0.2400960 0.3488372 0.3488372 0.2730997 0.3224073 0.2863962
##  [148] 0.2373418 0.2484472 0.1998002 0.3612282 0.3634161 0.2400960 0.2480364
##  [155] 0.1988731 0.3612282 0.2246350 0.2535926 0.3255562 0.2976190 0.3799873
##  [162] 0.3623188 0.2714932 0.2962963 0.2590674 0.3095975 0.1857010 0.3757044
##  [169] 0.2216476 0.3614458 0.3255562 0.2477291 0.3612282 0.3612282 0.2310358
```

```
##  [176] 0.2310358 0.3612282 0.2214022 1.0000000 0.2876318 0.2876318 0.3010537
##  [183] 0.2400960 0.2656042 0.3001501 0.2670227 0.2962963 0.2443992 0.6153846
##  [190] 0.2962963 0.3790272 0.7272727 0.1861620 1.0000000 1.0000000 0.2286585
##  [197] 0.2568493 0.2976190 0.2193784 0.2714932 0.2193784 1.0000000 0.2590674
##  [204] 1.0000000 0.2590674 0.2748511 0.2235469 0.2748511 0.3656307 0.2748511
##  [211] 0.2748511 0.2671416 0.2108963 0.2897151 0.2235469 0.2235469 0.2748511
##  [218] 0.2897151 0.2954210 0.2235469 0.2748511 0.2985075 0.3260870 0.3260870
##  [225] 0.3683241 0.3722084 0.2730997 0.3612282 0.3799873 0.5833333 0.3647416
##  [232] 0.2974715 0.3050330 0.2858504 0.3041054 0.2737226 0.2976190 0.3799873
##  [239] 1.0000000 0.3634161 0.3731343 0.2437043 0.3024194 0.2293578 0.3757044
##  [246] 0.3757044 0.2929688 0.3064351 0.3612282 0.1996672 0.2031832 0.2460025
##  [253] 0.3612282 0.2400960 0.3833866 0.3084833 0.2621232 0.3259098 1.0000000
##  [260] 0.3612282 0.2730997 0.3612282 0.2977667 1.0000000 0.3262643 0.3680982
##  [267] 0.3799873 0.4026846 0.3893576 0.3194888 0.3361345 0.2531646 0.3024194
##  [274] 0.2748511 0.2748511 0.3710575 0.3612282 0.2443992 0.2801120 0.2985075
##  [281] 0.2770083 0.2944063 0.3687769 0.2944063 0.3612282 0.3013561 0.2877698
##  [288] 0.3833866 0.2646670 0.3048780 0.3500000 0.2535926 0.5000000 0.7000000
##  [295] 0.4666667 0.7000000 0.2973241 0.3731343 0.2437043 0.2437043 0.1962067
##  [302] 0.2535926 0.3054990 0.2373418 0.2702703 0.3324100 0.2437043 0.2702703
##  [309] 0.2437043 0.3324100 0.2957122 0.2247191 0.2436054 0.2436054 0.2497918
##  [316] 0.2445985 0.3225806 0.2285714 0.5000000 0.1968504 0.2497918 0.2727273
##  [323] 0.2727273 0.3225806 0.3201708 0.7500000 0.3054990 0.3084833 0.3225806
##  [330] 0.3058104 0.2957122 0.3134796 0.3488372 0.3519062 1.0000000 0.3051882
##  [337] 0.2497918 0.3054990 0.3054990 0.2957122 0.3208556 0.3305785 0.2476269
##  [344] 0.2897151 0.3051882 0.2761160 0.2976190 0.1990710 0.2976190 0.2957122
##  [351] 0.2954210 0.2476269 1.0000000 0.3025719 0.2954210 0.2727273 0.3028773
##  [358] 0.2476269 0.2475248 0.2954210 0.3134796 0.2727273 0.2282237 0.2954210
##  [365] 0.3095975 0.2954210 0.5000000 0.2444988 0.3612282 0.2955665 0.3488372
##  [372] 0.3612282 1.0000000 0.3412969 0.3838772 0.3488372 0.2535926 0.2497918
##  [379] 0.3612282 0.2962963 0.2497918 0.3033367 1.0000000 0.3033367 0.2976190
##  [386] 0.2727273 0.3612282 0.2954210 0.3034901 1.0000000 0.3134796 0.2954210
##  [393] 0.2575107 0.2468120 0.1990710 0.3612282 0.2535926 0.8000000 0.2976190
##  [400] 0.3612282 0.2252252 0.1816530 0.3255562 0.1990710 0.3033367 0.2436054
##  [407] 0.2962963 0.3033367 0.2150538 0.2962963 0.2670227 1.0000000 0.2791996
##  [414] 0.2727273 0.2532714 0.3189793 0.3612282 0.3612282 0.3612282 0.2958580
##  [421] 0.3506721 0.3612282 0.2902758 0.2897151 0.2535926 0.2954210 0.3084833
##  [428] 0.1996672 0.3893576 0.3134796 0.3084833 0.3084833 0.2954210 0.3084833
##  [435] 0.3084833 0.2902758 0.2954210 0.2954210 0.3506721 0.3084833 0.3391747
##  [442] 0.2284844 0.4026846 0.2897151 0.2897151 0.2340094 0.2964427 0.2897151
##  [449] 0.2801120 0.2954210 0.2954210 0.3084833 0.3255562 0.2770083 0.3255562
##  [456] 0.3255562 0.1996672 0.3084833 0.2476269 0.1667593 0.1998668 0.1998668
##  [463] 0.2535926 0.2536998 0.2024291 0.2535926 1.0000000 0.2408671 0.2583979
##  [470] 0.1747234 0.3134796 0.3255562 0.1932990 0.2108963 0.3612282 0.3612282
##  [477] 0.7500000 0.2436054 0.2262443 0.3612282 0.3612282 0.1816530 0.2216476
##  [484] 0.2752294 0.3255562 0.3612282 0.2962963 0.3033367 0.3033367 0.2340094
##  [491] 1.0000000 0.2207506 1.0000000 0.2437043 1.0000000 0.2150538 1.0000000
##  [498] 0.3255562 1.0000000 0.3612282 0.2730997 0.3973510 0.2400960 0.2400960
##  [505] 0.2340094 1.0000000 1.0000000 0.2730997 0.1988731 0.3255562 0.2188982
##  [512] 0.2710027 0.2063983 0.2401922 0.3255562 0.2400960 1.0000000 0.2400960
##  [519] 0.2400960 0.2445985 0.2702703 0.1990710 0.1835424 0.2702703 0.2702703
##  [526] 0.2188982 0.2876318 0.2876318 0.3612282 0.2535926 0.2535926 0.3612282
##  [533] 0.2535926 0.2824859 0.2278769 0.3255562 0.2477291 0.2425222 0.2425222
##  [540] 0.2816901 0.1953125 0.2748511 0.3612282 0.2539145 0.2748511 0.2748511
##  [547] 0.2748511 0.2748511 0.2748511 1.0000000 0.5833333 1.0000000 0.2235469
```

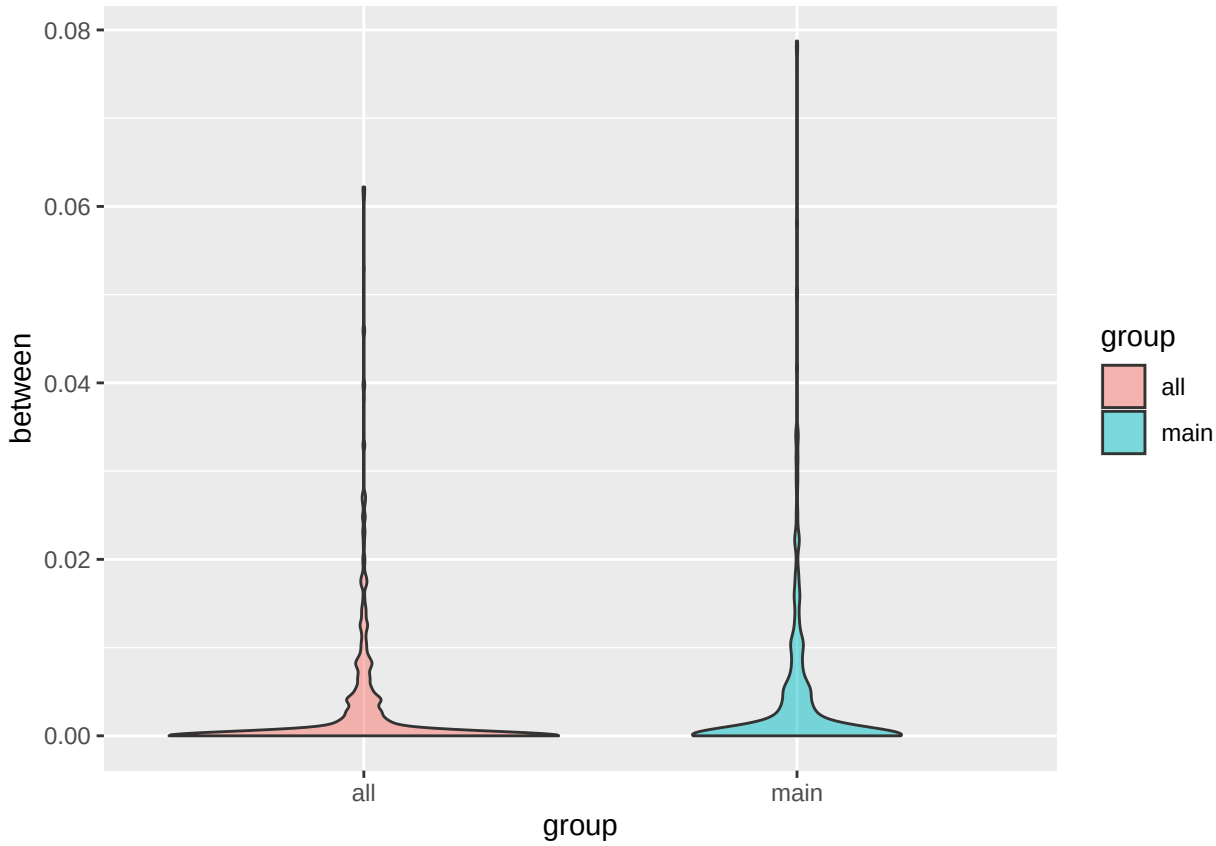
```
##  [554] 0.2235469 0.2053388 0.2535926 0.2535926 0.2670227 0.1667593 0.2235469
##  [561] 0.2278769 0.2770083 0.2053388 0.1667593 0.2816901 0.2235469 0.2770083
##  [568] 0.2670227 0.2770083 0.2770083 0.2150538 0.2770083 0.2535926 0.2150538
##  [575] 0.2535926 0.2535926 0.3255562 0.2334630 0.2027712 0.2713704 0.2535926
##  [582] 0.2535926 0.3255562 0.2334630 0.3255562 0.2334630 0.3255562 0.2846300
##  [589] 0.3206841 0.2535926 0.2535926 0.3255562 0.2846300 0.3105590 0.3022670
##  [596] 0.3022670 0.3022670 0.3022670 0.3022670 0.3022670 1.0000000 0.2976190
##  [603] 0.2468120 0.3022670 0.3022670 0.2445985 0.2548853 0.3105590 0.3105590
##  [610] 0.2444988 0.2590674 1.0000000 0.3255562 0.2443992 0.2846300 0.3105590
##  [617] 0.2446982 0.2446982 0.3105590 0.2727273 1.0000000 0.3359462 0.3255562
##  [624] 0.3255562 0.3255562 0.2548853 0.3000000 0.3255562 1.0000000 0.3612282
##  [631] 0.2248033 0.1837110 0.2248033 0.2670227 1.0000000 0.4666667 0.6000000
##  [638] 1.0000000 0.6153846 0.5000000 0.5000000 0.6153846 1.0000000 0.7272727
##  [645] 0.7272727 0.7272727 0.7272727 1.0000000 1.0000000 1.0000000 1.0000000
##  [652] 0.1988731 0.2475248 0.6666667 0.8000000 0.3612282 0.2670227 0.2714932
##  [659] 0.2295333 0.1965924 0.3612282 0.2285714 0.2285714 0.2759890 0.3033367
##  [666] 0.2714932 0.3033367 1.0000000 0.1645639 0.3255562 0.2235469 0.2730997
##  [673] 0.3084833 0.3218884 0.3224073 0.2670227 0.3084833 0.2474227 0.2193784
##  [680] 0.2730997 0.2670227 0.2437043 0.2835539 0.2671416 0.2638522 0.3387916
##  [687] 0.3134796 0.2827521 0.2961500 0.2770083 0.2831524 0.2613240 0.2613240
##  [694] 0.1973035 0.2970297 0.2970297 0.2721088 0.2568493 0.2965892 0.2568493
##  [701] 0.2922552 0.2970297 0.2877698 0.2425222 0.2492730 0.2426203 0.3025719
##  [708] 0.2568493 0.2568493 0.2922552 0.2439024 0.2568493 0.3255562 0.2884615
##  [715] 0.3255562 0.3255562 0.3033367 0.1969150 0.2710027 0.2828854 0.2732240
##  [722] 0.3799873 0.3001501 0.3612282 0.2732240 0.2478315 0.2302379 0.2481390
##  [729] 0.2702703 0.3255562 0.2477291 0.2732240 0.3007519 0.2539145 0.2556455
##  [736] 0.2027712 0.2027712 0.2670227 0.3134796 0.3134796 0.2408671 0.2714932
##  [743] 0.2436054 0.2894356 0.1816530 0.3255562 0.3255562 0.3255562 0.2976190
##  [750] 0.2954210 0.3033367 0.3816794 0.2954210 0.2714932 0.3095975 0.3612282
##  [757] 0.2670227 0.2193784 0.2193784 0.2193784 0.2193784 0.3201708 0.3225806
##  [764] 0.2439024 0.3215434 0.2278769 0.2436054 0.2346500 0.2346500 0.3500583
##  [771] 0.2902758 0.2436054 0.3293085 0.2902758 0.2254791 0.2154399 0.2897151
##  [778] 0.2604167 0.1932990 0.2897151 0.2954210 0.2979146 0.2439024 0.2752294
##  [785] 0.2408671 0.2588438 0.2824859 0.2898551 0.2955665 0.2309469 0.2214022
##  [792] 0.2827521 0.3733665 0.3612282 0.3836317 0.2732240 0.2652520 0.2752294
##  [799] 0.2400960 0.3007519 0.2400960 0.3224073 0.2962963 0.1837110 0.3224073
##  [806] 0.2400960 0.3488372 0.3488372 0.2730997 0.3224073 0.2863962 0.2373418
##  [813] 0.2484472 0.1998002 0.3612282 0.3634161 0.2400960 0.2480364 0.1988731
##  [820] 0.3612282 0.2246350 0.2535926 0.3255562 0.2976190 0.3799873 0.3623188
##  [827] 0.2714932 0.2962963 0.2590674 0.3095975 0.1857010 0.3757044 0.2216476
##  [834] 0.3614458 0.3255562 0.2477291 0.3612282 0.3612282 0.2310358 0.2310358
##  [841] 0.3612282 0.2214022 0.2876318 0.2876318 0.3010537 0.2400960 0.2656042
##  [848] 0.3001501 0.2670227 0.2962963 0.2443992 0.2962963 0.3790272 0.1861620
##  [855] 0.2286585 0.2568493 0.2976190 0.2193784 0.2714932 0.2193784 0.2590674
##  [862] 0.2590674 0.2748511 0.2235469 0.2748511 0.3656307 0.2748511 0.2748511
##  [869] 0.2671416 0.2108963 0.2897151 0.2235469 0.2235469 0.2748511 0.2897151
##  [876] 0.2954210 0.2235469 0.2748511 0.2985075 0.3260870 0.3260870 0.3683241
##  [883] 0.3722084 0.2730997 0.3612282 0.3799873 0.3647416 0.2974715 0.3050330
##  [890] 0.2858504 0.3041054 0.2737226 0.2976190 0.3799873 0.3634161 0.3731343
##  [897] 0.2437043 0.3024194 0.2293578 0.3757044 0.3757044 0.2929688 0.3064351
##  [904] 0.3612282 0.1996672 0.2031832 0.2460025 0.3612282 0.2400960 0.3833866
##  [911] 0.3084833 0.2621232 0.3259098 0.3612282 0.2730997 0.3612282 0.2977667
##  [918] 0.3262643 0.3680982 0.3799873 0.4026846 0.3893576 0.3194888 0.3361345
##  [925] 0.2531646 0.3024194 0.2748511 0.2748511 0.3710575 0.3612282 0.2443992
```

```
##  [932] 0.2801120 0.2985075 0.2770083 0.2944063 0.3687769 0.2944063 0.3612282
##  [939] 0.3013561 0.2877698 0.3833866 0.2646670 0.3048780 0.2535926 0.2973241
##  [946] 0.3731343 0.2437043 0.2437043 0.1962067 0.2535926 0.3054990 0.2373418
##  [953] 0.2702703 0.3324100 0.2437043 0.2702703 0.2437043 0.3324100 0.2957122
##  [960] 0.2247191 0.2436054 0.2436054 0.2497918 0.2445985 0.3225806 0.2285714
##  [967] 0.1968504 0.2497918 0.2727273 0.2727273 0.3225806 0.3201708 0.3054990
##  [974] 0.3084833 0.3225806 0.3058104 0.2957122 0.3134796 0.3488372 0.3519062
##  [981] 0.3051882 0.2497918 0.3054990 0.3054990 0.2957122 0.3208556 0.3305785
##  [988] 0.2476269 0.2897151 0.3051882 0.2761160 0.2976190 0.1990710 0.2976190
##  [995] 0.2957122 0.2954210 0.2476269 0.3025719 0.2954210 0.2727273 0.3028773
## [1002] 0.2476269 0.2475248 0.2954210 0.3134796 0.2727273 0.2282237 0.2954210
## [1009] 0.3095975 0.2954210 0.2444988 0.3612282 0.2955665 0.3488372 0.3612282
## [1016] 0.3412969 0.3838772 0.3488372 0.2535926 0.2497918 0.3612282 0.2962963
## [1023] 0.2497918 0.3033367 0.3033367 0.2976190 0.2727273 0.3612282 0.2954210
## [1030] 0.3034901 0.3134796 0.2954210 0.2575107 0.2468120 0.1990710 0.3612282
## [1037] 0.2535926 0.2976190 0.3612282 0.2252252 0.1816530 0.3255562 0.1990710
## [1044] 0.3033367 0.2436054 0.2962963 0.3033367 0.2150538 0.2962963 0.2670227
## [1051] 0.2791996 0.2727273 0.2532714 0.3189793 0.3612282 0.3612282 0.3612282
## [1058] 0.2958580 0.3506721 0.3612282 0.2902758 0.2897151 0.2535926 0.2954210
## [1065] 0.3084833 0.1996672 0.3893576 0.3134796 0.3084833 0.3084833 0.2954210
## [1072] 0.3084833 0.3084833 0.2902758 0.2954210 0.2954210 0.3506721 0.3084833
## [1079] 0.3391747 0.2284844 0.4026846 0.2897151 0.2897151 0.2340094 0.2964427
## [1086] 0.2897151 0.2801120 0.2954210 0.2954210 0.3084833 0.3255562 0.2770083
## [1093] 0.3255562 0.3255562 0.1996672 0.3084833 0.2476269 0.1667593 0.1998668
## [1100] 0.1998668 0.2535926 0.2536998 0.2024291 0.2535926 0.2408671 0.2583979
## [1107] 0.1747234 0.3134796 0.3255562 0.1932990 0.2108963 0.3612282 0.3612282
## [1114] 0.2436054 0.2262443 0.3612282 0.3612282 0.1816530 0.2216476 0.2752294
## [1121] 0.3255562 0.3612282 0.2962963 0.3033367 0.3033367 0.2340094 0.2207506
## [1128] 0.2437043 0.2150538 0.3255562 0.3612282 0.2730997 0.3973510 0.2400960
## [1135] 0.2400960 0.2340094 0.2730997 0.1988731 0.3255562 0.2188982 0.2710027
## [1142] 0.2063983 0.2401922 0.3255562 0.2400960 0.2400960 0.2400960 0.2445985
## [1149] 0.2702703 0.1990710 0.1835424 0.2702703 0.2702703 0.2188982 0.2876318
## [1156] 0.2876318 0.3612282 0.2535926 0.2535926 0.3612282 0.2535926 0.2824859
## [1163] 0.2278769 0.3255562 0.2477291 0.2425222 0.2425222 0.2816901 0.1953125
## [1170] 0.2748511 0.3612282 0.2539145 0.2748511 0.2748511 0.2748511 0.2748511
## [1177] 0.2748511 0.2235469 0.2235469 0.2053388 0.2535926 0.2535926 0.2670227
## [1184] 0.1667593 0.2235469 0.2278769 0.2770083 0.2053388 0.1667593 0.2816901
## [1191] 0.2235469 0.2770083 0.2670227 0.2770083 0.2770083 0.2150538 0.2770083
## [1198] 0.2535926 0.2150538 0.2535926 0.2535926 0.3255562 0.2334630 0.2027712
## [1205] 0.2713704 0.2535926 0.2535926 0.3255562 0.2334630 0.3255562 0.2334630
## [1212] 0.3255562 0.2846300 0.3206841 0.2535926 0.2535926 0.3255562 0.2846300
## [1219] 0.3105590 0.3022670 0.3022670 0.3022670 0.3022670 0.3022670 0.3022670
## [1226] 0.2976190 0.2468120 0.3022670 0.3022670 0.2445985 0.2548853 0.3105590
## [1233] 0.3105590 0.2444988 0.2590674 0.3255562 0.2443992 0.2846300 0.3105590
## [1240] 0.2446982 0.2446982 0.3105590 0.2727273 0.3359462 0.3255562 0.3255562
## [1247] 0.3255562 0.2548853 0.3000000 0.3255562 0.3612282 0.2248033 0.1837110
## [1254] 0.2248033 0.2670227 0.1988731 0.2475248 0.3612282 0.2670227 0.2714932
## [1261] 0.2295333 0.1965924 0.3612282 0.2285714 0.2285714 0.2759890 0.3033367
## [1268] 0.2714932 0.3033367 0.1645639 0.3255562 0.2235469 0.2730997 0.3084833
## [1275] 0.3218884 0.3224073 0.2670227 0.4482759 0.4588235 0.4588235 0.4588235
## [1282] 0.6782609 0.6782609 0.7222222 0.6782609 0.7358491 0.7358491 0.2965779
## [1289] 0.4193548 0.6782609 0.6782609 0.7222222 0.4171123 0.4431818 0.4431818
## [1296] 0.7222222 0.7090909 0.6782609 0.7222222 0.6902655 0.4588235 0.7358491
## [1303] 0.6782609 0.7090909 0.7090909 0.6902655 0.4333333 0.4561404 0.6902655
```

