Matrix and Venn Diagrams

Hannah Rogers

10/29/2018

First, I read in the list of the TAIR10 gene ID's.

```
rm(list=ls())
getwd()
```

```
## [1] "/Users/hannahrogers/Downloads"
```

```
setwd("/Users/hannahrogers/downloads")
getwd()
```

```
## [1] "/Users/hannahrogers/Downloads"
```

```
geneIDs <- read.table("TAIR10_all_gene_models")
head(geneIDs)
```

```
##            V1
## 1 AT1G01010.1
## 2 AT1G01020.1
## 3 AT1G01020.2
## 4 AT1G01030.1
## 5 AT1G01040.1
## 6 AT1G01040.2
```

Then renamed the column "geneIDs".

```
colnames(geneIDs)<-"geneIDs"
head(geneIDs)
```

```
##       geneIDs
## 1 AT1G01010.1
## 2 AT1G01020.1
## 3 AT1G01020.2
## 4 AT1G01030.1
## 5 AT1G01040.1
## 6 AT1G01040.2
```

This data then needed to be trimmed in order to match the TAIR 10 model to the list of targets, so the last two characters were removed.

```
geneIDs$geneIDs <- substr(as.character(geneIDs$geneIDs),
                                       start= 1,
                                       stop=
nchar(as.character(geneIDs$geneIDs) )-2 )
```

The above portion of code creates duplicate values that must be deleted to have an accurate representation of the dataset to compare to the list fo targets. This following code removes the duplicate values and creates a new data set without duplicates.

```r
geneIDs2<-geneIDs[!duplicated(geneIDs),]

IDs<-data.frame(geneIDs2)
```

Next, the targets for each gene are read into a dataframe and matched to the gene ID's from the TAIR 10 genome. This created a logical vector, which was converted to a numeric vector using the function "as.numeric". The zero's indicate that there is not a matching ID for this region.

```r
#TOC1
TOC1 <-read.csv("TOC1.csv")

IDsandTOC1<-IDs[,1]%in%TOC1[,1]

NumTOC1<-as.numeric(IDsandTOC1)

#CCA1
CCA1 <-read.csv("CCA1.csv")

IDsandCCA1<-IDs[,1]%in%CCA1[,1]

NumCCA1<-as.numeric(IDsandCCA1)

#PRR5
PRR5 <-read.csv("PRR5.csv")

IDsandPRR5<-IDs[,1]%in%PRR5[,1]

NumPRR5<-as.numeric(IDsandPRR5)

#PRR7
PRR7 <-read.csv("PRR7.csv")

IDsandPRR7<-IDs[,1]%in%PRR7[,1]

NumPRR7<-as.numeric(IDsandPRR7)

#PRR9
PRR9 <-read.csv("PRR9.csv")

IDsandPRR9<-IDs[,1]%in%PRR9[,1]

NumPRR9<-as.numeric(IDsandPRR9)
```

These numeric vectors are now used to create a data frame, then from the data frame it is converted to a matrix for ease of use and the geneIDs2 character vector is used to name all of the columns based on the gene ID names.

```
dataframeALL<-data.frame(NumCCA1,NumTOC1,NumPRR5,NumPRR7,NumPRR9)
matrixALL<-as.matrix(dataframeALL)
rownames(matrixALL)<-geneIDs2
head(matrixALL)

##           NumCCA1 NumTOC1 NumPRR5 NumPRR7 NumPRR9
## AT1G01010       0       0       0       0       0
## AT1G01020       0       0       0       0       0
## AT1G01030       0       0       1       0       0
## AT1G01040       1       0       1       0       0
## AT1G01046       0       0       0       0       0
## AT1G01050       0       0       0       0       0

dim(matrixALL)

## [1] 33603     5
```

The rows were summed based on how many 1's were present in the entire column, then all of the sums were calculated for all of the column combinations in preparation for the quintuple venn diagram.

```
length(grep("1",matrixALL[,1]))

## [1] 1761

length(grep("1",matrixALL[,2]))

## [1] 334

length(grep("1",matrixALL[,3]))

## [1] 6326

length(grep("1",matrixALL[,4]))

## [1] 1096

length(grep("1",matrixALL[,5]))

## [1] 132

sum(matrixALL[,1] == "1" & grepl("1", matrixALL[,2]))

## [1] 89

sum(matrixALL[,1] == "1" & grepl("1", matrixALL[,3]))

## [1] 984

sum(matrixALL[,1] == "1" & grepl("1", matrixALL[,4]))
```

