Some Plot in R

Just a note

blade_jack163.com

2018.2.13

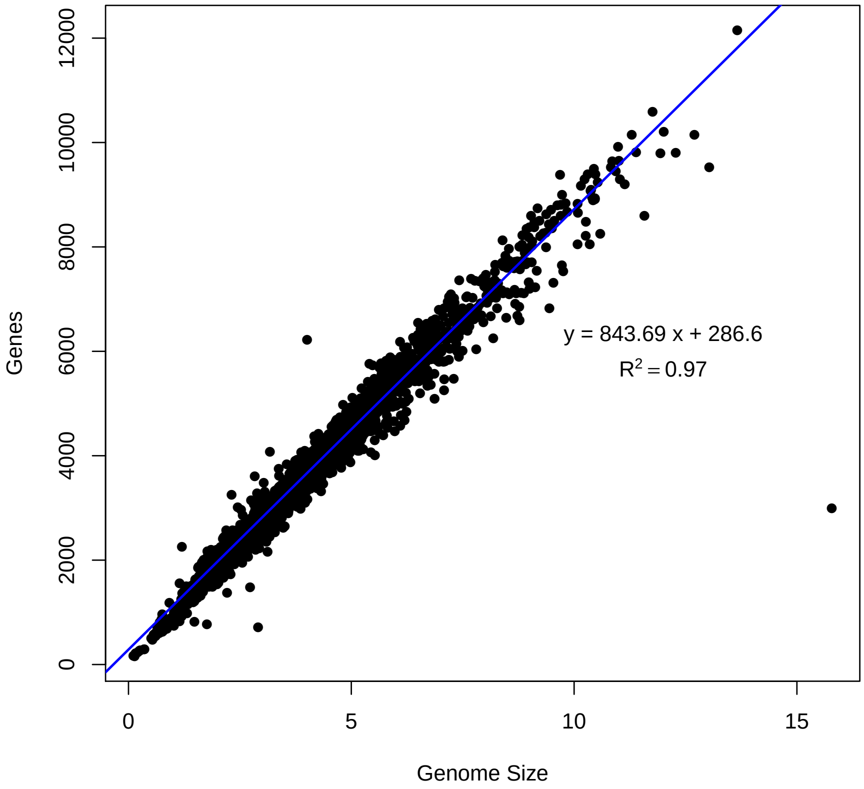
Contents

Chapter 1

R plot

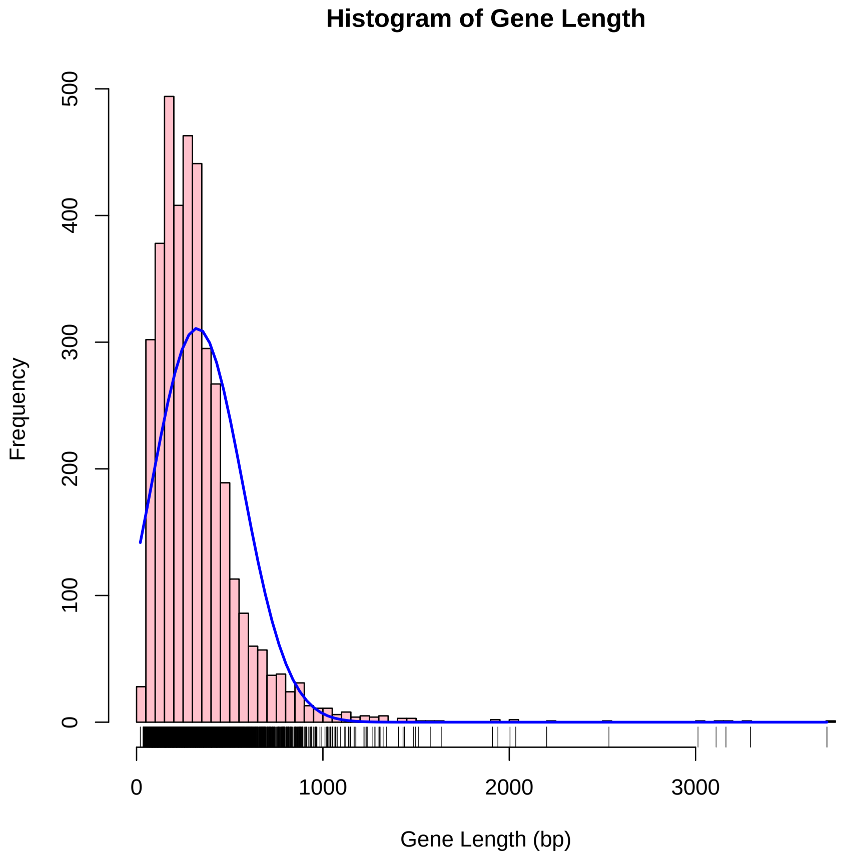
1.1 Scatter

```r
m=read.table("./figure/01.ScatterPlot/prok_representative.txt",
             sep="\t");
x=m[,2];
y=m[,4];
plot(x,y,pch=16,
        xlab="Genome Size",ylab="Genes");
fit <- lm(y~x);
abline( fit,col="blue",lwd=1.8 );
rr <- round( summary(fit)$adj.r.squared,2);
intercept <- round( summary(fit)$coefficients[1],2);
slope <- round( summary(fit)$coefficients[2],2);
eq <- bquote( atop( "y = " * .(slope) * " x + " * .(intercept),
                   R^2 == .(rr) ) );
#legend("topleft",legend=eq,bty="n");
text(12,6e3,eq);
```

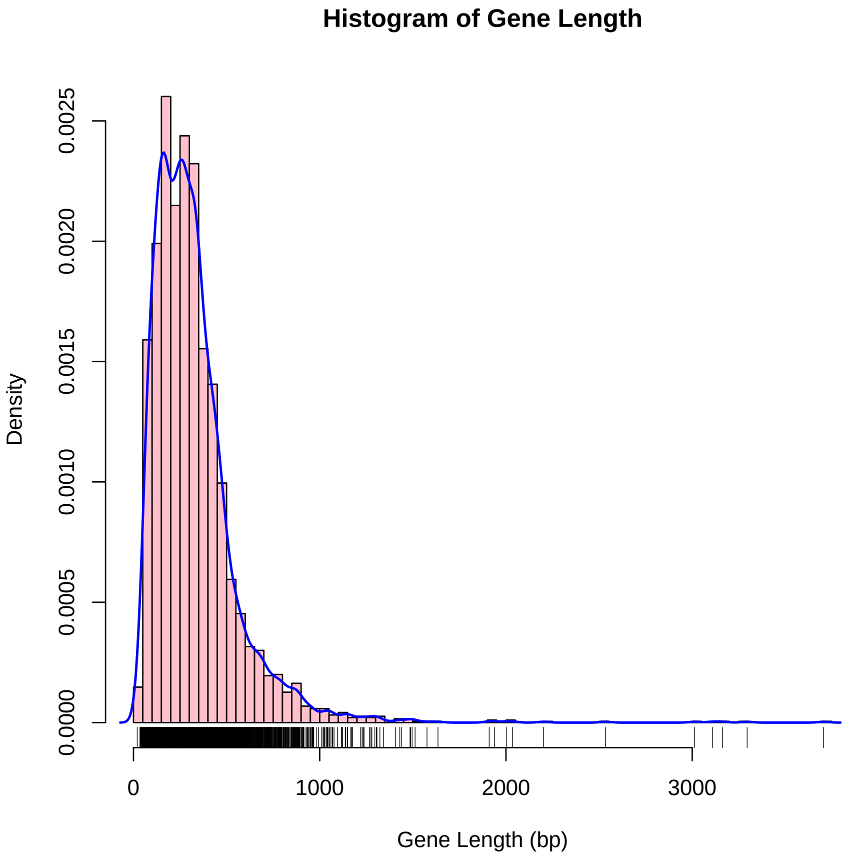


1.2 Histgram

```
m=read.table("./figure/02.Histogram/gene.length.txt",
             sep="\t",head=T);
x=m[,1];
h=hist(x,nclass=80,
        col="pink",
        xlab="Gene Length (bp)",
        main="Histogram of Gene Length");
rug(x);
xfit<-seq(min(x),max(x),length=100);
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x));
yfit <- yfit*diff(h$mids[1:2])*length(x);
lines(xfit, yfit, col="blue", lwd=2);
```

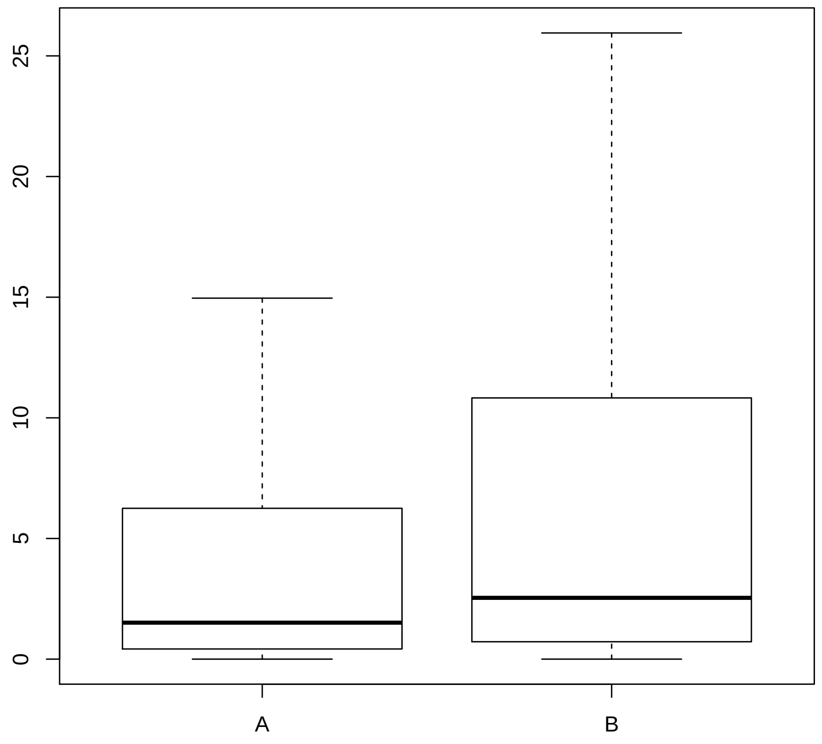


```
h=hist(x,nclass=80,
        probability=T,
        col="pink",
        xlab="Gene Length (bp)",
        main="Histogram of Gene Length");
lines(density(x),lwd=1.8,col="blue" );
rug(x);
```

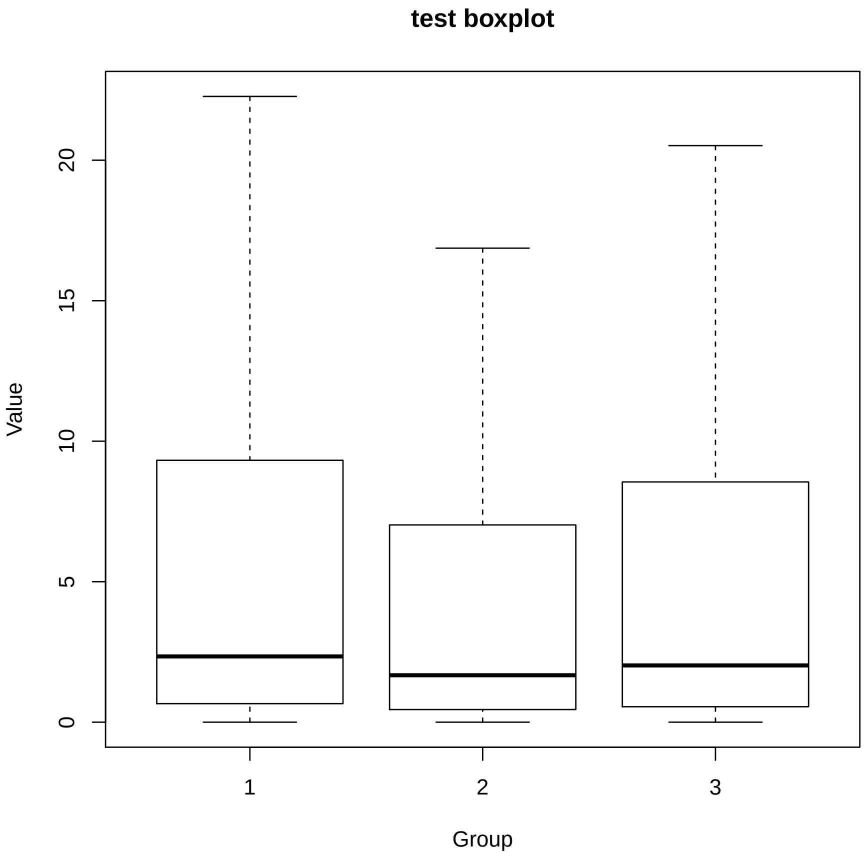


1.3 Boxplot

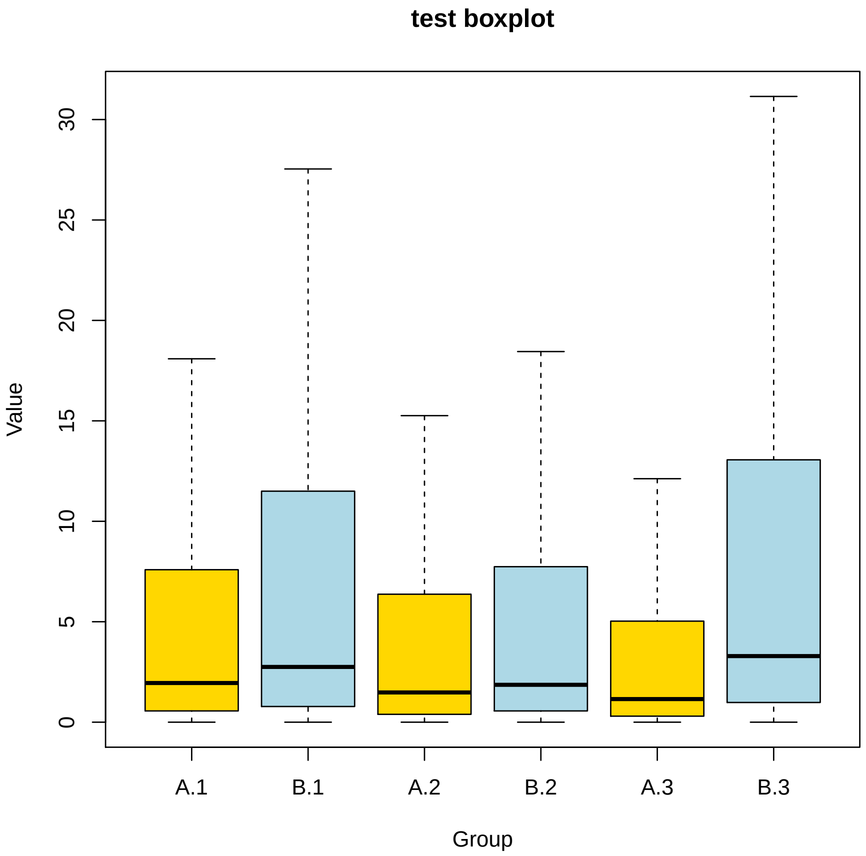
```
options(warn=-1)
m <- read.table("./figure/03.Boxplot/test.txt");
x <- m[,1];
y <- m[,2];
z <- m[,3];

boxplot(z~x,outline=F);
```



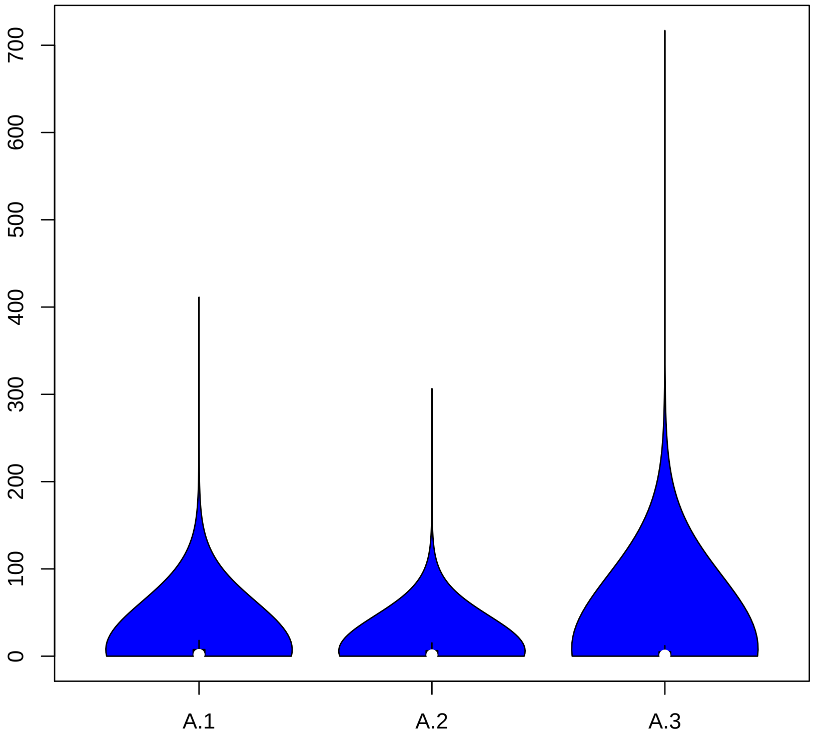
```
boxplot(z~y,outline=F,
        main="test boxplot",
        xlab="Group",ylab="Value");
```



```
boxplot(z~x*y,outline=F,
        main="test boxplot",
        xlab="Group",ylab="Value",
        col=(c("gold","lightblue")),
        );
```

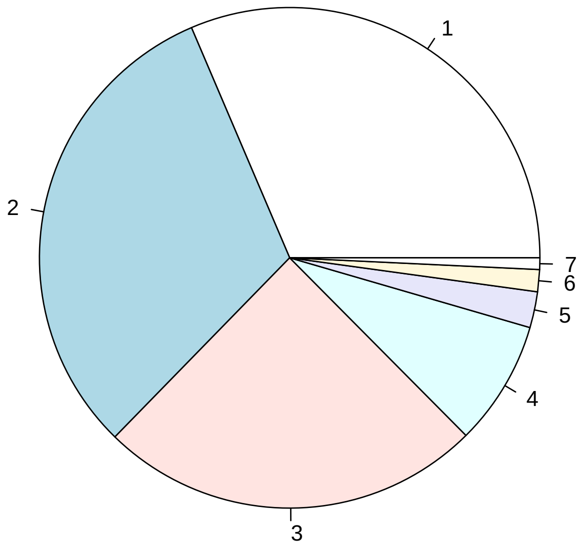


```
library(vioplot);

## Loading required package:  sm
## Package 'sm', version 2.2-5.4:  type help(sm) for summary information

m=read.table("./figure/03.Boxplot/test.matrix",head=T);
vioplot(m[,1],m[,2],m[,3],
        names=colnames(m)[1:3],col="blue");
```

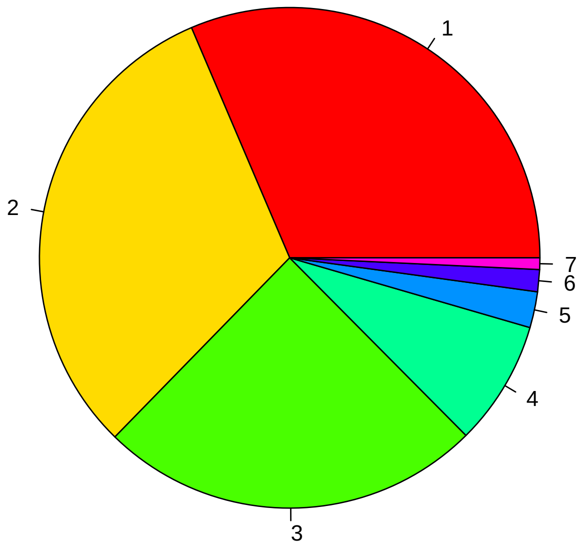


1.4 Pie

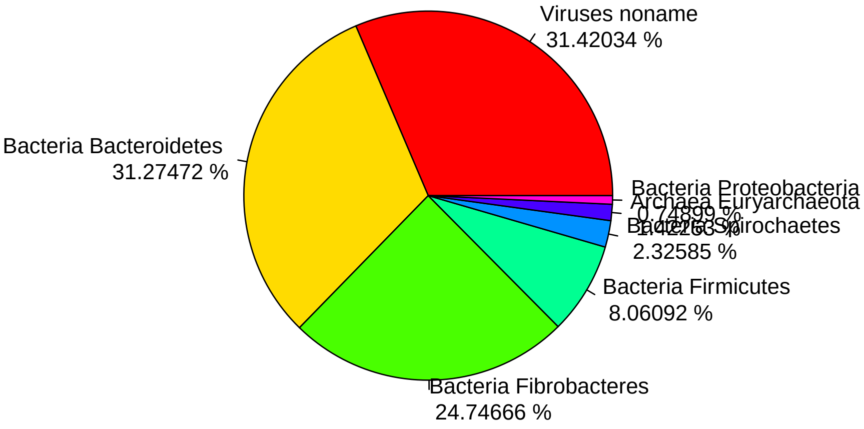
```
m=read.table("./figure/04.Pie/Species.txt");
pie(m[,3]);
```



```
pie(m[,3],
    col=rainbow(nrow(m) ),
    );
```

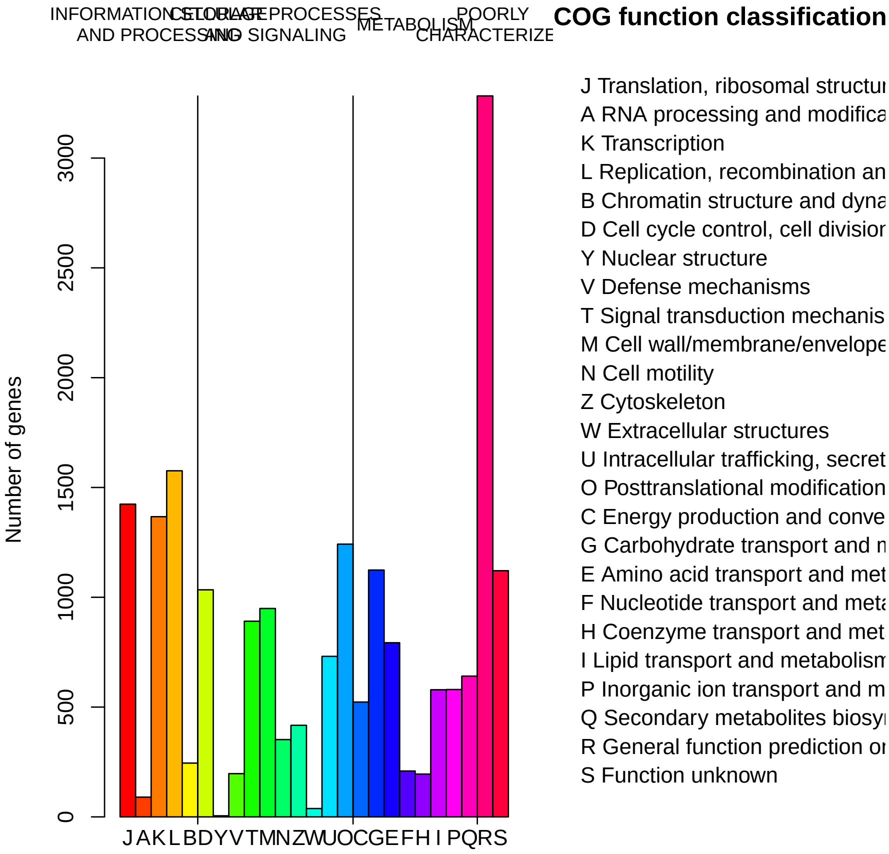


```
par(mar=rep(8,4) );
pie(m[,3],
    col=rainbow(nrow(m) ),
    labels=paste(m[,1],m[,2],"\n",m[,3],"%" ),
    );
```



1.5 Barplot

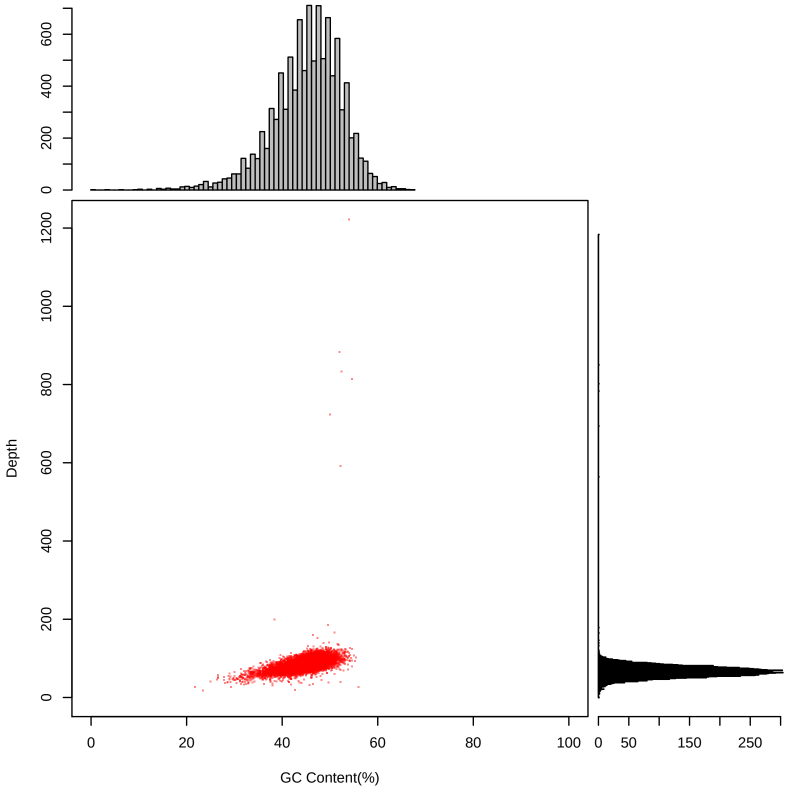
```
m=read.table("./figure/05.BarPlot/cog.class.annot.txt",
             head=T,sep="\t");
layout(matrix(c(1,2),nr=1),widths=c(20,13));
par( mar=c(3,4,4,1)+0.1 );

class <- c(
        "J","A","K","L","B",
        "D","Y","V","T","M","N","Z","W","U","O",
        "C","G","E","F","H","I","P","Q",
        "R","S");
t = factor( as.character(m[,1]),levels=class );
m <- m[order(t),]
x <- m[,3];

barplot(x,
        space=F,col=rainbow(25),
        ylab="Number of genes",
       );
mtext( m[,1],side=1,at=1:25-0.5 );

l <- c(0,5,15,23,25);
id<- c("INFORMATION STORAGE\nAND PROCESSING",
       "CELLULAR PROCESSES\nAND SIGNALING",
       "METABOLISM",
       "POORLY\nCHARACTERIZED"
      );
abline( v = l[c(-1,-5)] );
for( i in 2:length(l) ){
    text( (l[i-1]+l[i])/2,max(m[,3])*1.1,
            id[i-1],
            cex=0.8,xpd=T,
        );
}

par(mar=c(2,0,2,1)+0.1 );
plot(0,0,type="n",
        xlim=c(0,1),ylim=c(0,26),
        bty="n",axes=F,xlab="",ylab="",
    );

for( i in 1:length(class) ){
    text(0,26-i+0.5,paste(m[i,1],m[i,2]),
            pos=4,cex=1,pty=T,
        );
}
title("COG function classification");
```