```
## [1310] 0.6782609 0.4333333 0.6782609 0.6782609 0.4431818 0.6782609 0.4333333
## [1317] 0.6782609 0.4588235 0.4431818 0.4083770 0.6782609 0.6782609 0.6842105
## [1324] 0.7358491 0.6782609 0.7358491 0.4431818 0.6782609 0.6782609 0.4588235
## [1331] 0.6782609 0.4431818 0.4727273 0.6964286 0.7358491 0.6782609 0.4482759
## [1338] 0.4588235 0.4482759 0.4482759 0.6782609 0.4482759 0.4482759 0.6782609
## [1345] 0.6782609 0.7358491 0.4482759 0.6782609 0.6782609 0.4482759 0.4482759
## [1352] 0.4333333 0.4588235 0.4431818 0.4171123 0.4171123 0.3461538 0.7200000
## [1359] 0.7826087 0.6428571 0.5000000 0.5142857 0.5142857 0.6428571 0.7200000
## [1366] 0.7200000 0.5142857 0.3461538 0.6428571 0.6428571 0.5142857 0.7200000
## [1373] 0.7200000 0.7200000 0.5142857 0.3725490 0.5937500 0.5937500 0.4000000
## [1380] 0.3725490 0.3725490 0.3725490 0.3725490 0.6909091 0.6229508 0.6909091
## [1387] 0.5846154 0.6909091 0.5846154 0.6909091 0.3725490 0.3725490 0.5846154
## [1394] 0.5352113 0.3958333 0.5937500 0.6229508 0.6909091 0.6229508 0.3800000
## [1401] 0.4130435 0.4871795 0.4871795 0.6909091 0.4130435 0.4130435 0.4130435
## [1408] 0.6229508 0.5937500 0.5846154 0.6129032 0.6129032 0.3800000 0.2878788
## [1415] 0.6000000 0.5571429 0.6393443 0.4239130 0.7500000 0.7500000 0.6000000
## [1422] 0.7500000 0.5571429 0.5571429 0.7500000 0.7500000 0.6000000 0.4431818
## [1429] 0.5571429 0.6500000 0.5571429 0.3095238 0.6000000 0.6610169 0.6000000
## [1436] 0.5652174 0.6190476 0.5342466 0.4382022 0.3900000 0.3900000 0.5571429
## [1443] 0.4382022 0.5571429 0.6000000 0.6000000 0.5571429 0.4382022 0.5200000
## [1450] 0.5200000 0.3644860 0.6000000 0.6000000 0.7500000 0.7000000 0.7078652
## [1457] 0.7000000 0.7000000 0.7682927 0.7000000 0.3727811 0.4921875 0.4921875
## [1464] 0.7682927 0.4921875 0.7000000 0.3727811 0.5779817 0.4921875 0.4921875
## [1471] 0.7000000 0.4921875 0.8289474 0.5625000 0.8289474 0.8289474 0.8289474
## [1478] 0.7078652 0.4172185 0.5625000 0.5625000 0.8289474 0.5625000 0.8289474
## [1485] 0.7000000 0.7000000 0.8289474 0.8289474 0.5625000 0.7682927 0.7682927
## [1492] 0.7682927 0.7682927 0.7000000 0.3727811 0.4172185 0.7682927 0.7682927
## [1499] 0.8289474 0.8289474 0.8289474 0.8289474 0.8289474 0.8289474 0.5625000
## [1506] 0.5625000 0.7000000 0.5625000 0.5625000 0.7000000 0.7000000 0.7000000
## [1513] 0.4921875 0.7682927 0.4921875 0.7682927 0.5625000 0.7000000 0.2417582
## [1520] 0.2500000 0.3577236 0.3055556 0.3859649 0.3859649 0.3055556 0.3859649
## [1527] 0.3859649 0.3859649 0.4000000 0.2716049 0.3188406 0.3577236 0.2500000
## [1534] 0.2500000 0.3464567 0.3859649 0.2500000 0.2500000 0.2417582 0.2417582
## [1541] 0.2933333 0.3636364 0.3055556 0.2417582 0.2417582 0.3606557 0.2933333
## [1548] 0.3577236 0.2933333 0.2352941 0.2993197 0.2993197 0.3235294 0.4000000
## [1555] 0.4271845 0.2417582 0.2666667 0.2352941 0.2352941 0.2500000 0.3636364
## [1562] 0.2716049 0.3636364 0.6470588 0.6666667 0.4400000 0.4400000 0.6666667
## [1569] 0.4000000 0.5945946 0.4400000 0.4400000 0.8148148 0.6285714 0.6285714
## [1576] 0.6285714 0.6285714 0.6285714 0.6285714 0.6285714 0.6285714 0.8148148
## [1583] 0.8148148 0.8148148 0.8148148 0.5945946 0.2914573 0.3333333 0.4000000
## [1590] 0.4000000 0.5225225 0.4603175 0.4000000 0.3945578 0.3972603 0.4603175
## [1597] 0.4360902 0.4833333 0.3295455 0.3295455 0.4202899 0.3918919 0.4296296
## [1604] 0.4172662 0.3295455 0.2660550 0.4715447 0.4027778 0.3473054 0.4793388
## [1611] 0.3295455 0.4393939 0.3222222 0.4264706 0.3020833 0.3536585 0.4172662
## [1618] 0.4000000 0.4000000 0.3431953 0.2914573 0.4603175 0.4754098 0.4754098
## [1625] 0.3068783 0.4172662 0.4172662 0.3068783 0.2660550 0.3068783 0.5420561
## [1632] 0.5420561 0.2660550 0.3536585 0.3295455 0.3020833 0.3068783 0.4172662
## [1639] 0.3945578 0.3945578 0.3670886 0.2857143 0.3670886 0.3020833 0.3431953
## [1646] 0.7826087 0.5192308 0.6750000 0.8307692 0.5192308 0.8307692 0.8307692
## [1653] 0.8307692 0.8571429 0.8307692 0.8307692 0.8307692 0.6750000 0.5192308
## [1660] 0.5684211 0.5192308 0.8571429 0.5192308 0.5192308 0.8307692 0.8307692
## [1667] 0.5192308 0.8307692 0.7297297 0.7297297 0.6750000 0.8307692 0.8307692
## [1674] 0.8307692 0.8307692 0.8307692 0.8307692 0.5192308 0.5192308 0.8307692
## [1681] 0.8307692 0.5192308 0.8307692 0.8307692 0.8307692 0.8307692 0.8307692
```

```
## [1688] 0.6750000 0.6750000 0.8307692 0.6750000 0.8307692 0.6750000 0.6750000
## [1695] 0.8307692 0.8307692 0.8307692 0.8307692 0.8307692 0.7297297 0.2525253
## [1702] 0.5681818 0.6250000 0.3731343 0.3731343 0.3184713 0.4464286 0.4464286
## [1709] 0.5494505 0.3289474 0.3289474 0.4347826 0.6410256 0.4347826 0.5494505
## [1716] 0.5494505 0.5494505 0.4347826 0.5494505 0.5555556 0.3597122 0.5494505
## [1723] 0.3184713 0.3289474 0.3246753 0.2525253 0.3246753 0.5494505 0.5494505
## [1730] 0.5494505 0.4504505 0.3184713 0.5681818 0.4201681 0.5494505 0.5494505
## [1737] 0.3184713 0.4347826 0.4201681 0.4347826 0.4347826 0.4347826 0.4347826
## [1744] 0.5494505 0.5494505 0.3731343 0.5494505 0.5494505 0.5494505 0.5494505
## [1751] 0.5494505 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1758] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1765] 1.0000000 0.9859155 0.9859155 0.9859155 0.5035971 0.9859155 0.9859155
## [1772] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1779] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1786] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1793] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1800] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1807] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1814] 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155 0.9859155
## [1821] 0.9859155 0.9859155 0.9166667 0.9166667 0.9166667 1.0000000 1.0000000
## [1828] 0.6111111 1.0000000 1.0000000 0.9166667 0.9166667 0.9166667 0.9166667
## [1835] 0.2986111 0.4018692 0.5890411 0.5890411 0.5890411 0.3909091 0.5584416
## [1842] 0.4056604 0.5584416 0.2986111 0.3333333 0.5584416 0.5584416 0.5584416
## [1849] 0.3333333 0.6323529 0.6323529 0.4257426 0.3909091 0.3909091 0.5584416
## [1856] 0.5972222 0.5584416 0.3909091 0.3873874 0.5584416 0.5584416 0.5584416
## [1863] 0.3909091 0.3909091 0.5584416 0.2529412 0.5584416 0.5584416 0.5810811
## [1870] 0.5584416 0.3307692 0.3307692 0.2514620 0.3333333 0.3771930 0.3771930
## [1877] 0.3771930 0.4574468 0.7500000 1.0000000 0.7500000 0.6000000 1.0000000
## [1884] 1.0000000 1.0000000 1.0000000 1.0000000 0.6153846 0.7272727 0.6153846
## [1891] 0.6153846 0.7272727 0.7272727 0.7272727 0.7272727 1.0000000 1.0000000
## [1898] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [1905] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.8000000 0.6666667
## [1912] 0.8000000 1.0000000 0.5833333 0.3500000 0.5000000 0.7000000 0.4666667
## [1919] 0.7000000 0.5833333 0.4666667 1.0000000 1.0000000 1.0000000 0.5000000
## [1926] 0.7500000 0.5000000 0.7500000 0.6000000 0.5000000 0.5000000 1.0000000
## [1933] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [1940] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [1947] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
```

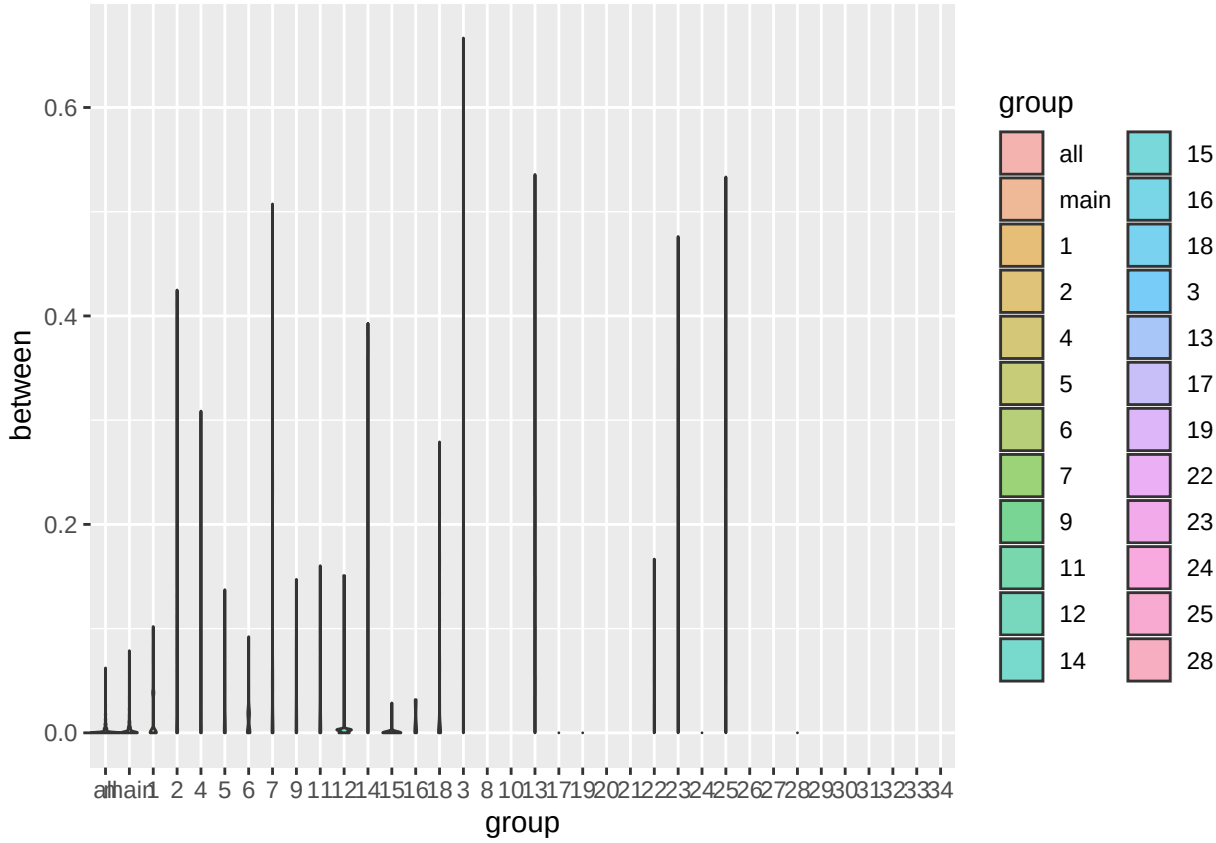
```
#
ggplot(values[values$group%in%c("all","main"),],aes(x=group,y=between,fill=group))+geom_violin(alpha=0.
```




```
ggplot(values,aes(x=group,y=between,fill=group))+geom_violin(alpha=0.5,width=1.25)
```

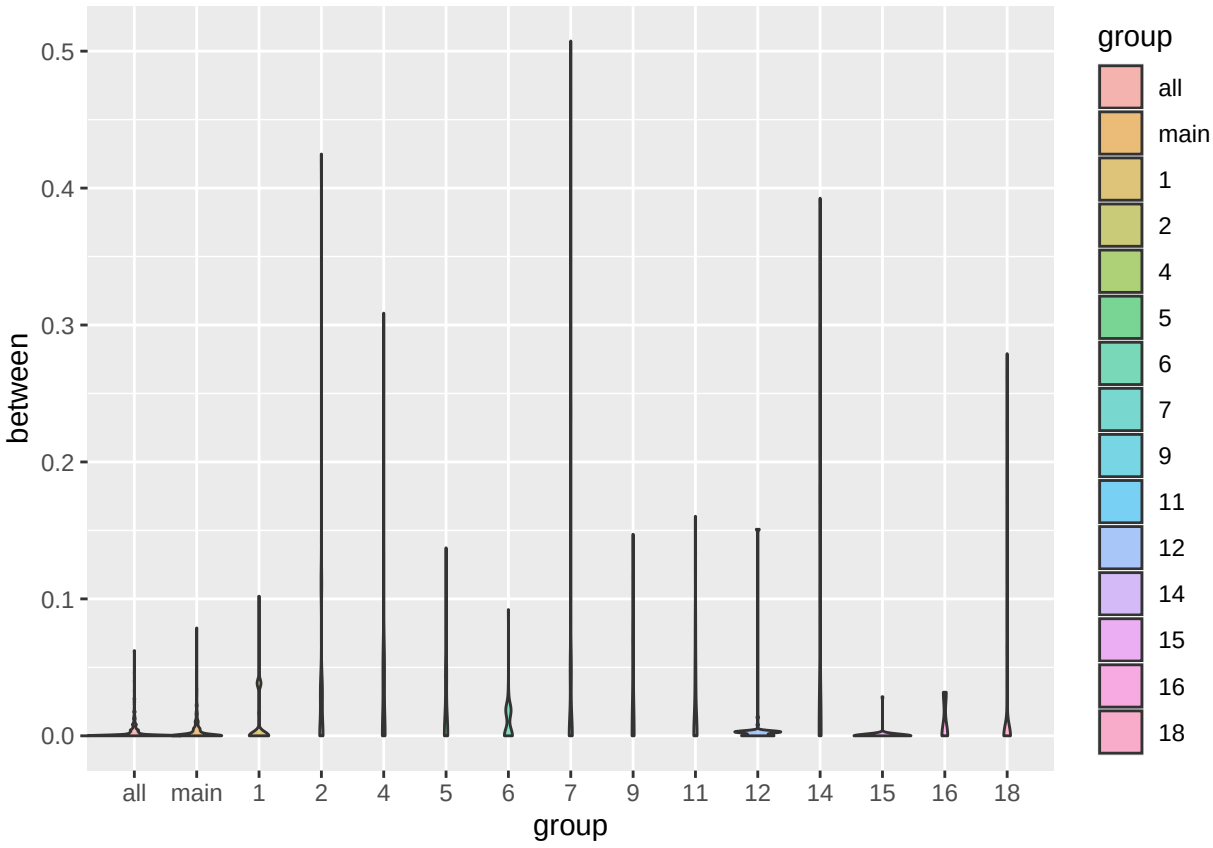
```
## Warning: Removed 24 rows containing non-finite values (stat_ydensity).
```

```
## Warning: position_dodge requires non-overlapping x intervals
```




```
ggplot(values[values$group%in%c("all","main",1,2,4,5,6,7,9,11,12,14,15,16,18),],
       aes(x=group,y=between,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



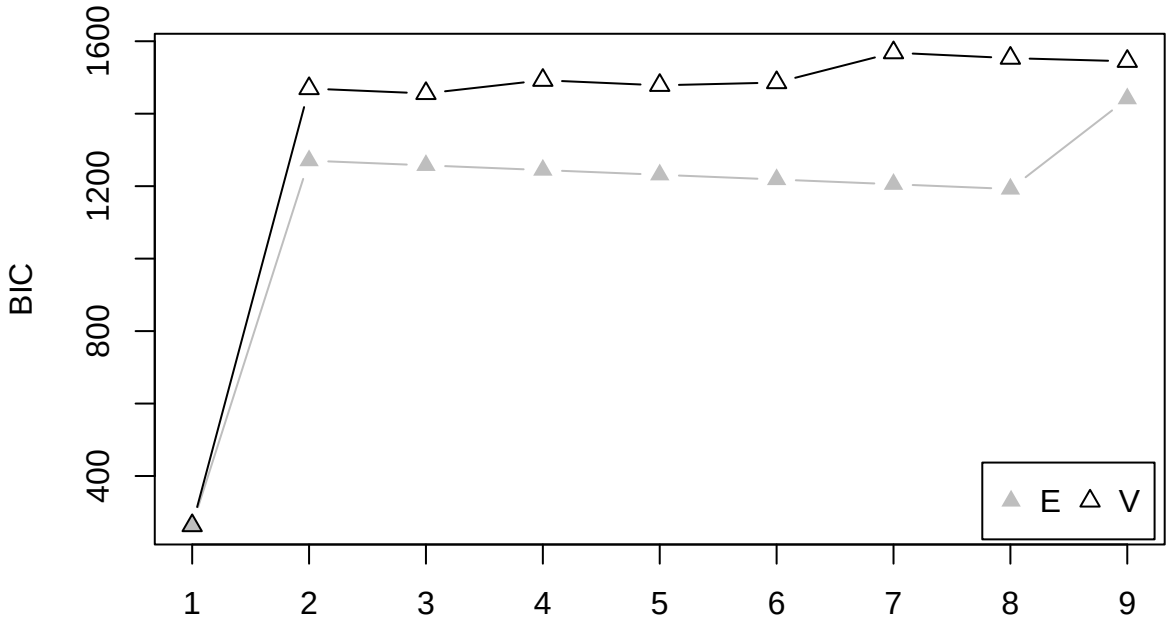

```
#
#
#
#
# Cuantos grupos de closeness hay usando una mezcla de normales?
#
library(mclust)
```

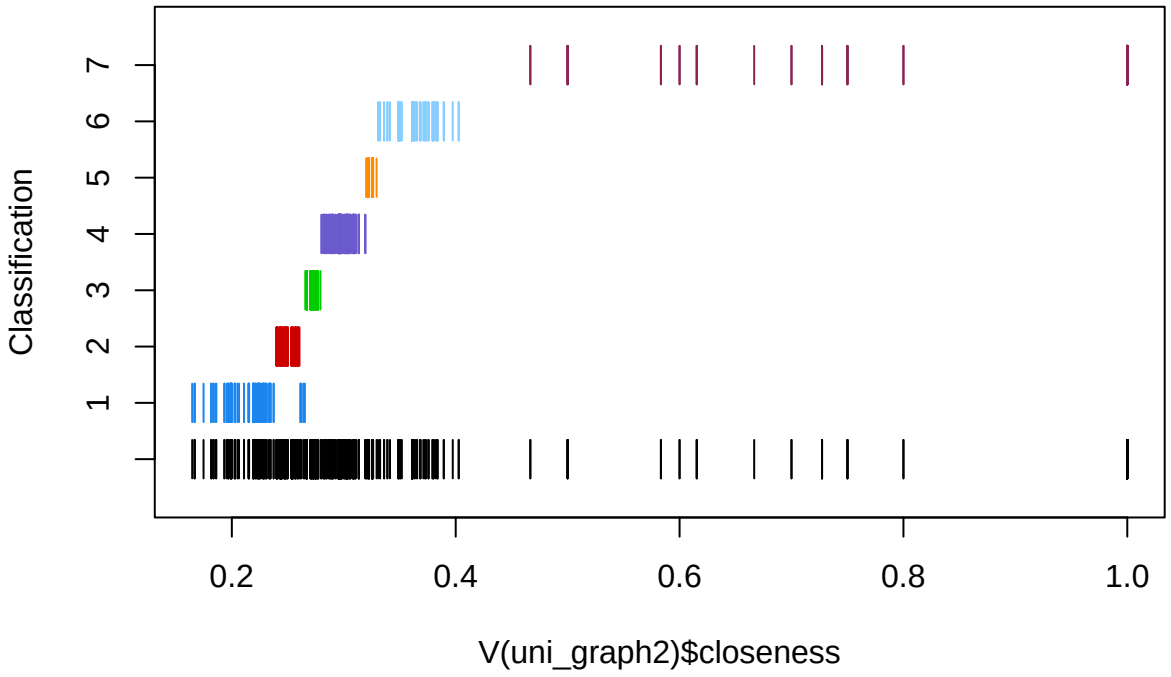
```
## Package 'mclust' version 5.4.9
## Type 'citation("mclust")' for citing this R package in publications.
```

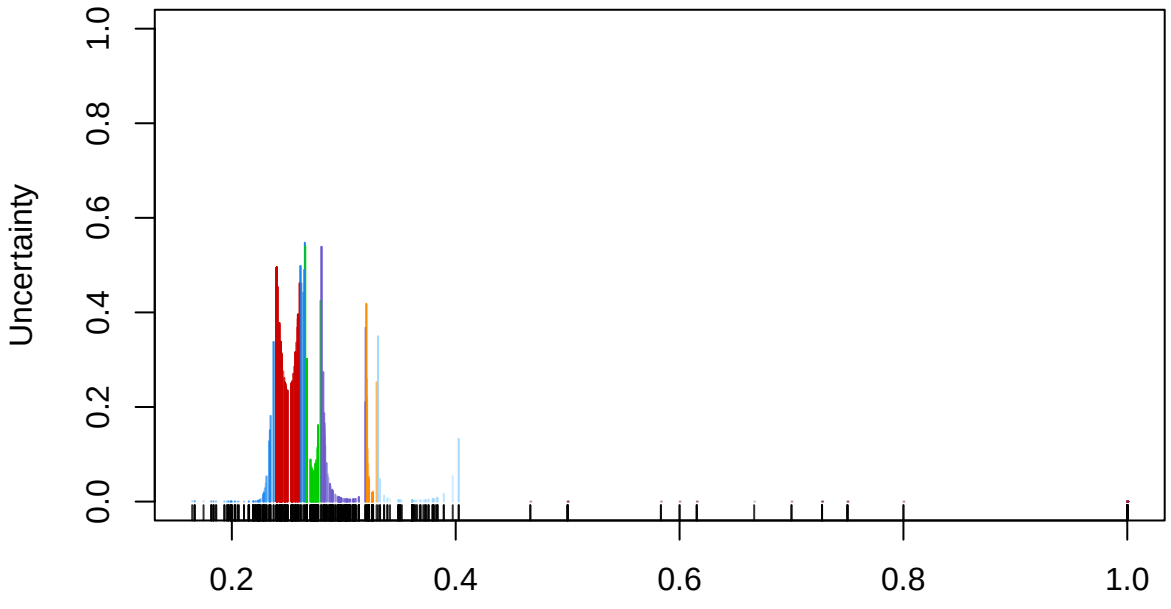
```
##
## Attaching package: 'mclust'
```

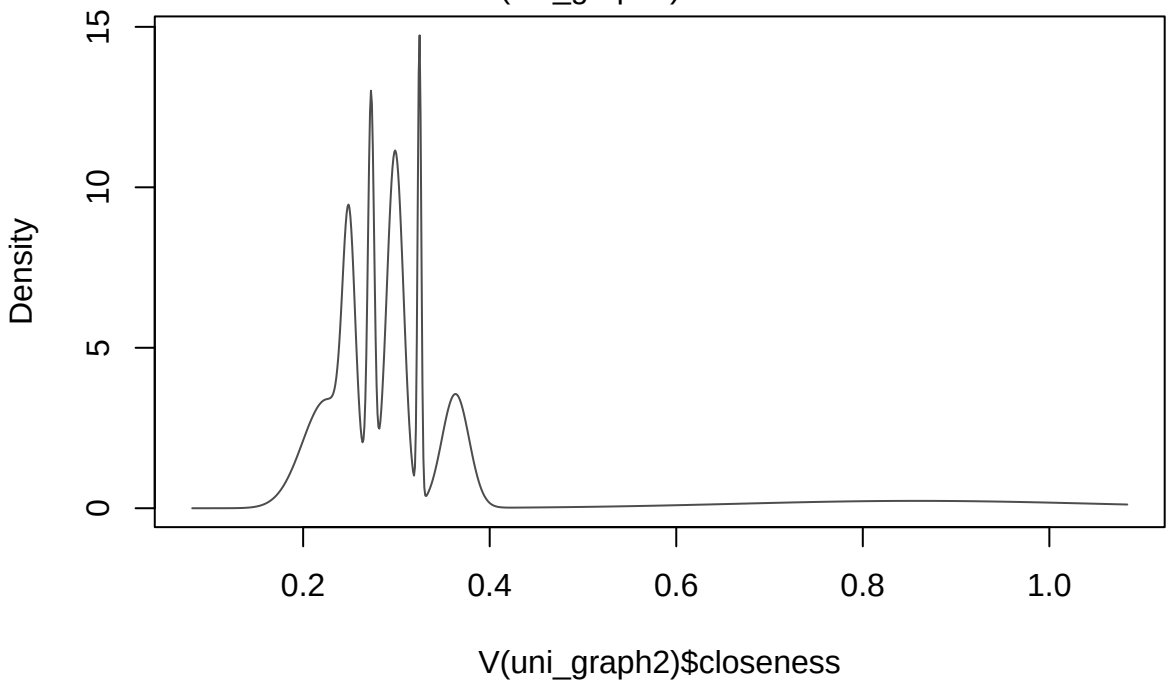
```
## The following object is masked from 'package:maps':
##
##     map
```

```
V(uni_graph2)$outdegree  <- degree(uni_graph2, mode = "out")
V(uni_graph2)$closeness  <- closeness(uni_graph2, mode = "total", normalized = TRUE)
V(uni_graph2)$betweeness <- betweenness(uni_graph2, normalized = TRUE)
clustering_closeness<-Mclust(V(uni_graph2)$closeness)
plot(clustering_closeness)
```