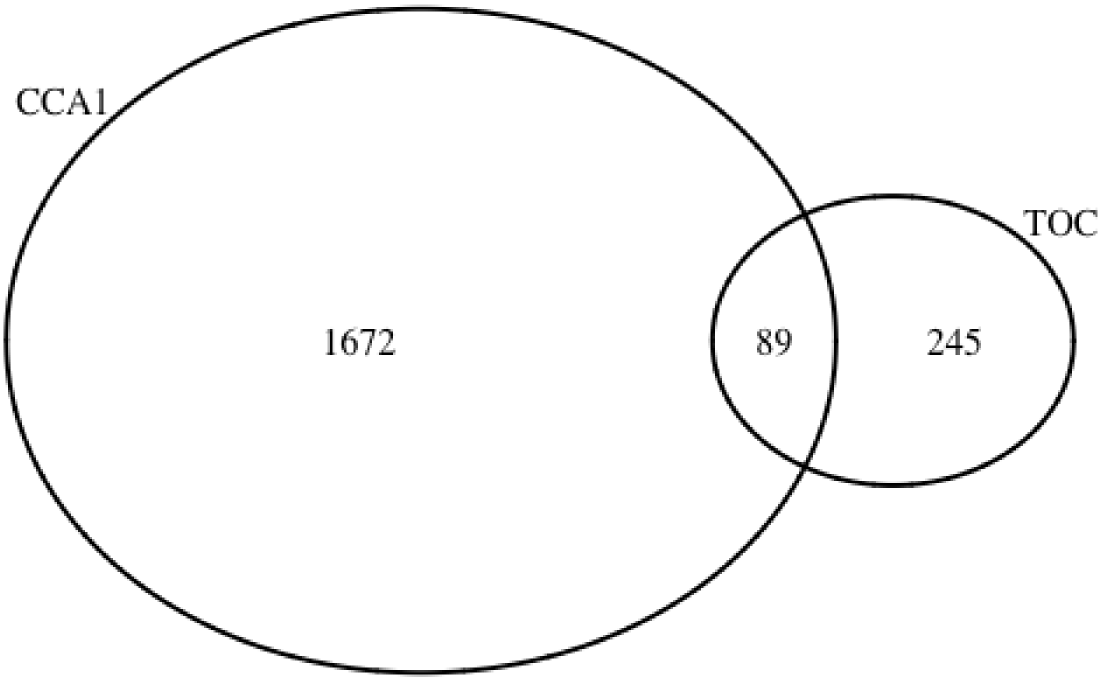
```
## [1] 310

sum(matrixALL[,1] == "1" & grepl("1", matrixALL[,5]))

## [1] 50

sum(matrixALL[,2] == "1" & grepl("1", matrixALL[,3]))

## [1] 300

sum(matrixALL[,2] == "1" & grepl("1", matrixALL[,4]))

## [1] 137

sum(matrixALL[,2] == "1" & grepl("1", matrixALL[,5]))

## [1] 31

sum(matrixALL[,3] == "1" & grepl("1", matrixALL[,4]))

## [1] 1046

sum(matrixALL[,3] == "1" & grepl("1", matrixALL[,5]))

## [1] 107

sum(matrixALL[,4] == "1" & grepl("1", matrixALL[,5]))

## [1] 103

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,3]))

## [1] 83

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,4]))

## [1] 48

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,5]))

## [1] 17

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,4]))

## [1] 298

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,5]))

## [1] 39
```

```r
sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,4]) & grepl("1",
matrixALL[,5]))

## [1] 40

sum(grepl("1", matrixALL[,2]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,4]))

## [1] 135

sum(grepl("1", matrixALL[,2]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,5]))

## [1] 28

sum(grepl("1", matrixALL[,2]) & grepl("1", matrixALL[,4]) & grepl("1",
matrixALL[,5]))

## [1] 27

sum(grepl("1", matrixALL[,3]) & grepl("1", matrixALL[,4]) & grepl("1",
matrixALL[,5]))

## [1] 99

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,3])
  & grepl("1", matrixALL[,4]))

## [1] 47

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,3])
    & grepl("1", matrixALL[,5]))

## [1] 15

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,4])
    & grepl("1", matrixALL[,5]))

## [1] 15

sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,4])
    & grepl("1", matrixALL[,5]))

## [1] 39

sum(grepl("1", matrixALL[,2]) & grepl("1", matrixALL[,3]) & grepl("1",
matrixALL[,4])
    & grepl("1", matrixALL[,5]))

## [1] 27
```

```
sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,3])
    & grepl("1", matrixALL[,4]) & grepl("1", matrixALL[,5]))

## [1] 15
```

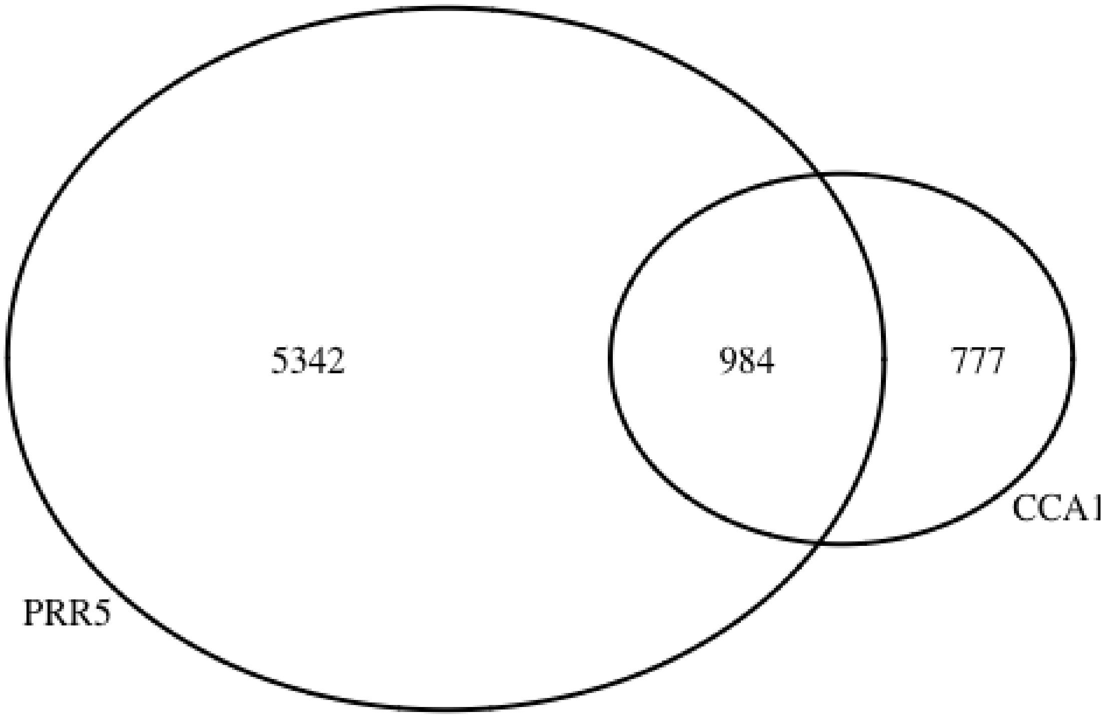
The pairwise venn diagrams were created.

```
library(VennDiagram)

## Loading required package: grid

## Loading required package: futile.logger

grid.newpage()
draw.pairwise.venn(area1 = 1761, area2 = 334, cross.area = 89, category =
c("CCA1",

"TOC1"))
```