1.6 Multi figures

```
m=read.table("./figure/06.MultiFigure/GC-depth.txt");
nf <- layout(matrix(c(0,2,0,0,1,3),2,3,byrow=T),
             c(0.5,3,1),c(1,3,0.5),TRUE);
par(mar=c(5,5,0.5,0.5));
x=m[,1];
y=m[,2];
plot(x,y,
        xlab='GC Content(%)',ylab='Depth',
        pch=46,col="#FF000077",
        xlim=c(0,100),ylim=c(0,max(y)),
    );

xbreaks <- 100;
ybreaks <- floor( max(y) - 0 );
xhist <- hist(x,breaks=xbreaks,plot=FALSE);
yhist <- hist(y,breaks=ybreaks,plot=FALSE);
```

```
par(mar=c(0,5,1,1));
barplot(xhist$counts,space=0,xlim=c(0,100) );

par(mar=c(5,0,1,1));
barplot(yhist$counts,space=0,horiz=TRUE,ylim=c(0,max(y)));
```



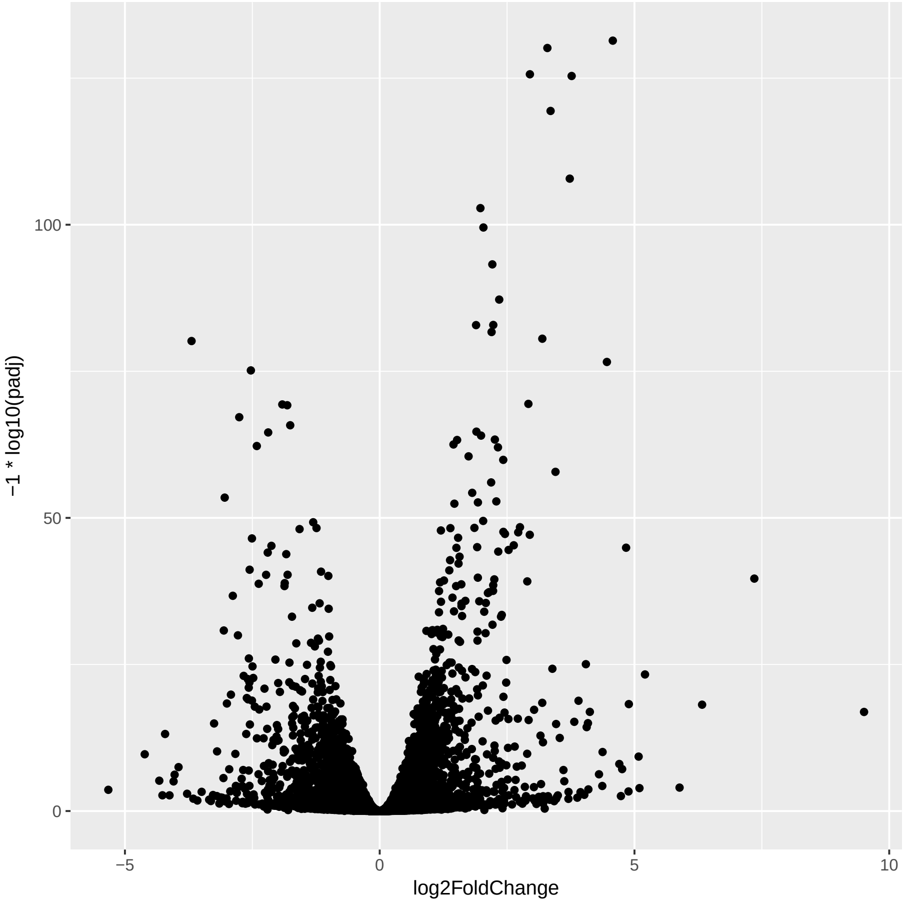
1.7 Volcano

```
library("ggplot2")
library("DESeq2")

## Loading required package:  S4Vectors
## Loading required package:  methods
## Loading required package:  stats4
## Loading required package:  BiocGenerics
## Loading required package:  parallel
##
## Attaching package:  'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##     IQR, mad, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, as.vector, cbind,
##     colnames, do.call, duplicated, eval, evalq, Filter, Find, get,
##     grep, grepl, intersect, is.unsorted, lapply, lengths, Map,
##     mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, Position, rank, rbind, Reduce, rownames, sapply,
##     setdiff, sort, table, tapply, union, unique, unlist, unsplit
## Loading required package:  IRanges
## Loading required package:  GenomicRanges
## Loading required package:  GenomeInfoDb
## Loading required package:  SummarizedExperiment
## Loading required package:  Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'.  To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package:  Rcpp
## Loading required package:  RcppArmadillo
## Note:  the specification for S3 class "AsIs" in package 'DBI' seems
equivalent to one from package 'BiocGenerics': not turning on duplicate
class definitions for this class.
```

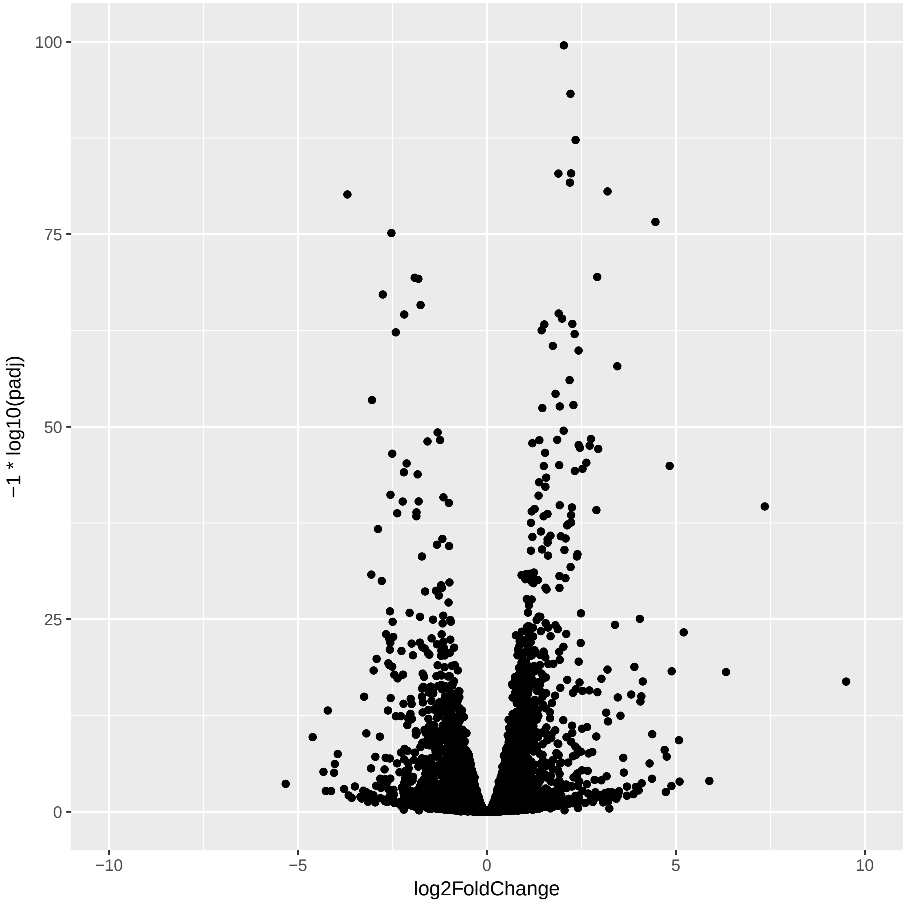
```
#
dta <- read.csv("data/des.csv",header = T)
head(dta)
```

```
##                 X     baseMean log2FoldChange      lfcSE       stat
## 1 ENSG00000000003  708.6021697    -0.38125397 0.10065597 -3.7876937
## 2 ENSG00000000419  520.2979006     0.20681259 0.11222180  1.8428915
## 3 ENSG00000000457  237.1630368     0.03792034 0.14345322  0.2643394
## 4 ENSG00000000460   57.9326331    -0.08816367 0.28716771 -0.3070111
## 5 ENSG00000000938    0.3180984    -1.37822703 3.49987280 -0.3937935
## 6 ENSG00000000971 5817.3528677     0.42640216 0.08831006  4.8284666
##         pvalue         padj
## 1 1.520521e-04 0.0012815112
## 2 6.534485e-02 0.1962308610
## 3 7.915184e-01 0.9112208706
## 4 7.588349e-01 0.8946714454
## 5 6.937335e-01           NA
## 6 1.375884e-06 0.0000181641
```

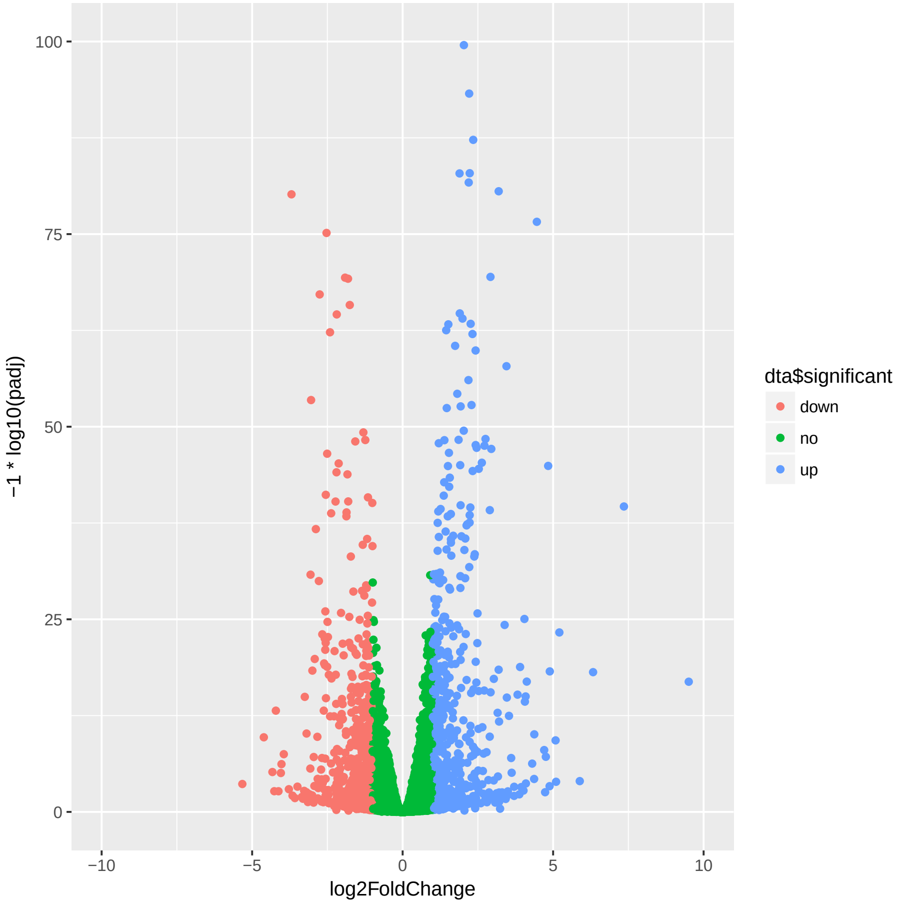
```
dta <- na.omit(dta)
volcano <- ggplot(dta,aes(log2FoldChange,-1*log10(padj)))
volcano+geom_point()
```



```
volcano+geom_point()+xlim(-10,10) + ylim(0,100)
```



```
dta$significant[dta$log2FoldChange<=-1] ="down"
dta$significant[dta$log2FoldChange>=1] ="up"
dta$significant[dta$log2FoldChange>-1 & dta$log2FoldChange <1] ="no"
volcano +geom_point(aes(color=dta$significant)) +
    xlim(-10,10) + ylim(0,100)
```

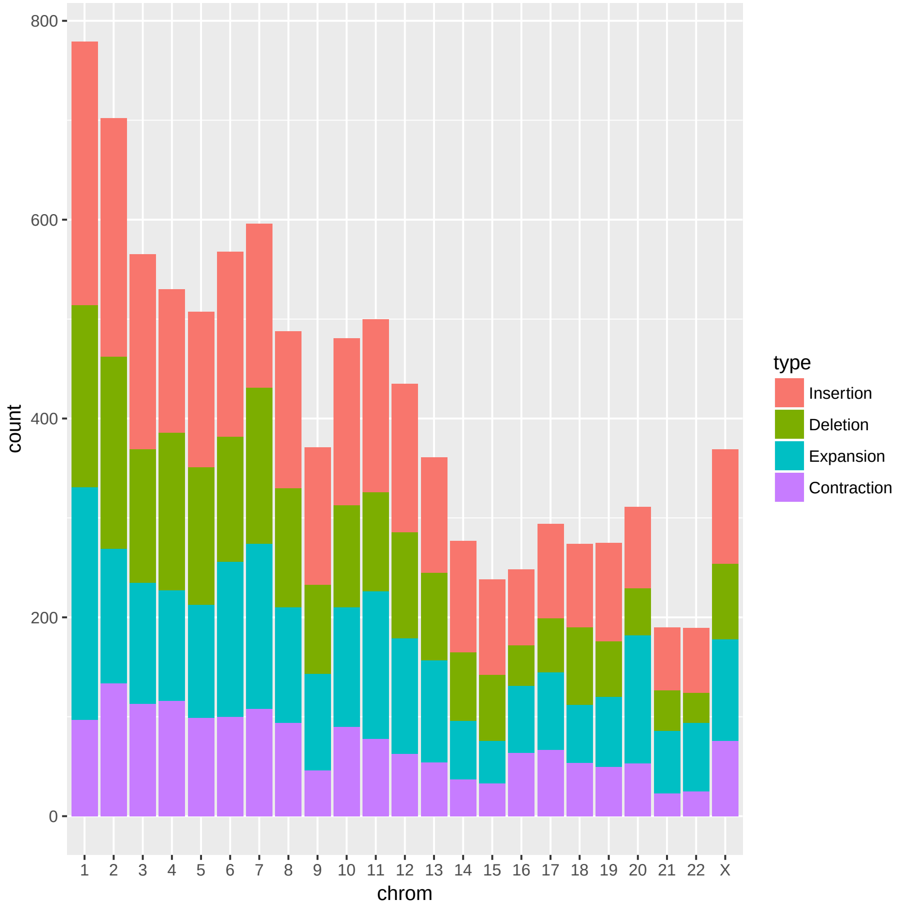

100
75
50
25
0
−1 * log10(padj)
−10
−5
0
5
10
log2FoldChange
dta$significant
down
no
up

Chapter 2

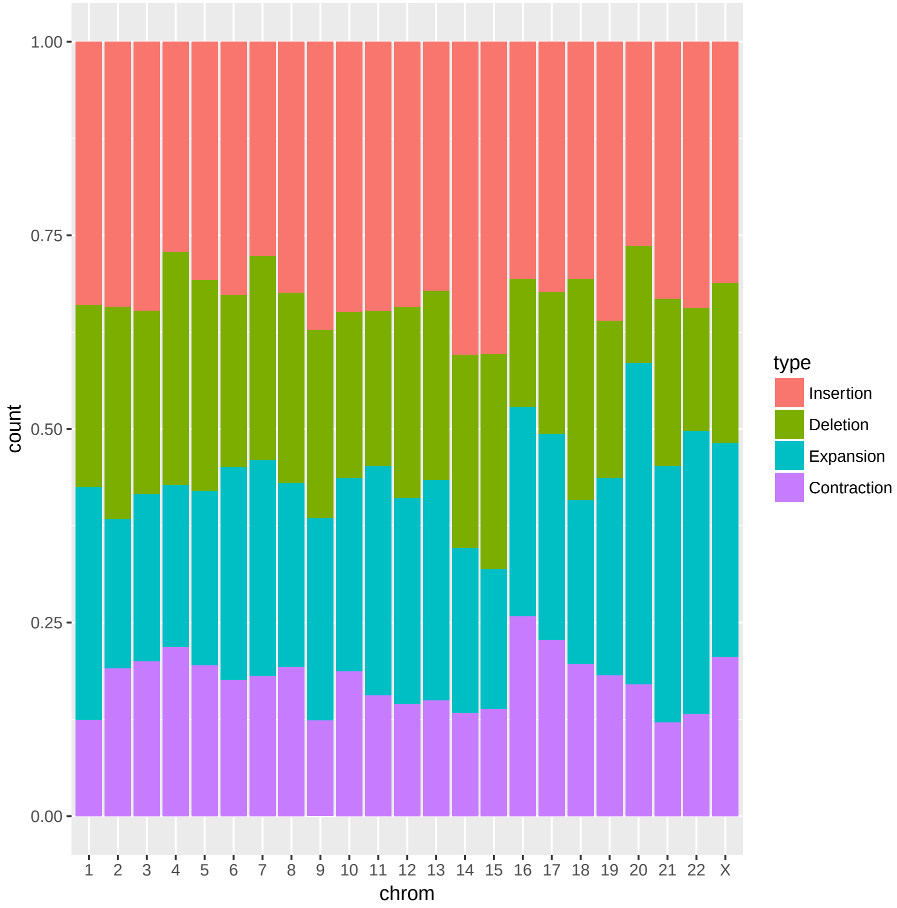
ggplot

2.1 SV statistics & plot

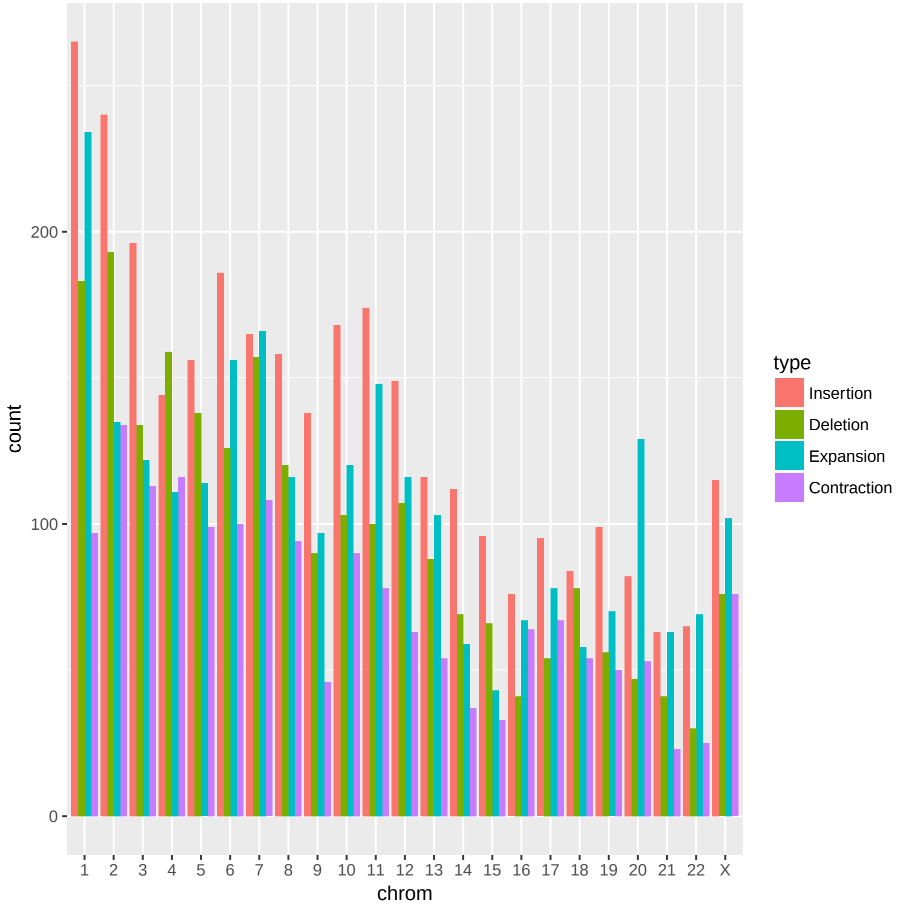
```r
library(ggplot2)

my_sv <- read.csv("data/variants_bed.csv",
                  stringsAsFactors=TRUE,header=TRUE)
#head(my_sv)
#colnames(my_sv)
#summary(my_sv$chrom)

#
my_sv$chrom <- factor(my_sv$chrom,
                       levels=c(seq(1,22),"X","Y"))
#  Type
my_sv$type <- factor(my_sv$type,
        levels=c("Insertion","Deletion","Expansion","Contraction"))


#
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar()
```



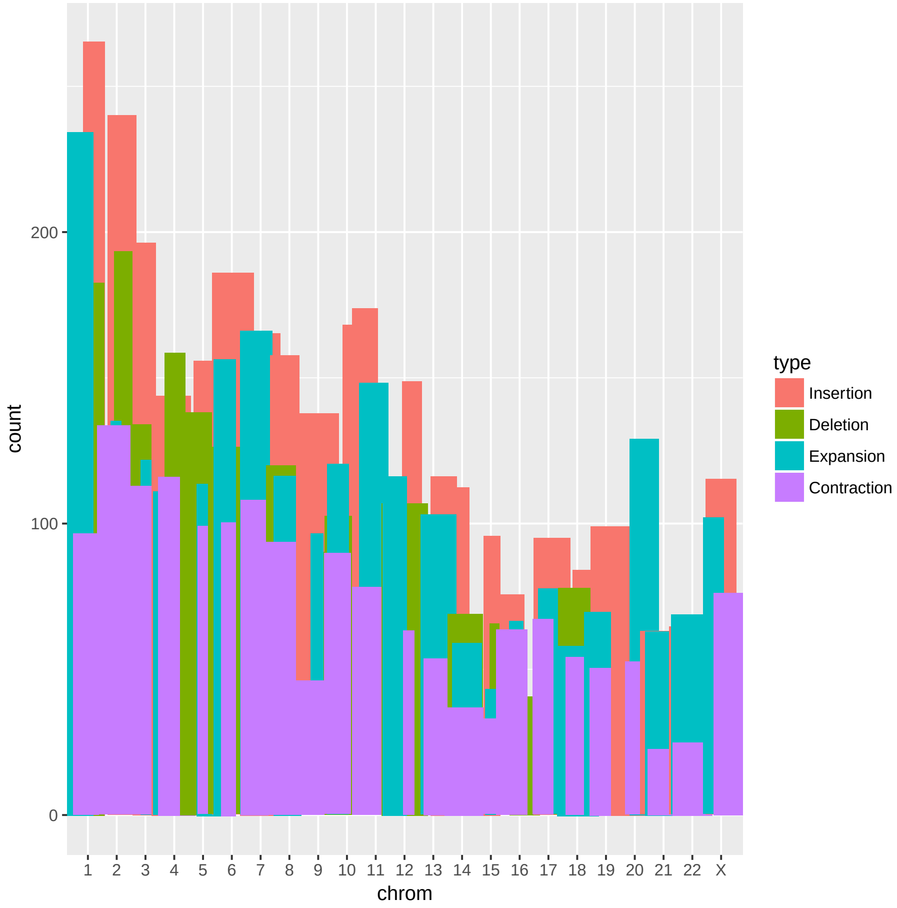
```
#
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar(position = "fill")
```



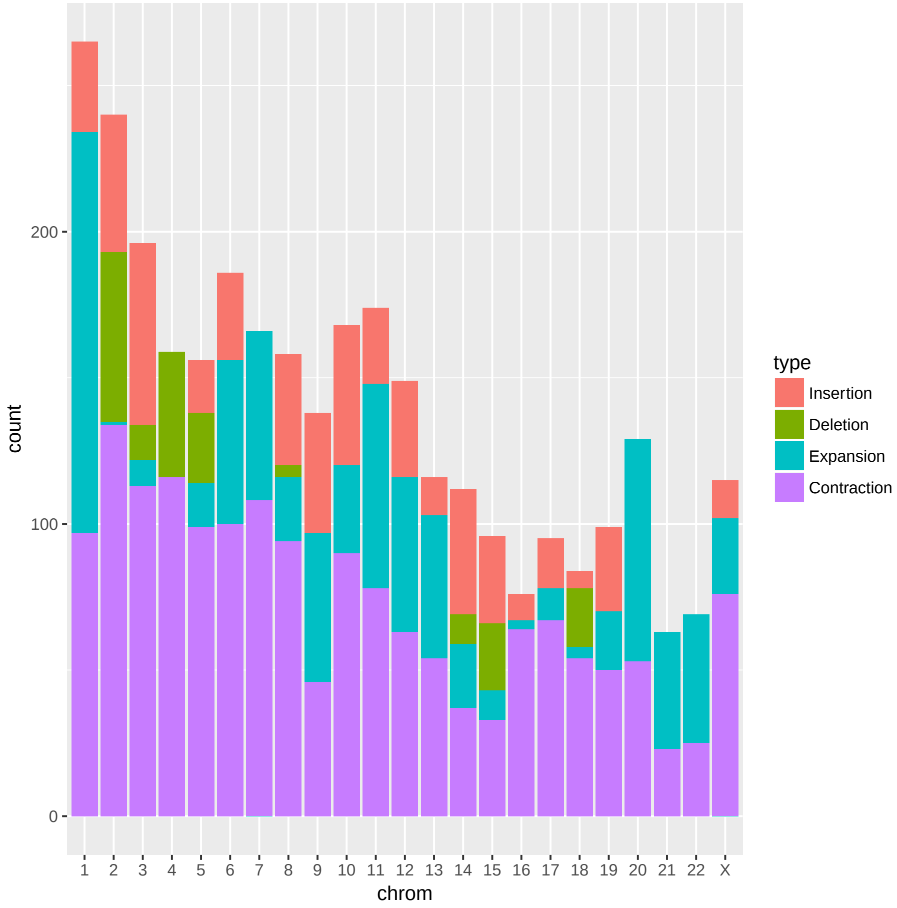
```
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar(position = "dodge")
```