BIC
1600
1200
800
400
1
2
3
4
5
6
7
8
9
Number of components
E
V


Classification
1
2
3
4
5
6
7
0.2
0.4
0.6
0.8
1.0
V(uni_graph2)$closeness



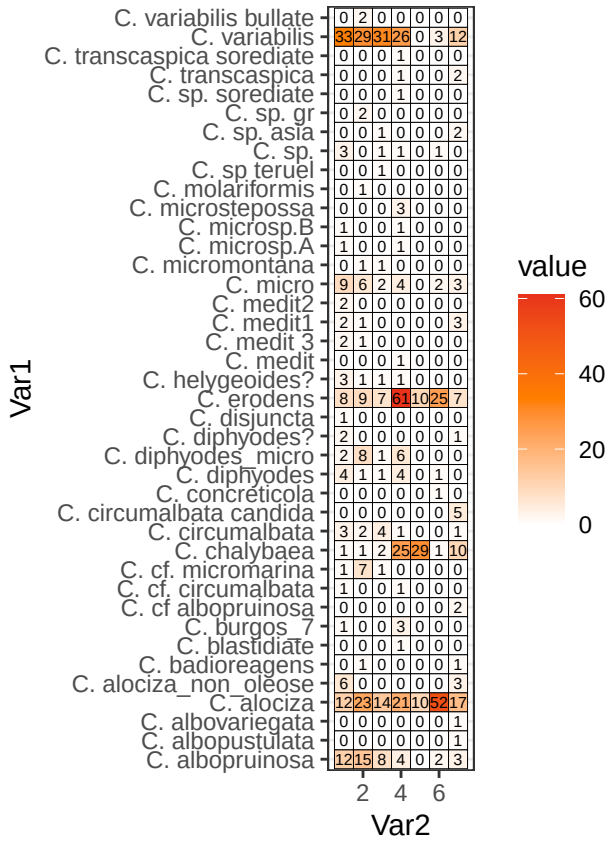



```
table(clustering_closeness$classification)
```

```
##
##   1   2   3   4   5   6   7
## 110 111  76 168  49  88  74
```

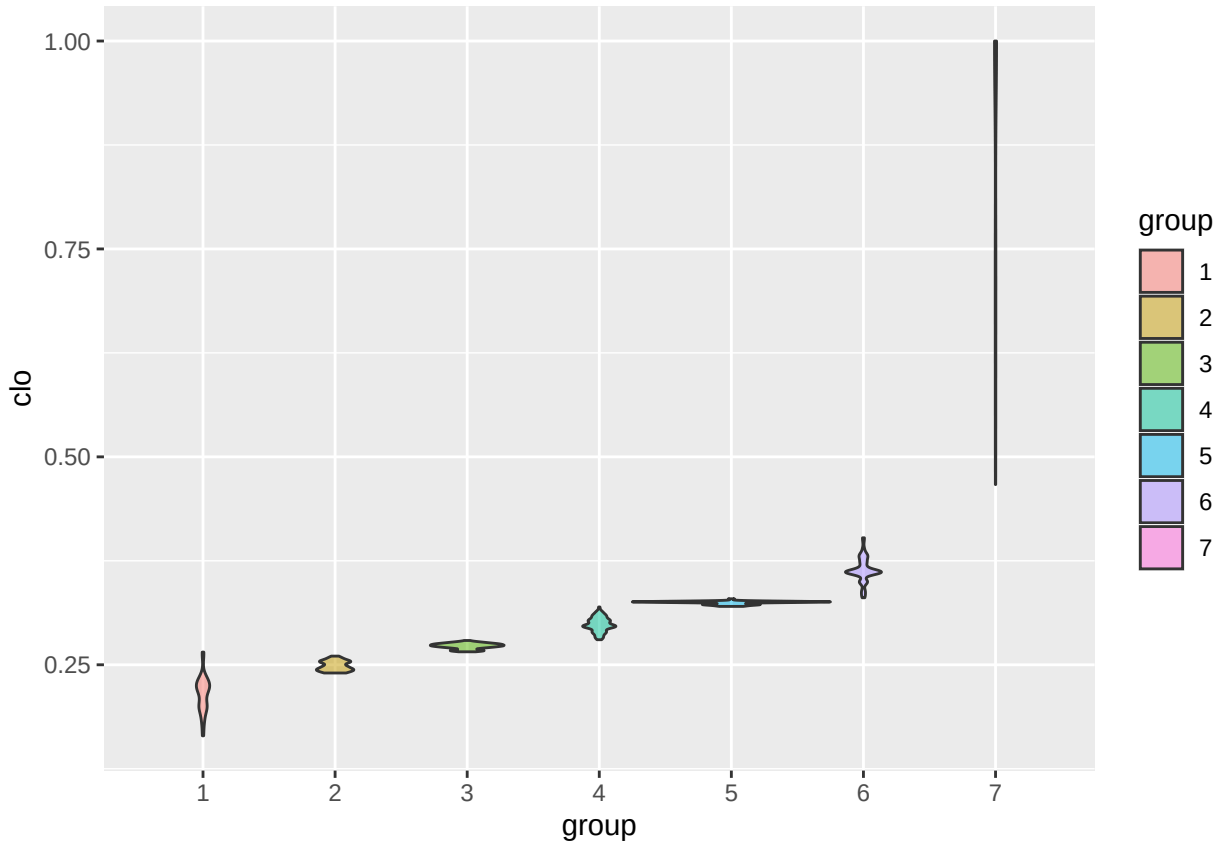
```
foo_tabla<-melt(table(condensed_baps_adm[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2),2],clustering_
ggplot(foo_tabla,aes(x=Var2,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = val
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 33,
```

```
                      space = "Lab",
                      na.value = "#FFFFFF",
                      guide = "colourbar",
                      aesthetics = "fill"
  ) + coord_fixed() + theme_bw()
```




```
ggplot(data.frame(clo = V(uni_graph2)$closeness,group=factor(clustering_closeness$classification)),aes(
```

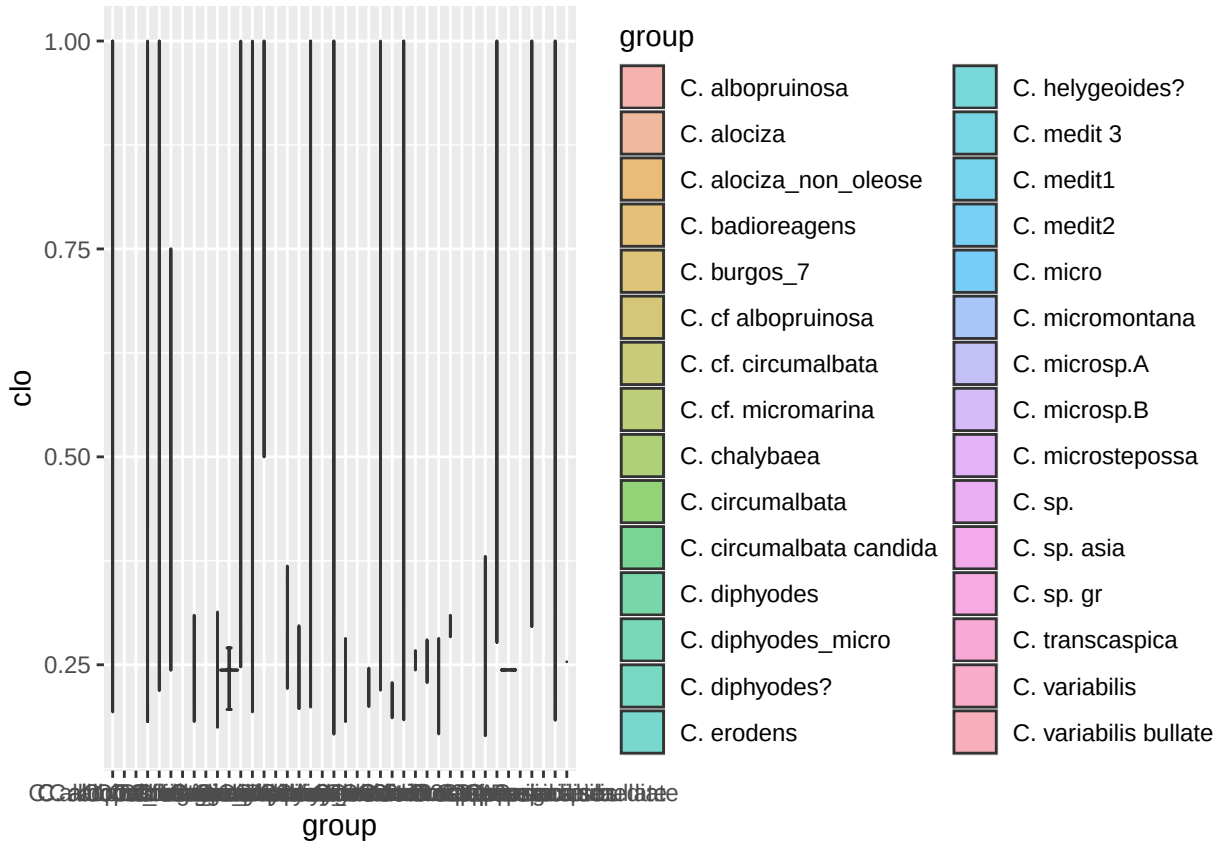
```
## Warning: position_dodge requires non-overlapping x intervals
```




```
ggplot(data.frame(clo = V(uni_graph2)$closeness,group=factor(condensed_baps_adm[sapply(strsplit(names(V
```

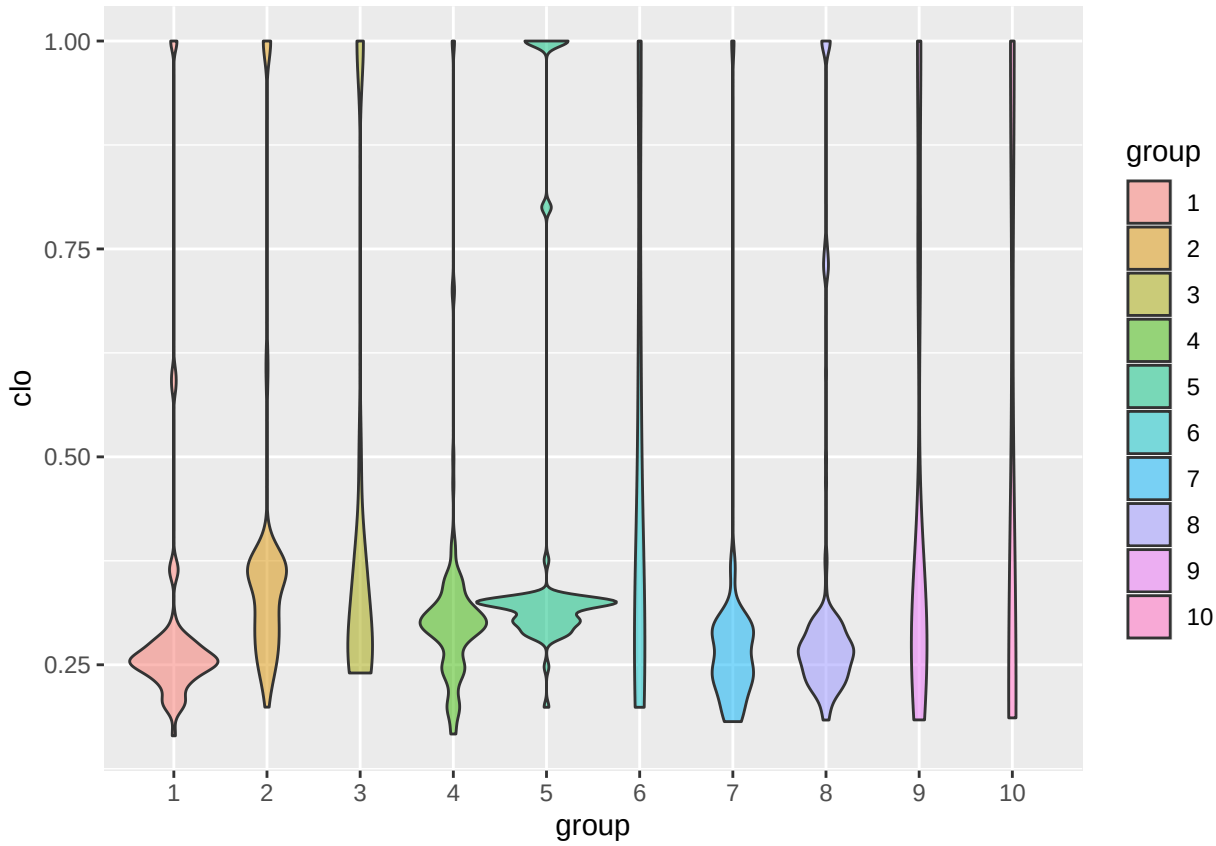
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



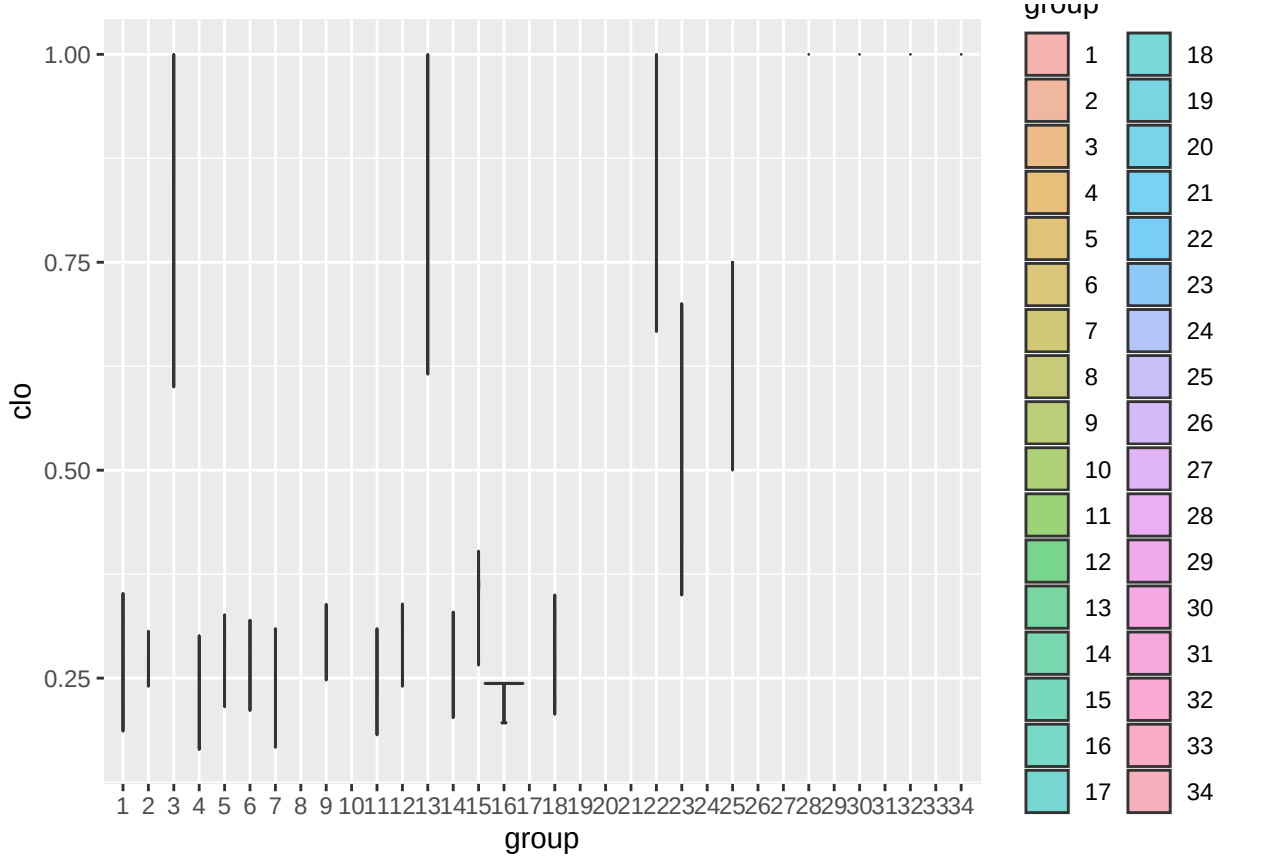

```
ggplot(data.frame(clo = V(uni_graph2)$closeness,group=factor(condensed_baps_adm[sapply(strsplit(names(V
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

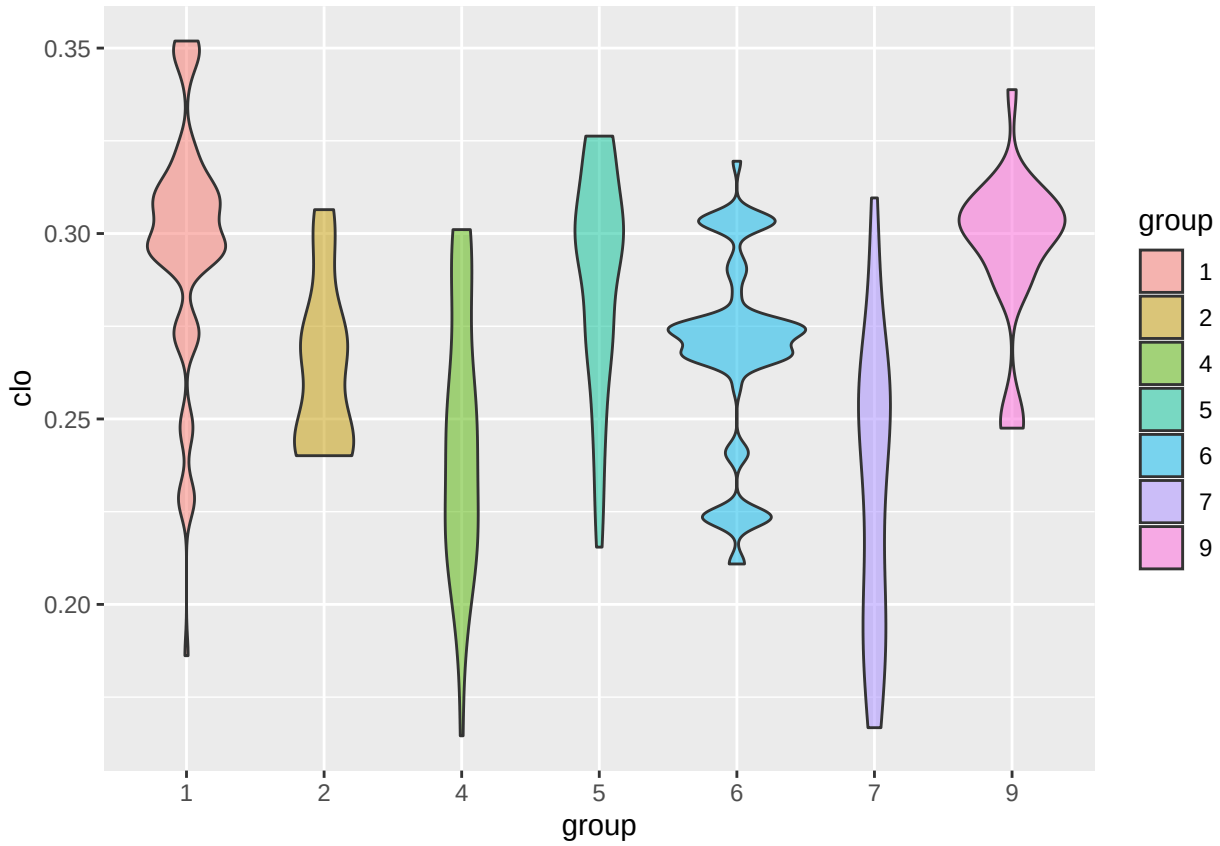



```
ferdframe<-data.frame(clo = V(uni_graph2)$closeness,group=factor(foo_modules[sapply(strsplit(names(V(uni
ggplot(ferdframe,
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

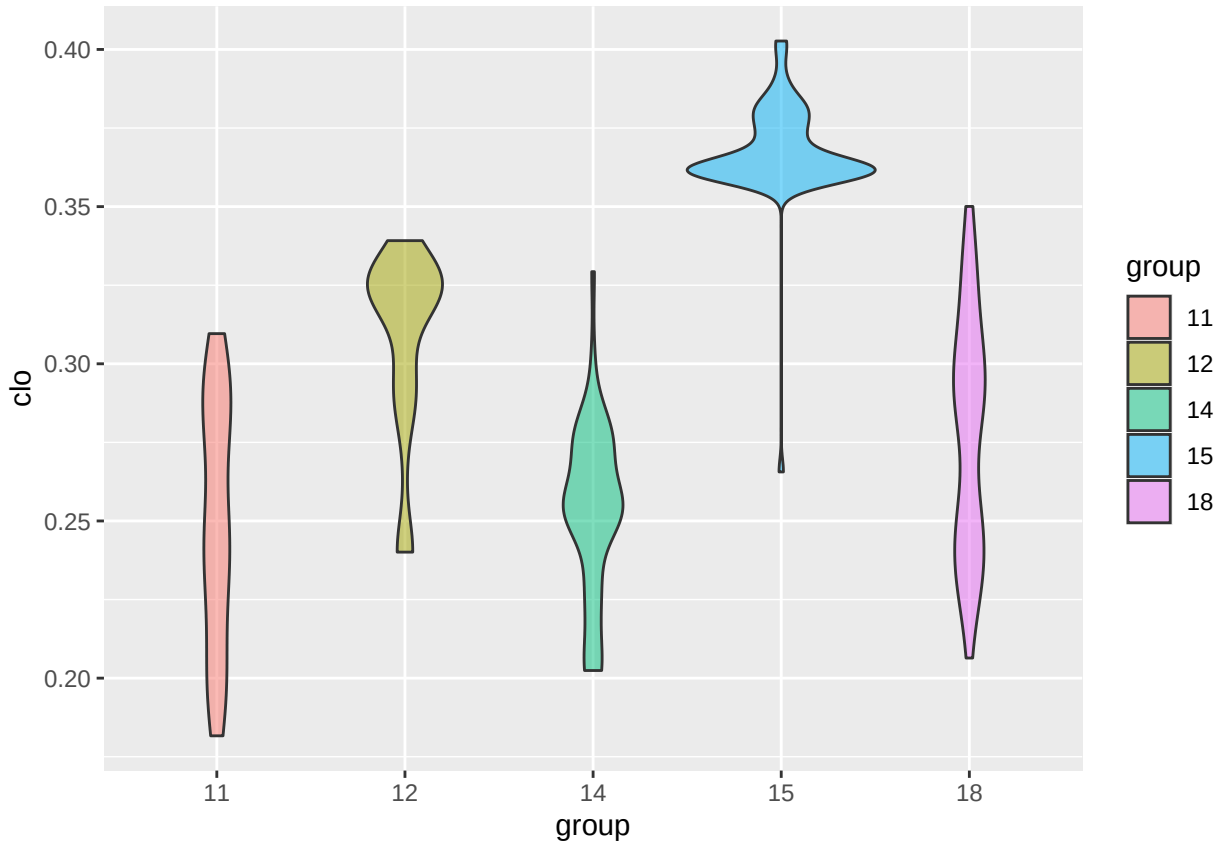
```
## Warning: position_dodge requires non-overlapping x intervals
```