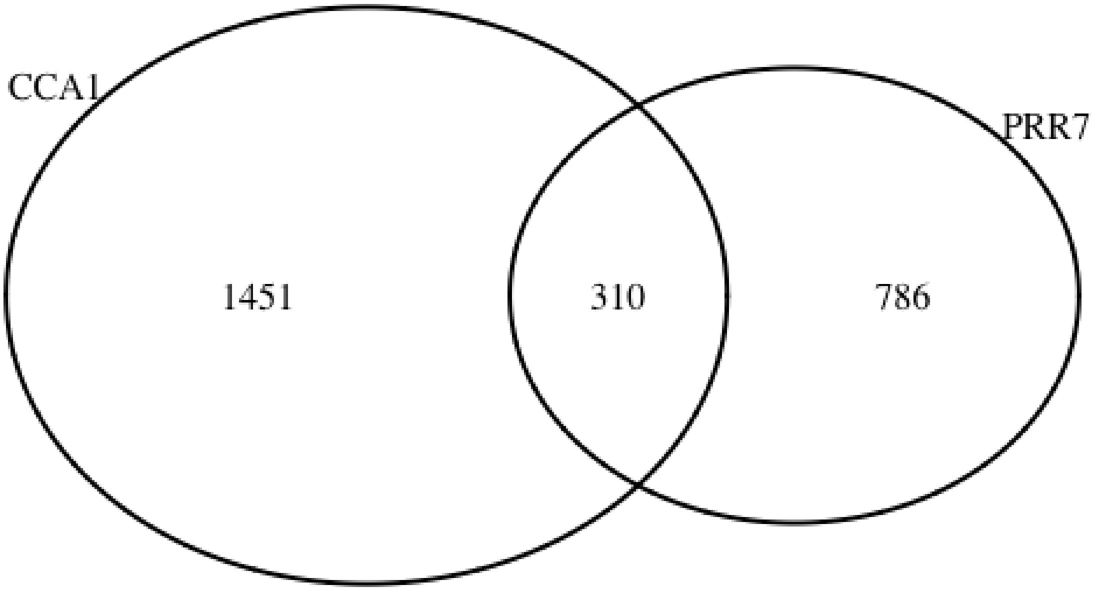
```
## (polygon[GRID.polygon.1], polygon[GRID.polygon.2],
polygon[GRID.polygon.3], polygon[GRID.polygon.4], text[GRID.text.5],
text[GRID.text.6], text[GRID.text.7], text[GRID.text.8], text[GRID.text.9])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 1761, area2 = 6326, cross.area = 984, category =
c("CCA1",

"PRR5"))
```

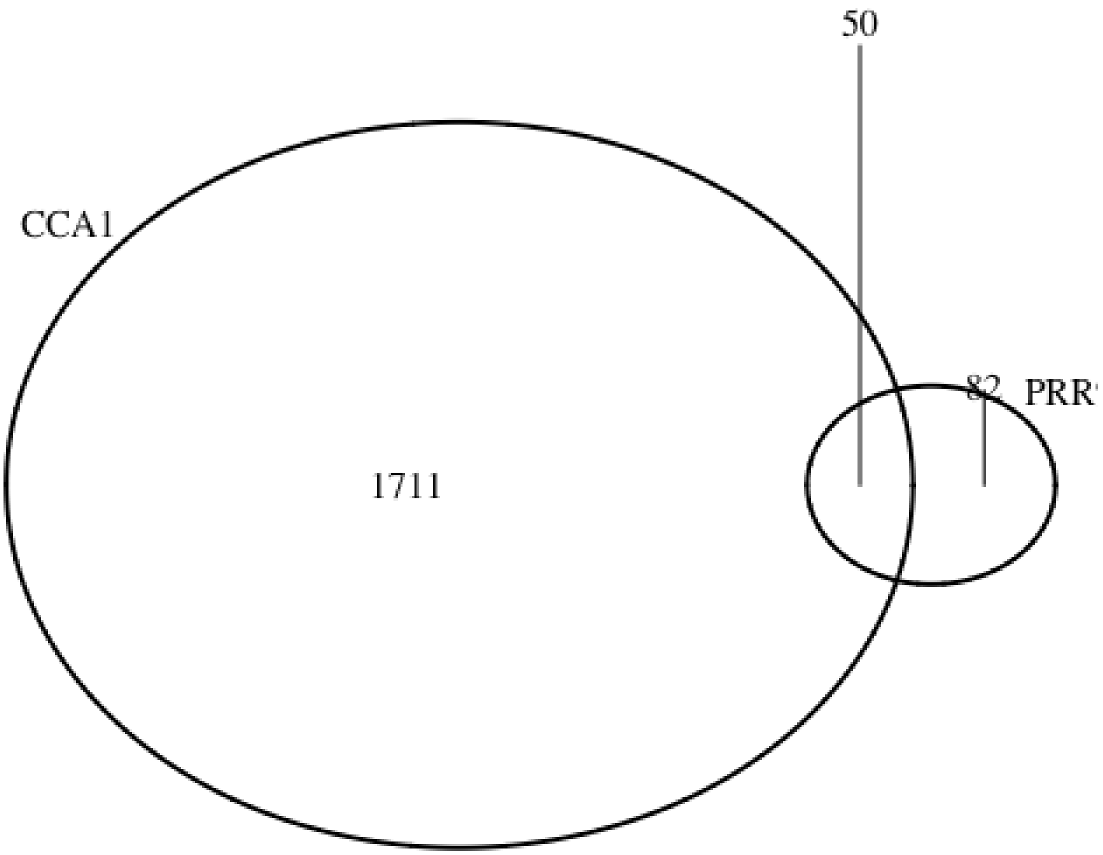


```
## (polygon[GRID.polygon.10], polygon[GRID.polygon.11],
polygon[GRID.polygon.12], polygon[GRID.polygon.13], text[GRID.text.14],
text[GRID.text.15], text[GRID.text.16], text[GRID.text.17],
text[GRID.text.18])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 1761, area2 = 1096, cross.area = 310, category =
c("CCA1",

"PRR7"))
```