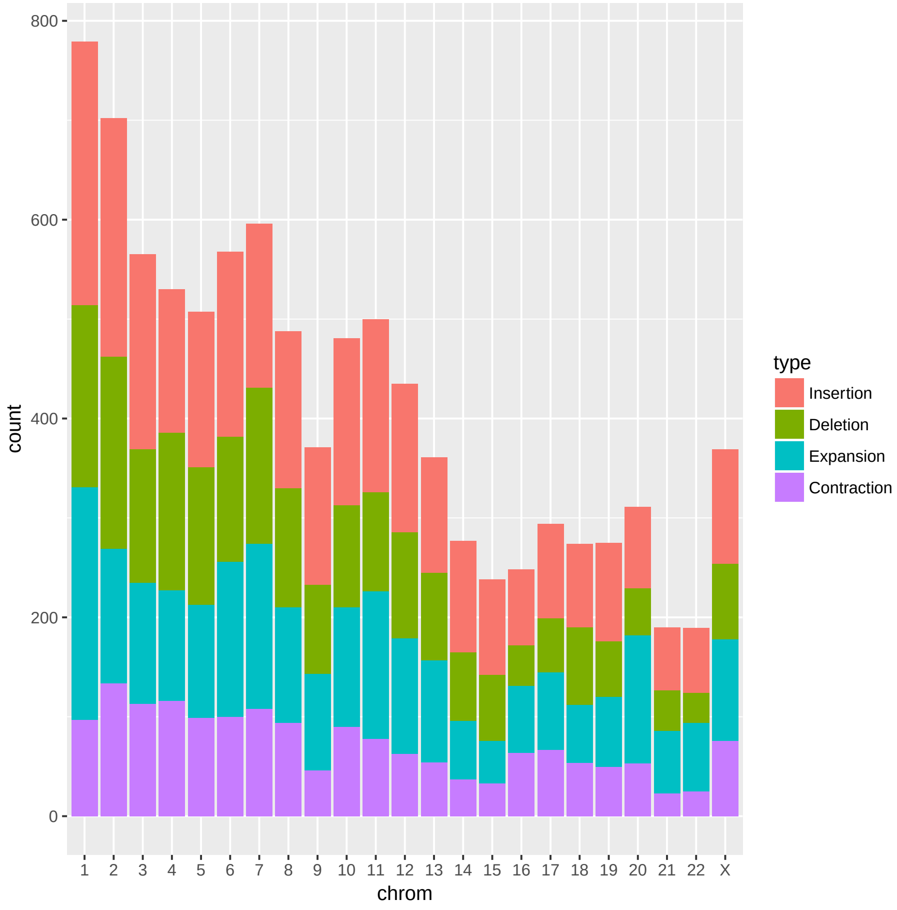
```
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar(position = "jitter")
```



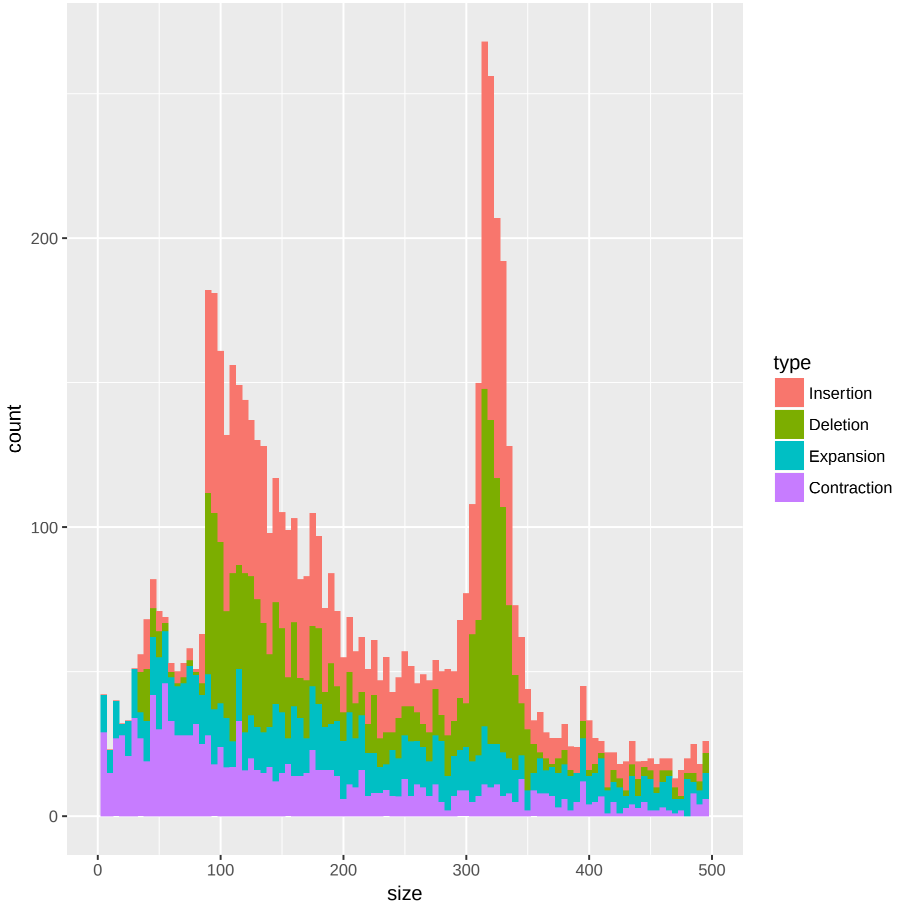
```
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar(position = "identity")
```



```
ggplot(my_sv, aes(x=chrom,fill=type)) + geom_bar(position = "stack")
```

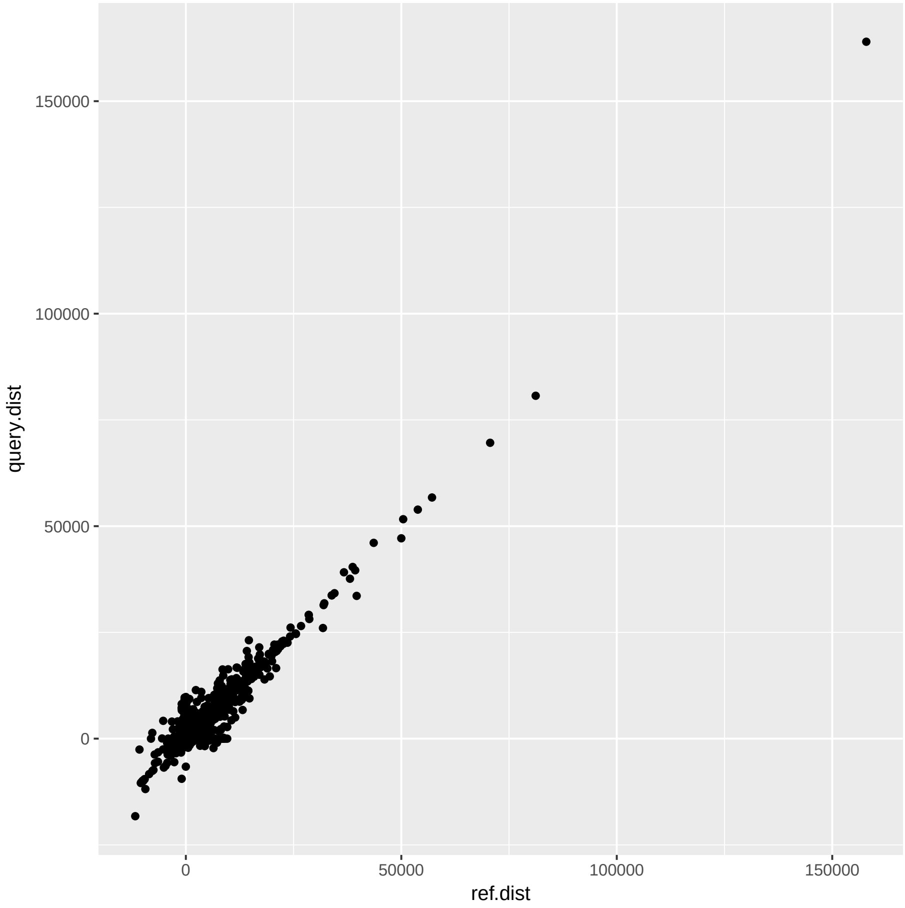


```
#
ggplot(my_sv, aes(x=size,fill=type)) +
    geom_bar(binwidth=5) + xlim(0,500)
```

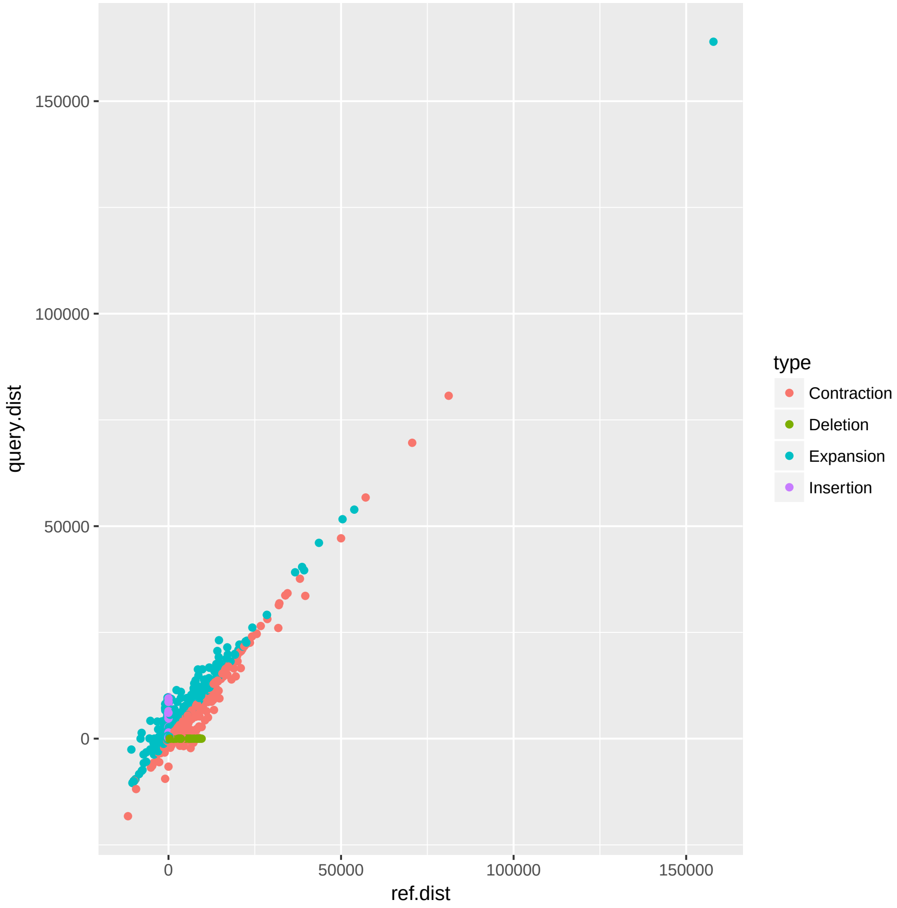


2.2 Scatter

```
my_sv<- read.csv("data/variants_bed.csv",
                 stringsAsFactors=TRUE,header=TRUE)
#
ggplot(my_sv, aes(x=ref.dist,y=query.dist)) +
    geom_point()
```

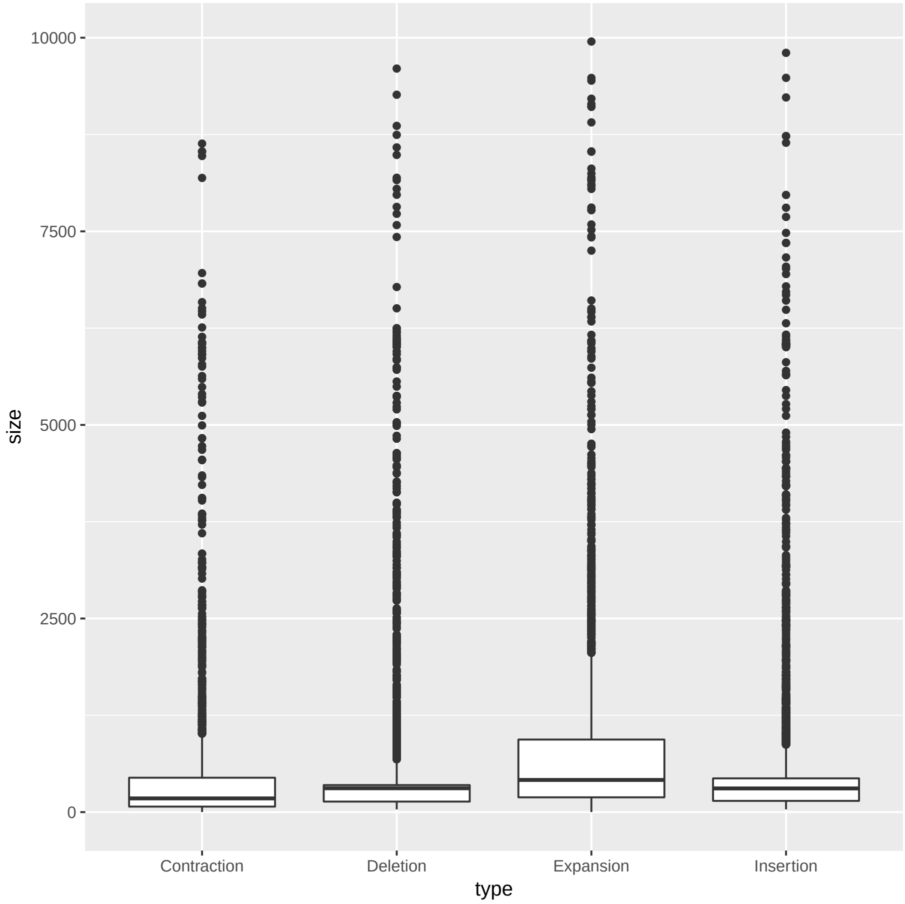


```
ggplot(my_sv, aes(x=ref.dist,y=query.dist,color=type)) +
    geom_point()
```

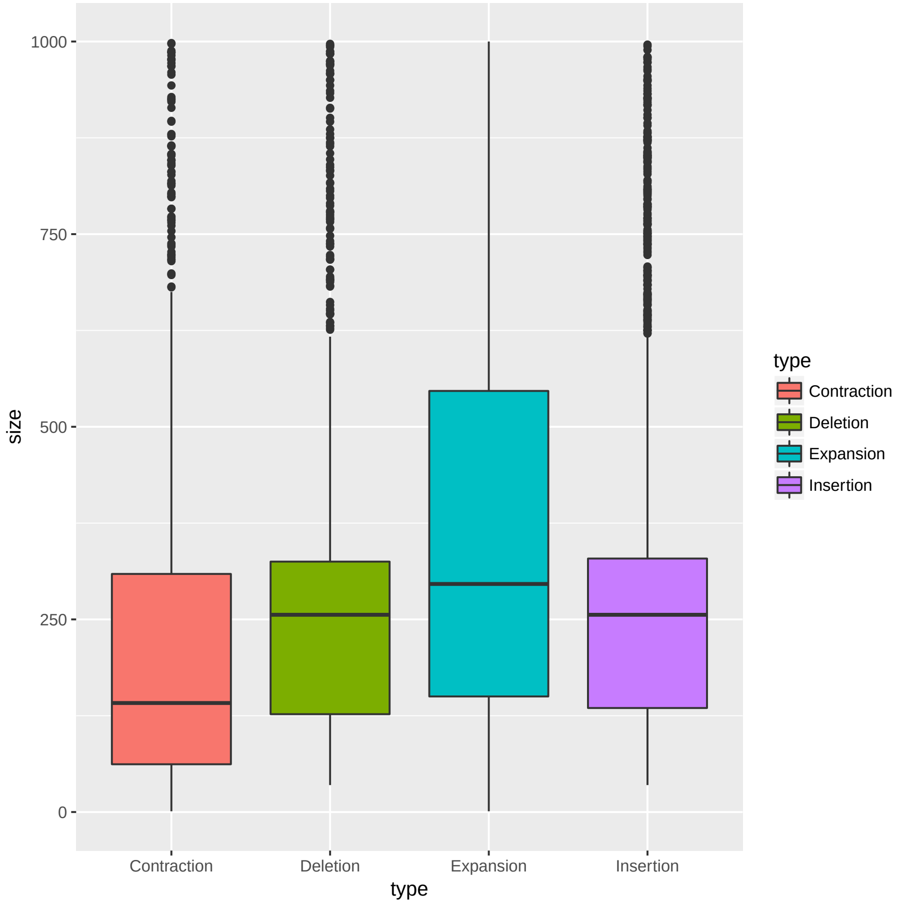


2.3 Boxplot

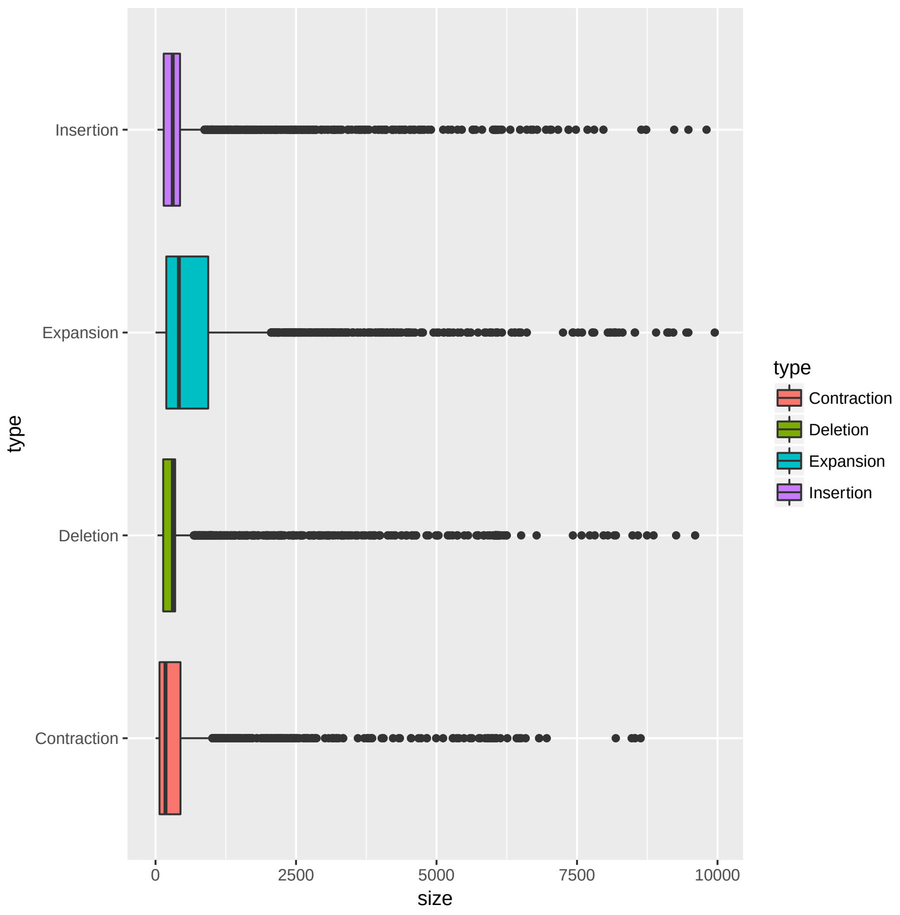
```
my_sv<- read.csv("data/variants_bed.csv",
                   stringsAsFactors=TRUE,header=TRUE)
ggplot(my_sv, aes(x=type,y=size)) + geom_boxplot()
```



```
ggplot(my_sv, aes(x=type,y=size,fill=type)) +
    geom_boxplot()+ylim(0,1000)
```

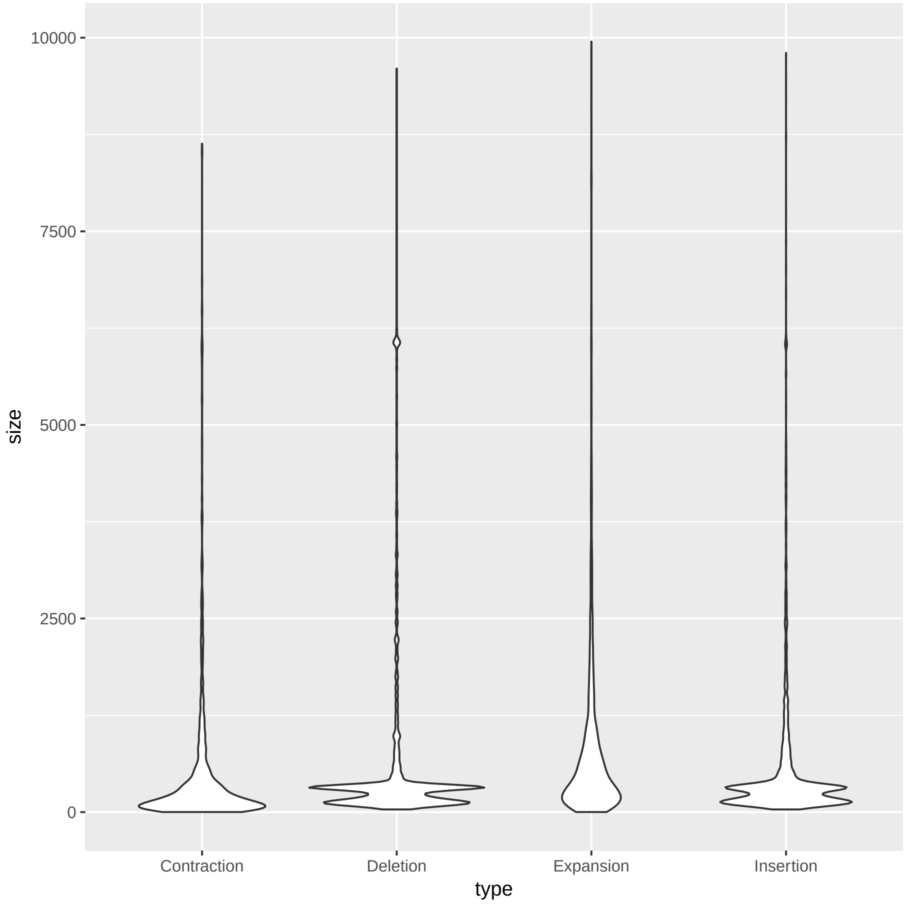


```
ggplot(my_sv, aes(x=type,y=size,fill=type)) +
    geom_boxplot() + coord_flip()
```

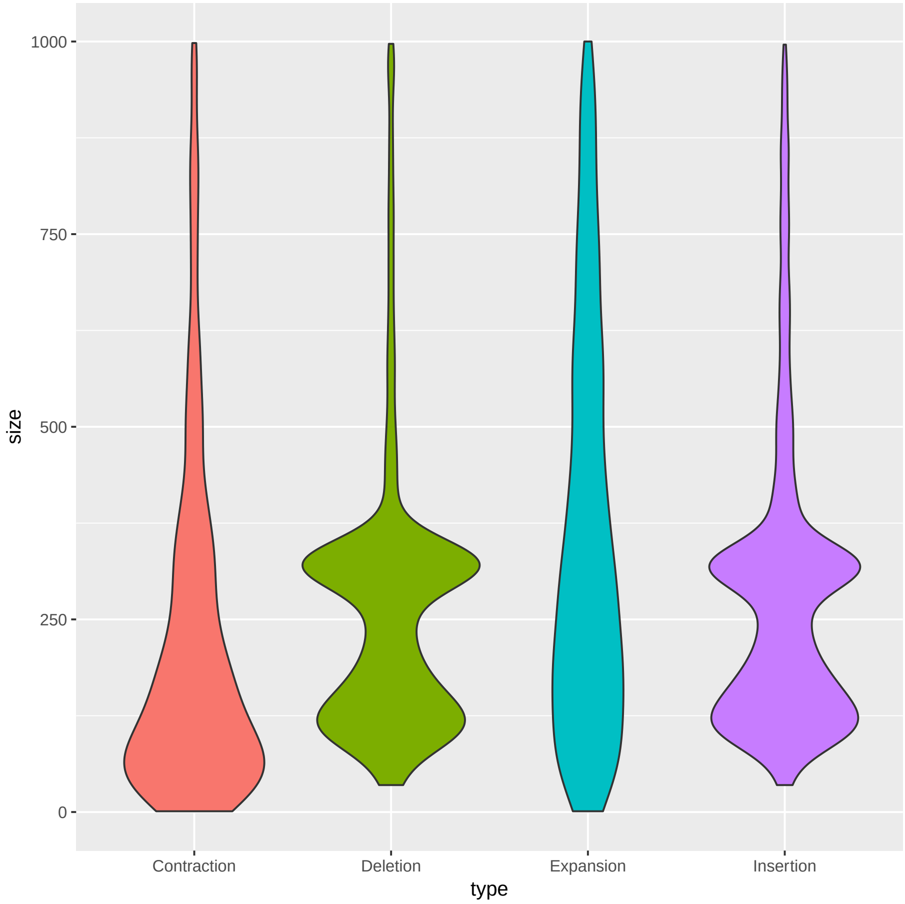


2.4 Violine

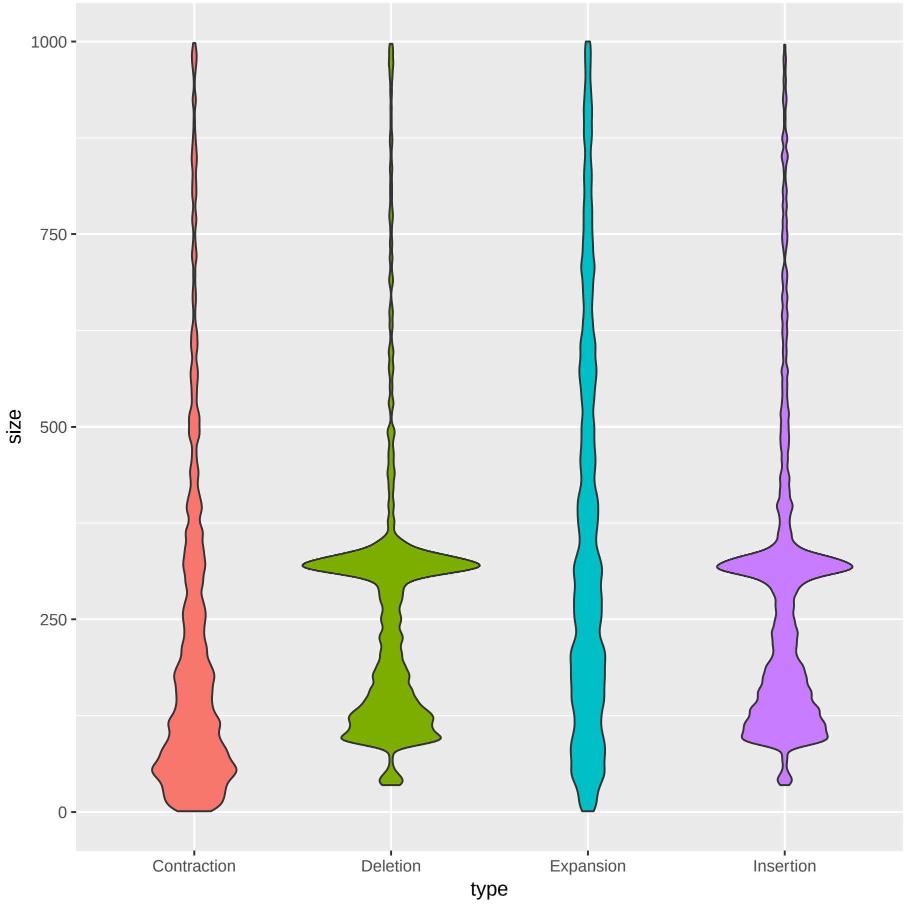
```
my_sv<- read.csv("data/variants_bed.csv",
                  stringsAsFactors=TRUE,header=TRUE)
ggplot(my_sv, aes(x=type,y=size)) + geom_violin()
```