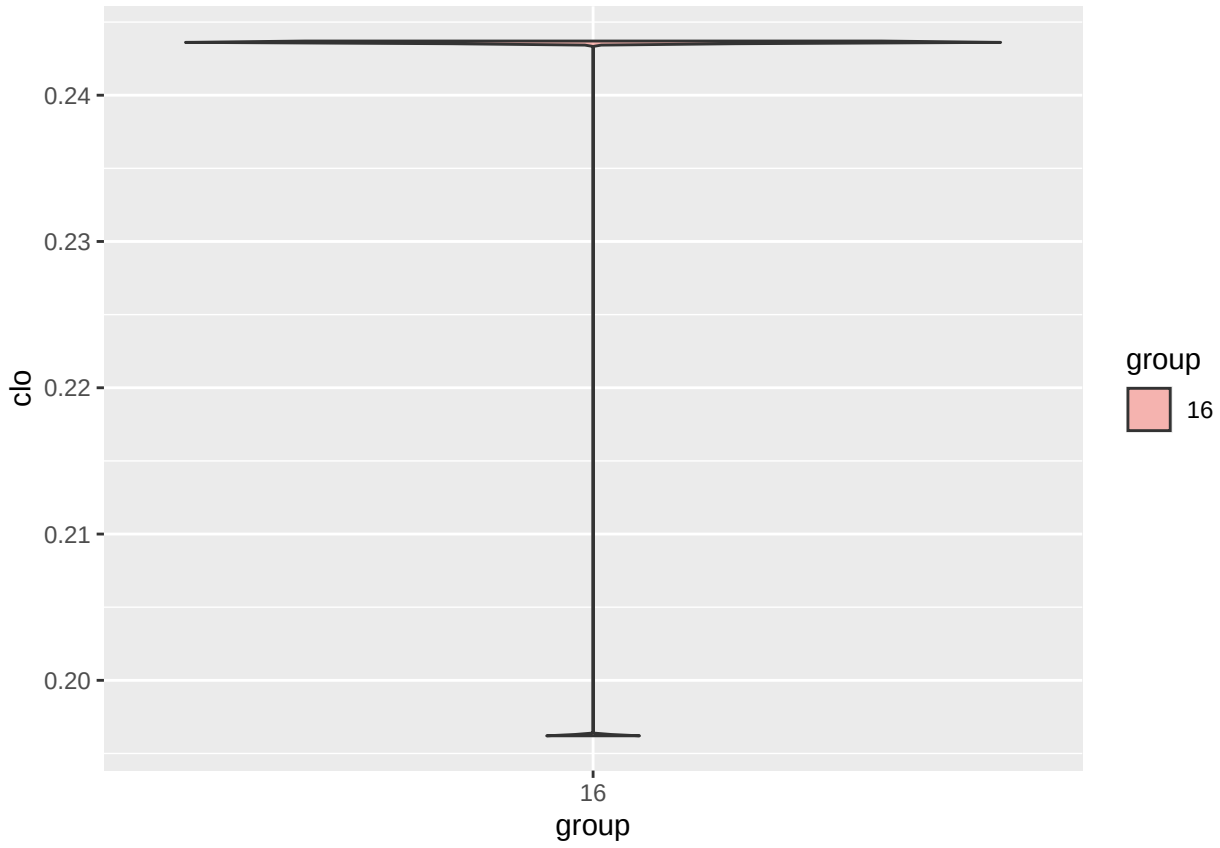

```
ggplot(ferdframe[ferdframe$group%in%c(1,2,4,5,6,7,9),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
ggplot(ferdframe[ferdframe$group%in%c(11,12,14,15,18),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```



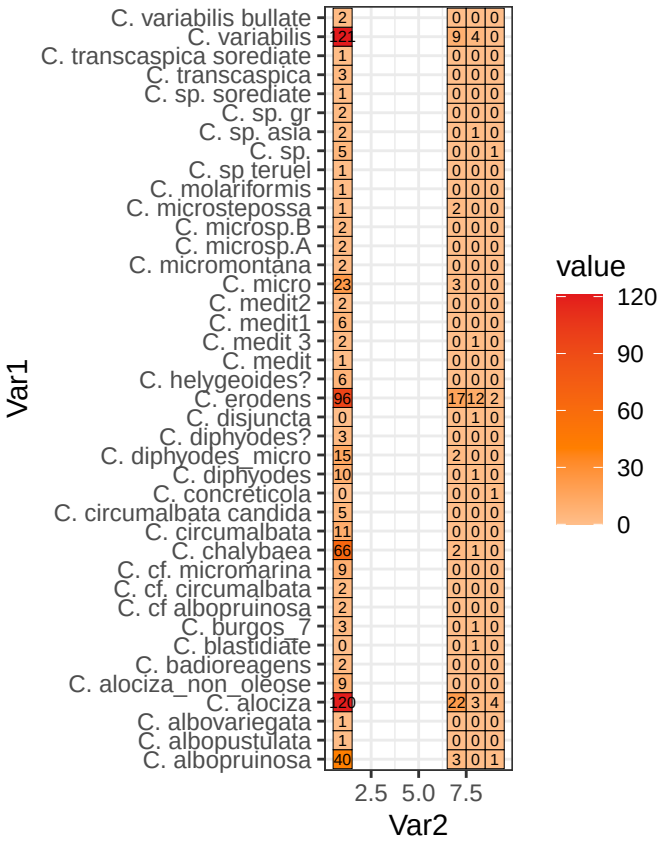

```
ggplot(ferdframe[ferdframe$group%in%c(16),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
#
# Betweenness
#
clustering_closeness<-Mclust(V(uni_graph2)$betweeness)
table(clustering_closeness$classification)
```

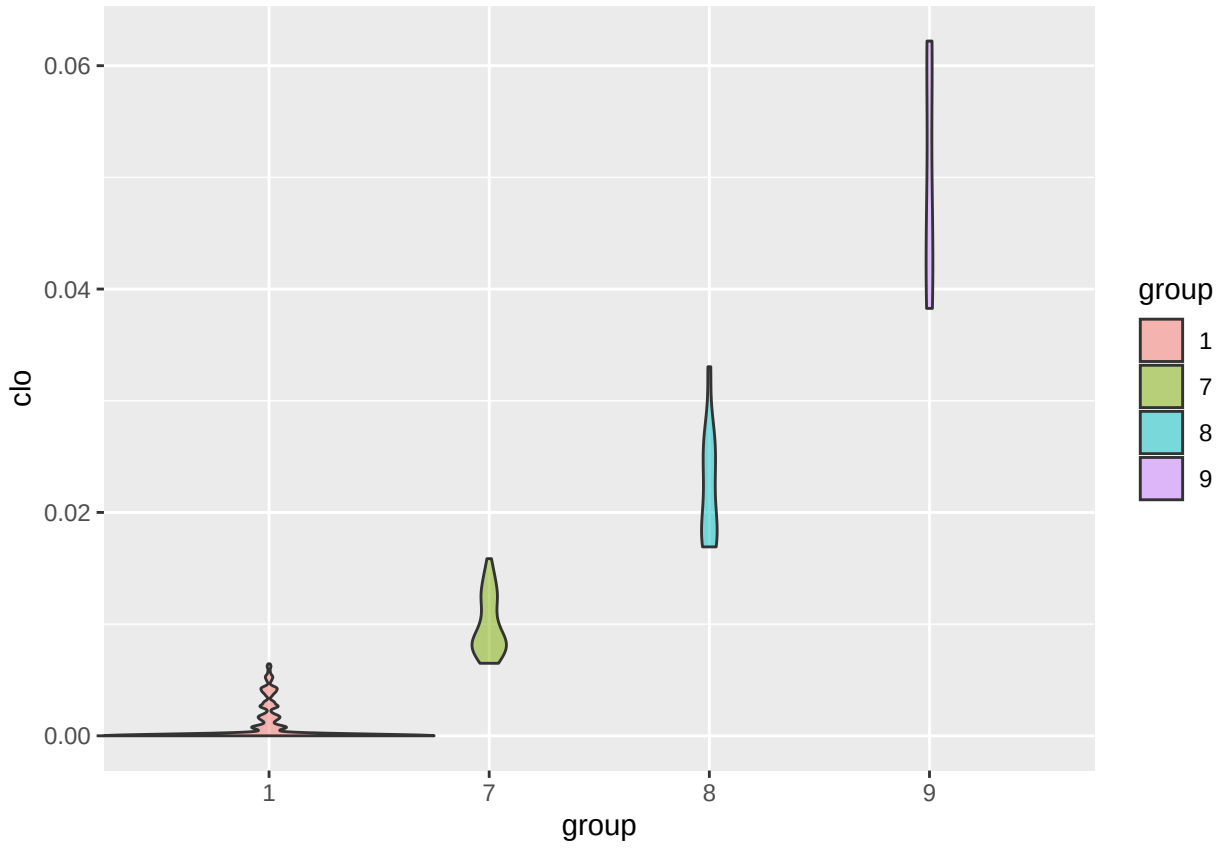
```
##
##   1   7   8   9
## 581  60  26   9
```

```
foo_tabla<-melt(table(condensed_baps_adm[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2),2],clustering_
ggplot(foo_tabla,aes(x=Var2,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = val
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 40,
                       space = "Lab",
                       na.value = "#FFFFFF",
                       guide = "colourbar",
                       aesthetics = "fill"
  ) + coord_fixed() + theme_bw()
```




```
ggplot(data.frame(clo = V(uni_graph2)$betweeness,group=factor(clustering_closeness$classification)),aes
```

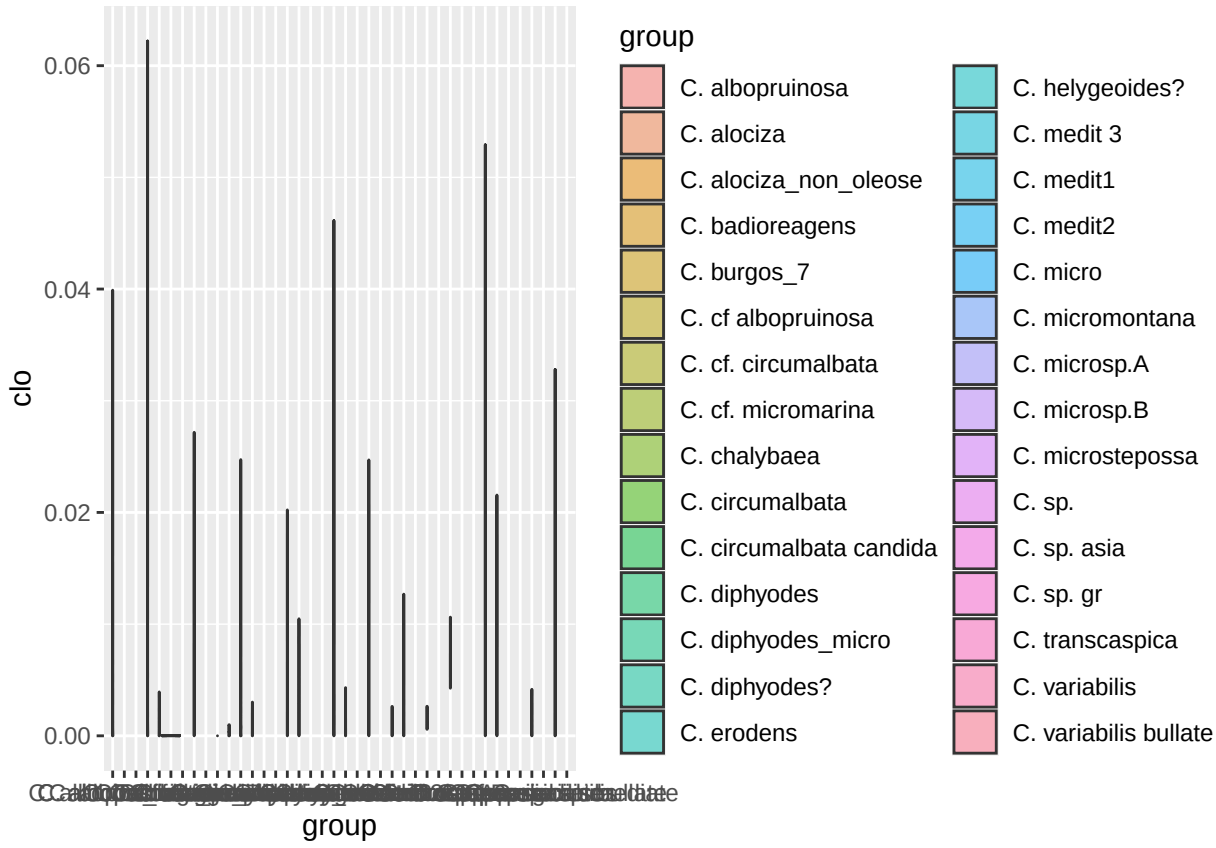
```
## Warning: position_dodge requires non-overlapping x intervals
```




```
#
#
ggplot(data.frame(clo = V(uni_graph2)$betweeness,group=factor(condensed_baps_adm[sapply(strsplit(names(V
```

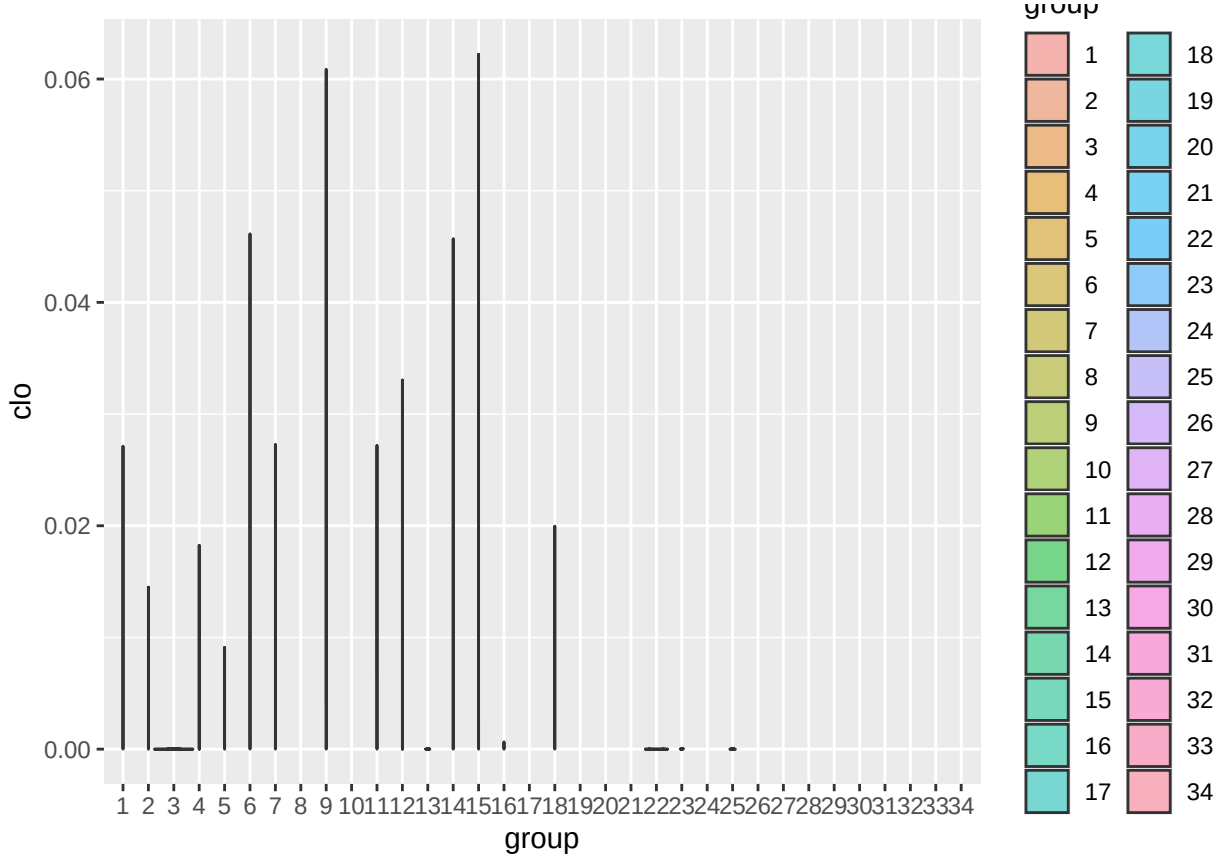
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



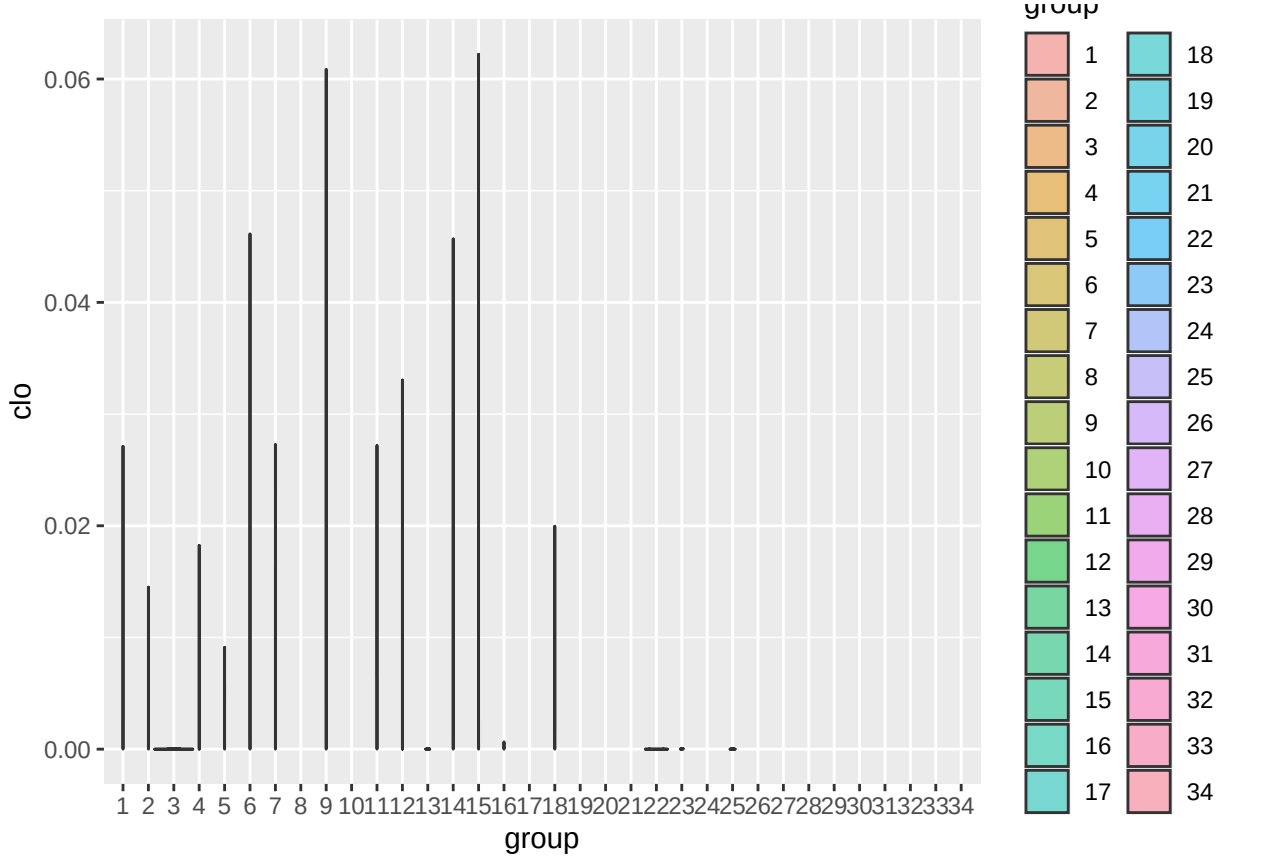

```
ggplot(data.frame(clo = V(uni_graph2)$betweeness,group=factor(foo_modules[sapply(strsplit(names(V(uni_g
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