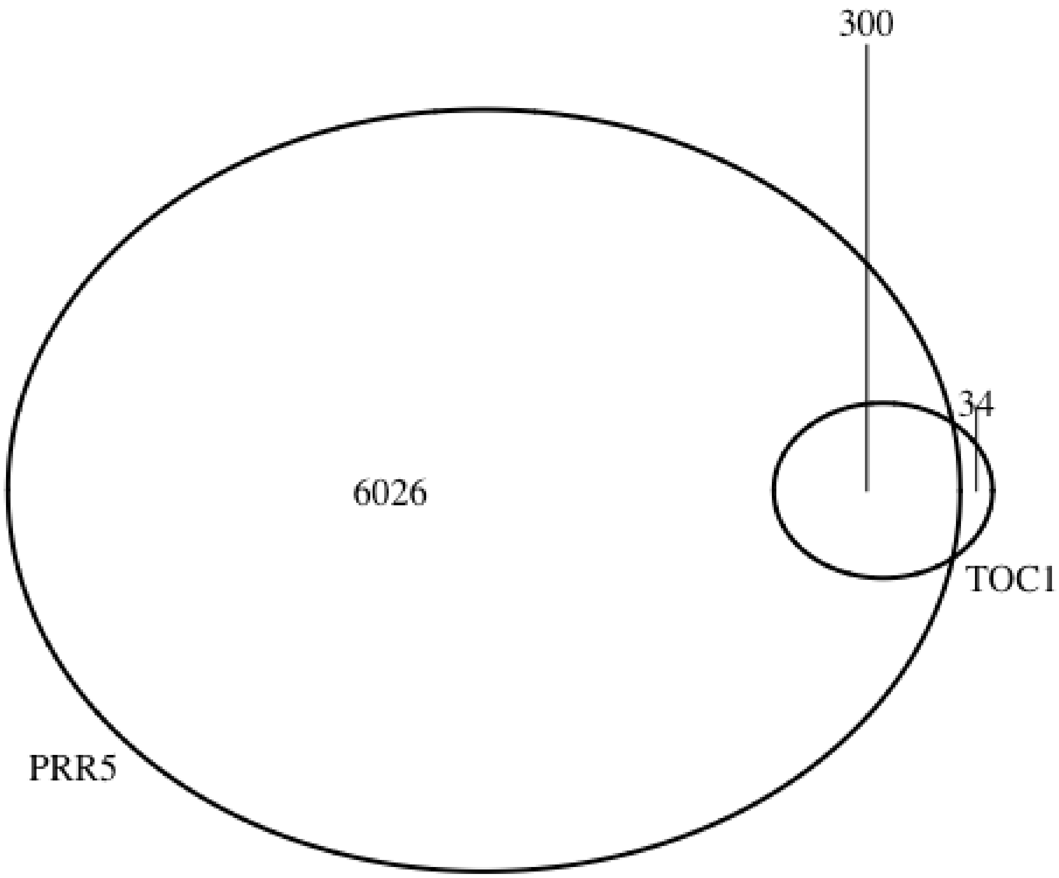


```
## (polygon[GRID.polygon.19], polygon[GRID.polygon.20],
polygon[GRID.polygon.21], polygon[GRID.polygon.22], text[GRID.text.23],
text[GRID.text.24], text[GRID.text.25], text[GRID.text.26],
text[GRID.text.27])

grid.newpage()
draw.pairwise.venn(area1 = 1761, area2 = 132, cross.area = 50, category =
c("CCA1",

"PRR9"))
```



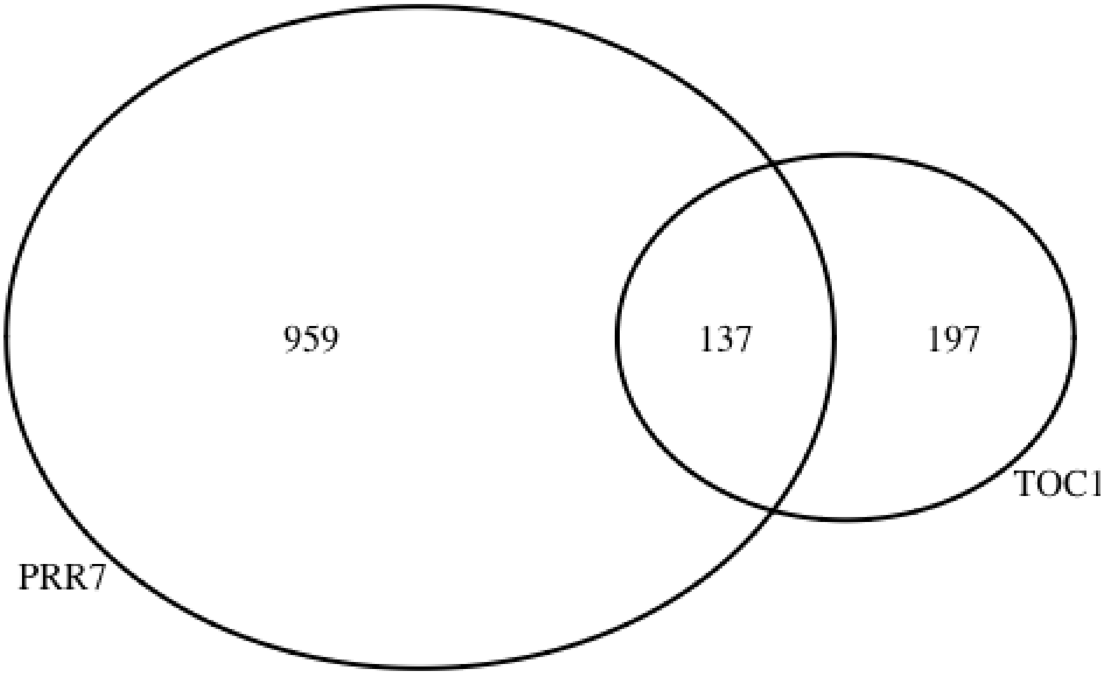
```
## (polygon[GRID.polygon.28], polygon[GRID.polygon.29],
polygon[GRID.polygon.30], polygon[GRID.polygon.31], text[GRID.text.32],
text[GRID.text.33], lines[GRID.lines.34], text[GRID.text.35],
lines[GRID.lines.36], text[GRID.text.37], text[GRID.text.38])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 334, area2 = 6326, cross.area = 300, category =
c("TOC1",

"PRR5"))
```