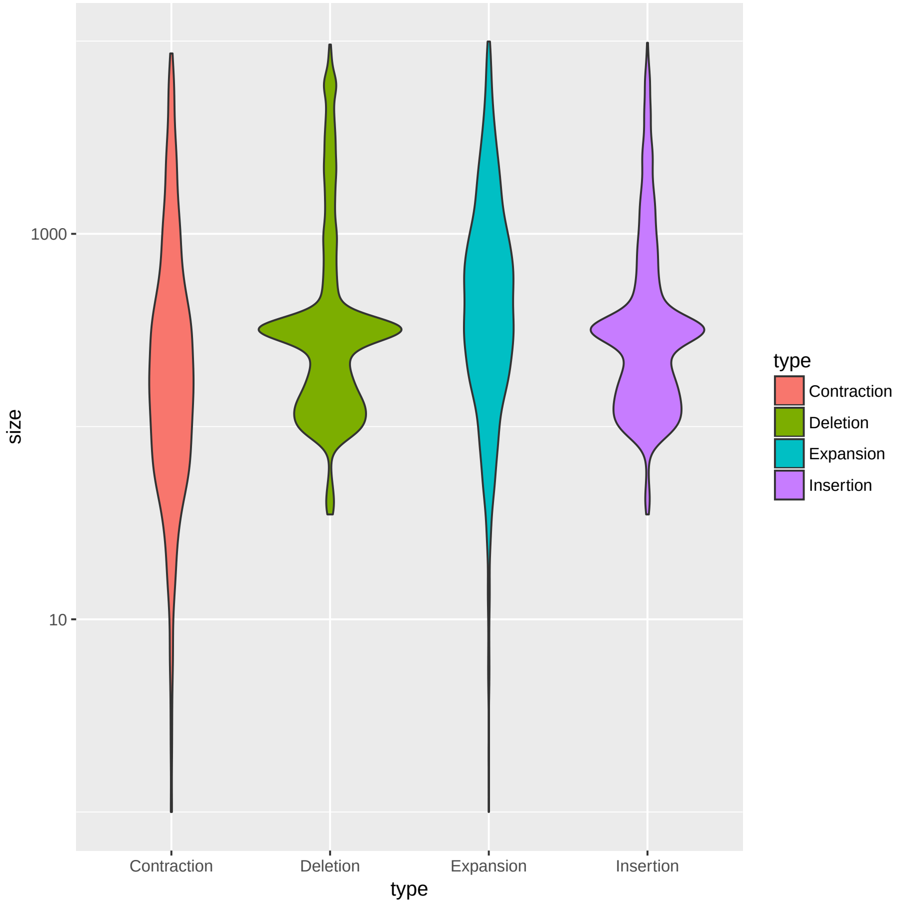
```
ggplot(my_sv, aes(x=type,y=size,fill=type)) +
    geom_violin() + ylim(0,1000) + guides(fill=FALSE)
```



```
ggplot(my_sv, aes(x=type,y=size,fill=type)) +
    geom_violin(adjust=0.2) + ylim(0,1000) + guides(fill=FALSE)
```

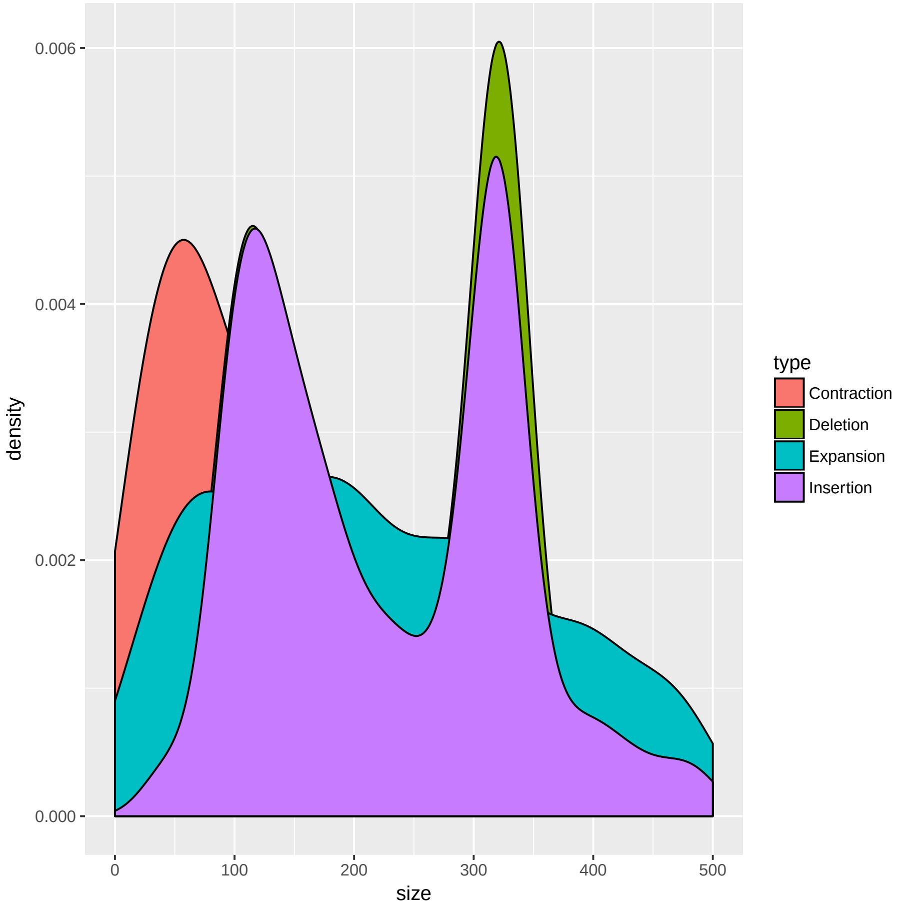


```
ggplot(my_sv, aes(x=type,y=size,fill=type)) +
    geom_violin() +  scale_y_log10()
```

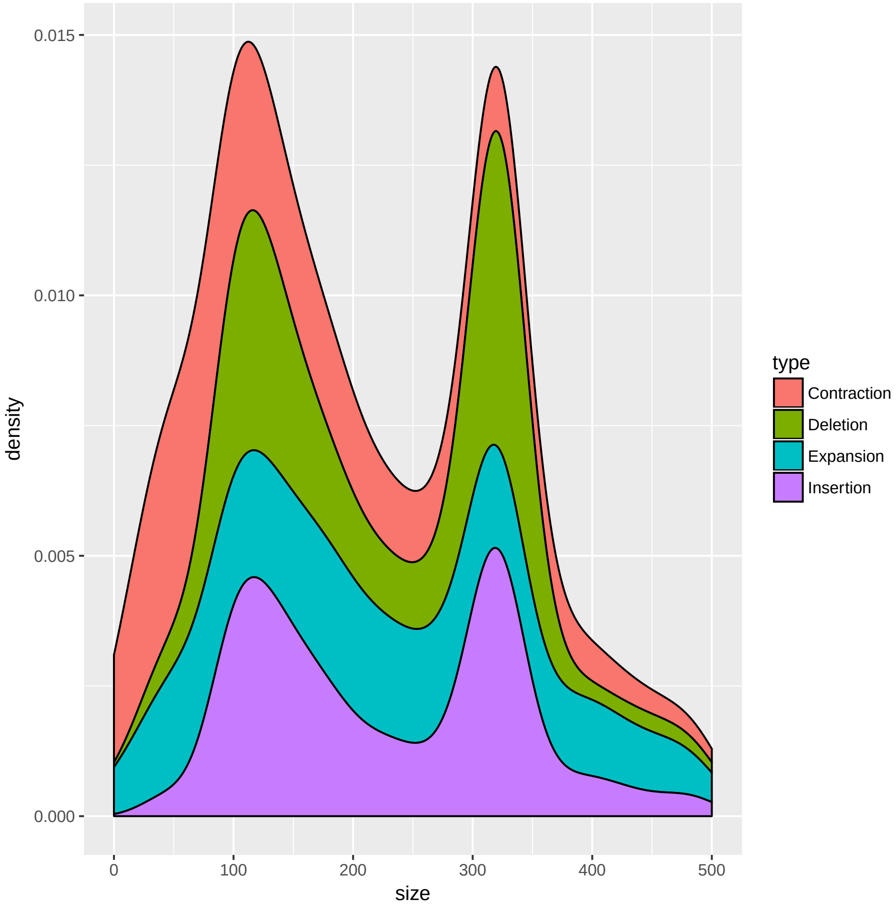


2.5 Density curve

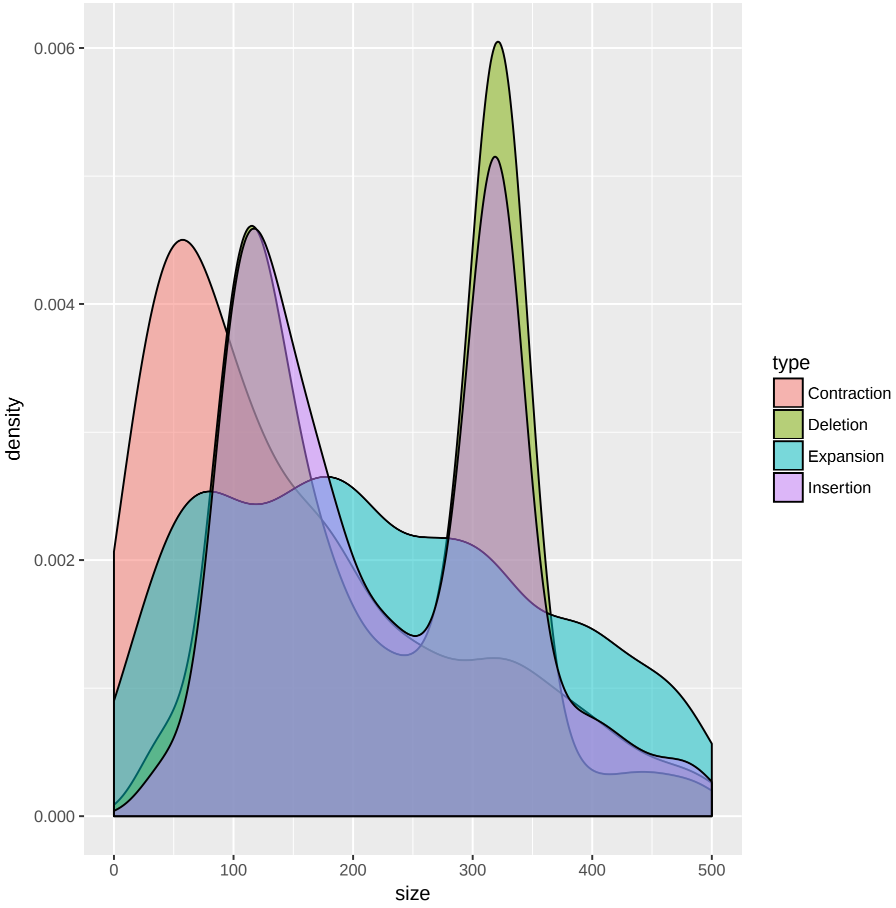
```
my_sv<- read.csv("data/variants_bed.csv",
                 stringsAsFactors=TRUE,header=TRUE)
ggplot(my_sv, aes(x=size,fill=type)) +
    geom_density() + xlim(0,500)
```



```
ggplot(my_sv, aes(x=size,fill=type)) +
    geom_density(position="stack") + xlim(0,500)
```

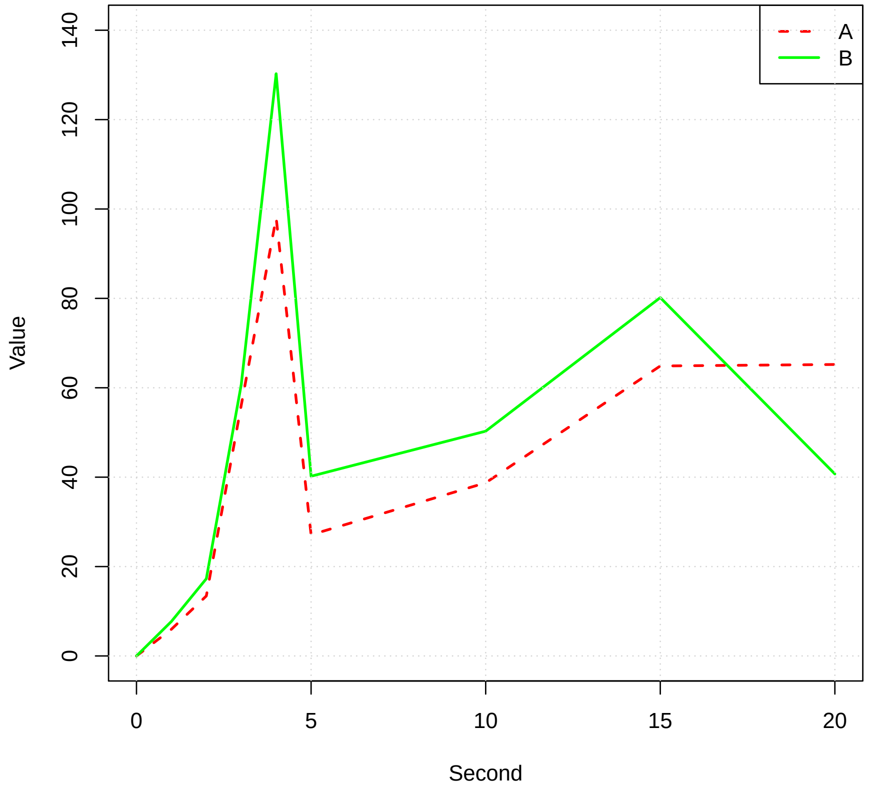


```
ggplot(my_sv, aes(x=size,fill=type)) +
    geom_density(alpha=0.5) + xlim(0,500)
```

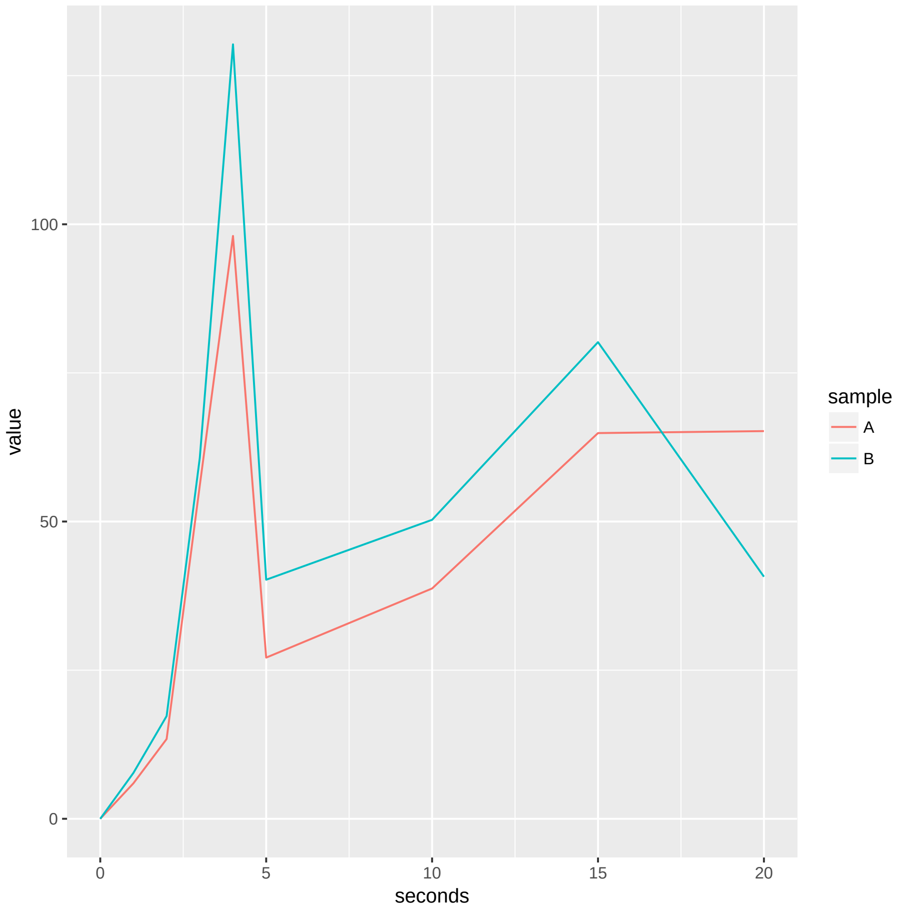


2.6 Time change plot

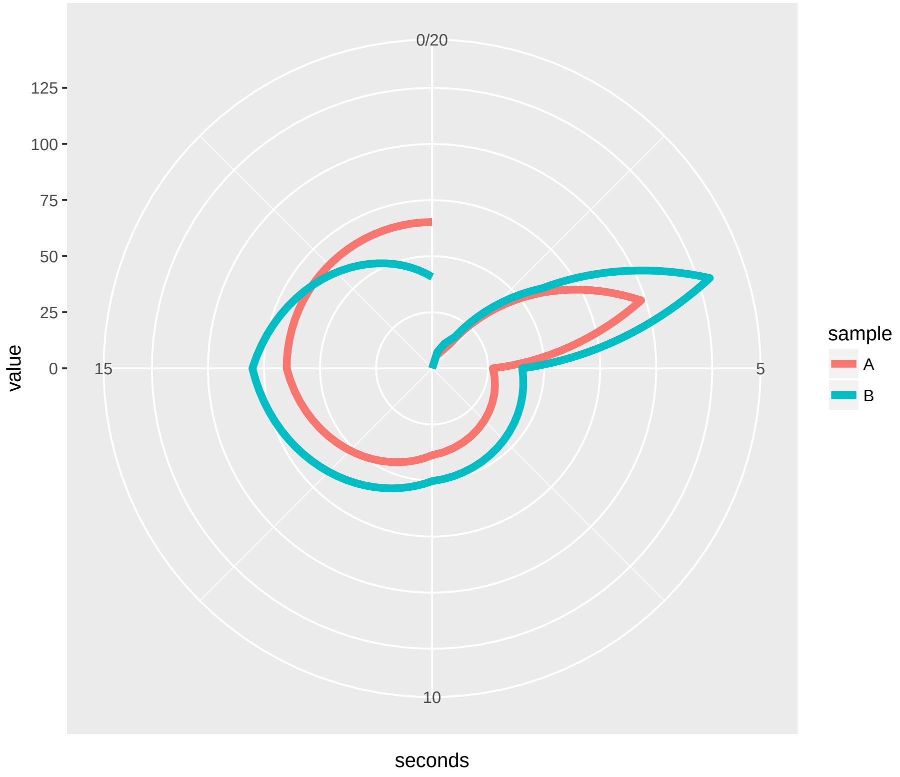
```
library(ggplot2)
time_course <- read.csv("data/time_course_data.txt",
                        stringsAsFactors=TRUE,header=TRUE)
time_course

##    seconds  value sample
## 1        0   0.00      A
## 2        1   5.97      A
## 3        2  13.42      A
## 4        3  56.08      A
## 5        4  98.04      A
## 6        5  27.11      A
## 7       10  38.74      A
## 8       15  64.88      A
## 9       20  65.21      A
## 10       0   0.00      B
## 11        1   7.72       B
## 12        2  17.28       B
## 13        3  60.72       B
## 14        4 130.27       B
## 15        5  40.21       B
## 16       10  50.29       B
## 17       15  80.17       B
## 18       20  40.72       B

library(reshape2)
times <- dcast(time_course,seconds ~ sample)

#
plot(times$seconds,times$A,
     type = "l",col="red",lwd=2,ylim=c(0,140),lty=2,
     xlab = "Second",ylab="Value")
lines(times$seconds,times$B,col="green",lwd=2)
legend("topright",legend = c("A","B"),
       lty = c(2,1),lwd=2,col = c("red","green"))
grid()
```



```
# ggplot2
#library(ggplot2)
ggplot(time_course, aes(x=seconds,y=value,colour=sample)) +
    geom_line()
```



```
#
ggplot(time_course, aes(x=seconds,y=value,colour=sample)) +
    geom_line(size=2) + coord_polar()
```



2.7 Variant type statistics and plot

```
#
my_var <- read.csv("data/Encode.Hmm.csv",
                   stringsAsFactors = TRUE,header=TRUE)
head(my_var)

##   chrom start  stop              type V5 V6    V7    V8          V9
## 1  chr1 10000 10600 15_Repetitive/CNV  0  . 10000 10600 245,245,245
## 2  chr1 10600 11137 13_Heterochrom/lo  0  . 10600 11137 245,245,245
## 3  chr1 11137 11737       8_Insulator  0  . 11137 11737  10,190,254
## 4  chr1 11737 11937       11_Weak_Txn  0  . 11737 11937 153,255,102
## 5  chr1 11937 12137    7_Weak_Enhancer  0  . 11937 12137   255,252,4
## 6  chr1 12137 14537       11_Weak_Txn  0  . 12137 14537 153,255,102

dim(my_var)

## [1] 571339      9
```

```
#
summary(my_var)
```

```
##      chrom            start                 stop
##  chr1   : 54467   Min.   :        0   Min.   :    10200
##  chr2   : 46499   1st Qu.: 33424623   1st Qu.: 33427336
##  chr3   : 37617   Median : 66145965   Median : 66150096
##  chr6   : 34846   Mean   : 77800396   Mean   : 77805350
##  chr5   : 30071   3rd Qu.:114147254   3rd Qu.:114148704
##  chr7   : 29420   Max.   :249229377   Max.   :249232977
##  (Other):338419
##                 type              V5    V6              V7
##  7_Weak_Enhancer  :109468   Min.   :0   .:571339   Min.   :        0
##  11_Weak_Txn      : 82312   1st Qu.:0              1st Qu.: 33424623
##  13_Heterochrom/lo: 75112   Median :0              Median : 66145965
##  6_Weak_Enhancer  : 69111   Mean   :0              Mean   : 77800396
##  5_Strong_Enhancer: 38604   3rd Qu.:0              3rd Qu.:114147254
##  2_Weak_Promoter  : 35065   Max.   :0              Max.   :249229377
##  (Other)          :161667
##        V8                     V9
##  Min.   :    10200   255,252,4  :178579
##  1st Qu.: 33427336   245,245,245: 89268
##  Median : 66150096   153,255,102: 82312
##  Mean   : 77805350   250,202,0  : 64090
##  3rd Qu.:114148704   0,176,80   : 42736
##  Max.   :249232977   255,105,105: 35065
##                      (Other)    : 79289
```

```
table(my_var$chrom)
```