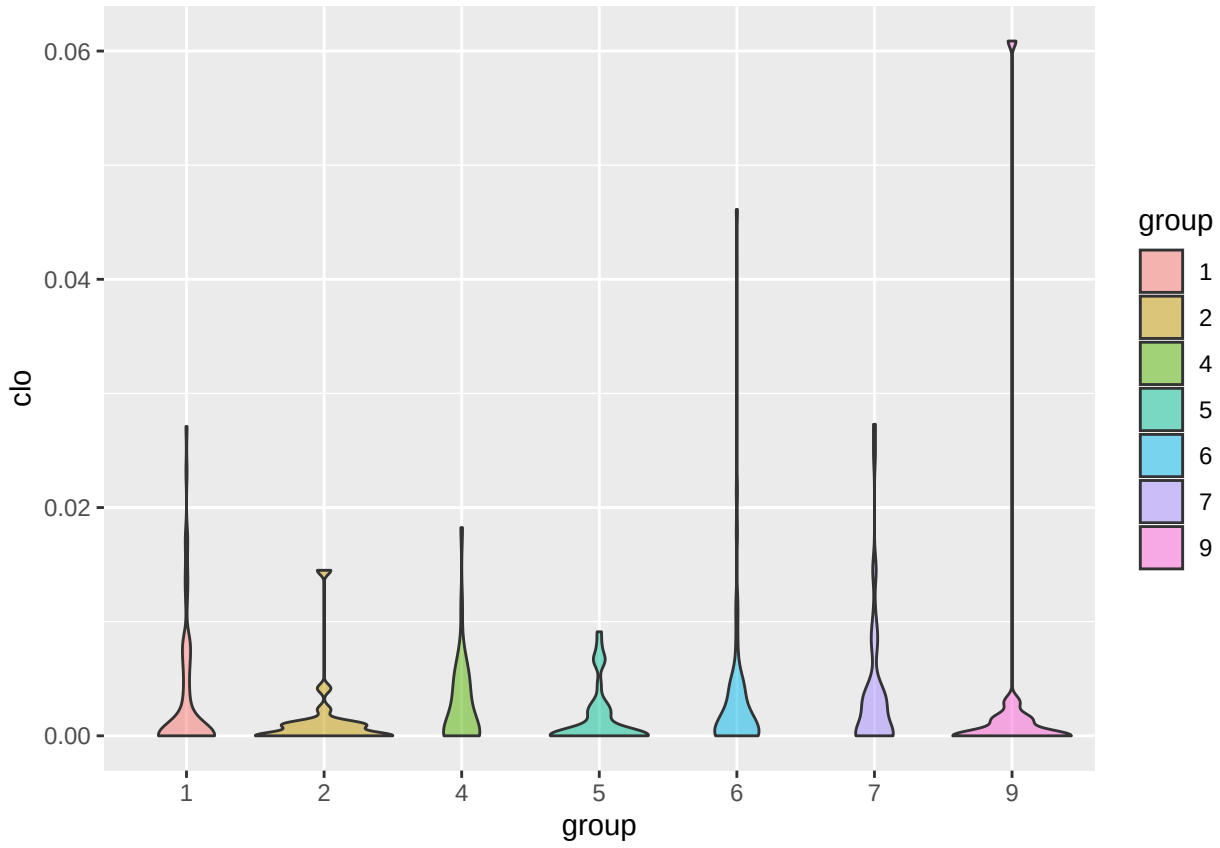



```
ferdframe<-data.frame(clo = V(uni_graph2)$betweeness,group=factor(foo_modules[sapply(strsplit(names(V(u
ggplot(ferdframe,
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

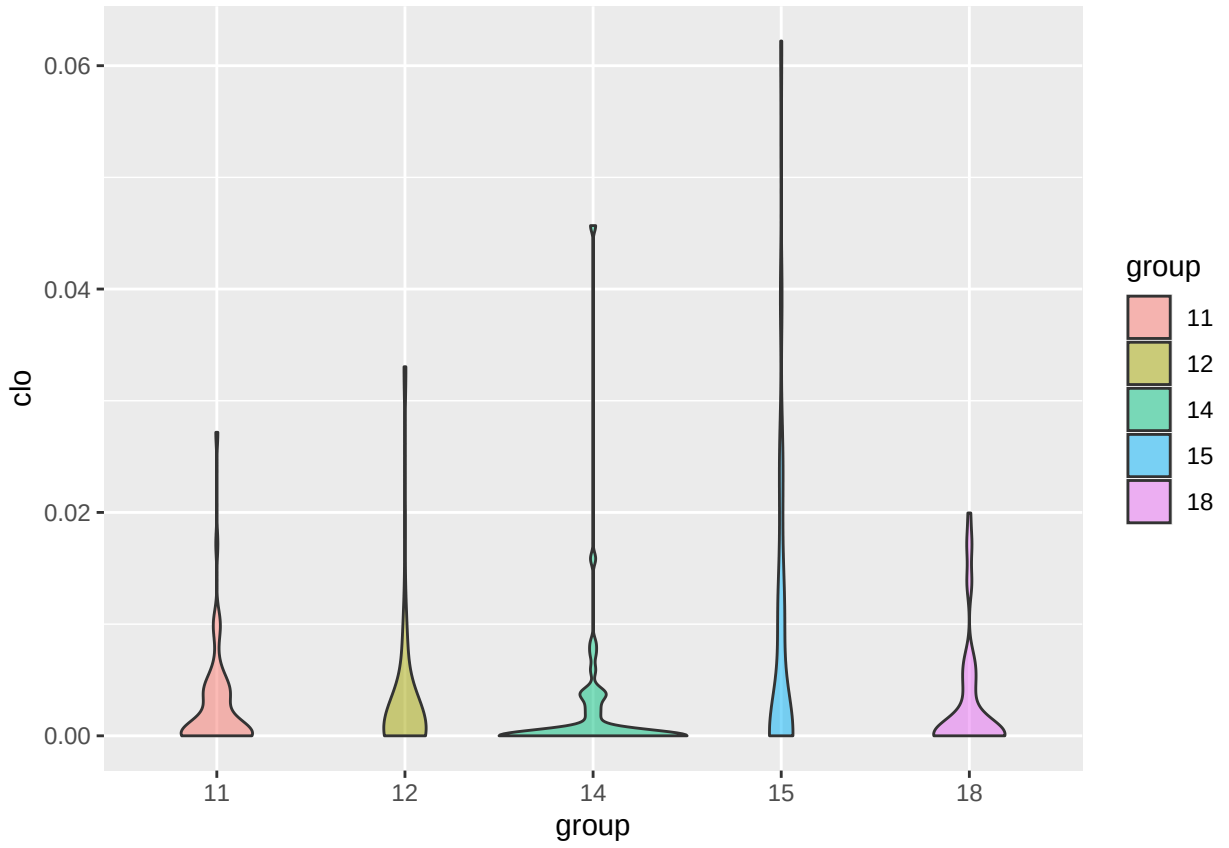
```
## Warning: position_dodge requires non-overlapping x intervals
```



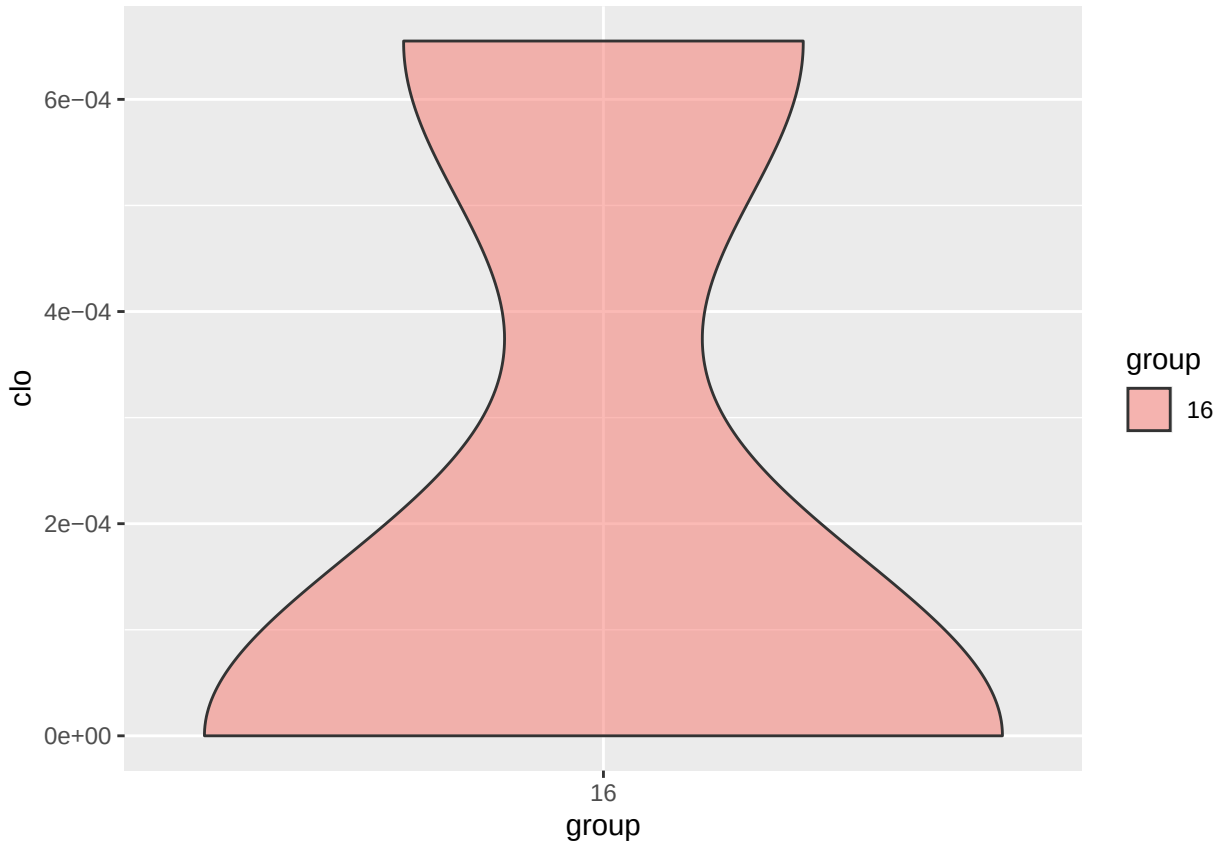

```
ggplot(ferdframe[ferdframe$group%in%c(1,2,4,5,6,7,9),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
ggplot(ferdframe[ferdframe$group%in%c(11,12,14,15,18),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```



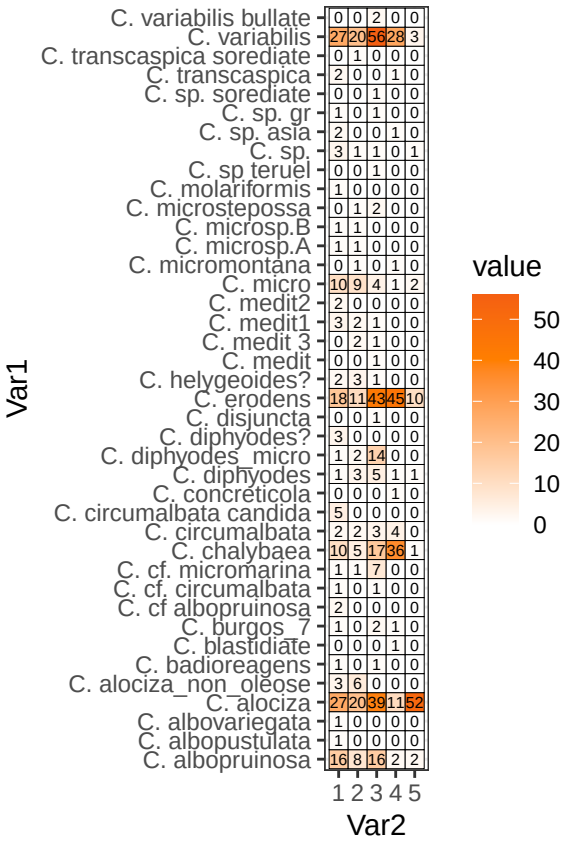

```
ggplot(ferdframe[ferdframe$group%in%c(16),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
#

# Degree
#
clustering_closeness<-Mclust(V(uni_graph2)$outdegree)
table(clustering_closeness$classification)
```

```
##
##   1   2   3   4   5
## 149 100 221 134  72
```

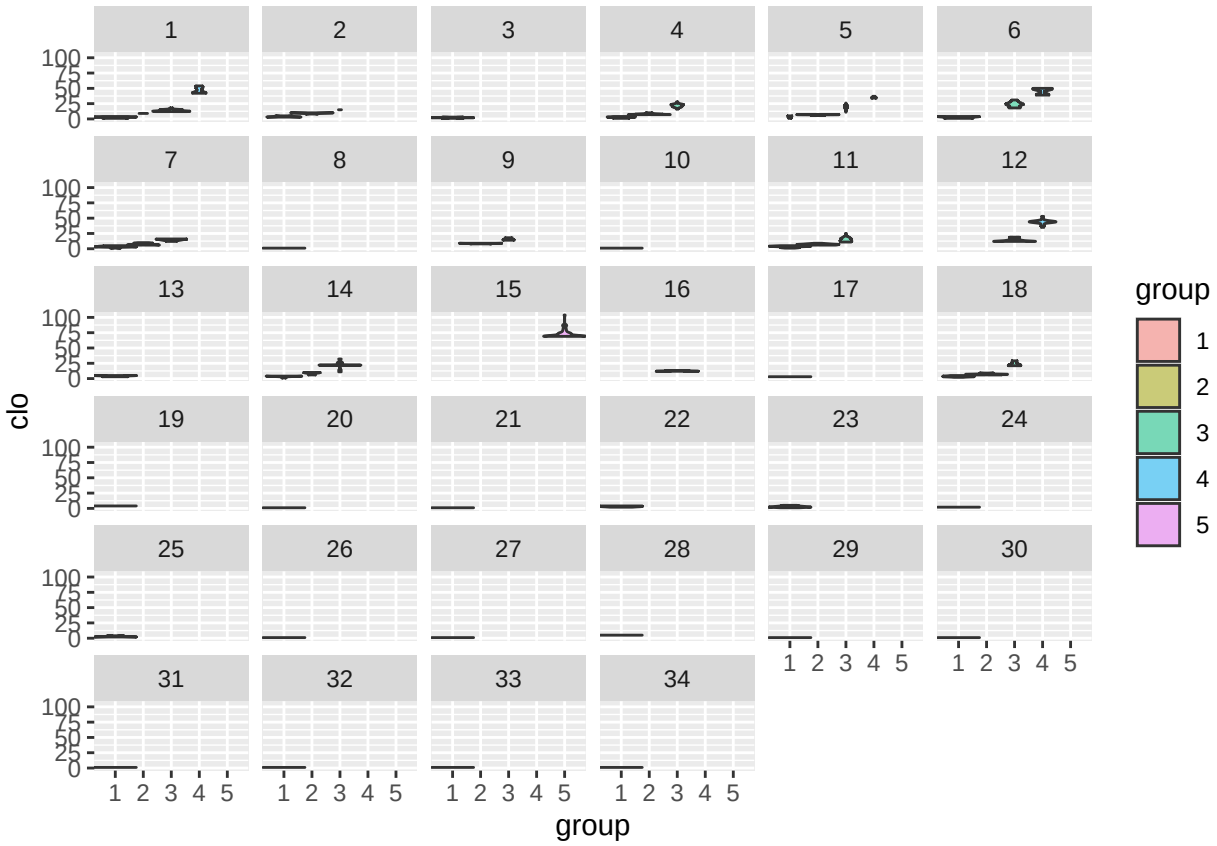
```
foo_tabla<-melt(table(condensed_baps_adm[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2),2],clustering_
ggplot(foo_tabla,aes(x=Var2,y=Var1,fill=value)) + geom_tile(color = "black") + geom_text(aes(label = va
  scale_fill_gradient2(low = "#FFFFFF",
                       mid = colorinos.bipolar[7],
                       high = colorinos.bipolar[6],
                       midpoint = 40,
                       space = "Lab",
                       na.value = "#FFFFFF",
                       guide = "colourbar",
                       aesthetics = "fill"
  ) + coord_fixed() + theme_bw()
```




```
ggplot(data.frame(clo = V(uni_graph2)$outdegree,group=factor(clustering_closeness$classification)),aes(
  facet_wrap(vars(factor(foo_modules[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2)])))
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

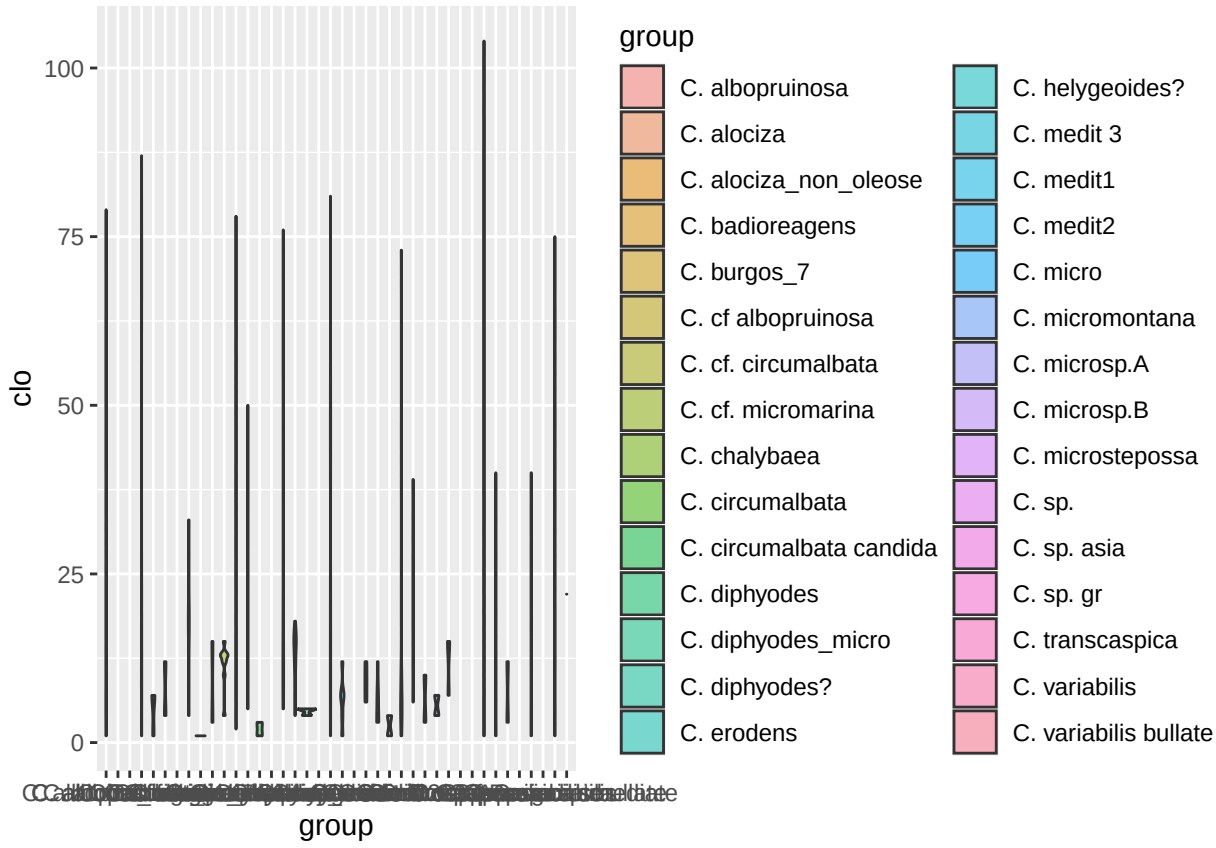
```
## Warning: position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
```




```
#
#
ggplot(data.frame(clo = V(uni_graph2)$outdegree,group=factor(condensed_baps_adm[sapply(strsplit(names(V
```

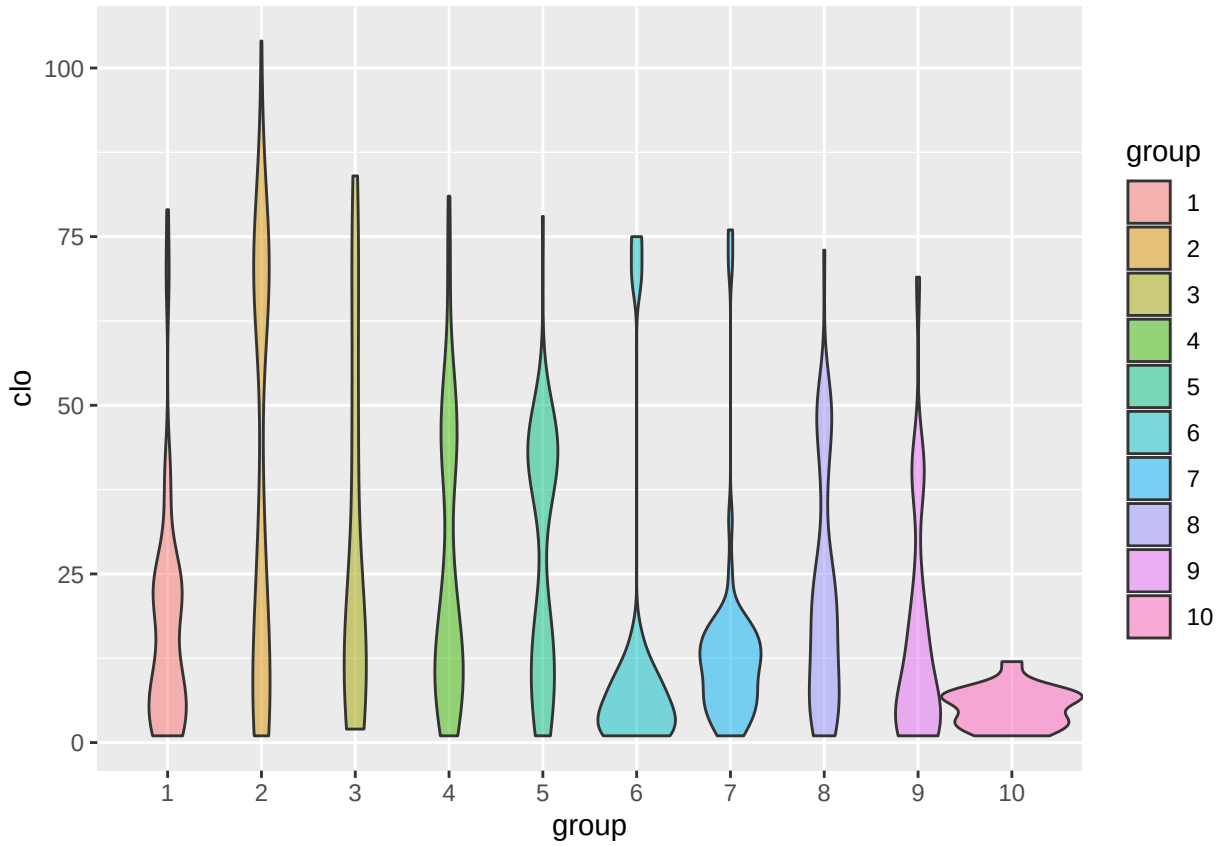
```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



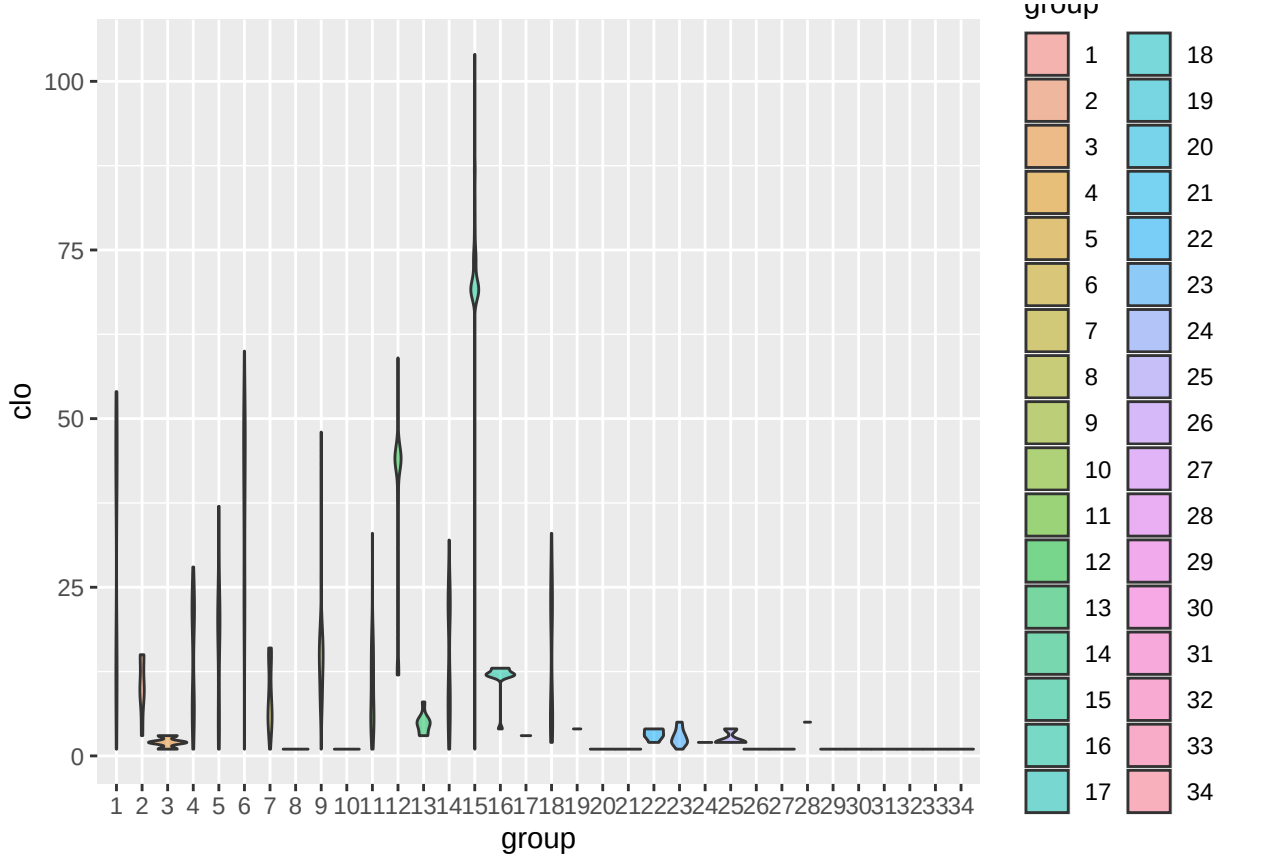

```
ggplot(data.frame(clo = V(uni_graph2)$outdegree,group=factor(condensed_baps_adm[sapply(strsplit(names(V
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

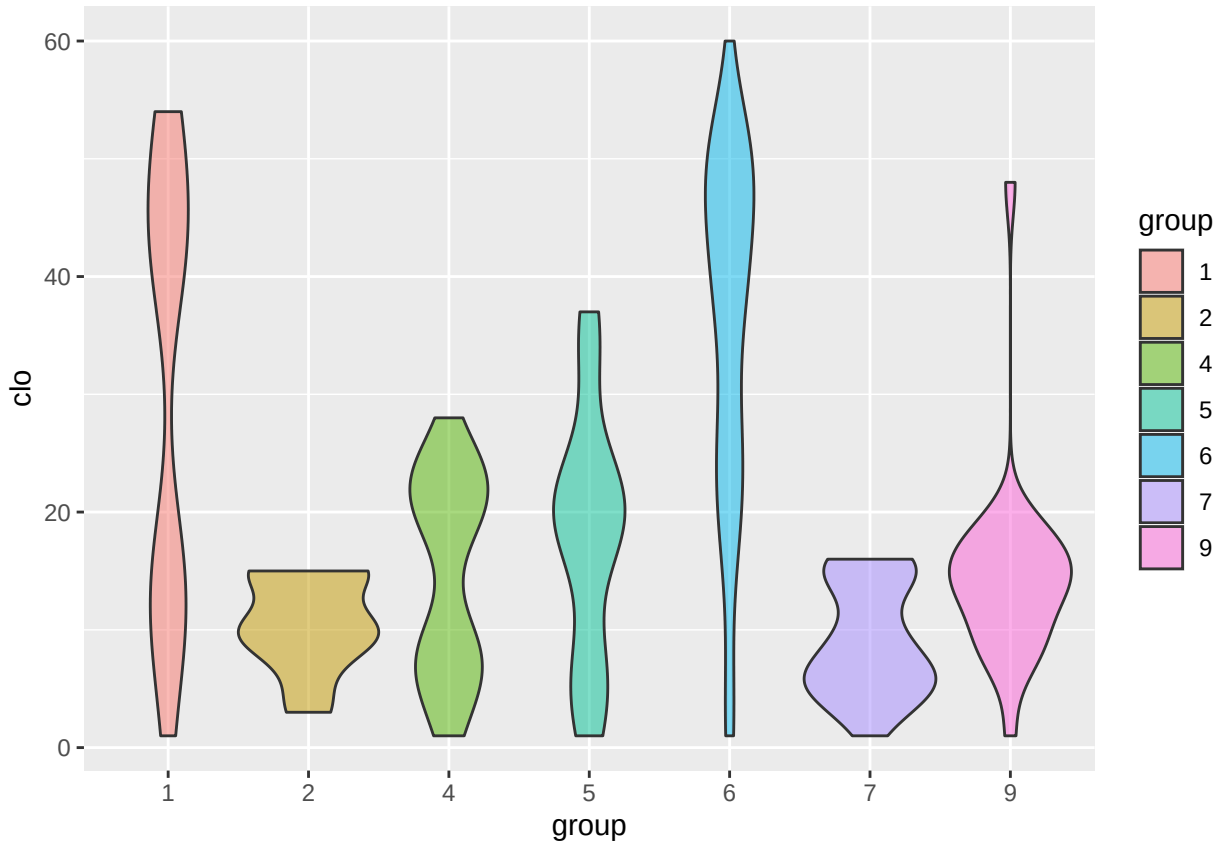