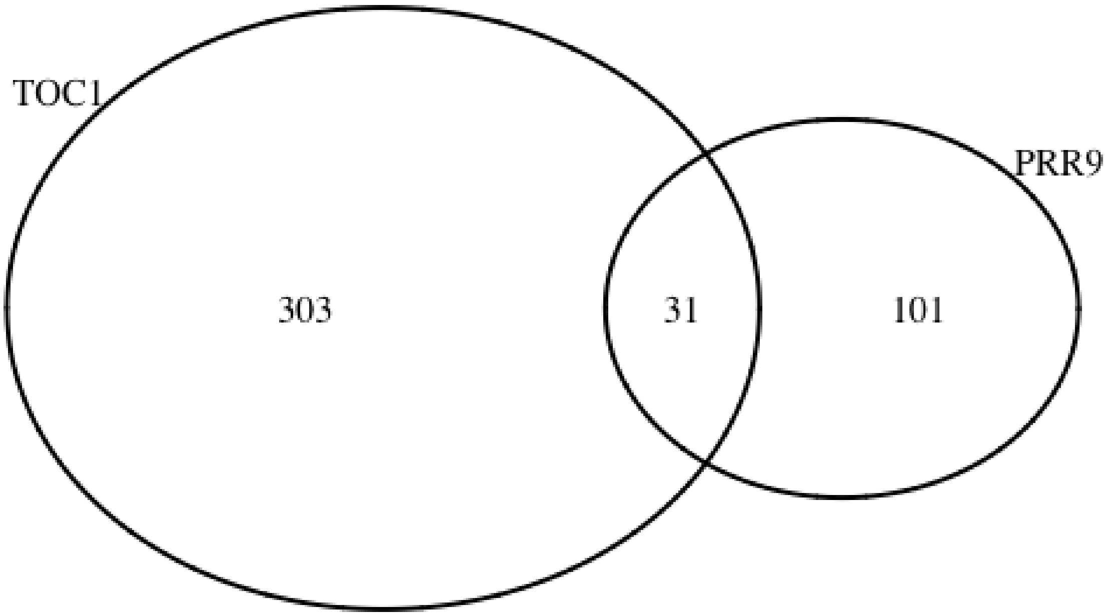
```
## (polygon[GRID.polygon.39], polygon[GRID.polygon.40],
polygon[GRID.polygon.41], polygon[GRID.polygon.42], text[GRID.text.43],
text[GRID.text.44], lines[GRID.lines.45], text[GRID.text.46],
lines[GRID.lines.47], text[GRID.text.48], text[GRID.text.49])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 334, area2 = 1096, cross.area = 137, category =
c("TOC1",

"PRR7"))
```



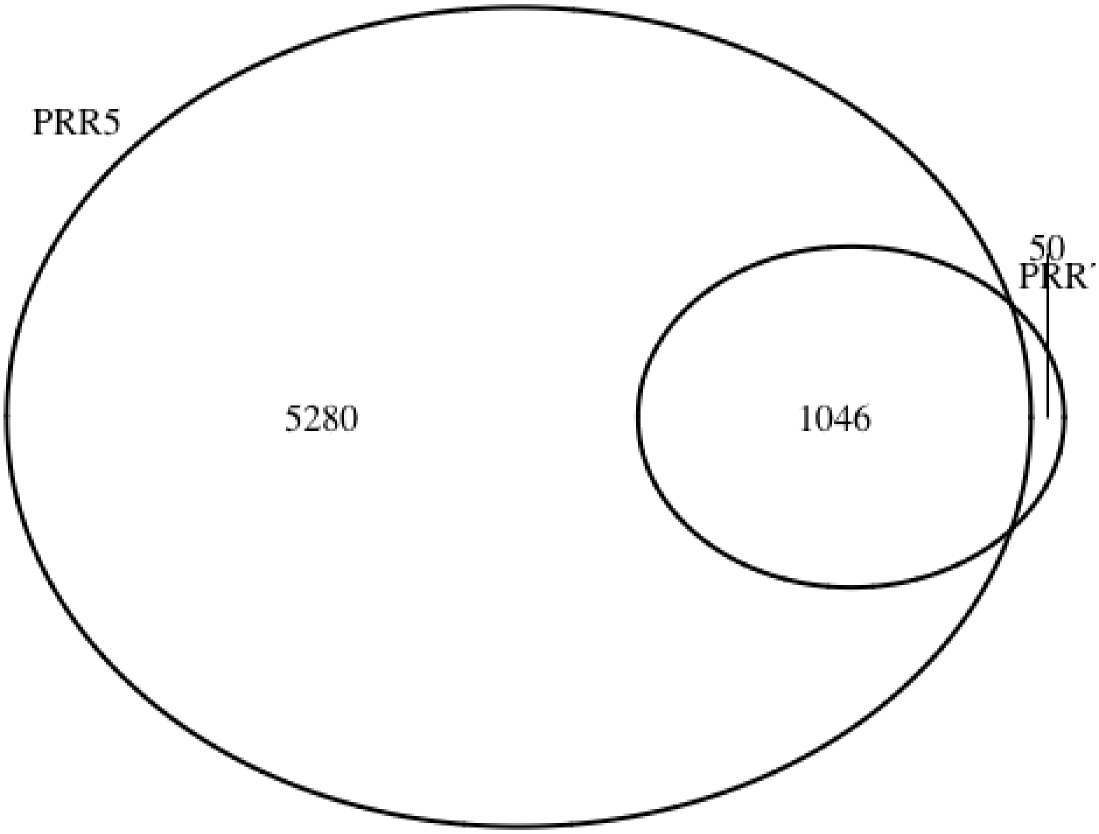
```
## (polygon[GRID.polygon.50], polygon[GRID.polygon.51],
polygon[GRID.polygon.52], polygon[GRID.polygon.53], text[GRID.text.54],
text[GRID.text.55], text[GRID.text.56], text[GRID.text.57],
text[GRID.text.58])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 334, area2 = 132, cross.area = 31, category =
c("TOC1",

"PRR9"))
```