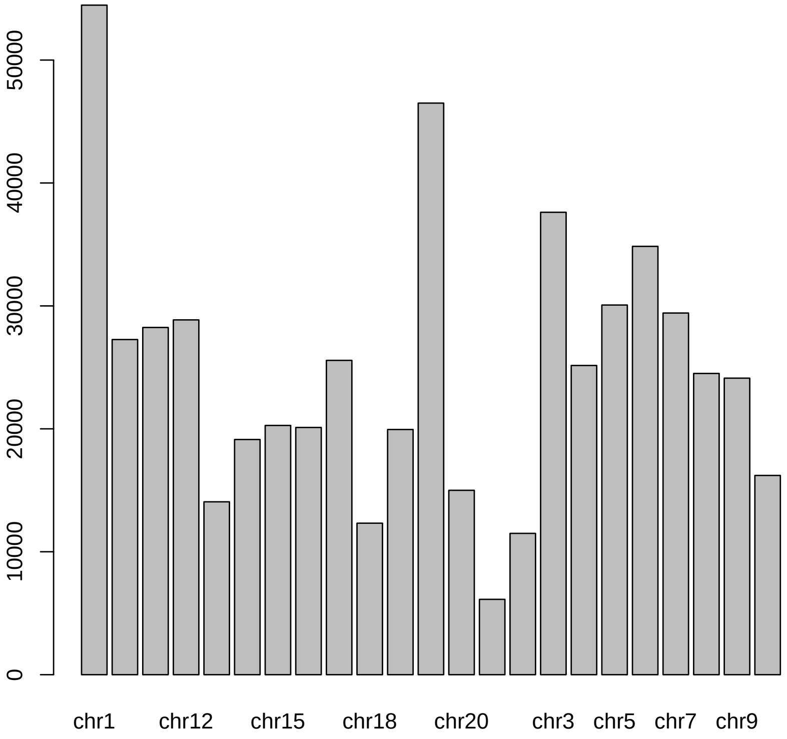
```
##
##  chr1 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19  chr2
## 54467 27263 28246 28863 14064 19133 20277 20113 25570 12324 19947 46499
## chr20 chr21 chr22  chr3  chr4  chr5  chr6  chr7  chr8  chr9  chrX
## 15000  6128 11497 37617 25155 30071 34846 29420 24506 24123 16210
```

```
levels(my_var$type)
```

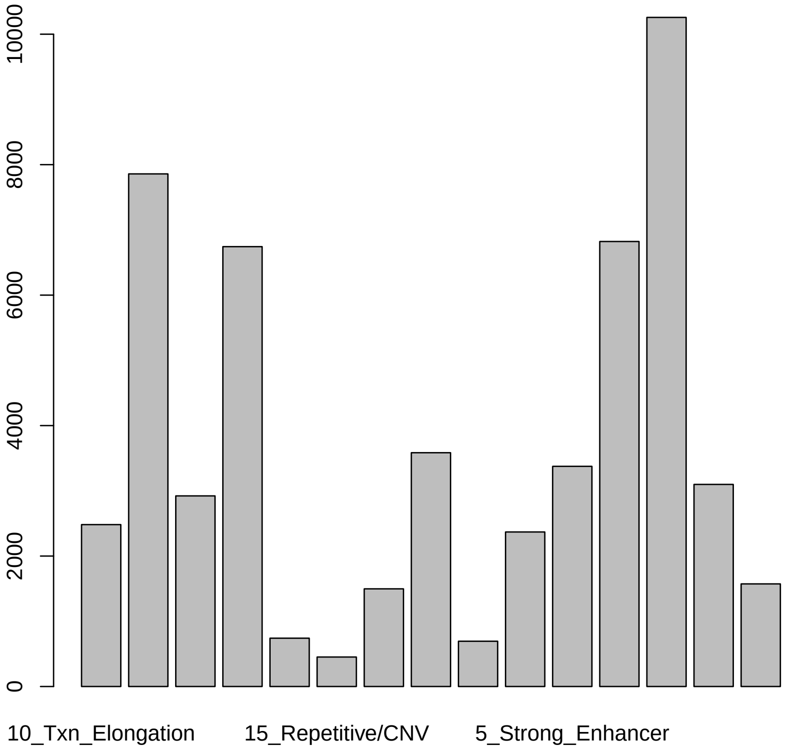
```
##  [1] "10_Txn_Elongation" "11_Weak_Txn"       "12_Repressed"
##  [4] "13_Heterochrom/lo" "14_Repetitive/CNV" "15_Repetitive/CNV"
##  [7] "1_Active_Promoter" "2_Weak_Promoter"   "3_Poised_Promoter"
## [10] "4_Strong_Enhancer" "5_Strong_Enhancer" "6_Weak_Enhancer"
## [13] "7_Weak_Enhancer"   "8_Insulator"       "9_Txn_Transition"
```

```
table(my_var$type)
```

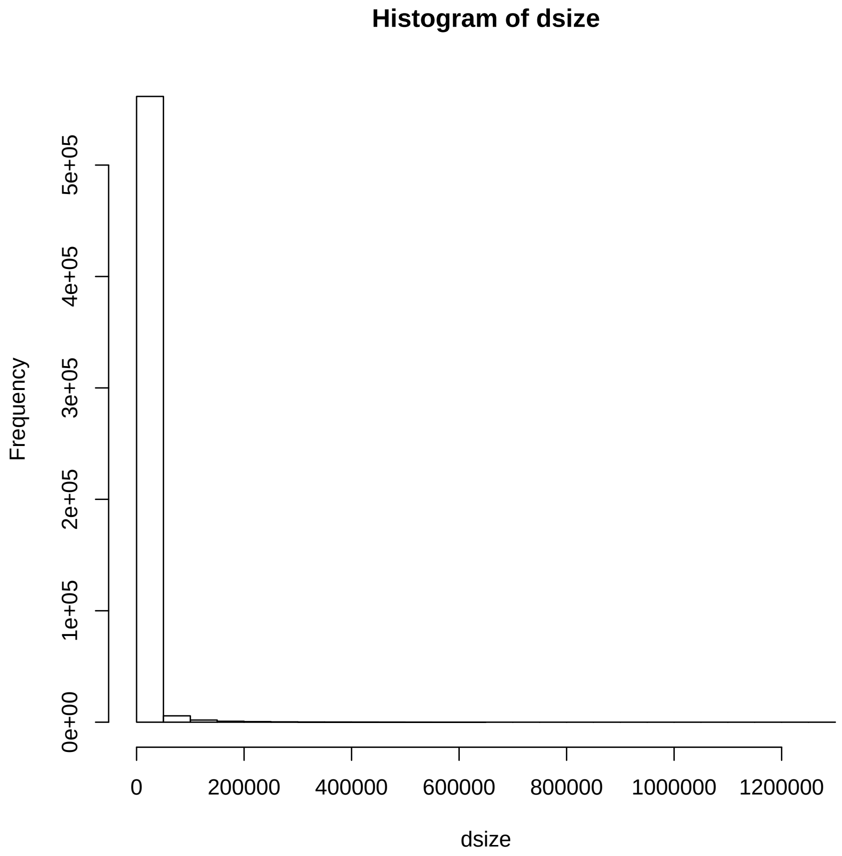
```
##
## 10_Txn_Elongation       11_Weak_Txn      12_Repressed 13_Heterochrom/lo
##             26509             82312             25483             75112
## 14_Repetitive/CNV 15_Repetitive/CNV 1_Active_Promoter   2_Weak_Promoter
##              8028              6128             15278             35065
## 3_Poised_Promoter 4_Strong_Enhancer 5_Strong_Enhancer   6_Weak_Enhancer
##               5263             25486             38604             69111
##   7_Weak_Enhancer       8_Insulator  9_Txn_Transition
##             109468             33265             16227

barplot(table(my_var$chrom))
```



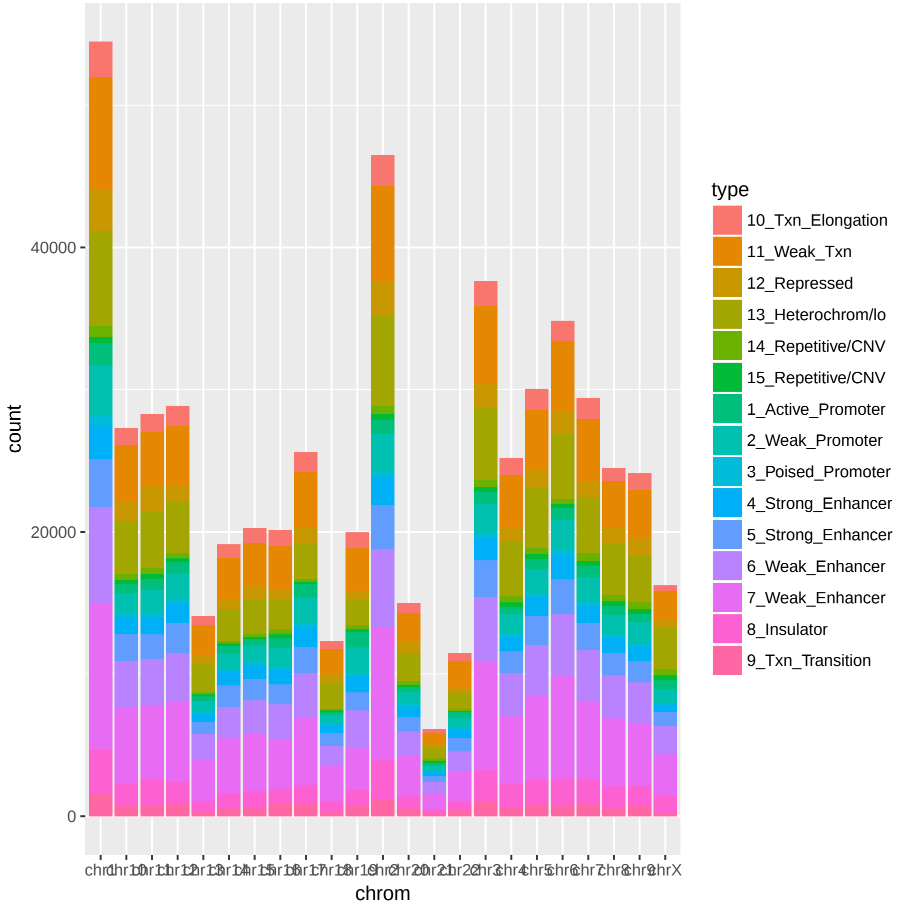
```
#
chr1 <- table(my_var[my_var$chrom=="chr1",]$type)
barplot(chr1)
```



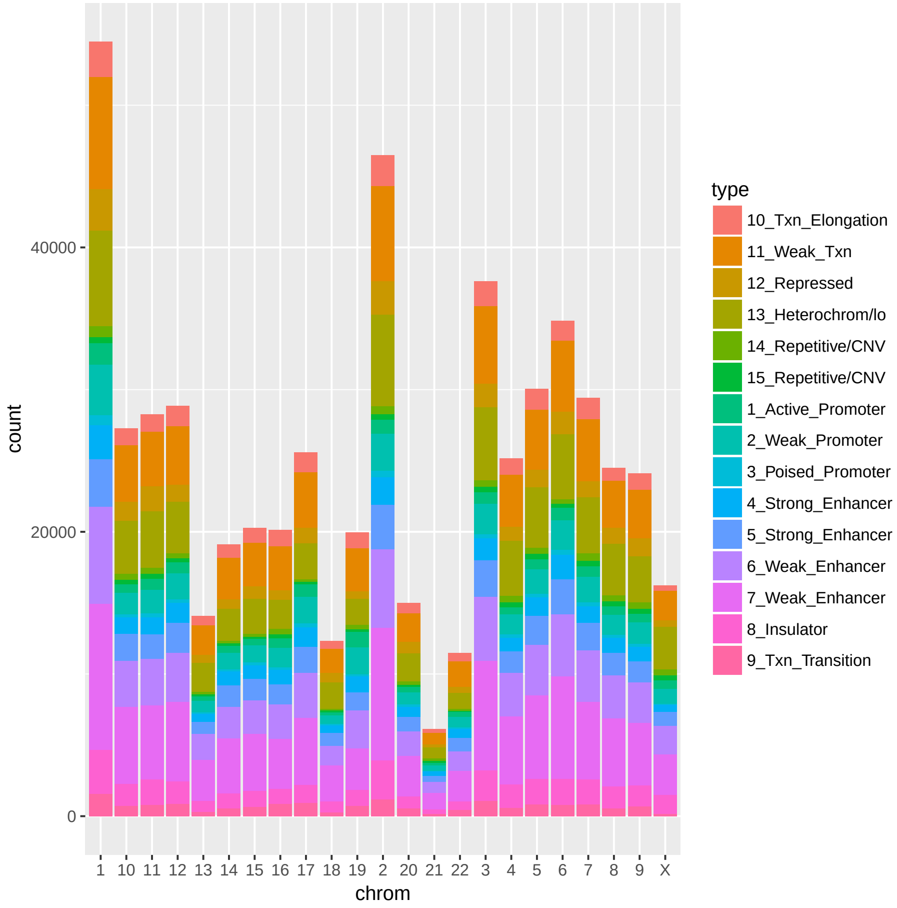
```
#
dsize = my_var$stop - my_var$start
hist(dsize)
```



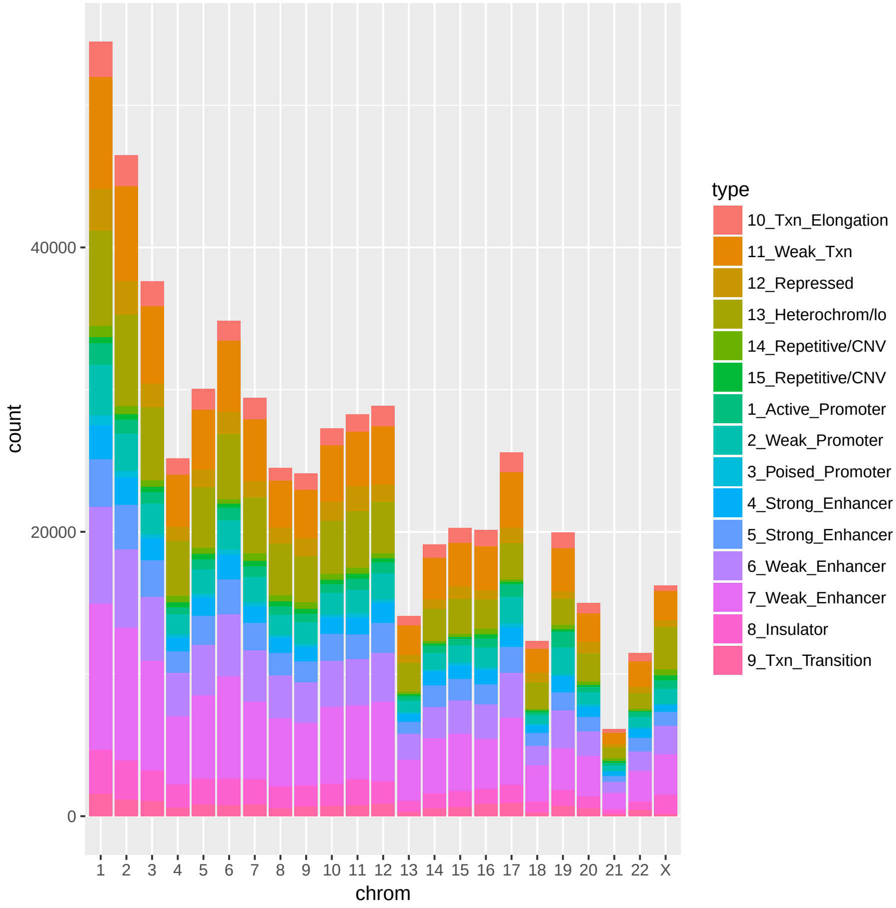
```
summary(dsize)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     400     800    4955    2400 1255000

# ggplot2
ggplot(data=my_var,mapping = aes(x=chrom,fill=type)) +
    geom_bar()
```



```
# 1
# 2  type
# 3 type

#       chr
my_var$chrom <- factor(gsub("chr", "", my_var$chrom))
ggplot(my_var,aes(x=chrom,fill=type)) + geom_bar()
```

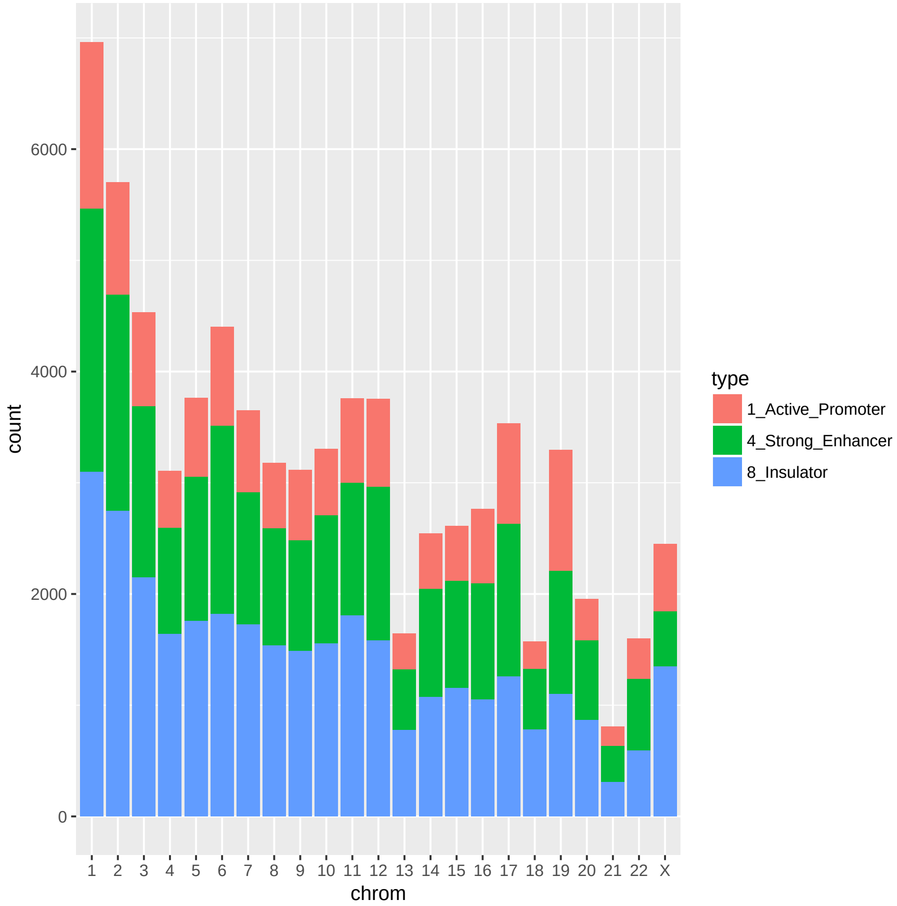


```
#
c(seq(1,22),"X","Y")

##  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"  "13"  "14"
## [15] "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "X"   "Y"

my_var$chrom <- factor(my_var$chrom, levels=c(seq(1,22),"X","Y"))
ggplot(my_var,aes(x=chrom,fill=type)) + geom_bar()
```

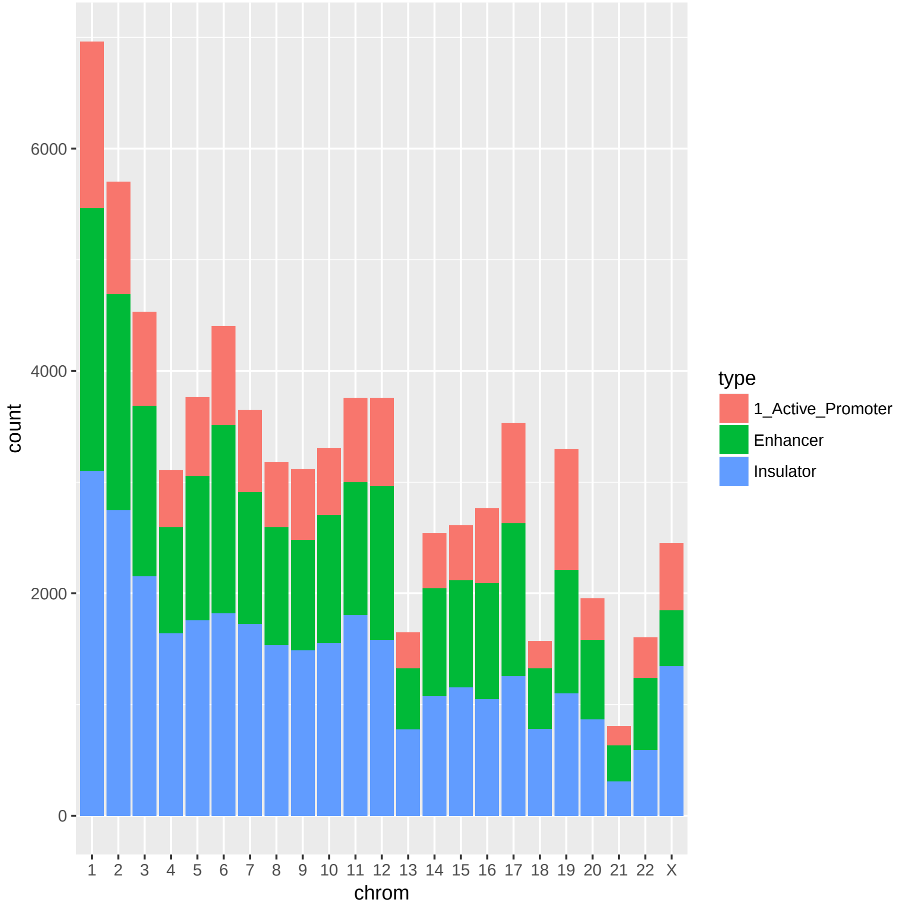


```
#    .

summary(my_var$type)

## 10_Txn_Elongation       11_Weak_Txn      12_Repressed 13_Heterochrom/lo
##             26509             82312             25483             75112
## 14_Repetitive/CNV 15_Repetitive/CNV 1_Active_Promoter   2_Weak_Promoter
##              8028              6128             15278             35065
## 3_Poised_Promoter 4_Strong_Enhancer 5_Strong_Enhancer   6_Weak_Enhancer
##              5263             25486             38604             69111
##   7_Weak_Enhancer       8_Insulator  9_Txn_Transition
##            109468             33265             16227

#            %in%
my_var <- my_var[my_var$type %in%
                 c("1_Active_Promoter",
                   "4_Strong_Enhancer",
                   "8_Insulator"), ]

summary(my_var$type)
```

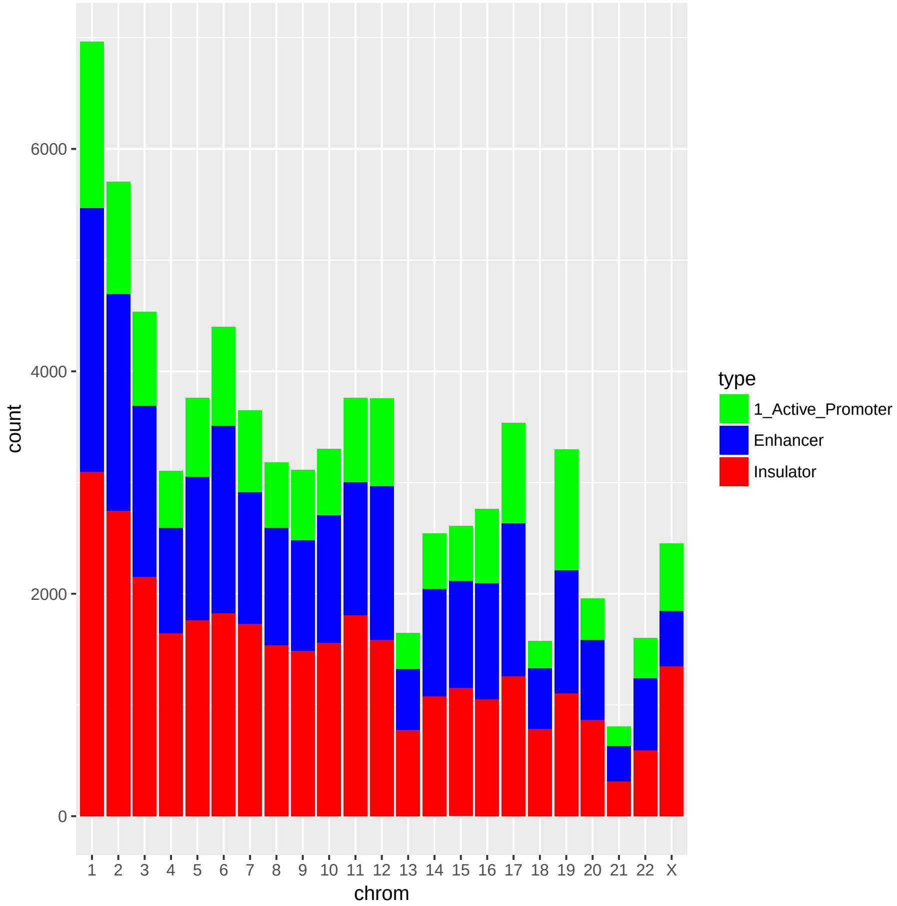
```
## 10_Txn_Elongation        11_Weak_Txn      12_Repressed 13_Heterochrom/lo
##                  0                 0                 0                 0
## 14_Repetitive/CNV 15_Repetitive/CNV 1_Active_Promoter   2_Weak_Promoter
##                  0                 0             15278                 0
## 3_Poised_Promoter 4_Strong_Enhancer 5_Strong_Enhancer   6_Weak_Enhancer
##                  0             25486                 0                 0
##    7_Weak_Enhancer       8_Insulator  9_Txn_Transition
##                  0             33265                 0

ggplot(my_var,aes(x=chrom,fill=type)) + geom_bar()
```