```
ferdframe<-data.frame(clo = V(uni_graph2)$outdegree,group=factor(foo_modules[sapply(strsplit(names(V(uni

ggplot(ferdframe,
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1.5)
```

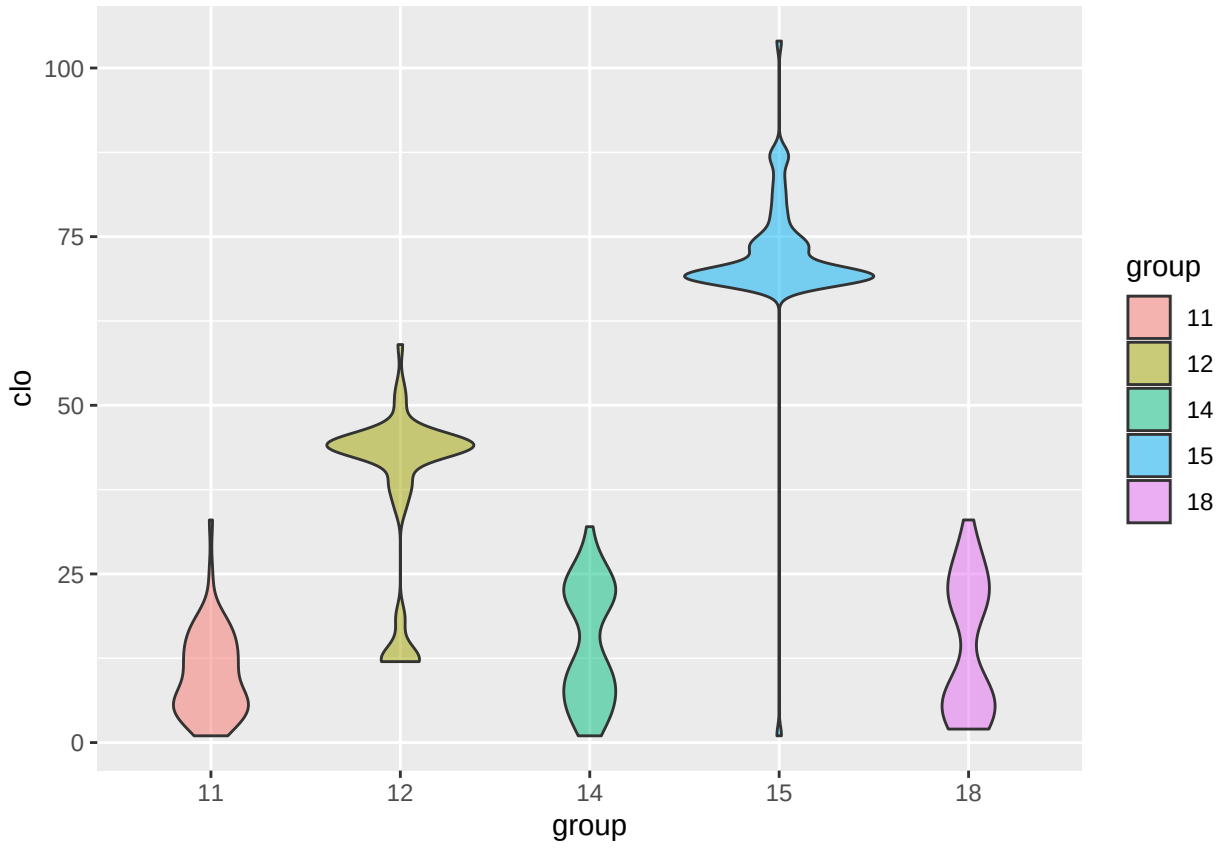
```
## Warning: position_dodge requires non-overlapping x intervals
```



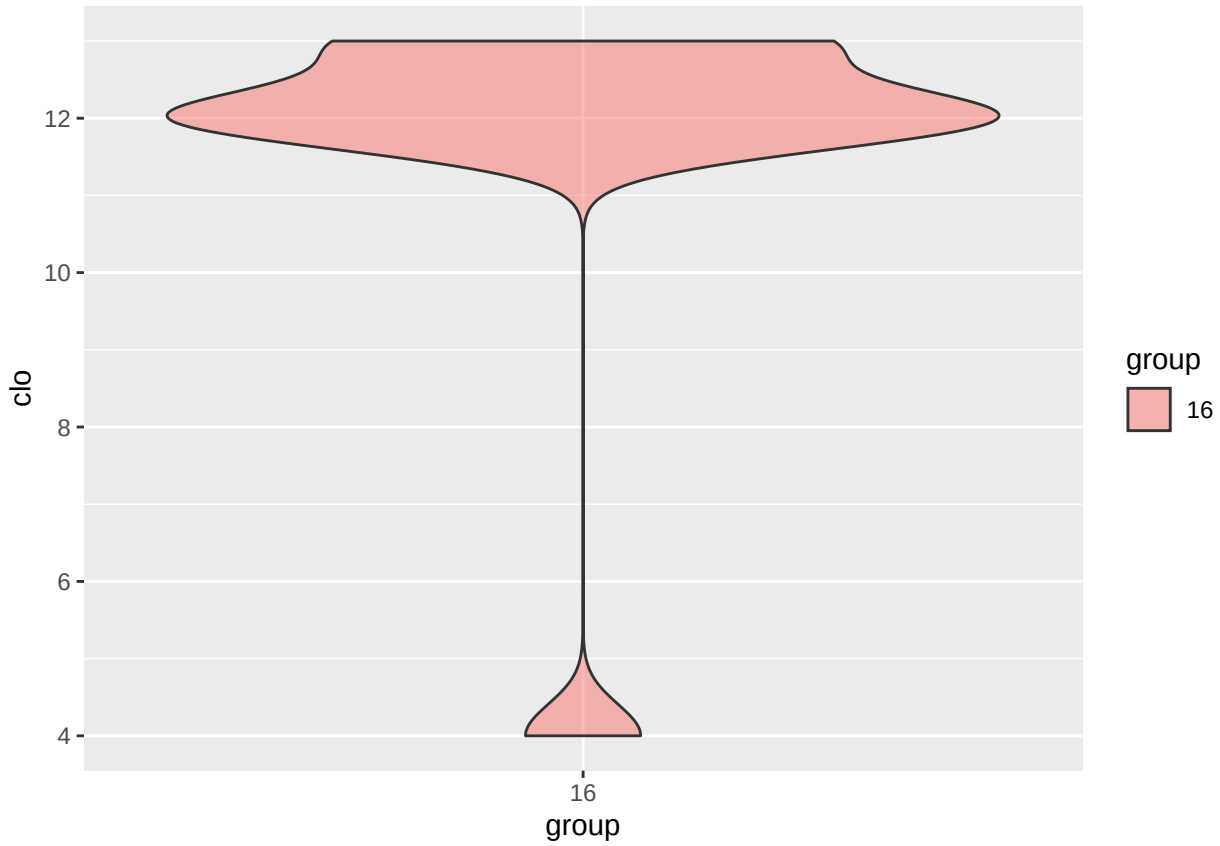

```
ggplot(ferdframe[ferdframe$group%in%c(1,2,4,5,6,7,9),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
ggplot(ferdframe[ferdframe$group%in%c(11,12,14,15,18),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
ggplot(ferdframe[ferdframe$group%in%c(16),],
       aes(x=group,y=clo,fill=group))+geom_violin(alpha=0.5,width=1)
```




```
#
```

```
V(uni_graph2)$closeness[condensed_baps_adm[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2),13]==1]
```

```
##  [1] 0.2670227 0.2437043 0.2539145 0.2556455 0.2027712 0.2027712 0.2670227
##  [8] 0.6000000 0.2439024 0.2278769 0.2346500 0.2346500 0.2439024 0.2671416
## [15] 0.2108963 0.3612282 0.3710575 0.2770083 0.3612282 0.3013561 0.2877698
## [22] 0.2535926 0.2535926 0.2535926 0.2770083 0.2476269 0.2535926 0.2536998
## [29] 0.2024291 0.2535926 0.2583979 0.2535926 0.2535926 0.2535926 0.2824859
## [36] 0.2278769 0.2539145 1.0000000 0.5833333 1.0000000 0.2053388 0.2535926
## [43] 0.2535926 0.2278769 0.2770083 0.2053388 0.2770083 0.2670227 0.2770083
## [50] 0.2770083 0.2770083 0.2535926 0.2535926 0.2334630 0.2027712 0.2713704
## [57] 0.2535926 0.2535926 0.2334630 0.2334630 0.2535926 0.2535926 0.1645639
```

```
#
# Not recalculate vertex stuff
#
stats_degree<-list()
stats_desc<-list()
V(uni_graph2)$outdegree  <- degree(uni_graph2, mode = "out")
V(uni_graph2)$closeness  <- closeness(uni_graph2, mode = "total", normalized = TRUE)
V(uni_graph2)$betweeness <- betweenness(uni_graph2, normalized = TRUE)
for (i in 1:34)
{
  foo<-induced_subgraph(uni_graph2,which(color_clusters==i))
  # Extracting each vectex features as a data.frame
  stats <- as_data_frame(foo, what = "vertices")
  # Computing quantiles for each variable
```

```
  stats_degree[[i]] <- with(stats, {
    cbind(
      outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
      closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
      betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
    )
  })
  stats_desc[[i]] <- cbind(
    size    = vcount(foo),
    nedges  = ecount(foo),
    density = edge_density(foo),
    recip   = reciprocity(foo),
    centr   = centr_betw(foo)$centralization,
    pathLen = mean_distance(foo)
  )
}

foo<-induced_subgraph(uni_graph2,which(color_clusters%in%c(1,2,4,5,6,7,9,11,12,14,15,16,18)))
stats <- as_data_frame(foo, what = "vertices")
stats_degree[[35]] <- with(stats, {
  cbind(
    outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
    closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
    betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
  )
})
stats_desc[[35]] <- cbind(
  size    = vcount(foo),
  nedges  = ecount(foo),
  density = edge_density(foo),
  recip   = reciprocity(foo),
  centr   = centr_betw(foo)$centralization,
  pathLen = mean_distance(foo)
)

foo<-uni_graph2
stats <- as_data_frame(foo, what = "vertices")
stats_degree[[36]] <- with(stats, {
  cbind(
    outdegree  = quantile(outdegree, c(.025, .5, .975), na.rm = TRUE),
    closeness  = quantile(closeness, c(.025, .5, .975), na.rm = TRUE),
    betweeness = quantile(betweeness, c(.025, .5, .975), na.rm = TRUE)
  )
})
stats_desc[[36]] <- cbind(
  size    = vcount(foo),
  nedges  = ecount(foo),
  density = edge_density(foo),
  recip   = reciprocity(foo),
  centr   = centr_betw(foo)$centralization,
  pathLen = mean_distance(foo)
)
```

```
print(stats_degree)
```

```
## [[1]]
##       outdegree closeness betweeness
## 2.5%        2.9 0.2285540 0.00000000
## 50%        42.0 0.2958580 0.00000000
## 97.5%      54.0 0.3506721 0.01759249
##
## [[2]]
##       outdegree closeness betweeness
## 2.5%          3 0.2400960 0.00000000
## 50%          10 0.2702703 0.00000000
## 97.5%        15 0.3046280 0.00984555
##
## [[3]]
##       outdegree closeness   betweeness
## 2.5%      1.075   0.61125 0.000000e+00
## 50%       2.000   0.75000 0.000000e+00
## 97.5%     2.925   0.98125 8.132762e-06
##
## [[4]]
##       outdegree closeness betweeness
## 2.5%       1.95 0.1949910 0.00000000
## 50%       19.00 0.2444988 0.00131443
## 97.5%     24.20 0.2965342 0.01168070
##
## [[5]]
##       outdegree closeness  betweeness
## 2.5%          1 0.2290098 0.000000000
## 50%          19 0.2976190 0.000000000
## 97.5%        37 0.3260914 0.007906799
##
## [[6]]
##       outdegree closeness  betweeness
## 2.5%      2.725 0.2181704 0.000000000
## 50%      39.000 0.2714932 0.001698865
## 97.5%    50.000 0.3033367 0.019397525
##
## [[7]]
##       outdegree closeness  betweeness
## 2.5%          3 0.1667593 0.000000000
## 50%           9 0.2437043 0.002624464
## 97.5%        16 0.2985075 0.023682604
##
## [[8]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[9]]
##       outdegree closeness betweeness
## 2.5%       4.85 0.2475248 0.00000000
## 50%       14.00 0.3022670 0.00000000
```

```
## 97.5%     31.50 0.3232637 0.02902508
##
## [[10]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[11]]
##       outdegree closeness   betweeness
## 2.5%       2.00 0.1816530 0.0000000000
## 50%        8.00 0.2477291 0.0004457213
## 97.5%     22.75 0.3038569 0.0141659932
##
## [[12]]
##       outdegree closeness  betweeness
## 2.5%       12.0 0.2400960 0.000000000
## 50%        44.0 0.3255562 0.000772224
## 97.5%      52.3 0.3323097 0.028047272
##
## [[13]]
##       outdegree closeness   betweeness
## 2.5%        3.0 0.6153846 0.000000e+00
## 50%         5.0 0.7272727 0.000000e+00
## 97.5%       7.4 0.9454545 5.275305e-05
##
## [[14]]
##       outdegree closeness betweeness
## 2.5%        4.0 0.2027712 0.00000000
## 50%        10.0 0.2535926 0.00000000
## 97.5%      27.5 0.2979595 0.01396839
##
## [[15]]
##       outdegree closeness betweeness
## 2.5%         69 0.3612282 0.00000000
## 50%          69 0.3612282 0.00000000
## 97.5%        87 0.3986844 0.05511925
##
## [[16]]
##       outdegree closeness  betweeness
## 2.5%        6.2 0.2092413 0.000000000
## 50%        12.0 0.2436054 0.000000000
## 97.5%      13.0 0.2437043 0.000655017
##
## [[17]]
##       outdegree closeness betweeness
## 2.5%          3         1          0
## 50%           3         1          0
## 97.5%         3         1          0
##
## [[18]]
##       outdegree closeness betweeness
## 2.5%          3 0.2257045 0.00000000
## 50%           9 0.2897151 0.00000000
```

```
## 97.5%        29 0.3358551 0.01729053
##
## [[19]]
##       outdegree closeness betweeness
## 2.5%          4         1          0
## 50%           4         1          0
## 97.5%         4         1          0
##
## [[20]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[21]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[22]]
##       outdegree closeness   betweeness
## 2.5%        2.1      0.68 0.000000e+00
## 50%         3.0      0.80 0.000000e+00
## 97.5%       4.0      1.00 4.396087e-06
##
## [[23]]
##       outdegree closeness   betweeness
## 2.5%      1.175 0.3704167 0.000000e+00
## 50%       2.500 0.5416667 0.000000e+00
## 97.5%     4.825 0.7000000 4.396087e-05
##
## [[24]]
##       outdegree closeness betweeness
## 2.5%          2         1          0
## 50%           2         1          0
## 97.5%         2         1          0
##
## [[25]]
##       outdegree closeness  betweeness
## 2.5%          2      0.50 0.00000e+00
## 50%           2      0.50 0.00000e+00
## 97.5%         4      0.75 3.51687e-05
##
## [[26]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[27]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
```

```
## 97.5%         1         1          0
##
## [[28]]
##       outdegree closeness betweeness
## 2.5%          5         1          0
## 50%           5         1          0
## 97.5%         5         1          0
##
## [[29]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[30]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[31]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[32]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[33]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[34]]
##       outdegree closeness betweeness
## 2.5%          1         1          0
## 50%           1         1          0
## 97.5%         1         1          0
##
## [[35]]
##       outdegree closeness betweeness
## 2.5%          2 0.1953125 0.00000000
## 50%          18 0.2816901 0.00000000
## 97.5%        74 0.3790272 0.02635734
##
## [[36]]
##       outdegree closeness betweeness
## 2.5%        1.0 0.1965442 0.00000000
## 50%        15.5 0.2936875 0.00000000
```

```
## 97.5%      74.0 1.0000000 0.02476439
```

```
print(stats_desc)
```

```
## [[1]]
##      size nedges   density recip      centr  pathLen
## [1,]   79   1124 0.3648166     1 0.09235636 1.824732
##
## [[2]]
##      size nedges   density recip     centr  pathLen
## [1,]   19     84 0.4912281     1 0.4015977 1.754386
##
## [[3]]
##      size nedges   density recip     centr  pathLen
## [1,]    4      4 0.6666667     1 0.6666667 1.333333
##
## [[4]]
##      size nedges  density recip     centr  pathLen
## [1,]   39    271 0.365722     1 0.2875271 2.055331
##
## [[5]]
##      size nedges   density recip     centr  pathLen
## [1,]   40    289 0.3705128     1 0.1180861 1.841026
##
## [[6]]
##      size nedges   density recip     centr  pathLen
## [1,]   64   1105 0.5481151     1 0.0840709 1.583333
##
## [[7]]
##      size nedges  density recip     centr  pathLen
## [1,]   45    188 0.189899     1 0.4630021 3.351515
##
## [[8]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[9]]
##      size nedges   density recip     centr  pathLen
## [1,]   23    132 0.5217391     1 0.1206218 1.667984
##
## [[10]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[11]]
##      size nedges   density recip     centr  pathLen
## [1,]   59    259 0.1513735     1 0.1326637 2.704267
##
## [[12]]
##      size nedges   density recip     centr  pathLen
## [1,]   55    979 0.6592593     1 0.1460444 1.394613
##
## [[13]]
##      size nedges   density recip     centr  pathLen
## [1,]    9     21 0.5833333     1 0.5357143 1.416667
```

```
##
## [[14]]
##      size nedges   density recip     centr  pathLen
## [1,]   51    355 0.2784314     1 0.3732898 2.307451
##
## [[15]]
##      size nedges   density recip      centr  pathLen
## [1,]   71   2416 0.9722334     1 0.02857143 1.027767
##
## [[16]]
##      size nedges   density recip     centr  pathLen
## [1,]   12     59 0.8939394     1 0.0231405 1.106061
##
## [[17]]
##      size nedges density recip centr pathLen
## [1,]    4      6       1     1     0       1
##
## [[18]]
##      size nedges   density recip     centr  pathLen
## [1,]   44    268 0.2832981     1 0.2535734 2.312896
##
## [[19]]
##      size nedges density recip centr pathLen
## [1,]    5     10       1     1     0       1
##
## [[20]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[21]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[22]]
##      size nedges density recip centr pathLen
## [1,]    5      8     0.8     1 0.125     1.2
##
## [[23]]
##      size nedges   density recip     centr  pathLen
## [1,]    8     11 0.3928571     1 0.3673469 1.928571
##
## [[24]]
##      size nedges density recip centr pathLen
## [1,]    3      3       1     1     0       1
##
## [[25]]
##      size nedges   density recip     centr  pathLen
## [1,]    7      9 0.4285714     1 0.4444444 1.761905
##
## [[26]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[27]]
```

```
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[28]]
##      size nedges density recip centr pathLen
## [1,]    6     15       1     1     0       1
##
## [[29]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[30]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[31]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[32]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[33]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[34]]
##      size nedges density recip centr pathLen
## [1,]    2      1       1     1   NaN       1
##
## [[35]]
##      size nedges    density recip     centr  pathLen
## [1,]  601   8104 0.04494731     1 0.0744372 3.655668
##
## [[36]]
##      size nedges    density recip      centr  pathLen
## [1,]  676   8203 0.03595442     1 0.05918156 3.653843
```

```
stats_networksmelt<-melt(stats_networks)
```

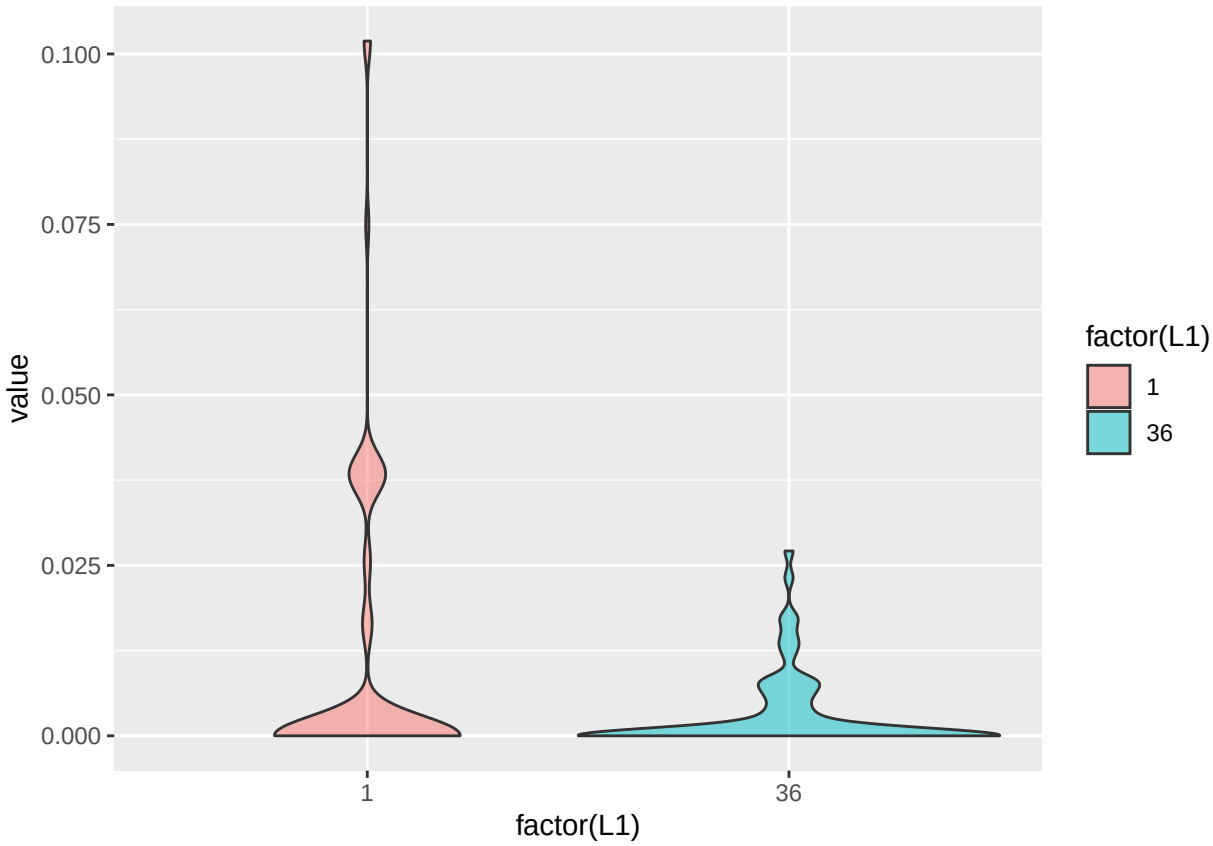
```
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
```