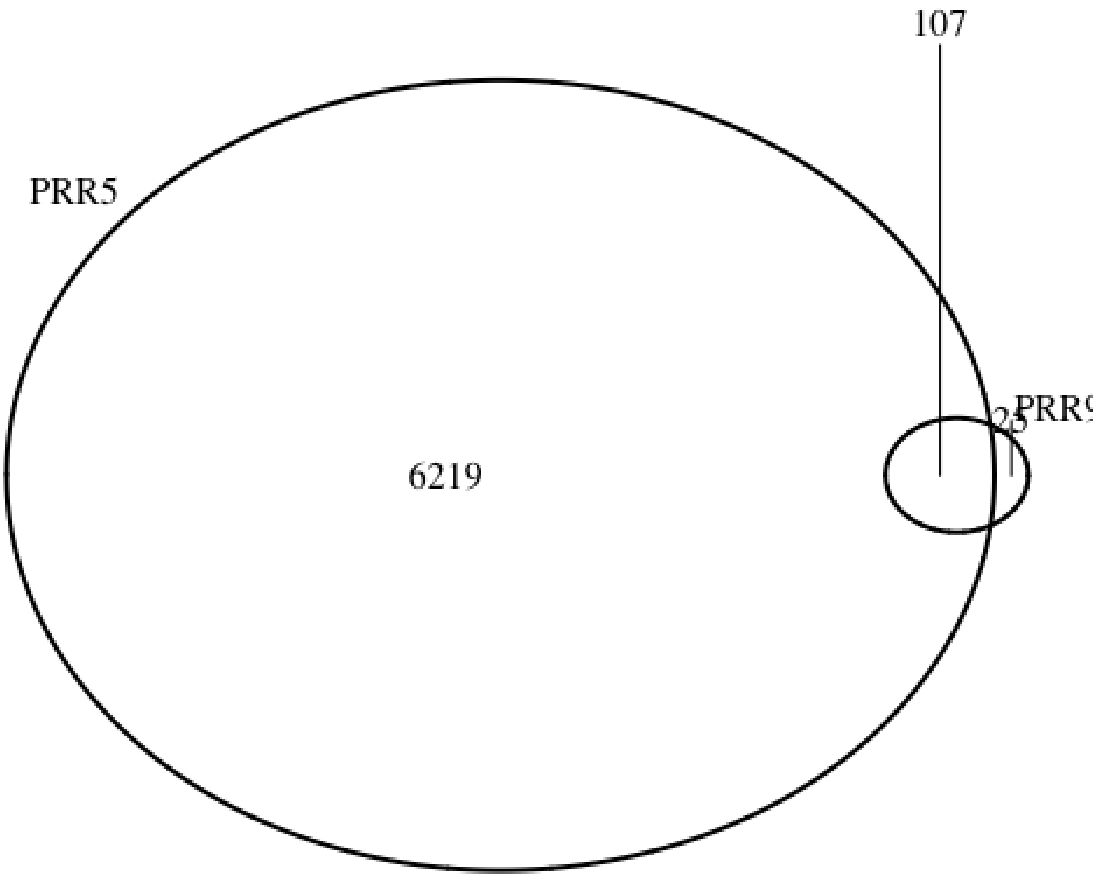
```
## (polygon[GRID.polygon.59], polygon[GRID.polygon.60],
polygon[GRID.polygon.61], polygon[GRID.polygon.62], text[GRID.text.63],
text[GRID.text.64], text[GRID.text.65], text[GRID.text.66],
text[GRID.text.67])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 6326, area2 = 1096, cross.area = 1046, category =
c("PRR5",

"PRR7"))
```