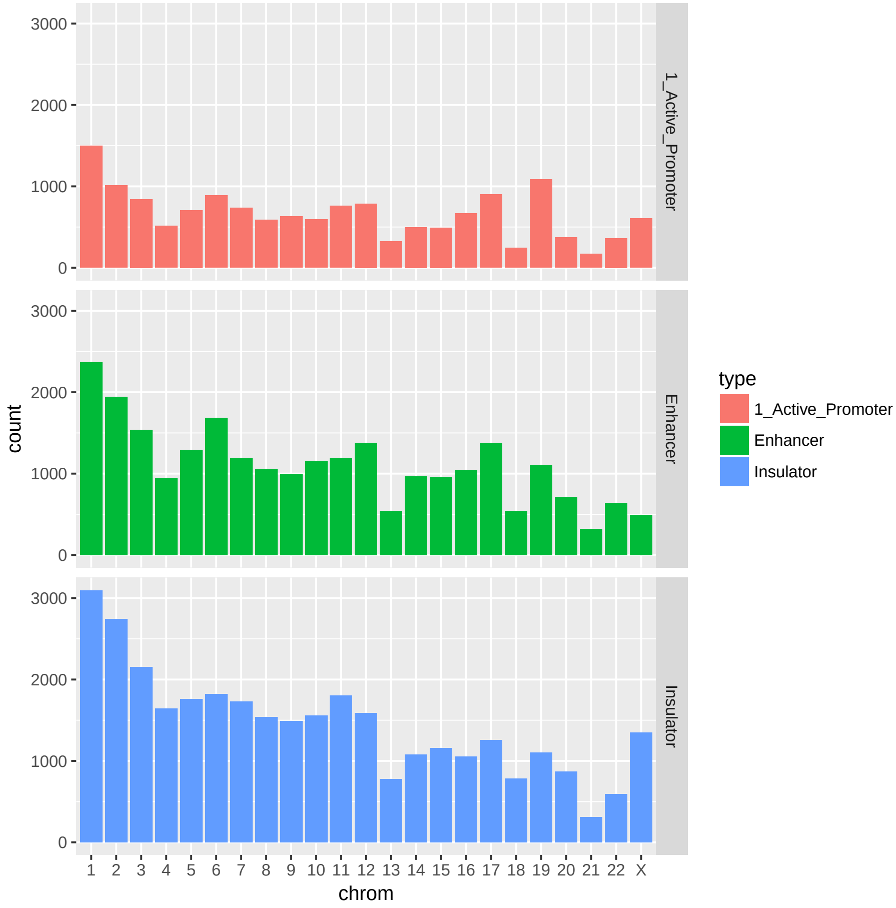
```
#
levels(my_var$type)[1]="Promoter"
levels(my_var$type)[10]="Enhancer"
levels(my_var$type)[14]="Insulator"

# plyr  revalue
#library(plyr)
#my_var$type <- revalue(my_var$type, c("1_Active_Promoter"="Promoter", "4_Strong_Enhancer"
summary(my_var$type)

##           Promoter        11_Weak_Txn      12_Repressed 13_Heterochrom/lo
##                  0                  0                 0                 0
## 14_Repetitive/CNV 15_Repetitive/CNV 1_Active_Promoter   2_Weak_Promoter
##                  0                  0             15278                 0
## 3_Poised_Promoter           Enhancer 5_Strong_Enhancer   6_Weak_Enhancer
##                  0              25486                 0                 0
##    7_Weak_Enhancer          Insulator  9_Txn_Transition
##                  0              33265                 0

ggplot(my_var,aes(x=chrom,fill=type)) + geom_bar()
```



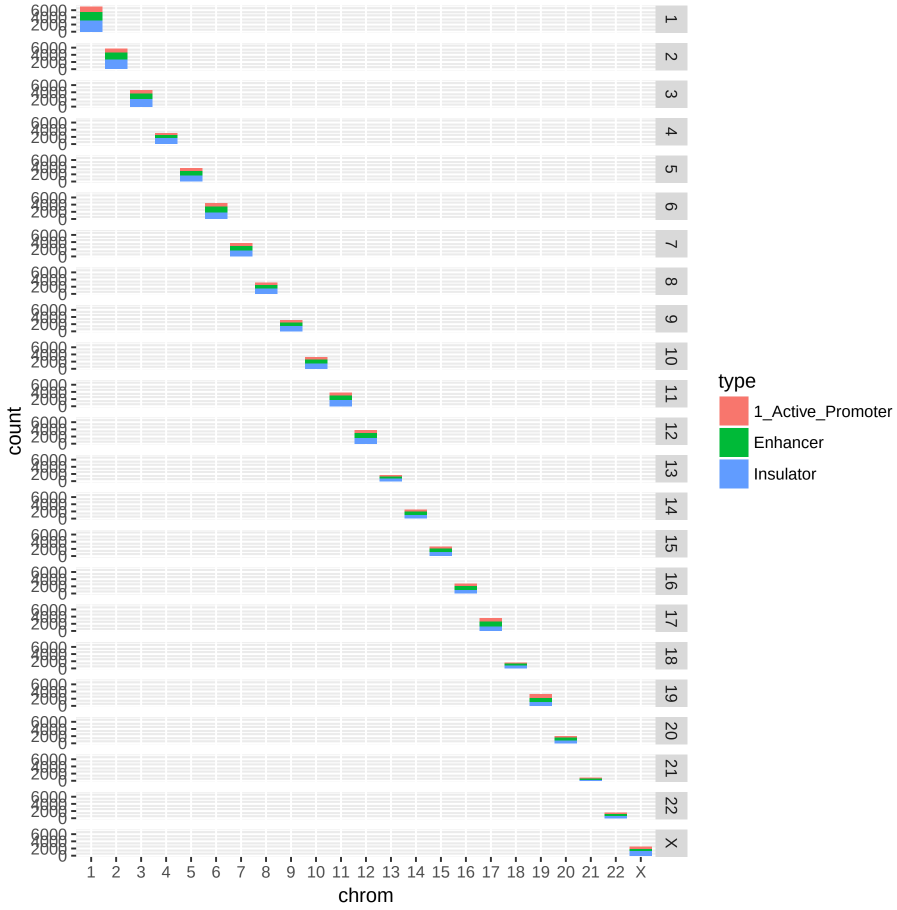
```
ggplot(my_var,aes(x=chrom,fill=type)) + geom_bar() +
    scale_fill_manual(values = c("green","blue","red"))
```



```
#  facet
head(my_var)

##     chrom  start   stop              type V5 V6     V7     V8         V9
## 3       1  11137  11737         Insulator  0  .  11137  11737 10,190,254
## 13      1  28537  29737 1_Active_Promoter  0  .  28537  29737    255,0,0
## 22      1  91137  91737         Insulator  0  .  91137  91737 10,190,254
## 24      1  92337  92537         Insulator  0  .  92337  92537 10,190,254
## 29      1 104737 105137         Insulator  0  . 104737 105137 10,190,254
## 31      1 134337 134737         Insulator  0  . 134337 134737 10,190,254

colnames(my_var)

## [1] "chrom" "start" "stop"  "type"  "V5"    "V6"    "V7"    "V8"    "V9"

ggplot(my_var, aes(x=chrom,fill=type)) + geom_bar() +
    facet_grid(type ~ .)
```

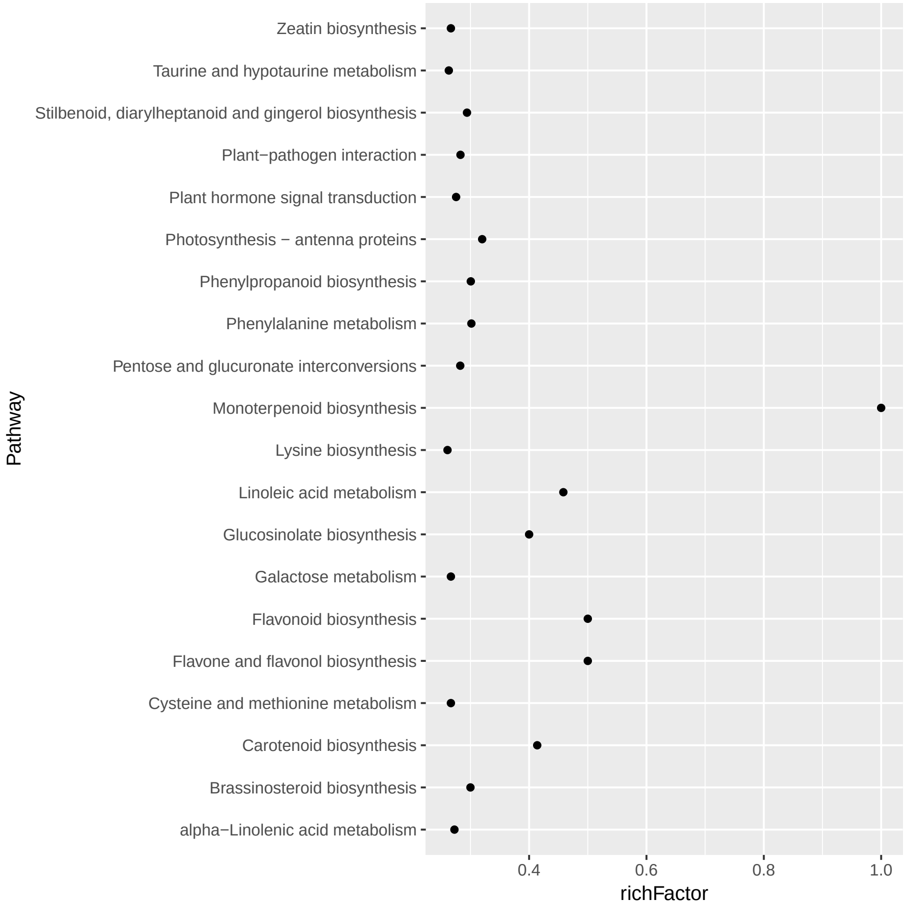


```
ggplot(my_var, aes(x=chrom,fill=type)) + geom_bar() +
    facet_grid(chrom ~ .)
```

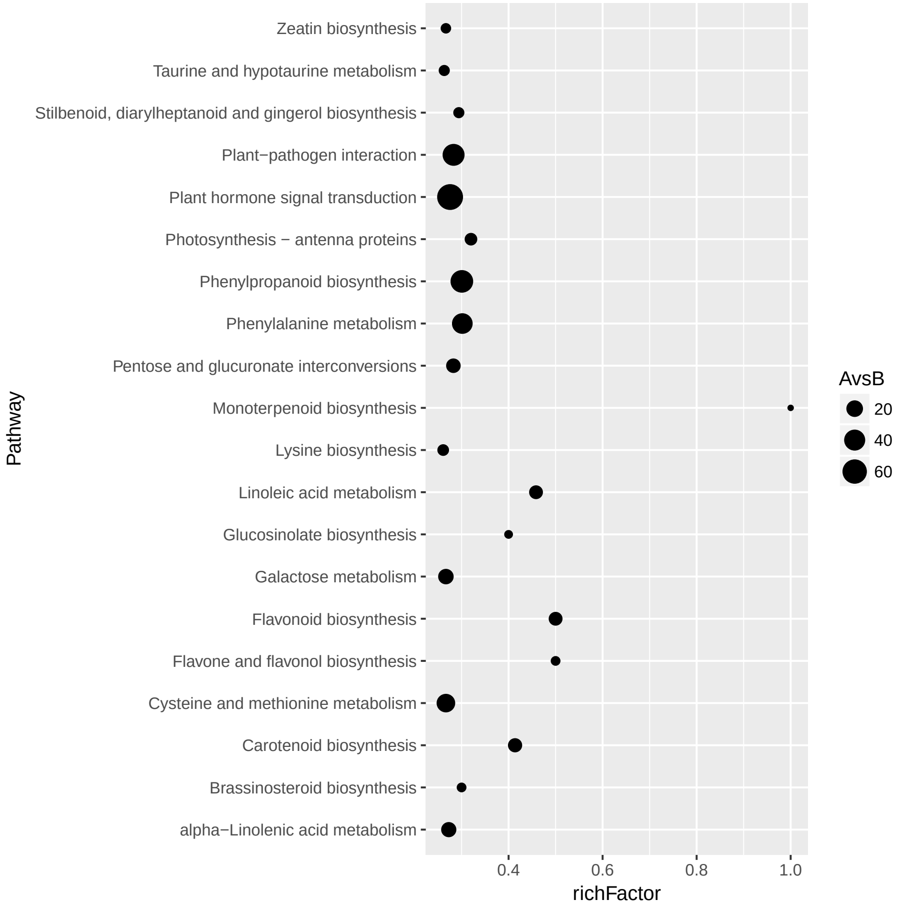


2.8 KEGG Plot

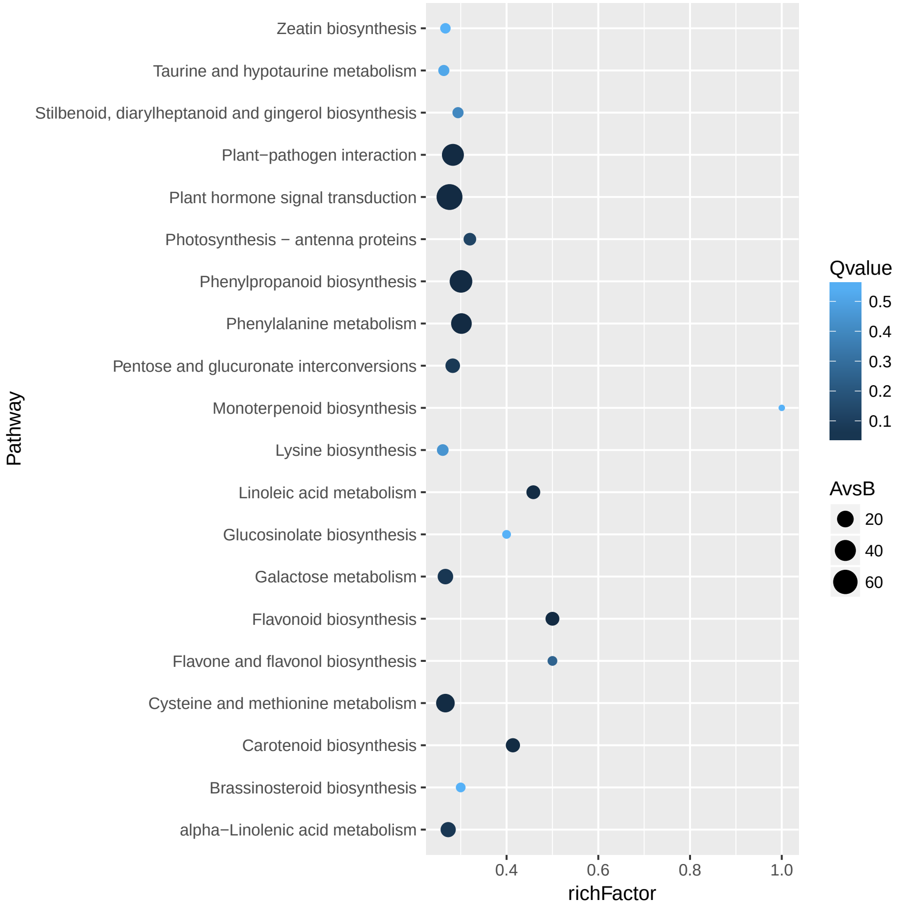
```
library(ggplot2)

pathway <-  read.csv("data/kegg.csv",header=T)
colnames(pathway)

## [1] "Pathway"     "AvsB"         "All_Unigene" "Pvalue"       "Qvalue"
## [6] "richFactor"  "Pathway.ID"   "Genes"       "KOs"

pp <-  ggplot(data=pathway,mapping = aes(x=richFactor,y=Pathway))
pp + geom_point()
```



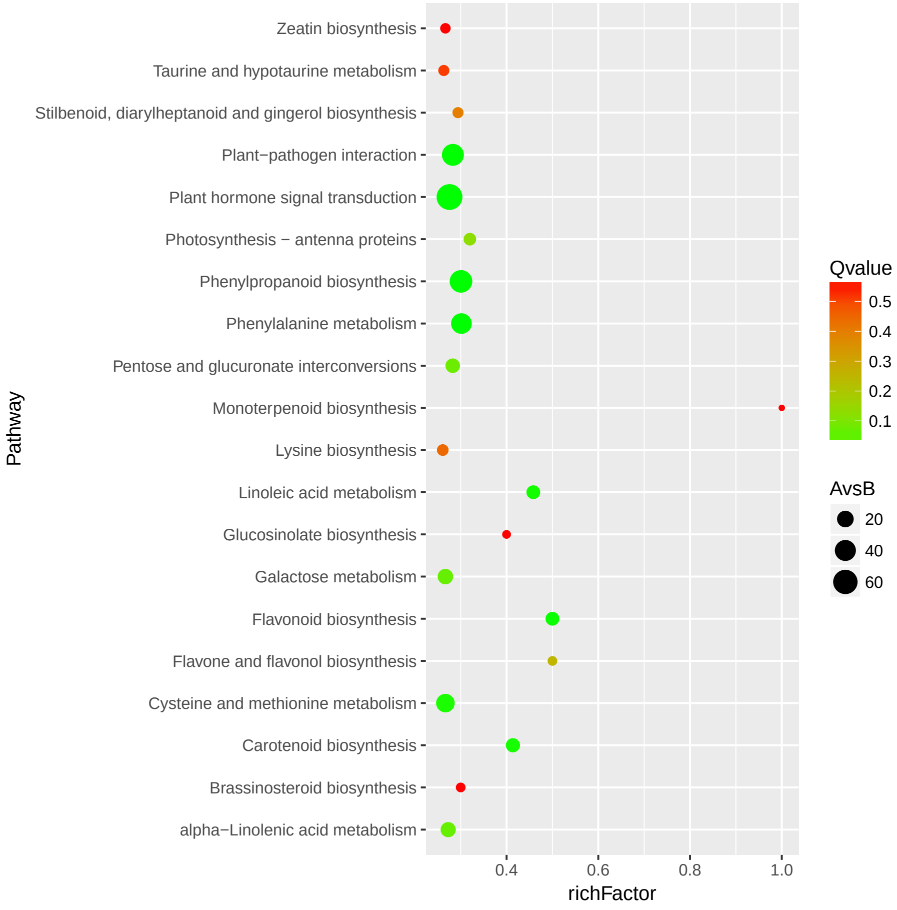
```
pp + geom_point(aes(size=AvsB))
```



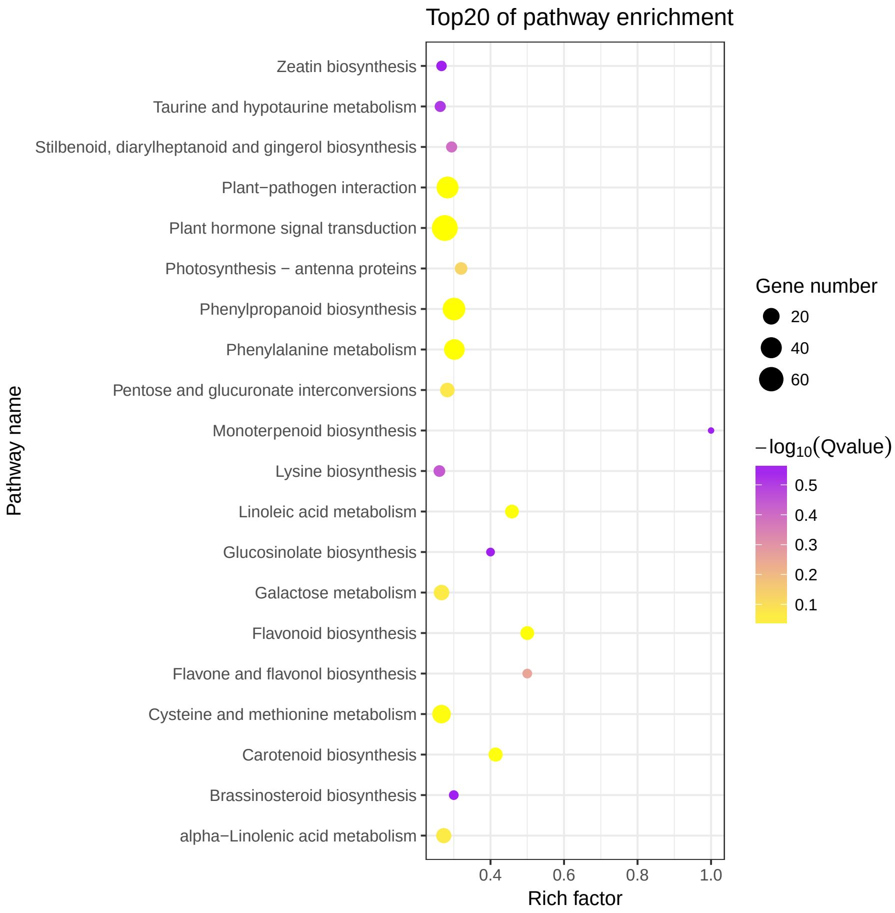
```
pp + geom_point(aes(size=AvsB,color=Qvalue))
```



```
pp + geom_point(aes(size=AvsB,color=Qvalue)) +
    scale_colour_gradient(low="green",high="red")
```



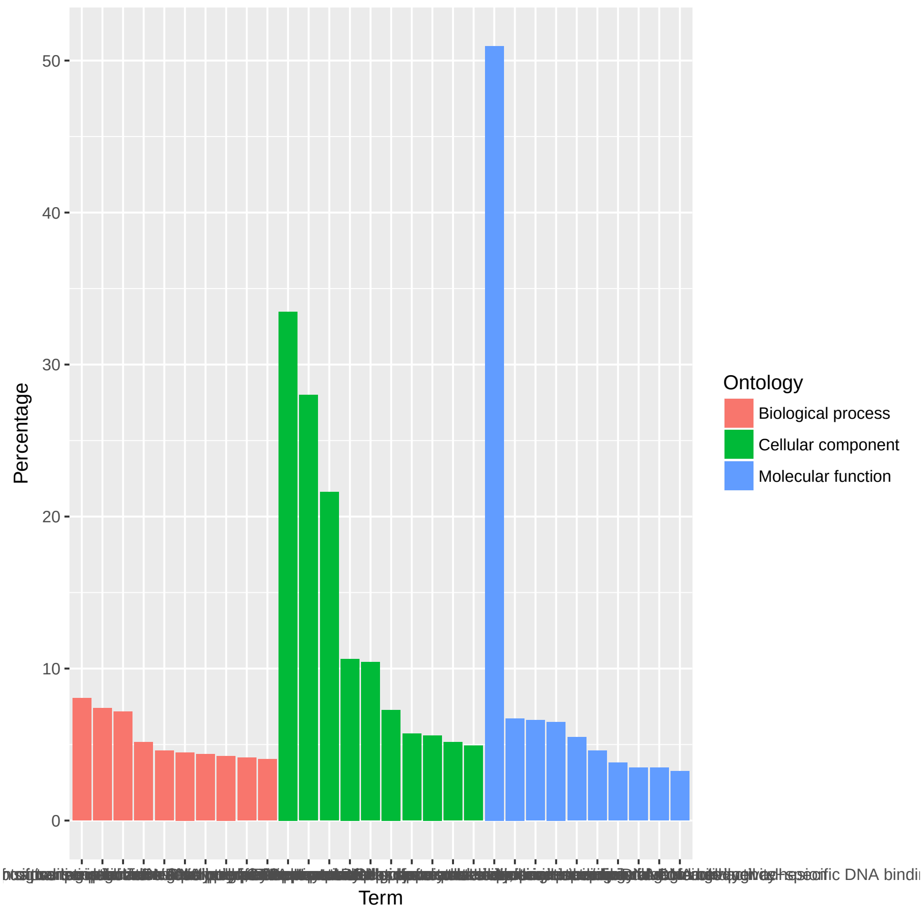
```
pr = pp + geom_point(aes(size=AvsB,color=Qvalue)) +
    scale_colour_gradient(low="yellow",high="purple") +
    labs(color=expression(-log[10](Qvalue)),
           size="Gene number",
           x="Rich factor",
           y="Pathway name",
           title="Top20 of pathway enrichment")
pr + theme_bw()
```



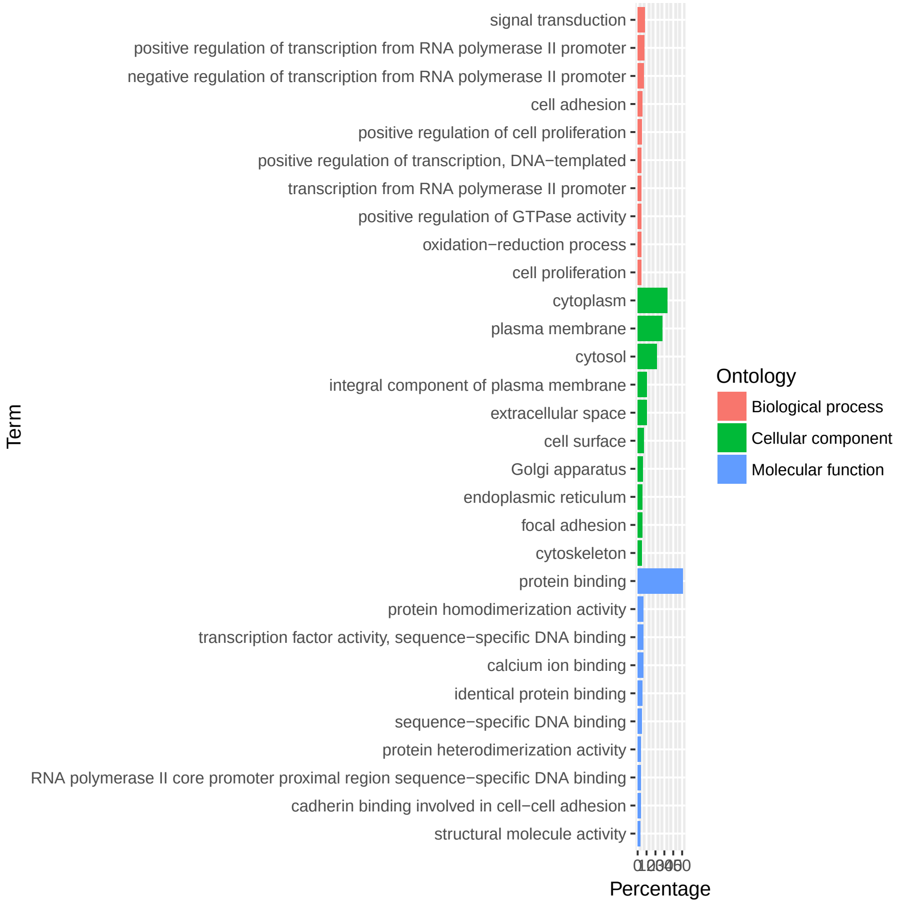
2.9 Go annotation plot

```
library(ggplot2)

go <- read.csv("data/go.csv",header = T)
go_sort <- go[order(go$Ontology,-go$Percentage),]
m <- go_sort[go_sort$Ontology=="Molecular function",][1:10,]
c <- go_sort[go_sort$Ontology=="Cellular component",][1:10,]
b <- go_sort[go_sort$Ontology=="Biological process",][1:10,]
slimgo <- rbind(b,c,m)

#   Trem
slimgo$Term=factor(slimgo$Term,levels=slimgo$Term)

colnames(slimgo)

## [1] "Ontology"     "Term"         "GO_ID"        "Input_number"
## [5] "Percentage"
```