```
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
## Using name as id variables
```

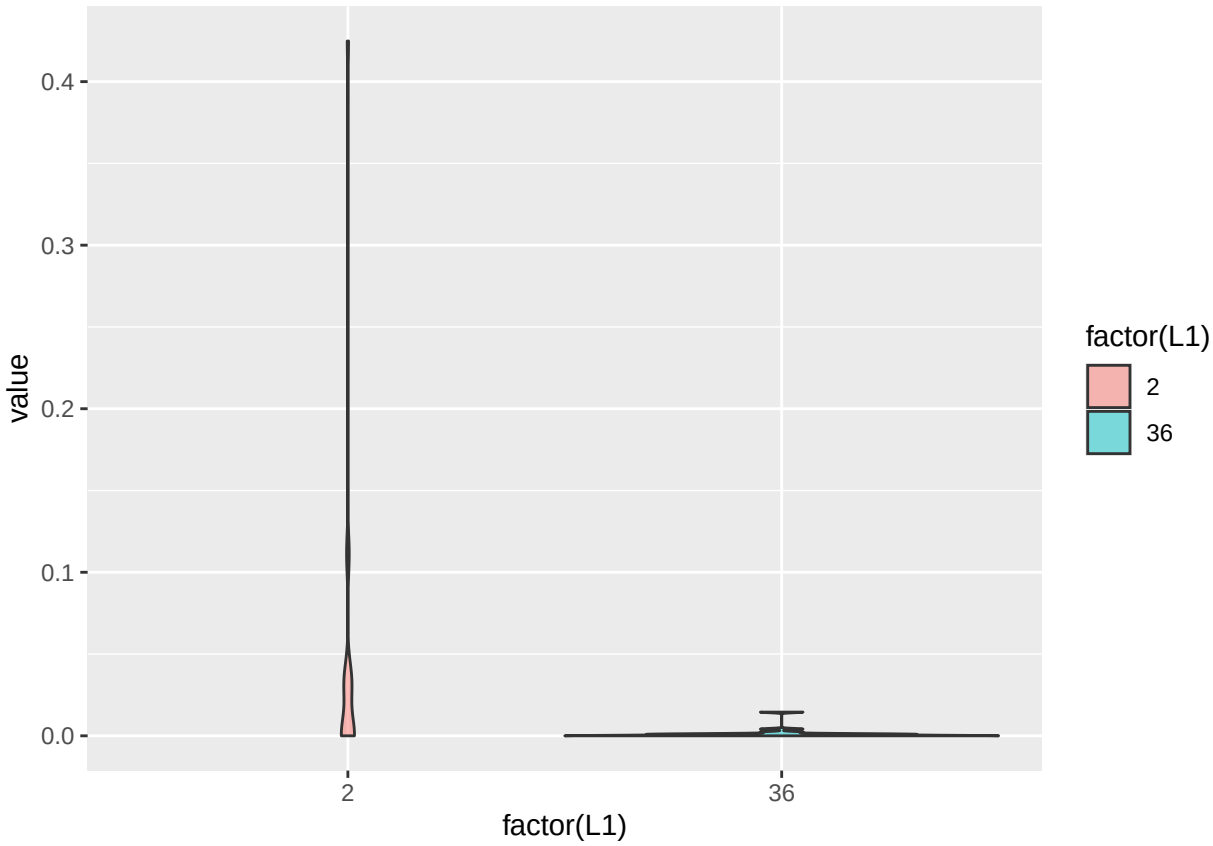
```
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

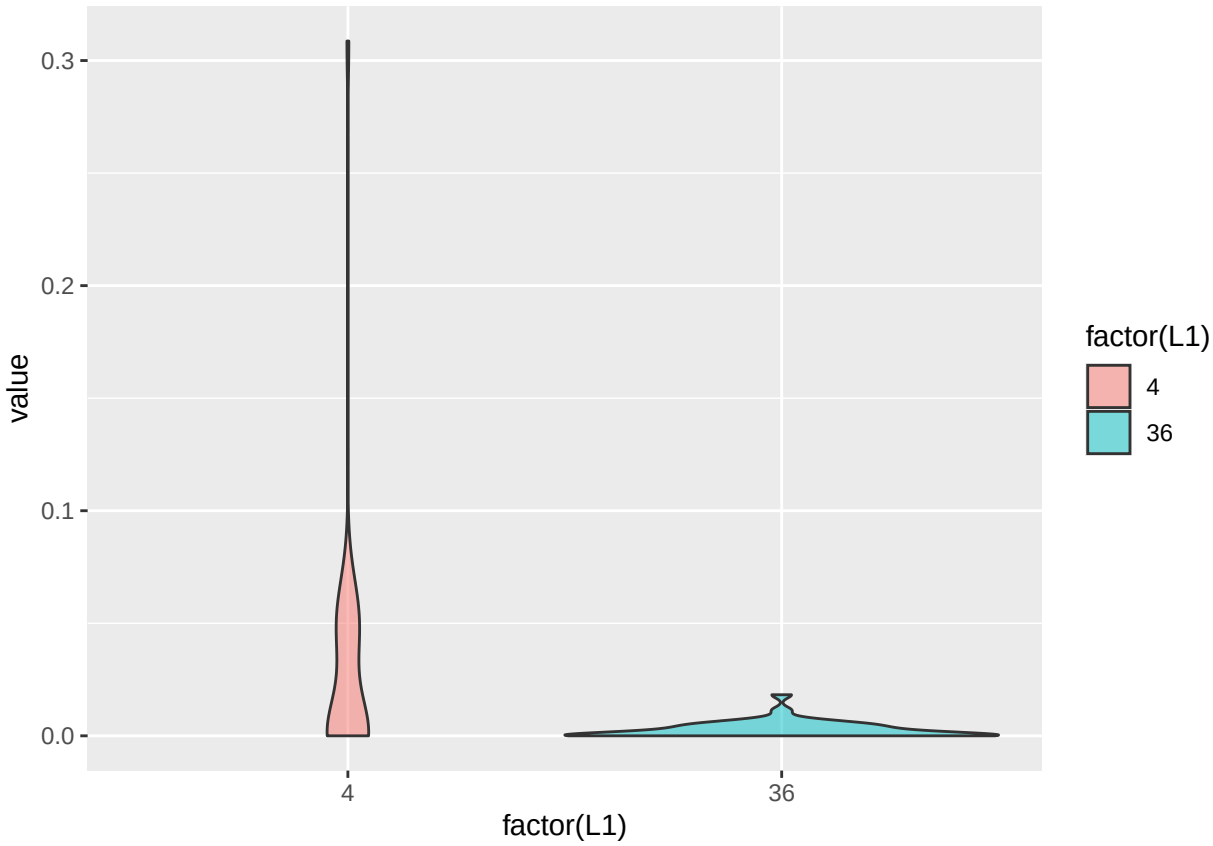
```
for (i in c(1,2,4,5,6,7,9,11,12,14,15,16,18))
{
vertices_select<-stats_networksmelt[stats_networksmelt$L1==i,1]
foo<-stats_networksmelt[stats_networksmelt[,1]%in%vertices_select,]
print(ggplot(foo[foo$variable=="betweeness"&foo$L1!=35,],
             aes(x=factor(L1),y=value,z=variable,fill=factor(L1)))+geom_violin(alpha=0.5,width=1))
stargazer(anova(lm(value~factor(L1),data=foo[foo$variable=="betweeness"&foo$L1!=35,])),type = "text")
}
```



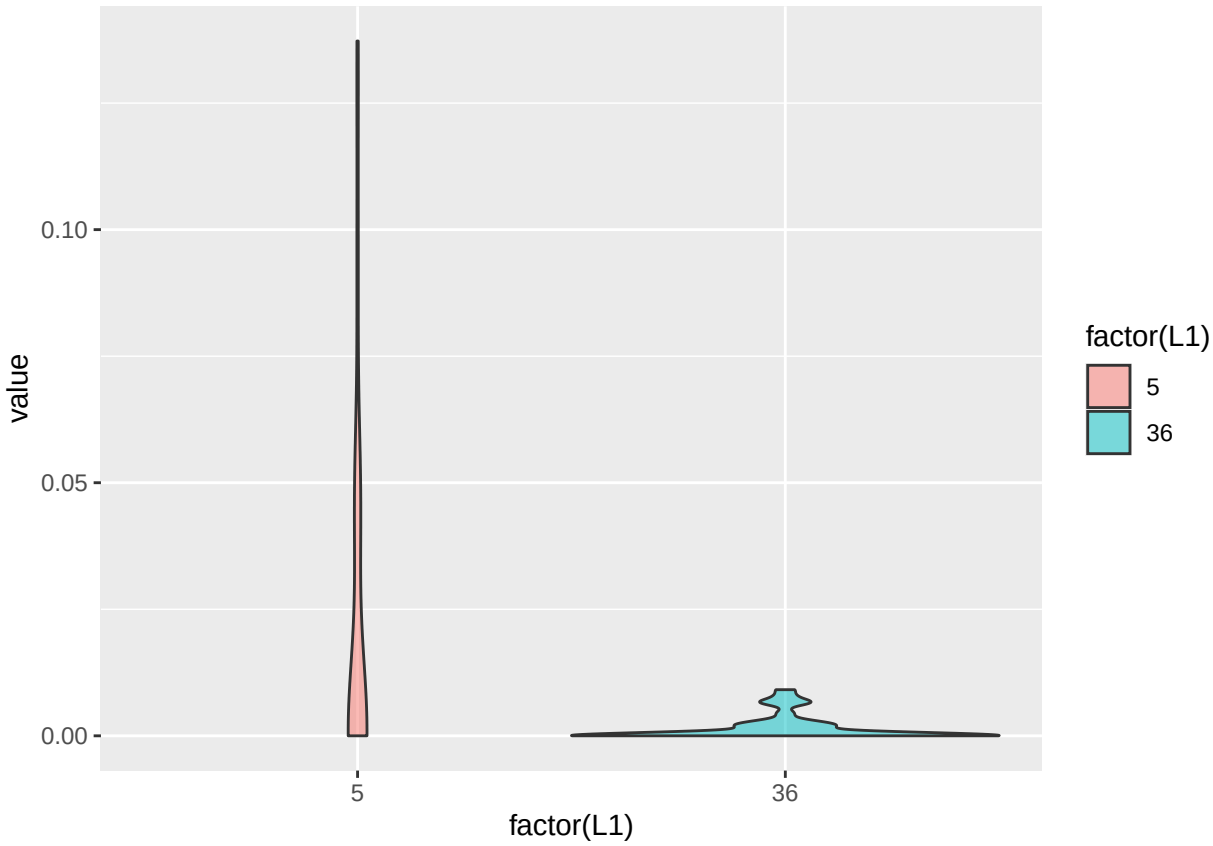

```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min    Max
## ----------------------------------------
## Df        2 78.500 109.602    1     156
## Sum Sq    2 0.021   0.026   0.002  0.040
## Mean Sq   2 0.001   0.001   0.0003 0.002
## F value   1 9.059           9.059  9.059
## Pr(> F)   1 0.003           0.003  0.003
## ----------------------------------------
```



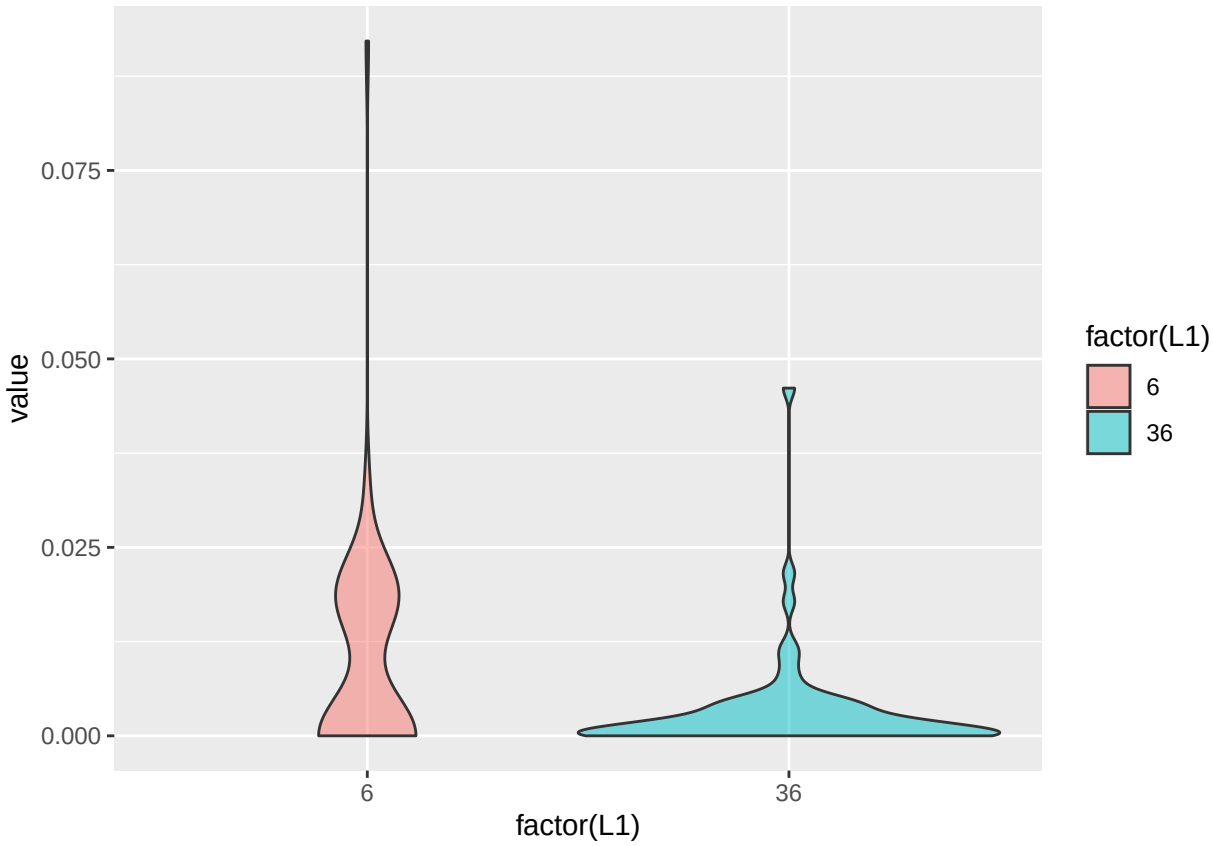

```
##
## ======================================
## Statistic N  Mean  St. Dev.  Min   Max
## --------------------------------------
## Df        2 18.500  24.749    1    36
## Sum Sq    2 0.096   0.111   0.018 0.174
## Mean Sq   2 0.011   0.009   0.005 0.018
## F value   1 3.620           3.620 3.620
## Pr(> F)   1 0.065           0.065 0.065
## --------------------------------------
```



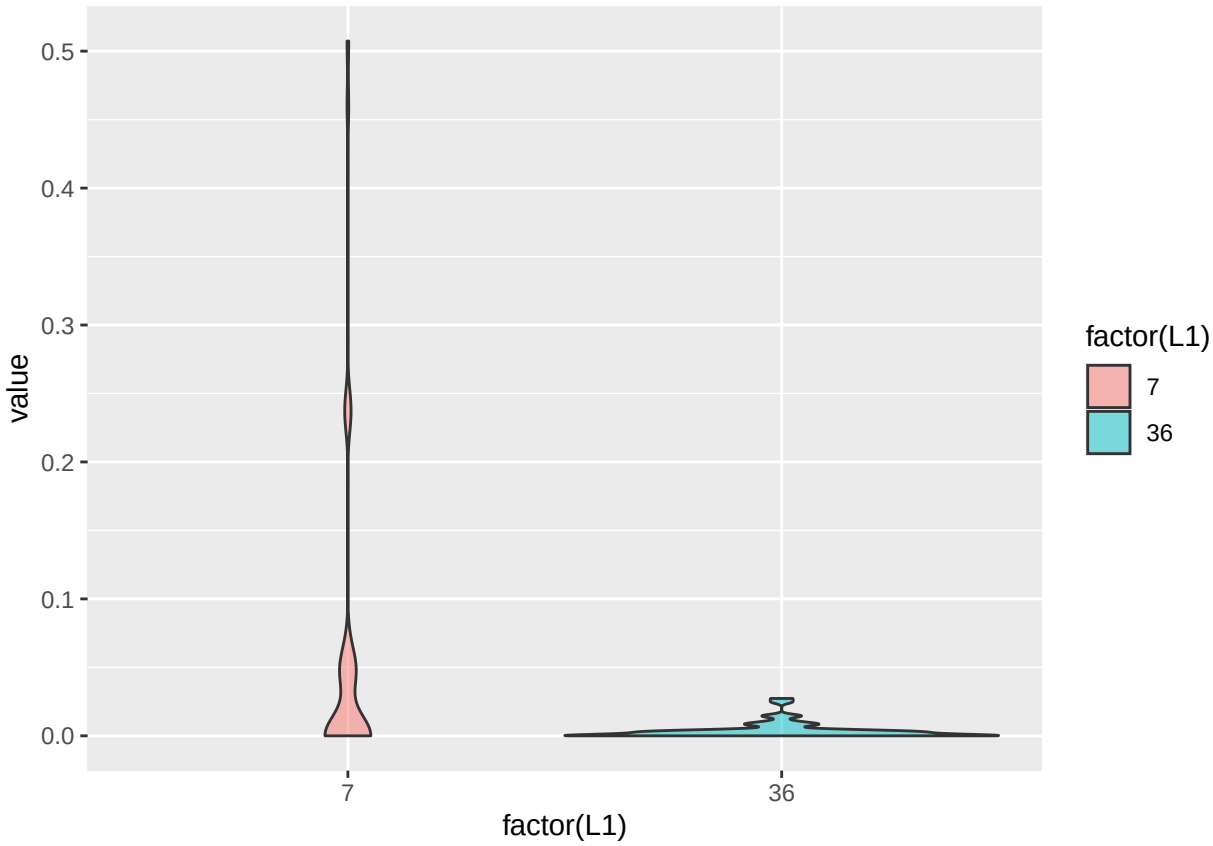

```
##
## ======================================
## Statistic N  Mean  St. Dev.  Min   Max
## --------------------------------------
## Df        2 38.500  53.033    1    76
## Sum Sq    2 0.059   0.065   0.013 0.106
## Mean Sq   2 0.007   0.008   0.001 0.013
## F value   1 9.411           9.411 9.411
## Pr(> F)   1 0.003           0.003 0.003
## --------------------------------------
```



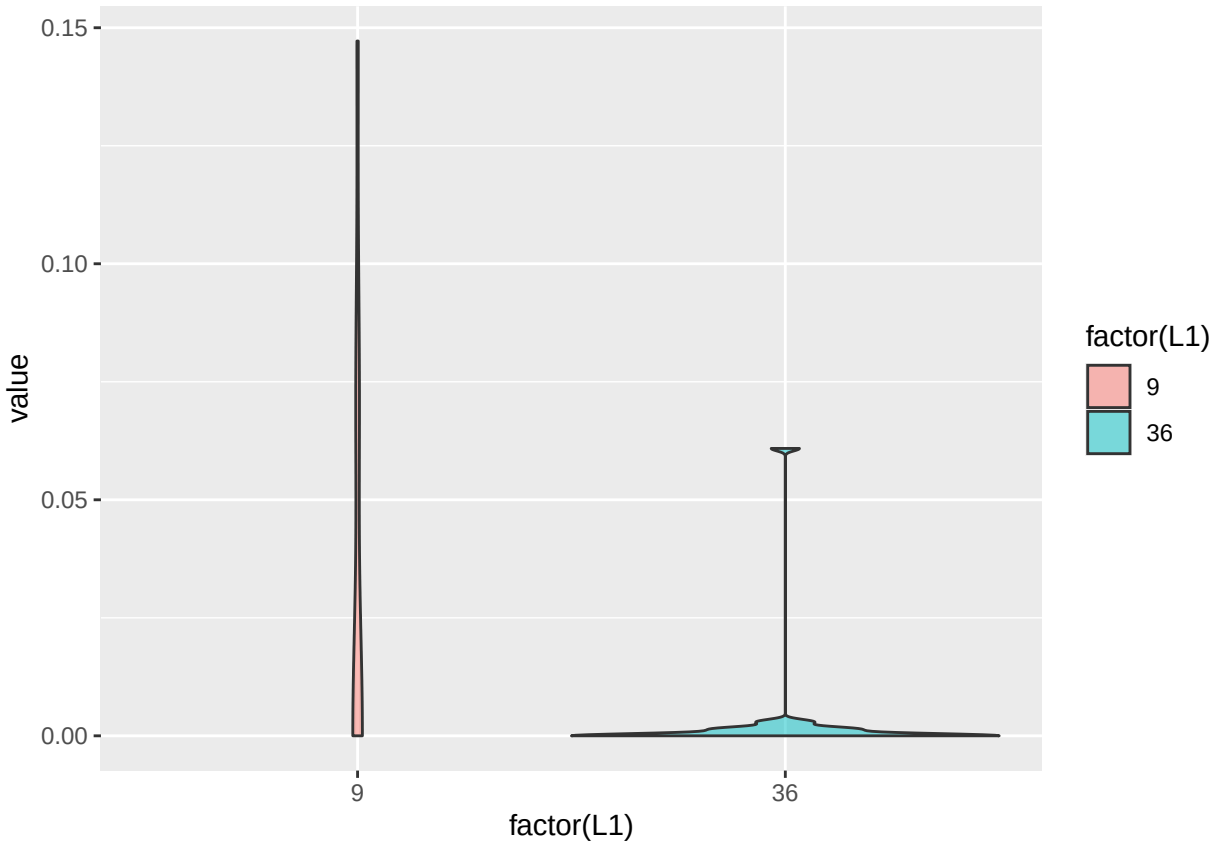

```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min    Max
## ----------------------------------------
## Df        2 39.500  54.447    1      78
## Sum Sq    2 0.030   0.031   0.009  0.052
## Mean Sq   2 0.005   0.006   0.001  0.009
## F value   1 12.712          12.712 12.712
## Pr(> F)   1 0.001           0.001  0.001
## ----------------------------------------
```



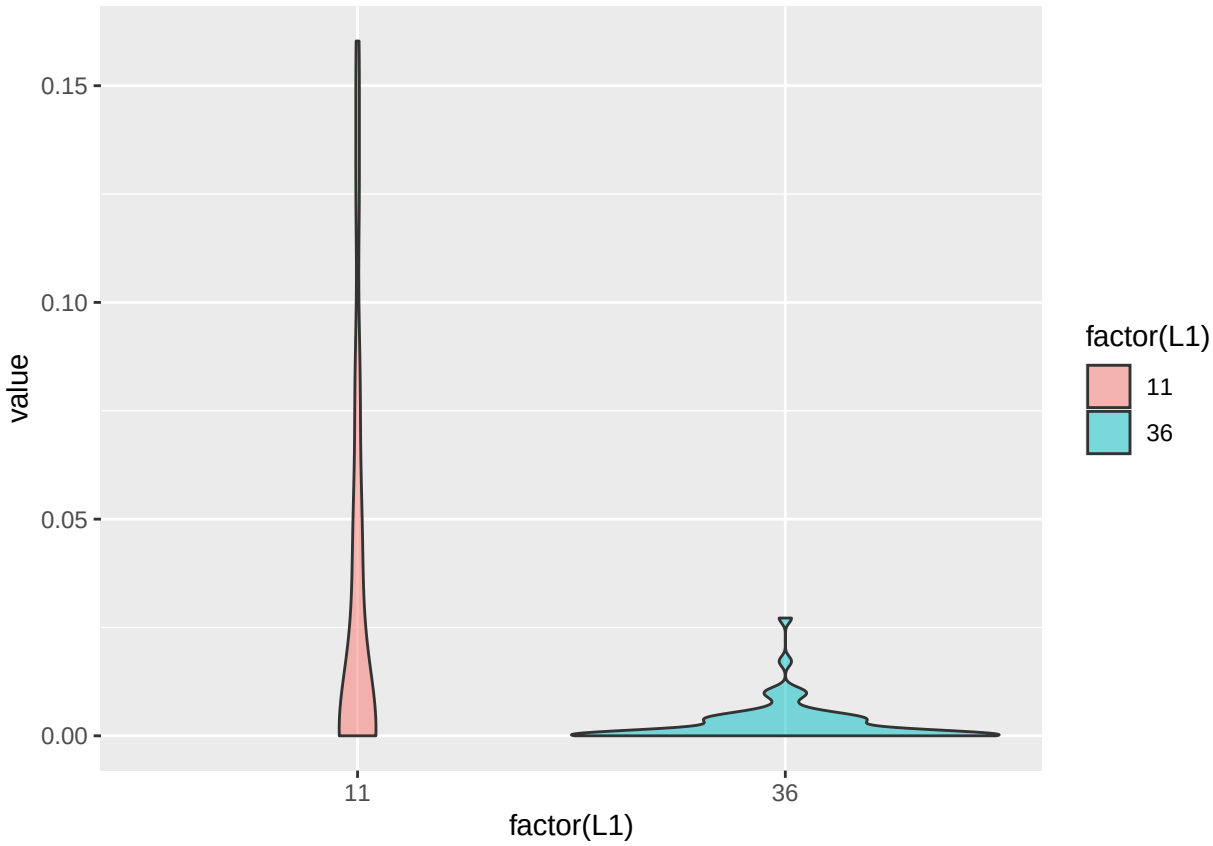

```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min    Max
## ----------------------------------------
## Df        2 63.500  88.388    1     126
## Sum Sq    2 0.009   0.010   0.001  0.016
## Mean Sq   2 0.001   0.001   0.0001 0.001
## F value   1 9.880           9.880  9.880
## Pr(> F)   1 0.002           0.002  0.002
## ----------------------------------------
```