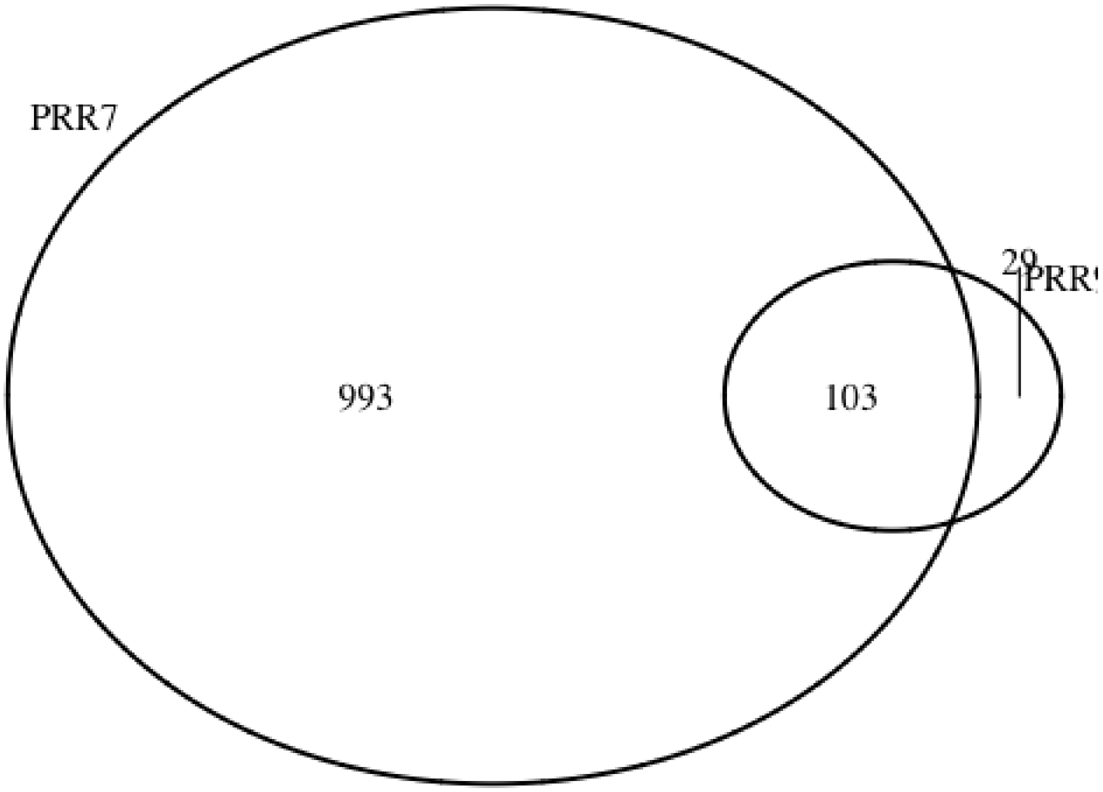
```
## (polygon[GRID.polygon.68], polygon[GRID.polygon.69],
polygon[GRID.polygon.70], polygon[GRID.polygon.71], text[GRID.text.72],
text[GRID.text.73], lines[GRID.lines.74], text[GRID.text.75],
text[GRID.text.76], text[GRID.text.77])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 6326, area2 = 132, cross.area = 107, category =
c("PRR5",

"PRR9"))
```



```
## (polygon[GRID.polygon.78], polygon[GRID.polygon.79],
polygon[GRID.polygon.80], polygon[GRID.polygon.81], text[GRID.text.82],
text[GRID.text.83], lines[GRID.lines.84], text[GRID.text.85],
lines[GRID.lines.86], text[GRID.text.87], text[GRID.text.88])
```

```
grid.newpage()
draw.pairwise.venn(area1 = 1096, area2 = 132, cross.area = 103, category =
c("PRR7",


"PRR9"))
```

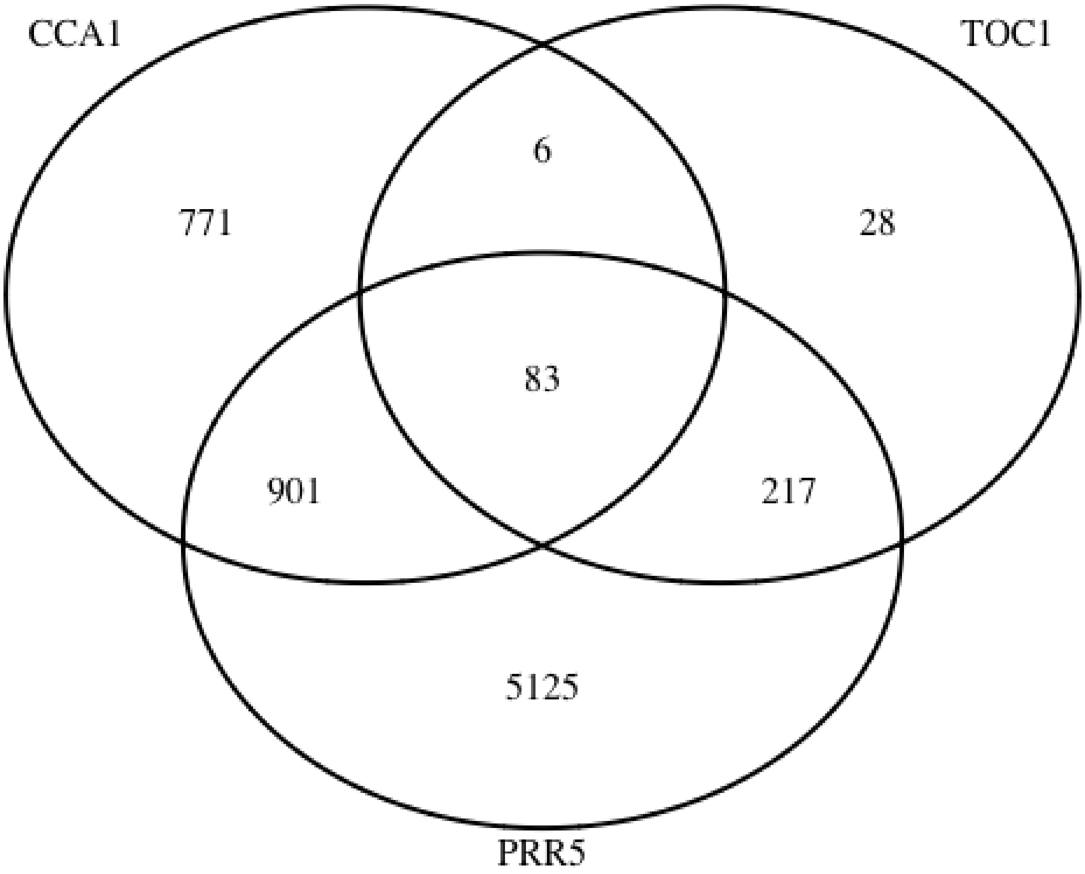


```
## (polygon[GRID.polygon.89], polygon[GRID.polygon.90],
polygon[GRID.polygon.91], polygon[GRID.polygon.92], text[GRID.text.93],
text[GRID.text.94], lines[GRID.lines.95], text[GRID.text.96],
text[GRID.text.97], text[GRID.text.98])
```

```
sum(grepl("1", matrixALL[,1]) & grepl("1", matrixALL[,2]) & grepl("1",
matrixALL[,3]))
```

```
## [1] 83
```