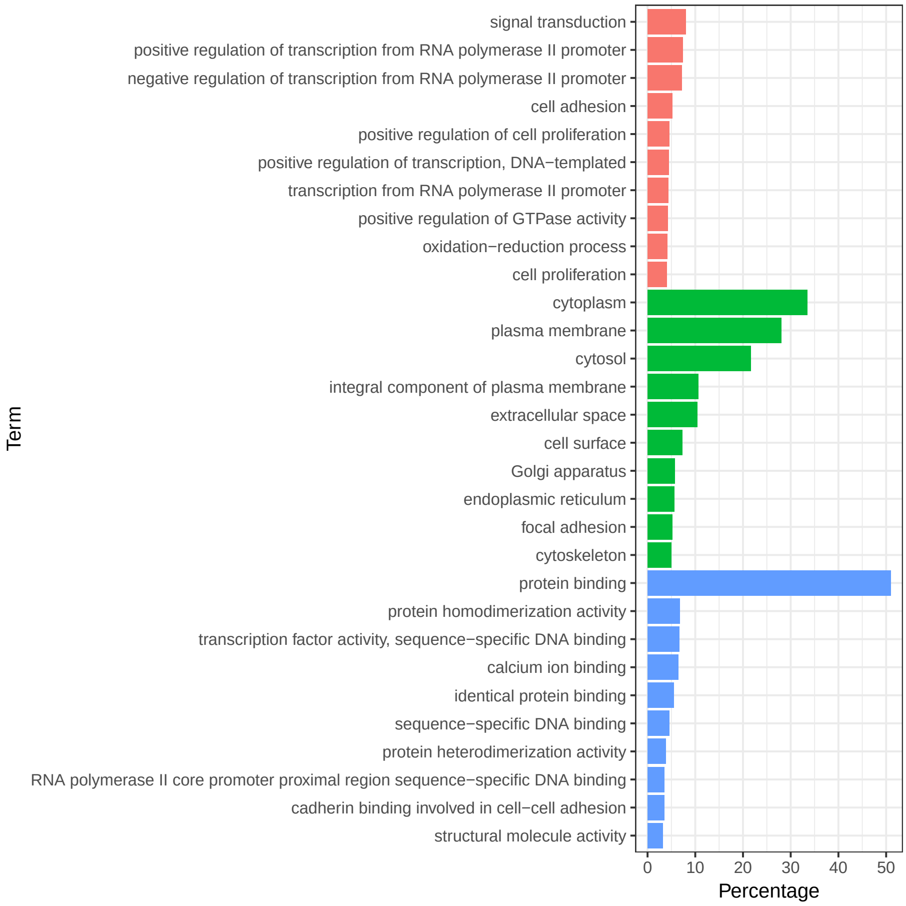
```
pp=ggplot(data = slimgo, mapping = aes(x=Term,y=Percentage,fill=Ontology))
pp+geom_bar(stat="identity")
```



```
pbar=pp+geom_bar(stat="identity")+coord_flip()
pbar=pp+geom_bar(stat="identity")+coord_flip() +
    scale_x_discrete(limits=rev(levels(slimgo$Term)))
pbar
```

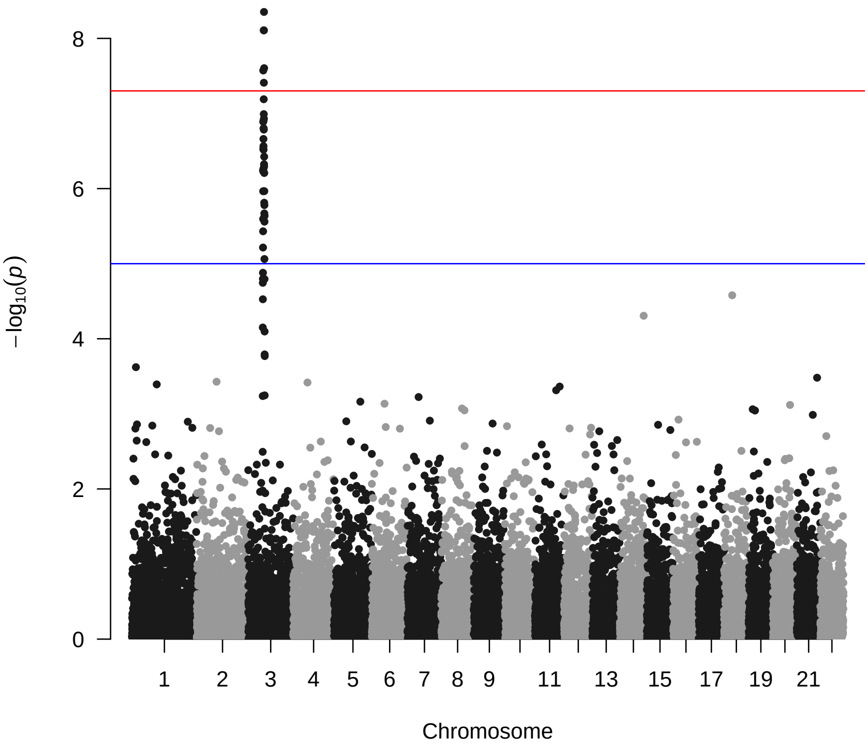


```
pr=pbar+scale_fill_discrete(name="Ontology",
                            breaks=c("Biological process",
                                     "Molecular function",
                                     "Cellular component"))

pr=pbar+scale_fill_discrete(name="Ontology",
                            breaks=c("Biological process",
                                     "Molecular function",
                                     "Cellular component")) +
    guides(fill=FALSE)
pr+theme_bw()
```

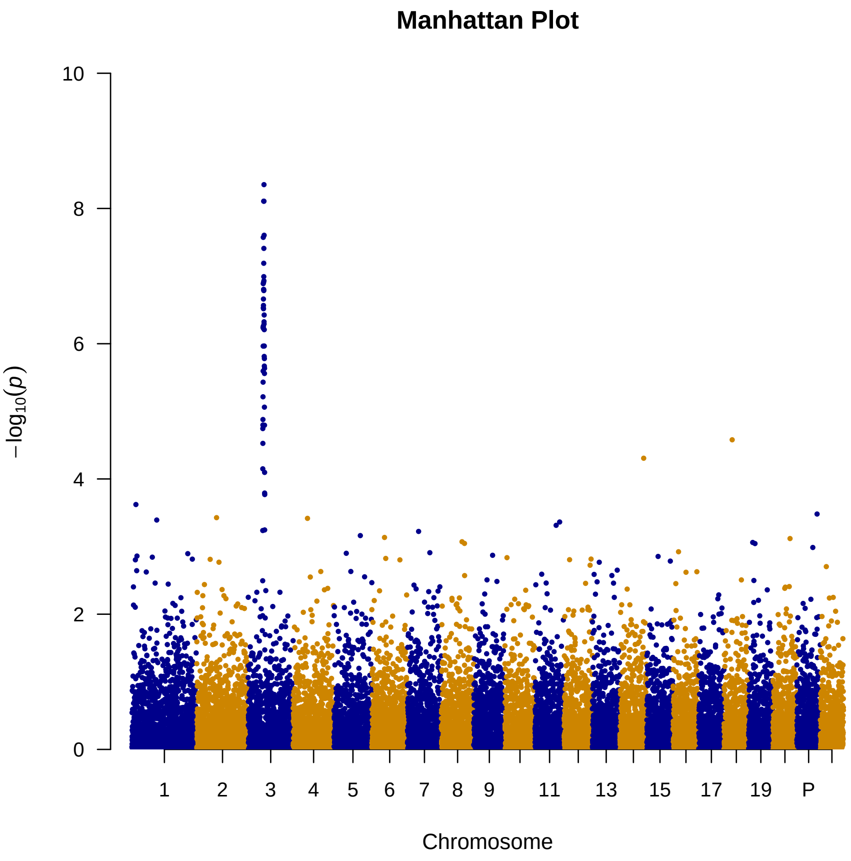


2.10 Manhattan

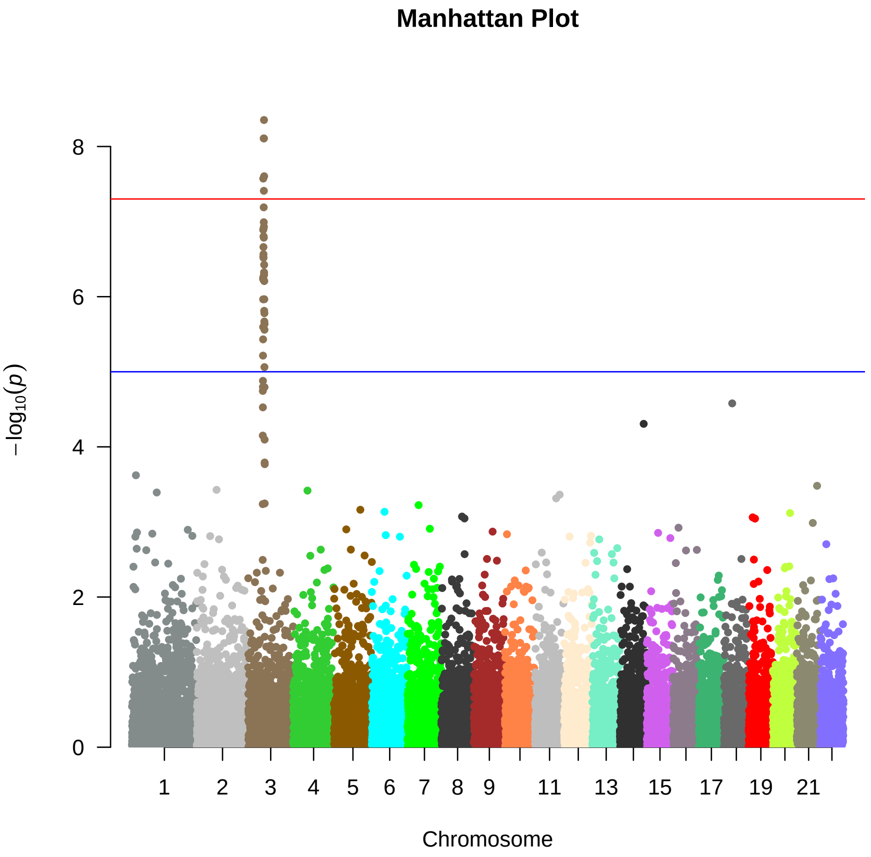
```
#install.packages("qqman")
library(qqman)

##
## For example usage please run: vignette('qqman')
##
## Citation appreciated but not required:
## Turner, S.D. qqman: an R package for visualizing GWAS results using
Q-Q and manhattan plots. biorXiv DOI: 10.1101/005165 (2014).
##

library(RColorBrewer)
str(gwasResults)

## 'data.frame': 16470 obs. of  4 variables:
##  $ SNP: chr  "rs1" "rs2" "rs3" "rs4" ...
##  $ CHR: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ BP : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ P  : num  0.915 0.937 0.286 0.83 0.642 ...

head(gwasResults)

##   SNP CHR BP         P
## 1 rs1   1  1 0.9148060
## 2 rs2   1  2 0.9370754
## 3 rs3   1  3 0.2861395
## 4 rs4   1  4 0.8304476
## 5 rs5   1  5 0.6417455
## 6 rs6   1  6 0.5190959

# Plot !
manhattan(gwasResults)
```



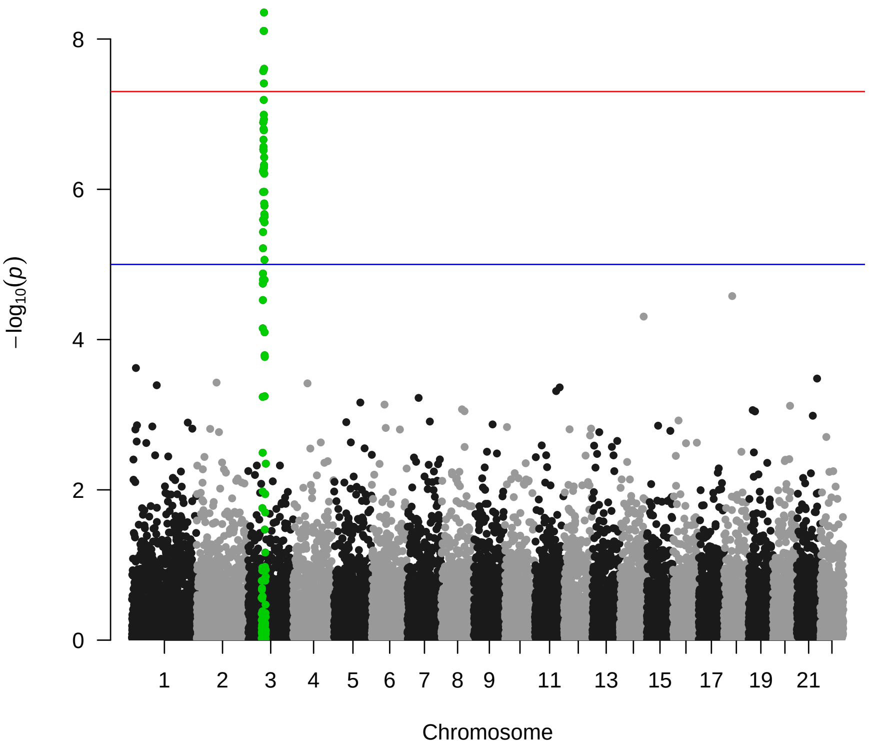
```
manhattan(gwasResults, main = "Manhattan Plot",
          ylim = c(0, 10),
          cex = 0.6,
          cex.axis = 0.9,
          col = c("blue4", "orange3"),
          suggestiveline = F,
          genomewideline = F,
          chrlabs = c(1:20, "P", "Q"))
```



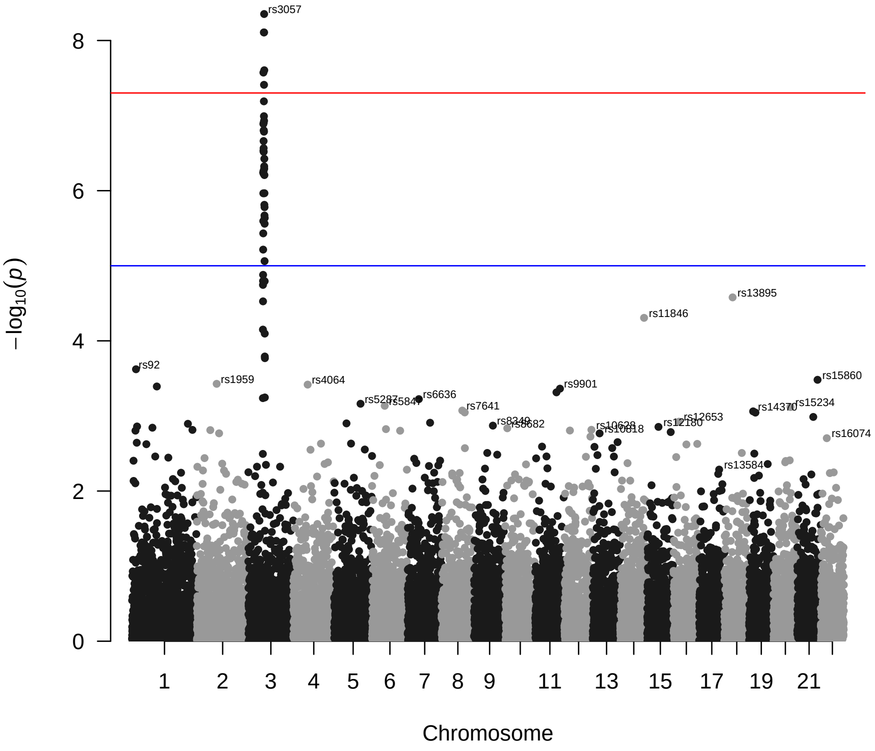
```
#
number <- length(unique(gwasResults$CHR))
set.seed(888)
yanse <- sample(colors(),number,replace = F)
manhattan(gwasResults,col = yanse,main = "Manhattan Plot")
```



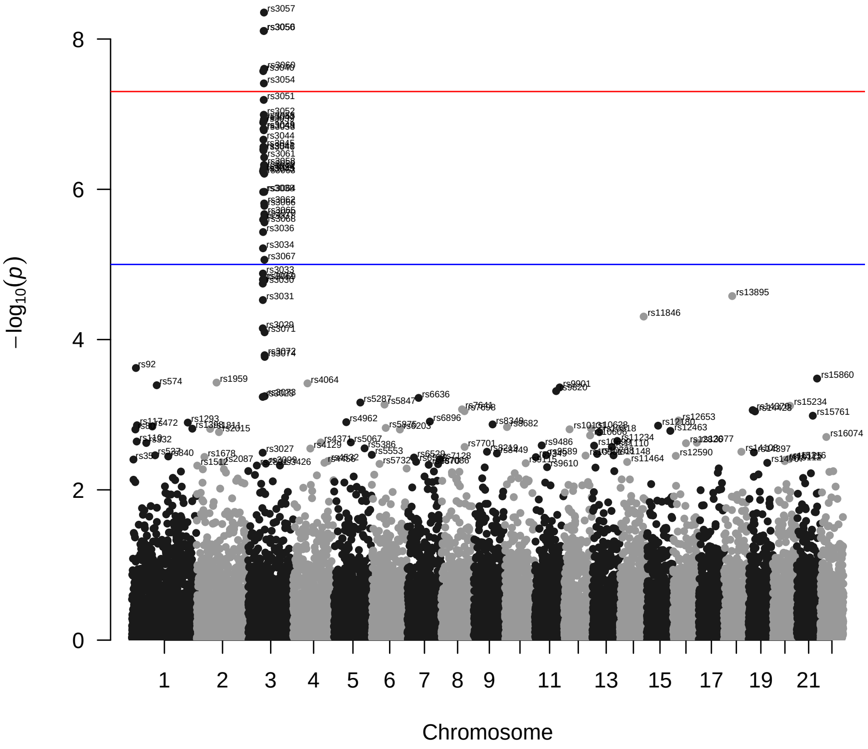
```
#   SNP
snpsOfInterest

##   [1] "rs3001" "rs3002" "rs3003" "rs3004" "rs3005" "rs3006" "rs3007"
##   [8] "rs3008" "rs3009" "rs3010" "rs3011" "rs3012" "rs3013" "rs3014"
##  [15] "rs3015" "rs3016" "rs3017" "rs3018" "rs3019" "rs3020" "rs3021"
##  [22] "rs3022" "rs3023" "rs3024" "rs3025" "rs3026" "rs3027" "rs3028"
##  [29] "rs3029" "rs3030" "rs3031" "rs3032" "rs3033" "rs3034" "rs3035"
##  [36] "rs3036" "rs3037" "rs3038" "rs3039" "rs3040" "rs3041" "rs3042"
##  [43] "rs3043" "rs3044" "rs3045" "rs3046" "rs3047" "rs3048" "rs3049"
##  [50] "rs3050" "rs3051" "rs3052" "rs3053" "rs3054" "rs3055" "rs3056"
##  [57] "rs3057" "rs3058" "rs3059" "rs3060" "rs3061" "rs3062" "rs3063"
##  [64] "rs3064" "rs3065" "rs3066" "rs3067" "rs3068" "rs3069" "rs3070"
##  [71] "rs3071" "rs3072" "rs3073" "rs3074" "rs3075" "rs3076" "rs3077"
##  [78] "rs3078" "rs3079" "rs3080" "rs3081" "rs3082" "rs3083" "rs3084"
##  [85] "rs3085" "rs3086" "rs3087" "rs3088" "rs3089" "rs3090" "rs3091"
##  [92] "rs3092" "rs3093" "rs3094" "rs3095" "rs3096" "rs3097" "rs3098"
##  [99] "rs3099" "rs3100"

manhattan(gwasResults, highlight = snpsOfInterest)
```



```
# SNP

manhattan(gwasResults, annotatePval = 0.01)
```



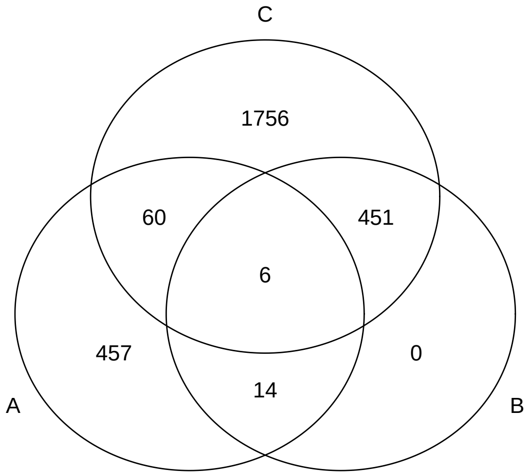
```r
manhattan(gwasResults, annotatePval = 0.005, annotateTop = FALSE)
```



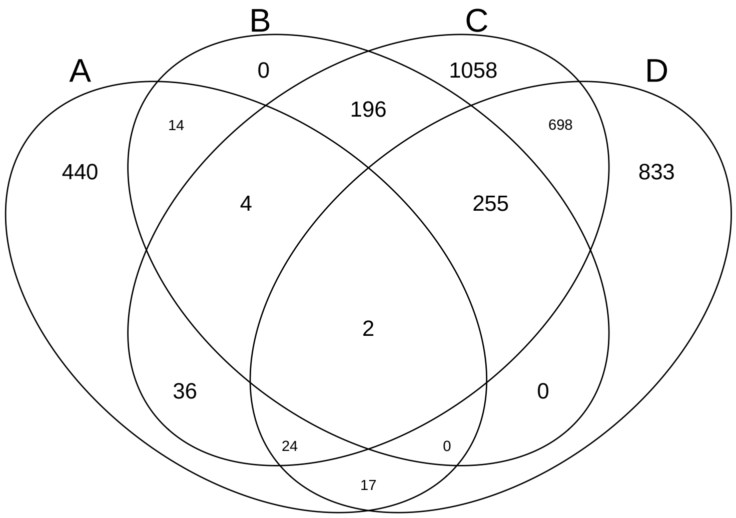
2.11 Venn

```
listA <- read.csv("data/genes_list_A.txt",header=FALSE)
A <- listA$V1
listB <- read.csv("data/genes_list_B.txt",header=FALSE)
B <- listB$V1
listC <- read.csv("data/genes_list_C.txt",header=FALSE)
C <- listC$V1
listD <- read.csv("data/genes_list_D.txt",header=FALSE)
D <- listD$V1

#length(A);length(B);length(C);length(D)
#intersect(A,B)
#setdiff(A,B)
#setdiff(B,A)
#union(C,D)

# gplots
```

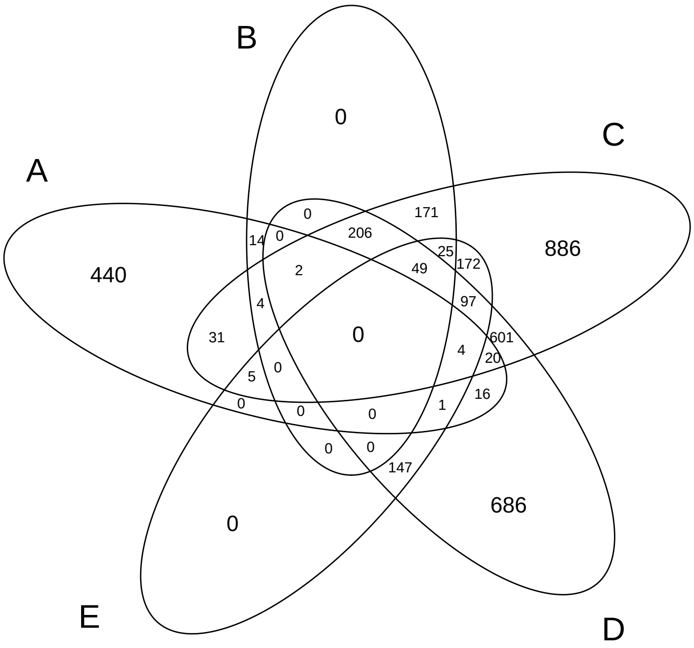
```
library(gplots)

##
## Attaching package:  'gplots'
## The following object is masked from 'package:IRanges':
##
##     space
## The following object is masked from 'package:stats':
##
##     lowess

E <- sample(unique(union(C,D)),500,replace = F)
vennlist <- list(A,B,C,D,E)
venn(vennlist[1:3])
```



```
venn(vennlist[1:4])
```



```r
venn(vennlist)
```



```
# VennDiagram
# install package VennDiagram
```

2.11.1 gplots Venn

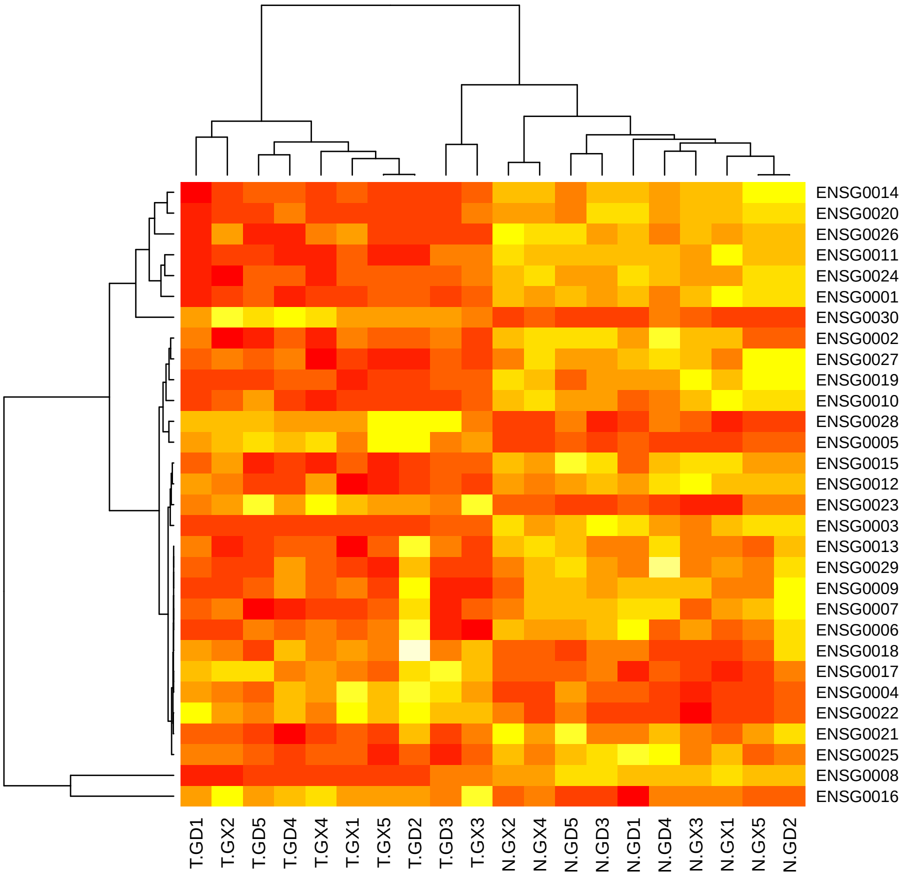
```
library(VennDiagram)

## Loading required package:  grid
## Loading required package:  futile.logger

# This function only works by saving directly to a file
venn.diagram(list("list C"=C, "list D"=D),
             fill = c("yellow","cyan"),
             cex = 1.5,filename = "venn2.png")

## [1] 1

venn.diagram(list(A = A, C = C, D = D),
             fill = c("yellow","red","cyan"),
             cex = 1.5,filename="venn3.png")
```

```
## [1] 1

venn.diagram(list(A = A, B = B, C = C, D = D),
             fill = c("yellow","red","cyan","forestgreen"),
             cex = 1.5,filename="venn4.png")

## [1] 1

venn.diagram(list(A = A, B = B, C = C, D = D , E = E ),
             fill = c("yellow","red","cyan",
                      "forestgreen","lightblue"),
             cex = 1.5,filename="venn5.png")

## [1] 1
```