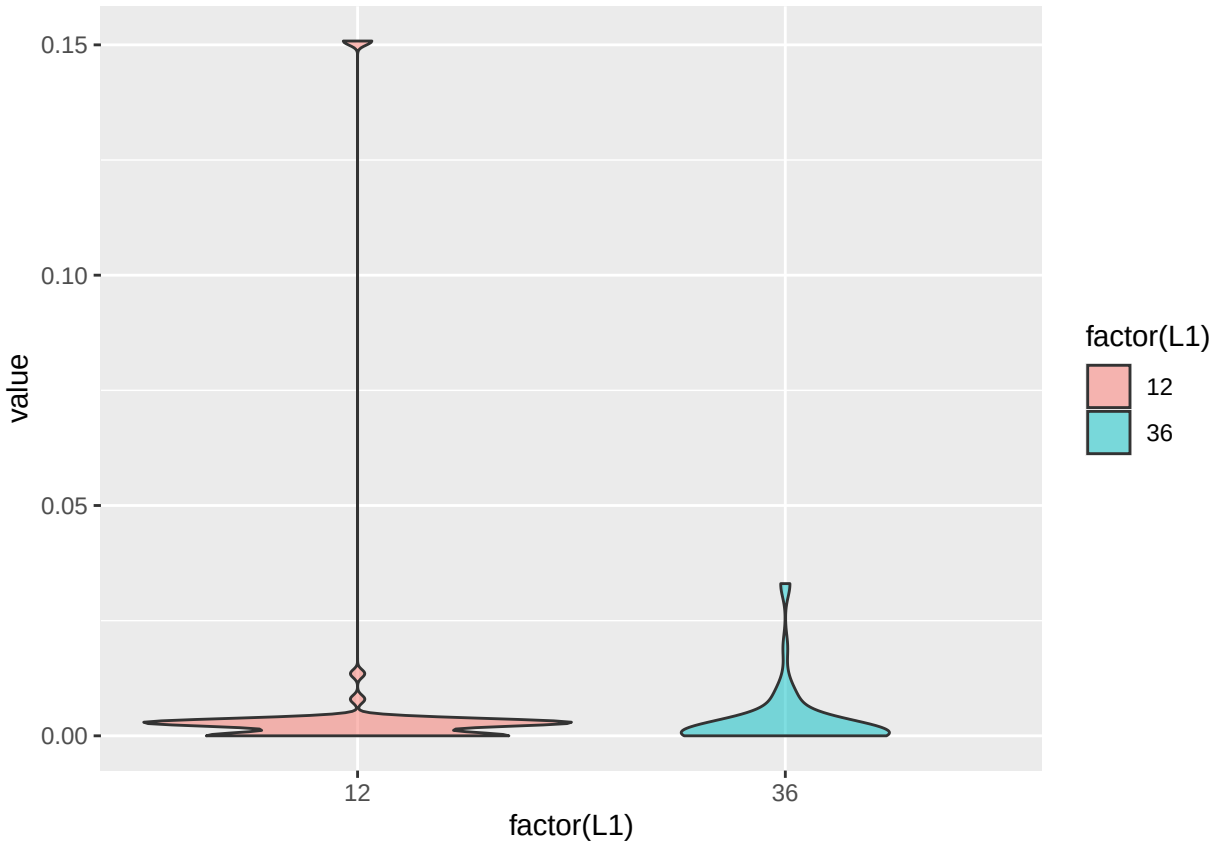

```
##
## ======================================
## Statistic N  Mean  St. Dev.  Min   Max
## --------------------------------------
## Df        2 44.500  61.518    1    88
## Sum Sq    2 0.323   0.375   0.058 0.589
## Mean Sq   2 0.032   0.036   0.007 0.058
## F value   1 8.692           8.692 8.692
## Pr(> F)   1 0.004           0.004 0.004
## --------------------------------------
```



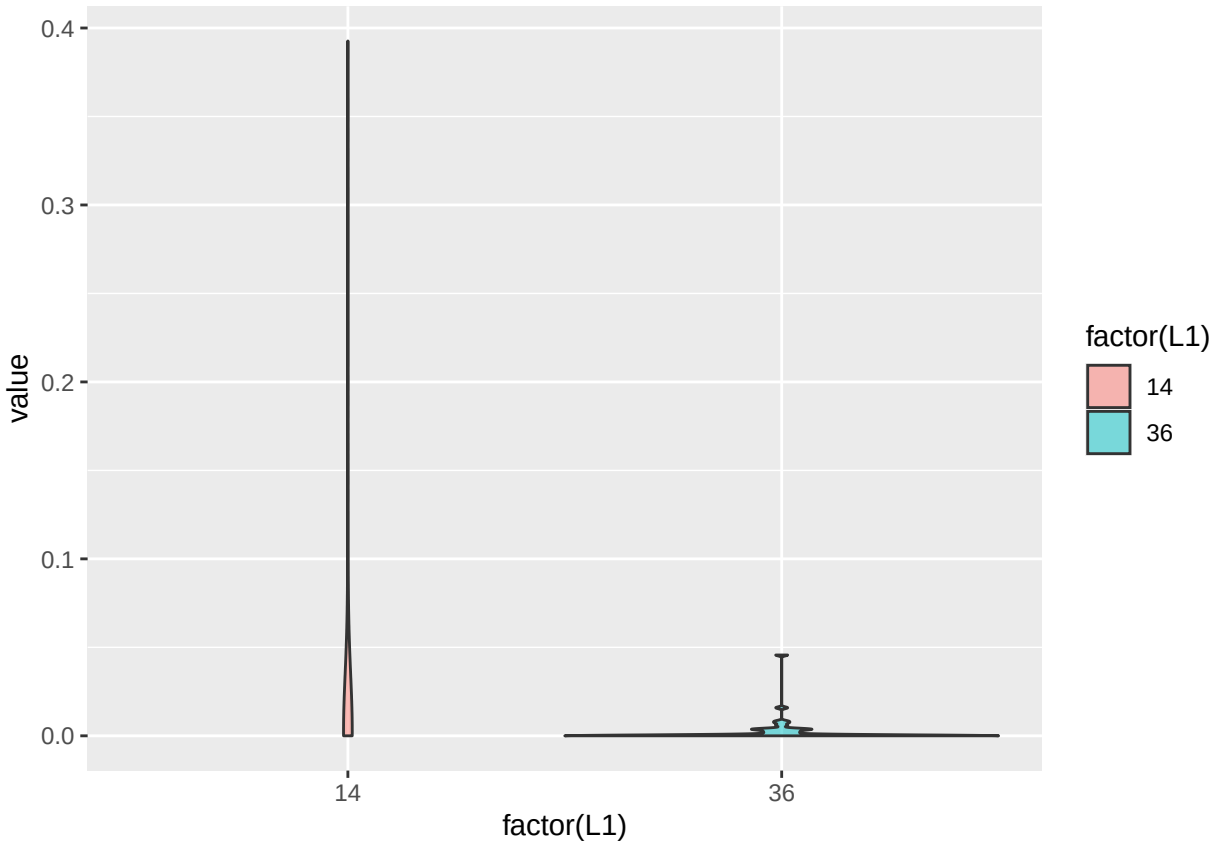

```
##
## ======================================
## Statistic N  Mean  St. Dev.  Min   Max
## --------------------------------------
## Df        2 22.500  30.406    1    44
## Sum Sq    2 0.033   0.033   0.009 0.056
## Mean Sq   2 0.005   0.006   0.001 0.009
## F value   1 7.411           7.411 7.411
## Pr(> F)   1 0.009           0.009 0.009
## --------------------------------------
```



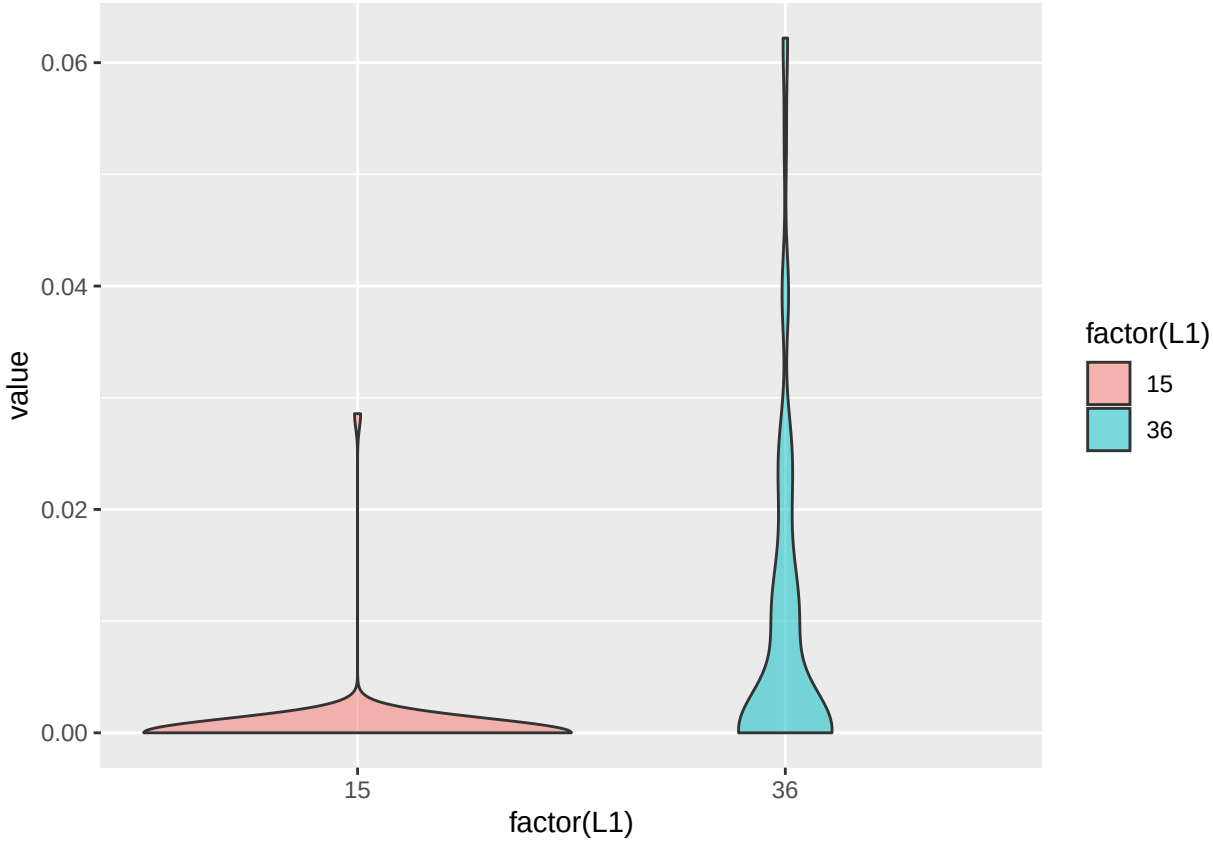

```
##
## ==========================================
## Statistic N  Mean   St. Dev.   Min     Max
## ------------------------------------------
## Df        2 58.500   81.317     1      116
## Sum Sq    2  0.071   0.070    0.022   0.121
## Mean Sq   2  0.011   0.015    0.001   0.022
## F value   1 20.911           20.911  20.911
## Pr(> F)   1 0.00001          0.00001 0.00001
## ------------------------------------------
```



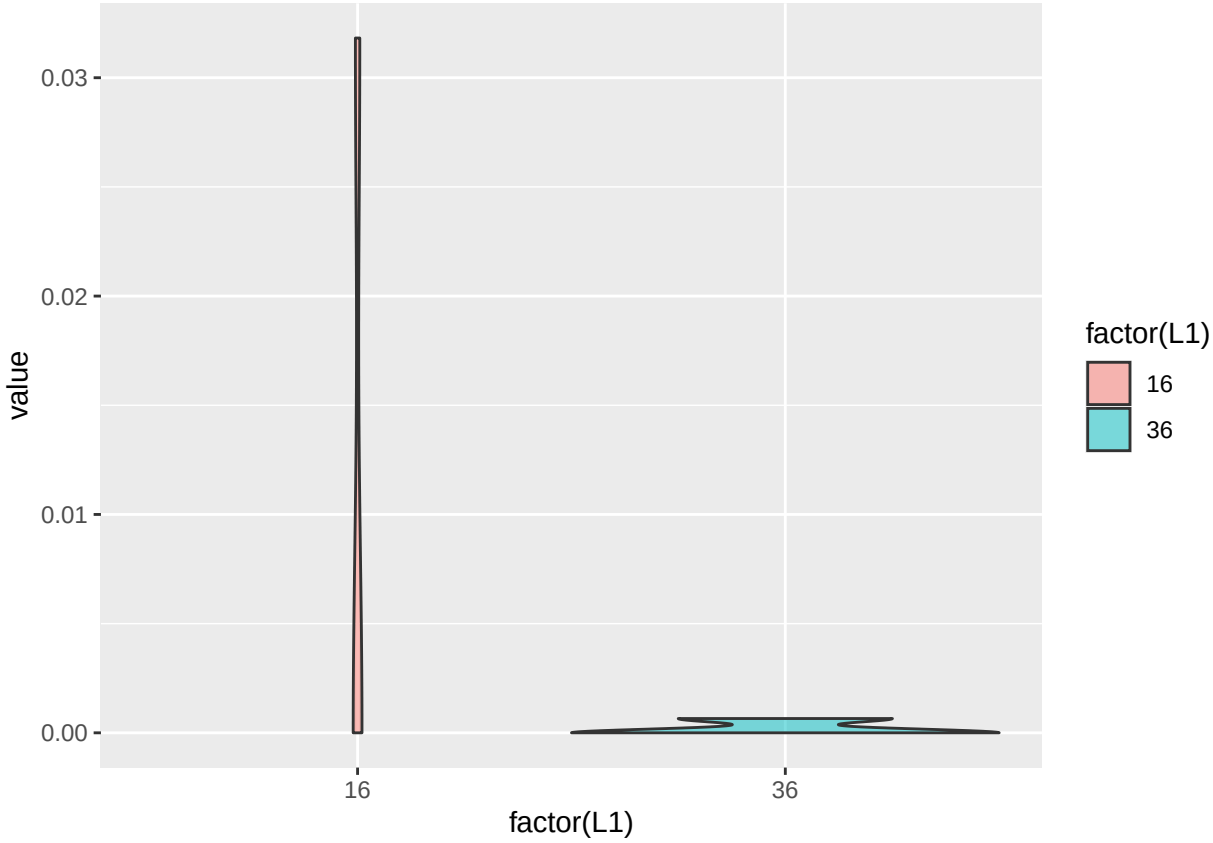

```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min    Max
## ----------------------------------------
## Df        2 54.500  75.660    1     108
## Sum Sq    2 0.023   0.032   0.001  0.045
## Mean Sq   2 0.0005  0.0001  0.0004 0.001
## F value   1 1.275           1.275  1.275
## Pr(> F)   1 0.261           0.261  0.261
## ----------------------------------------
```



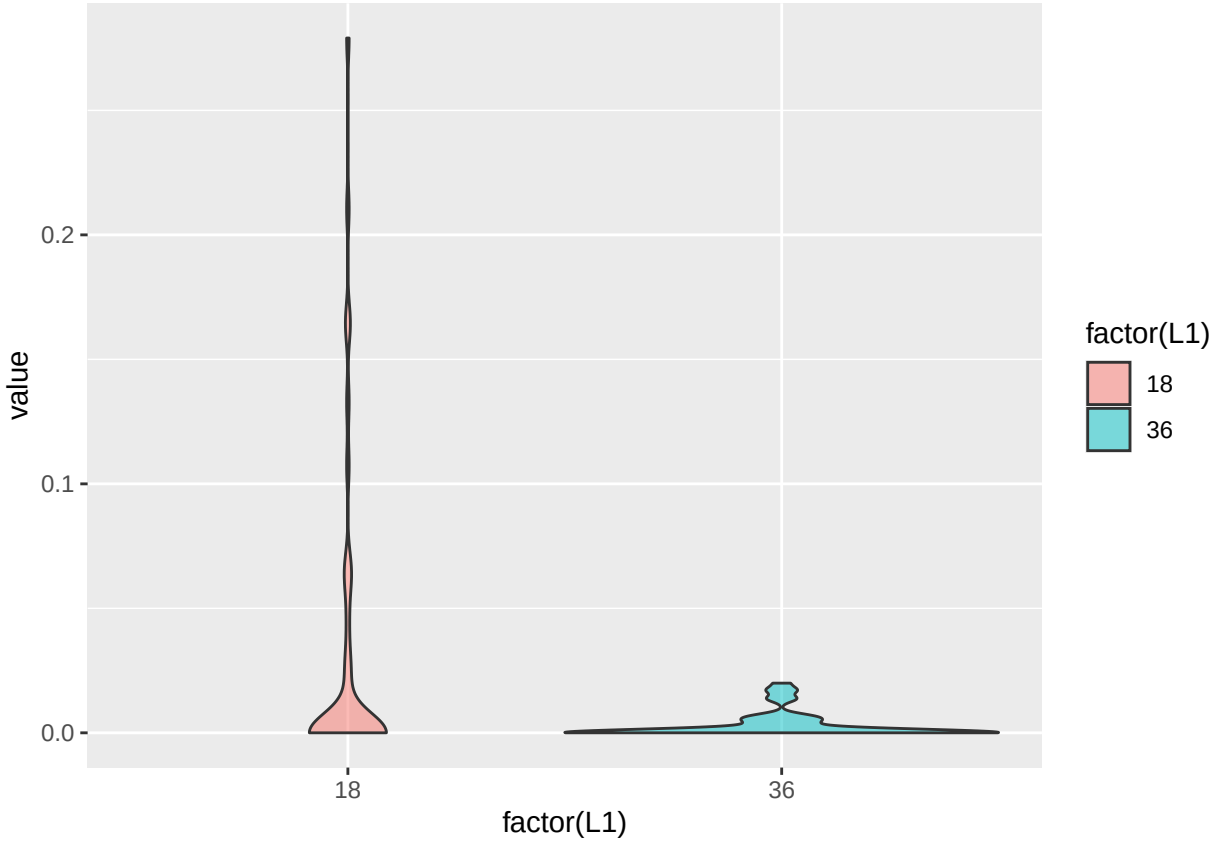

```
##
## ======================================
## Statistic N  Mean  St. Dev.  Min   Max
## --------------------------------------
## Df        2 50.500  70.004    1    100
## Sum Sq    2 0.160   0.204   0.015 0.304
## Mean Sq   2 0.009   0.009   0.003 0.015
## F value   1 5.032           5.032 5.032
## Pr(> F)   1 0.027           0.027 0.027
## --------------------------------------
```




```
##
## ==========================================
## Statistic N  Mean   St. Dev.   Min     Max
## ------------------------------------------
## Df        2 70.500   98.288     1      140
## Sum Sq    2  0.010   0.010    0.003   0.016
## Mean Sq   2  0.001   0.002   0.0001   0.003
## F value   1 23.646           23.646  23.646
## Pr(> F)   1 0.00000          0.00000 0.00000
## ------------------------------------------
```




```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min    Max
## ----------------------------------------
## Df        2 11.500  14.849    1     22
## Sum Sq    2 0.002   0.001   0.001  0.003
## Mean Sq   2 0.0004  0.0004  0.0001 0.001
## F value   1 5.274           5.274  5.274
## Pr(> F)   1 0.032           0.032  0.032
## ----------------------------------------
```



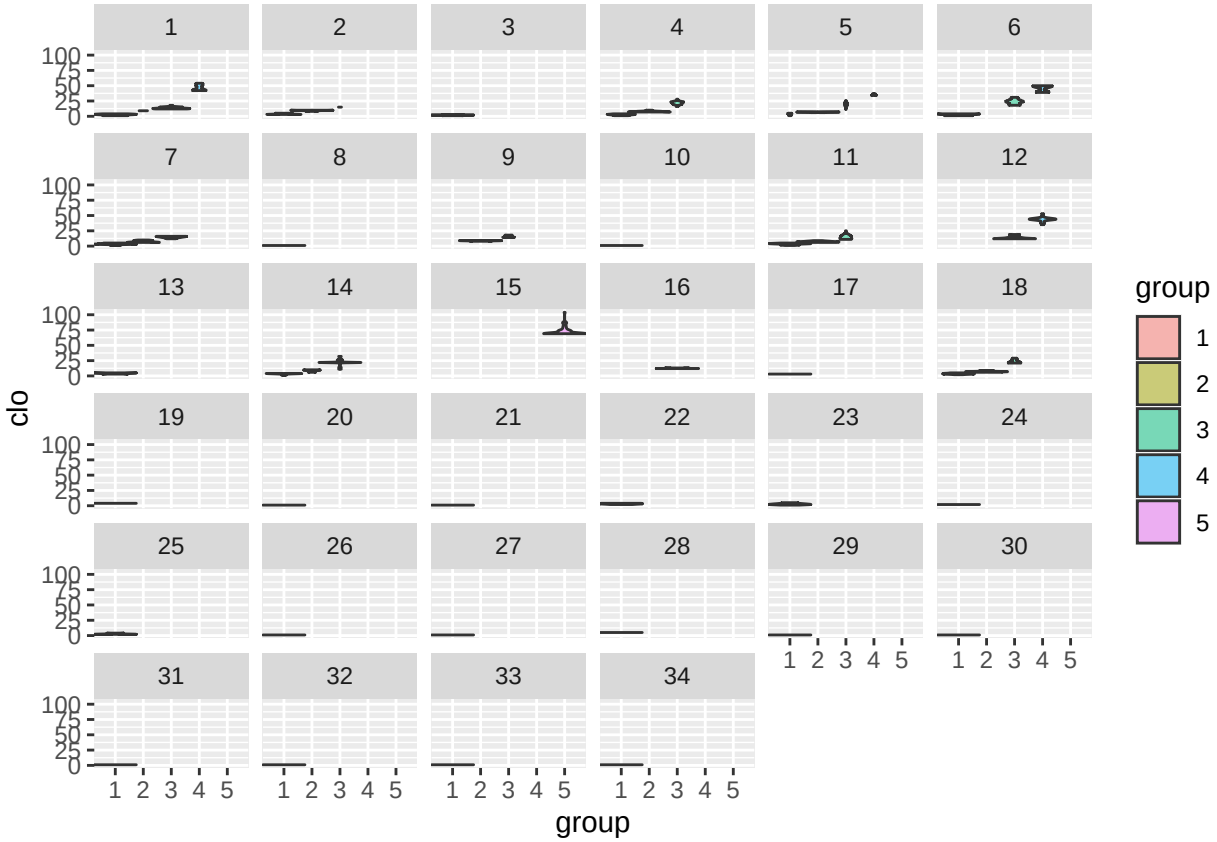

```
##
## ========================================
## Statistic N  Mean  St. Dev.  Min   Max
## ----------------------------------------
## Df        2 43.500  60.104    1    86
## Sum Sq    2 0.099   0.115   0.018 0.180
## Mean Sq   2 0.010   0.011   0.002 0.018
## F value   1 8.402           8.402 8.402
## Pr(> F)   1 0.005           0.005 0.005
## ----------------------------------------
```

```
ggplot(data.frame(clo = V(uni_graph2)$outdegree,group=factor(clustering_closeness$classification)),aes(
facet_wrap(vars(factor(foo_modules[sapply(strsplit(names(V(uni_graph2)),"_"),`[`,2)])))
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning: position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
```

```
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
```




```
ggplot(data.frame(clo = V(uni_graph2)$outdegree,group=factor(clustering_closeness$classification)),aes(
  geom_violin(alpha=0.5,width=1.5)+
  facet_wrap(vars(factor(condensed_baps_adm[as.character(sapply(strsplit(names(V(uni_graph2)),"_"),`[`,
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Warning in max(data$density): ningun argumento finito para max; retornando -Inf
## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0
## Warning: Groups with fewer than two data points have been dropped.
## Warning in max(data$density): ningun argumento finito para max; retornando -Inf
## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Warning in max(data$density): ningun argumento finito para max; retornando -Inf
## Warning: Computation failed in `stat_ydensity()`:
```

```
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf
```

```
## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.

## Warning in max(data$density): ningun argumento finito para max; retornando -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning: position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
```

```
## position_dodge requires non-overlapping x intervals
## position_dodge requires non-overlapping x intervals
```