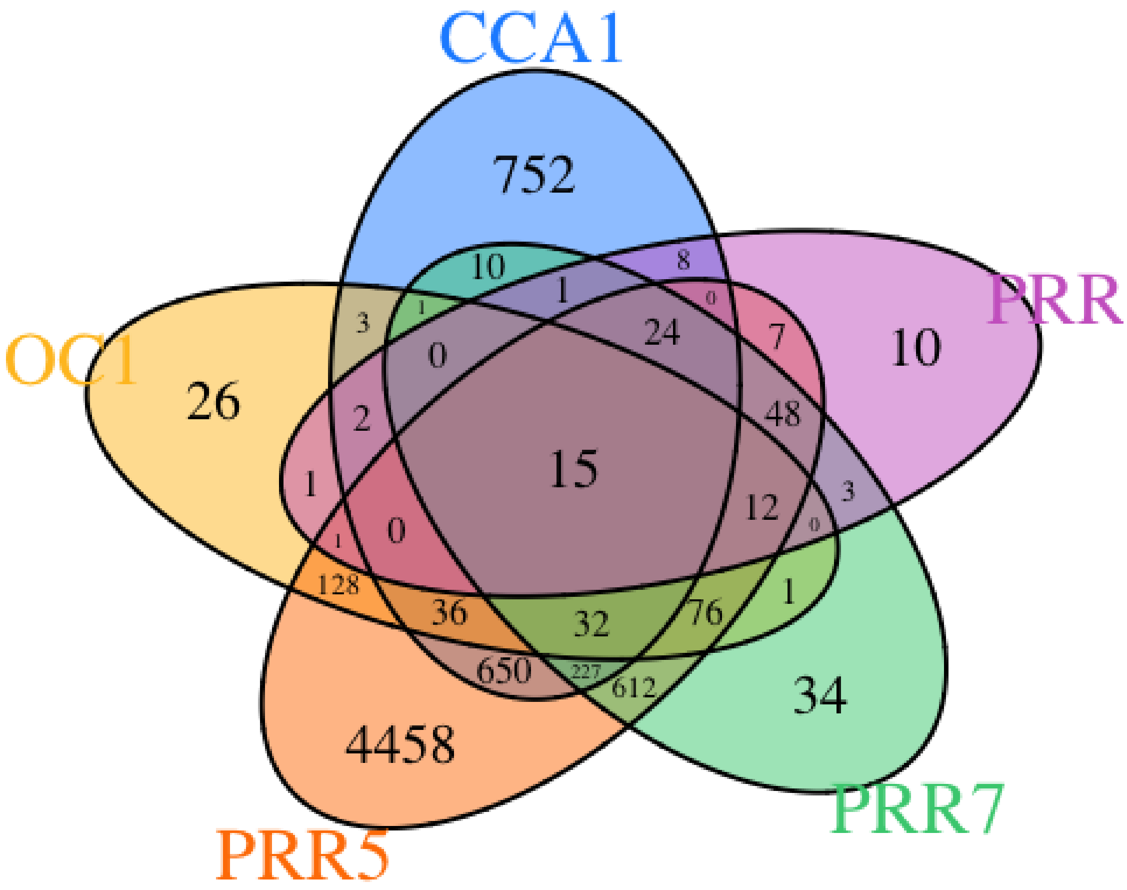
```
grid.newpage()
draw.triple.venn(area1 = 1761, area2 = 334, area3 = 6326, n12 = 89, n23 =
300, n13 = 984,
                 n123 = 83, category = c("CCA1", "TOC1", "PRR5"))
```



```
## (polygon[GRID.polygon.99], polygon[GRID.polygon.100],
polygon[GRID.polygon.101], polygon[GRID.polygon.102],
polygon[GRID.polygon.103], polygon[GRID.polygon.104], text[GRID.text.105],
text[GRID.text.106], text[GRID.text.107], text[GRID.text.108],
text[GRID.text.109], text[GRID.text.110], text[GRID.text.111],
text[GRID.text.112], text[GRID.text.113], text[GRID.text.114])
```

Lastly, the counts from above were used to plug into this code that creates the five-way venn diagram.

```
library(VennDiagram);
venn.plot <- draw.quintuple.venn(
  area1 = 1761, area2 = 334, area3 = 6326, area4 = 1096, area5 = 132,
  n12 = 89, n13 = 984, n14 = 310, n15 = 50,
  n23 = 300, n24 = 137, n25 = 31,
  n34 = 1046, n35 = 107, n45 = 103,
  n123 = 83, n124 = 48, n125 = 17,
  n134 = 298, n135 = 39, n145 = 40,
  n234 = 135, n235 = 28, n245 = 27,
  n345 = 99,
  n1234 = 47, n1235 = 15, n1245 = 15,
  n1345 = 39, n2345 = 27,
  n12345 = 15,
  category = c("CCA1", "TOC1", "PRR5", "PRR7", "PRR9"),
  fill = c("dodgerblue", "goldenrod1", "darkorange1", "seagreen3",
"orchid3"),
  cat.col = c("dodgerblue", "goldenrod1", "darkorange1", "seagreen3",
"orchid3"),
  cat.cex = 2,
  margin = 0.05,
  cex = c(
    1.5, 1.5, 1.5, 1.5, 1.5, 1, 0.8, 1, 0.8, 1, 0.8, 1, 0.8, 1, 0.8,
    1, 0.55, 1, 0.55, 1, 0.55, 1, 0.55, 1, 0.55, 1, 1, 1, 1, 1, 1.5),
  ind = TRUE);
```



```
dev.off()

## null device
##           1
```