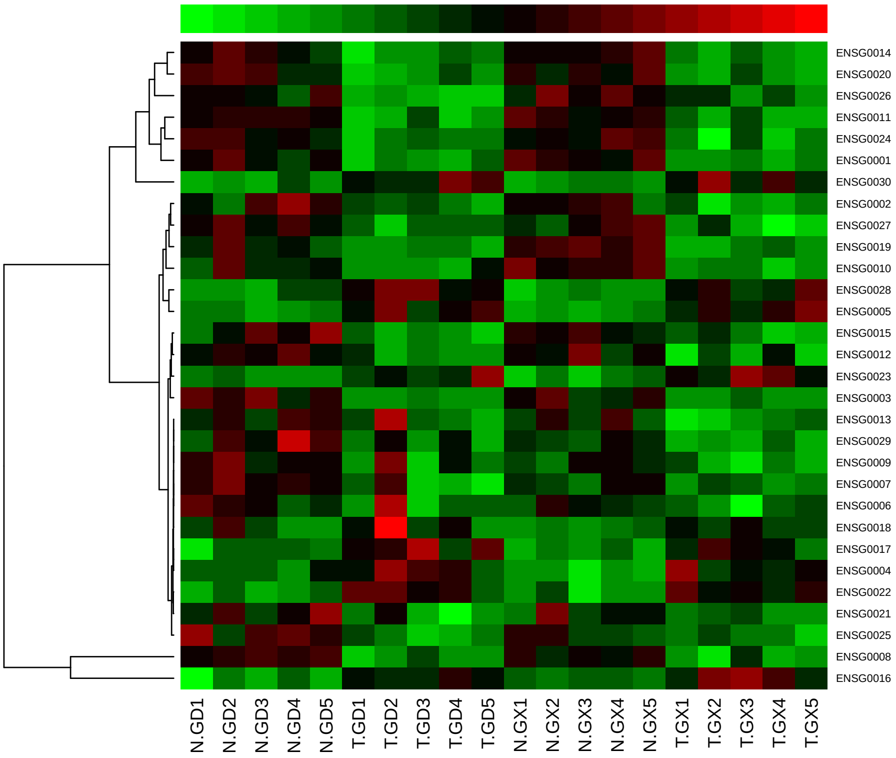
2.12 Heatmap

2.12.1 R heatmap

```
dta <- read.csv("data/heatmap.csv",header = T,row.names = 1)
heatmap(as.matrix(dta))
```

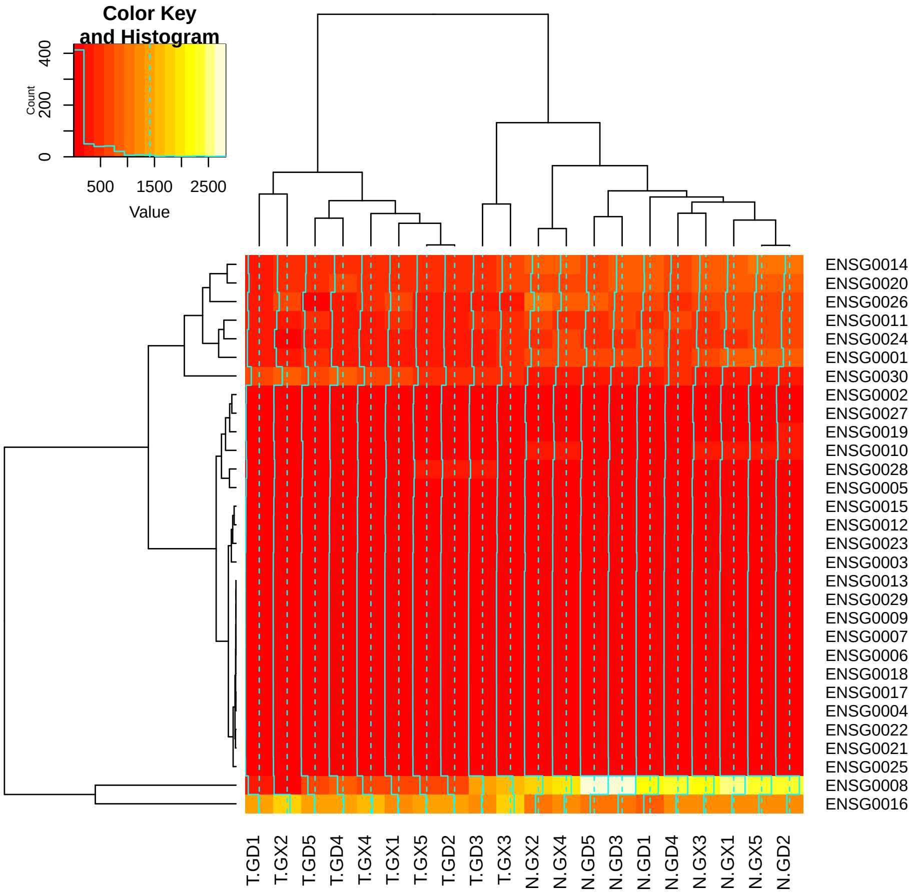


```
heatmap(as.matrix(dta),col=colorRampPalette(c("green","black","red"))(20),
        ColSideColors=colorRampPalette(c("green","black","red"))(20),
        Colv=NA,cexRow=0.8,cexCol=1.2)
```

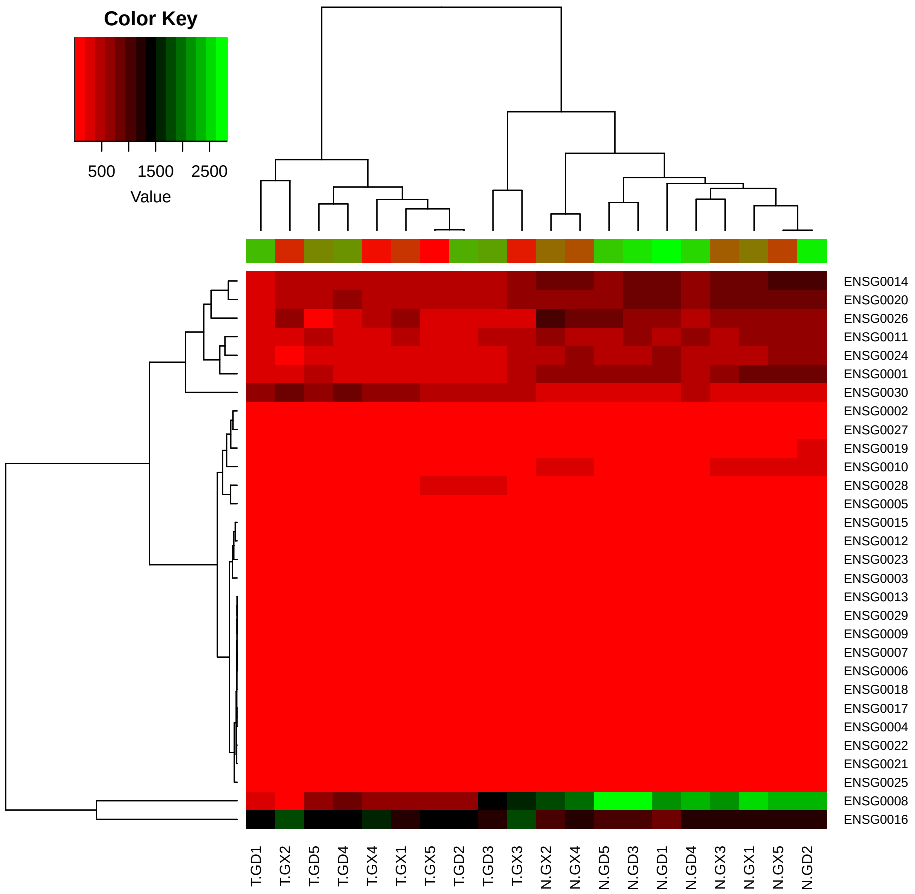


2.12.2 gplots heatmap

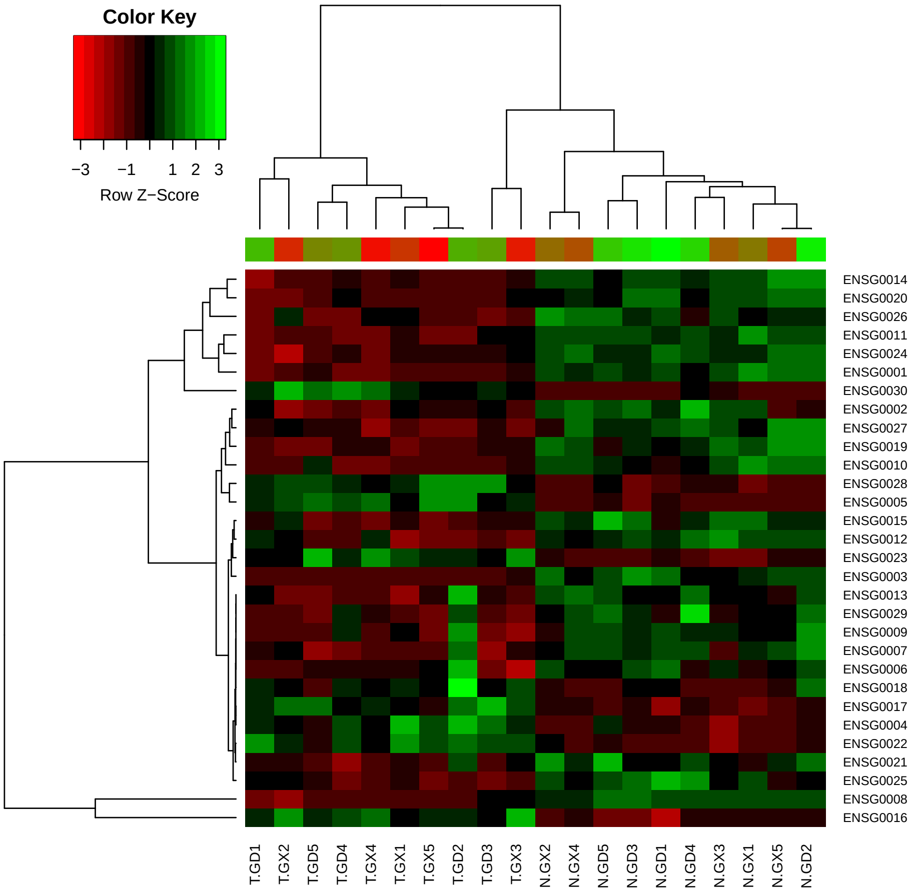
```
library(gplots)
dta <- read.csv("data/heatmap.csv",header = T,row.names = 1)
heatmap.2(as.matrix(dta))
```



```
group=colorRampPalette(c("green","red"))(ncol(dta))
heatmap.2(as.matrix(dta),col=redgreen,ColSideColors=group,
          key=TRUE,symkey=FALSE,density.info="none",trace="none")
```

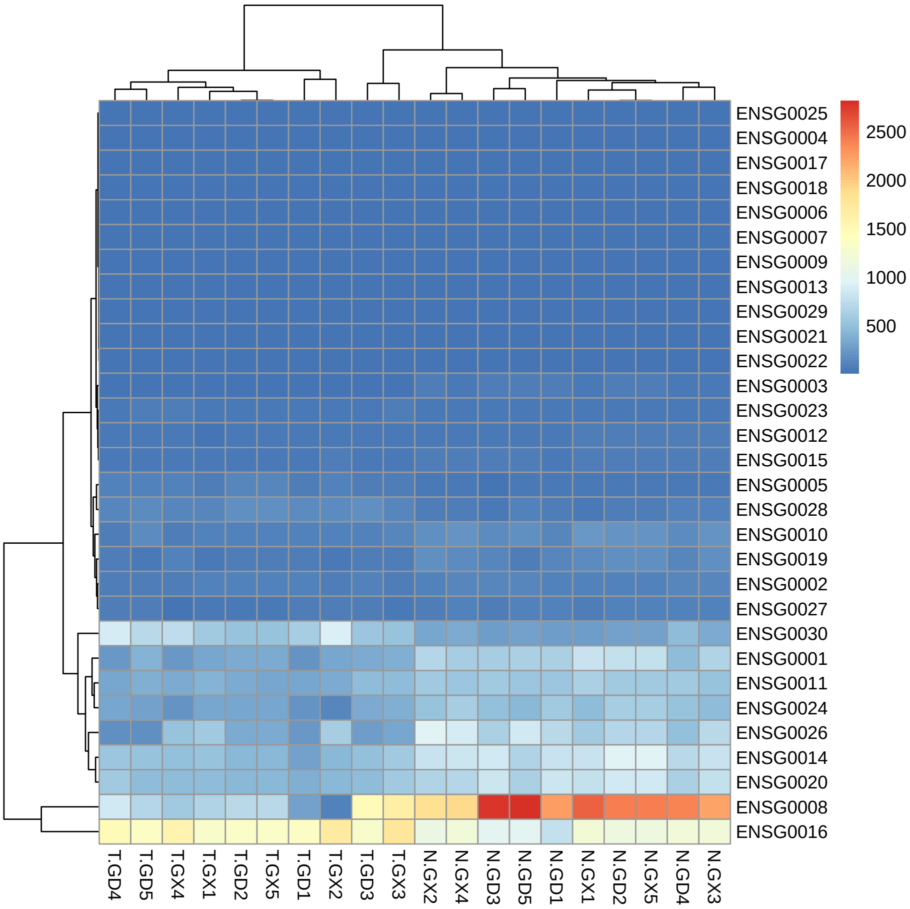


```
heatmap.2(as.matrix(dta),col=redgreen,ColSideColors=group,
          key=TRUE,symkey=FALSE,density.info="none",trace="none",scale = "row")
```

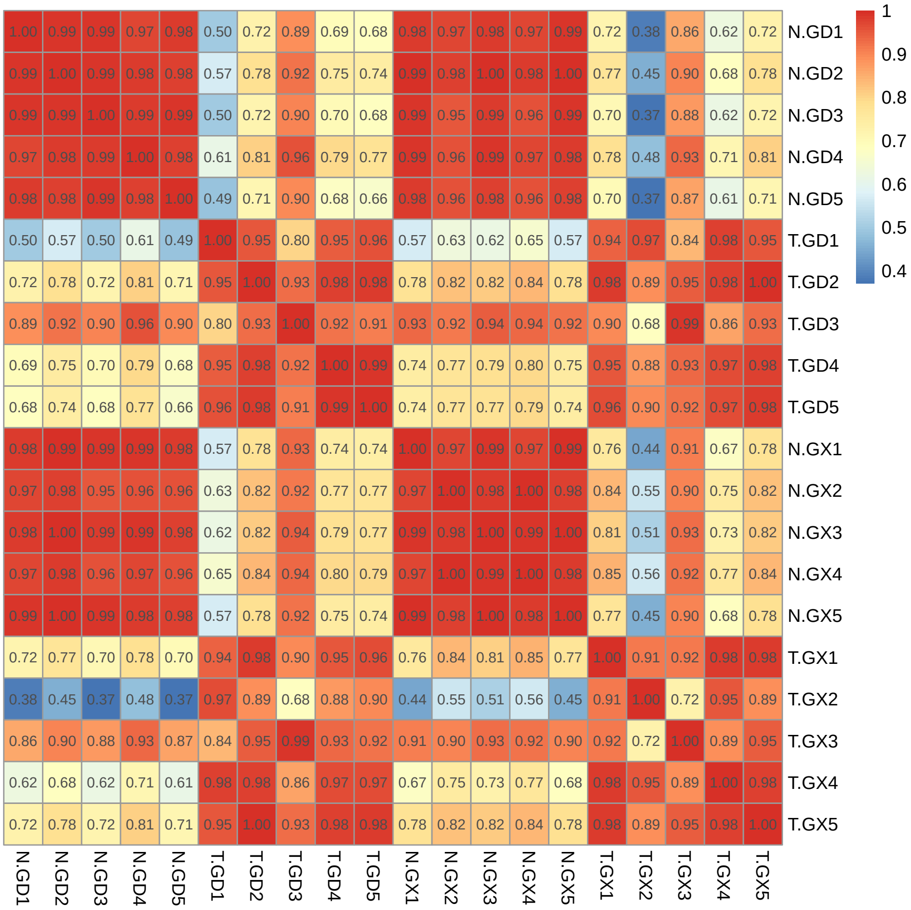


2.12.3 pheatmap heatmap

```
library(pheatmap)
dta=read.table("data/heatmap.csv",header=TRUE,row.names=1,sep=",")
pheatmap(dta)
```

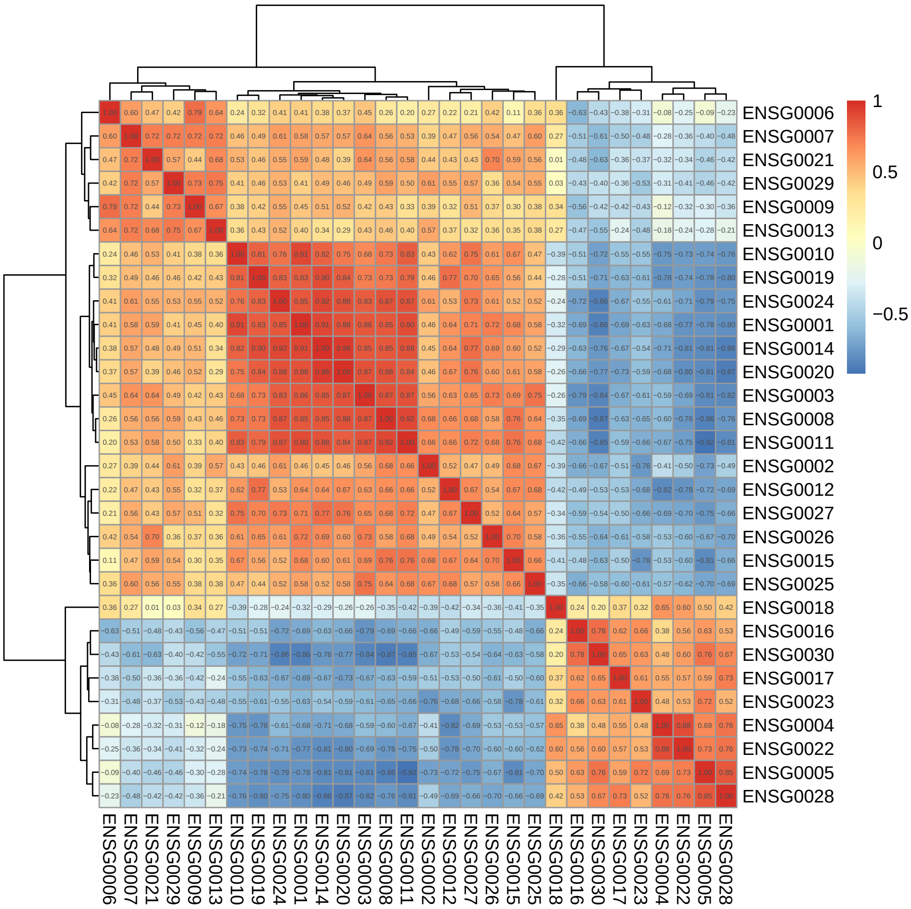


```
mat=cor(dta)
pheatmap(mat,cluster_rows=F,cluster_cols=F,display_numbers=TRUE)
```



	N.GD1	N.GD2	N.GD3	N.GD4	N.GD5	T.GD1	T.GD2	T.GD3	T.GD4	T.GD5	N.GX1	N.GX2	N.GX3	N.GX4	N.GX5	T.GX1	T.GX2	T.GX3	T.GX4	T.GX5
N.GD1	1.00	0.99	0.99	0.97	0.98	0.50	0.72	0.89	0.69	0.68	0.98	0.97	0.98	0.97	0.99	0.72	0.38	0.86	0.62	0.72
N.GD2	0.99	1.00	0.99	0.98	0.98	0.57	0.78	0.92	0.75	0.74	0.99	0.98	1.00	0.98	1.00	0.77	0.45	0.90	0.68	0.78
N.GD3	0.99	0.99	1.00	0.99	0.99	0.50	0.72	0.90	0.70	0.68	0.99	0.95	0.99	0.96	0.99	0.70	0.37	0.88	0.62	0.72
N.GD4	0.97	0.98	0.99	1.00	0.98	0.61	0.81	0.96	0.79	0.77	0.99	0.96	0.99	0.97	0.98	0.78	0.48	0.93	0.71	0.81
N.GD5	0.98	0.98	0.99	0.98	1.00	0.49	0.71	0.90	0.68	0.66	0.98	0.96	0.98	0.96	0.98	0.70	0.37	0.87	0.61	0.71
T.GD1	0.50	0.57	0.50	0.61	0.49	1.00	0.95	0.80	0.95	0.96	0.57	0.63	0.62	0.65	0.57	0.94	0.97	0.84	0.98	0.95
T.GD2	0.72	0.78	0.72	0.81	0.71	0.95	1.00	0.93	0.98	0.98	0.78	0.82	0.82	0.84	0.78	0.98	0.89	0.95	0.98	1.00
T.GD3	0.89	0.92	0.90	0.96	0.90	0.80	0.93	1.00	0.92	0.91	0.93	0.92	0.94	0.94	0.92	0.90	0.68	0.99	0.86	0.93
T.GD4	0.69	0.75	0.70	0.79	0.68	0.95	0.98	0.92	1.00	0.99	0.74	0.77	0.79	0.80	0.75	0.95	0.88	0.93	0.97	0.98
T.GD5	0.68	0.74	0.68	0.77	0.66	0.96	0.98	0.91	0.99	1.00	0.74	0.77	0.77	0.79	0.74	0.96	0.90	0.92	0.97	0.98
N.GX1	0.98	0.99	0.99	0.99	0.98	0.57	0.78	0.93	0.74	0.74	1.00	0.97	0.99	0.97	0.99	0.76	0.44	0.91	0.67	0.78
N.GX2	0.97	0.98	0.95	0.96	0.96	0.63	0.82	0.92	0.77	0.77	0.97	1.00	0.98	1.00	0.98	0.84	0.55	0.90	0.75	0.82
N.GX3	0.98	1.00	0.99	0.99	0.98	0.62	0.82	0.94	0.79	0.77	0.99	0.98	1.00	0.99	1.00	0.81	0.51	0.93	0.73	0.82
N.GX4	0.97	0.98	0.96	0.97	0.96	0.65	0.84	0.94	0.80	0.79	0.97	1.00	0.99	1.00	0.98	0.85	0.56	0.92	0.77	0.84
N.GX5	0.99	1.00	0.99	0.98	0.98	0.57	0.78	0.92	0.75	0.74	0.99	0.98	1.00	0.98	1.00	0.77	0.45	0.90	0.68	0.78
T.GX1	0.72	0.77	0.70	0.78	0.70	0.94	0.98	0.90	0.95	0.96	0.76	0.84	0.81	0.85	0.77	1.00	0.91	0.92	0.98	0.98
T.GX2	0.38	0.45	0.37	0.48	0.37	0.97	0.89	0.68	0.88	0.90	0.44	0.55	0.51	0.56	0.45	0.91	1.00	0.72	0.95	0.89
T.GX3	0.86	0.90	0.88	0.93	0.87	0.84	0.95	0.99	0.93	0.92	0.91	0.90	0.93	0.92	0.90	0.92	0.72	1.00	0.89	0.95
T.GX4	0.62	0.68	0.62	0.71	0.61	0.98	0.98	0.86	0.97	0.97	0.67	0.75	0.73	0.77	0.68	0.98	0.95	0.89	1.00	0.98
T.GX5	0.72	0.78	0.72	0.81	0.71	0.95	1.00	0.93	0.98	0.98	0.78	0.82	0.82	0.84	0.78	0.98	0.89	0.95	0.98	1.00

```r
dta=t(dta)
matrix=cor(dta)
pheatmap(matrix,cluster_rows=T,cluster_cols=T,display_numbers=TRUE,
         fontsize_number=4,number_format = "%.2f")
```


1
0.5
0
−0.5
ENSG0006
ENSG0007
ENSG0021
ENSG0029
ENSG0009
ENSG0013
ENSG0010
ENSG0019
ENSG0024
ENSG0001
ENSG0014
ENSG0020
ENSG0003
ENSG0008
ENSG0011
ENSG0002
ENSG0012
ENSG0027
ENSG0026
ENSG0015
ENSG0025
ENSG0018
ENSG0016
ENSG0030
ENSG0017
ENSG0023
ENSG0004
ENSG0022
ENSG0005
ENSG0028