

ggman

Veera M Rajagopal

12/5/2016

## Contents

ggman: R package to create Manhattan plot using ggplot.

## Create a basic Manhattan plot

A toy GWAS dataset is made available along with the package. Let's look at the dimensions, head and tail of the dataset.

```
library(ggman)
```

```
dim(toy.gwas)
```

```
## [1] 21751 7
```

```
##          snp chrom      bp pvalue      beta      or  gene
## 1 rs1_3120      1 2025837 0.9142 0.00796817 1.008 GENE1
## 2 rs1_4135      1 2528866 0.5928 0.07139000 1.074 GENE1
## 3 rs1_4125      1 2878321 0.7542 0.02176149 1.022 GENE1
## 4 rs1_4130      1 3192870 0.1324 0.10795714 1.114 GENE1
## 5 rs1_1590      1 3292731 0.5687 0.03633193 1.037 GENE1
## 6 rs1_185       1 3576288 0.6959 0.02566775 1.026 GENE1
```

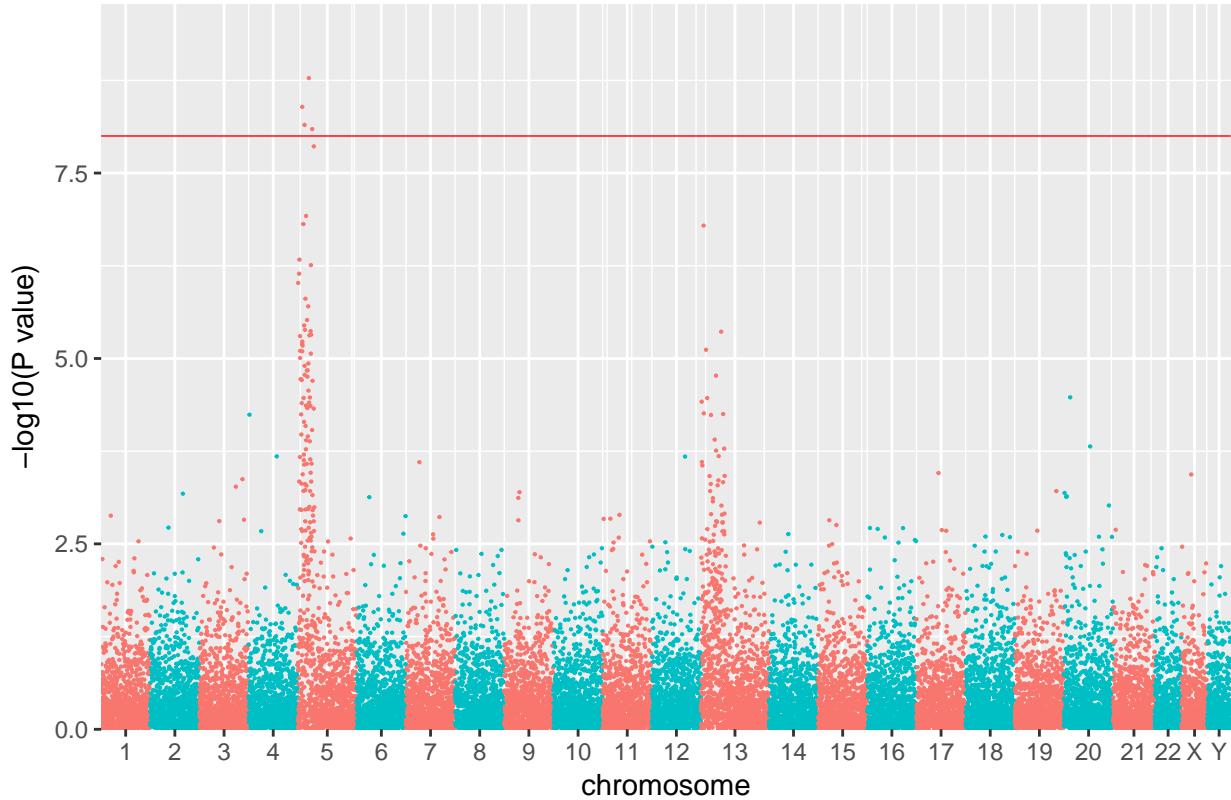
```
tail(toy.gwas)
```

```
##          snp chrom      bp pvalue      beta      or     gene
## 21746 rs22_838      Y 48821171 1.0000 0.000000000 1.0000 GENE544
## 21747 rs21_368      Y 48898458 0.7871 0.017839918 1.0180 GENE544
## 21748 rs21_373      Y 48906877 0.1455 -0.121151202 0.8859 GENE544
## 21749 rs22_428      Y 48946140 0.2218 0.088010877 1.0920 GENE544
## 21750 rs21_818      Y 49030475 0.9449 -0.004811557 0.9952 GENE544
## 21751 rs21_688      Y 49350919 0.4664 -0.053084371 0.9483 GENE544
```

To create a Manhattan plot, only the first 4 columns (chrom,snp,bp,pvalue) are required. Specific preformatting of the column classes is not required. The chromosome identifiers can be either numbers (1,2,3..) or strings("Chr1","Chr2"..).

```
ggman(toy.gwas, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "pvalue")
```

## Manhattan Plot

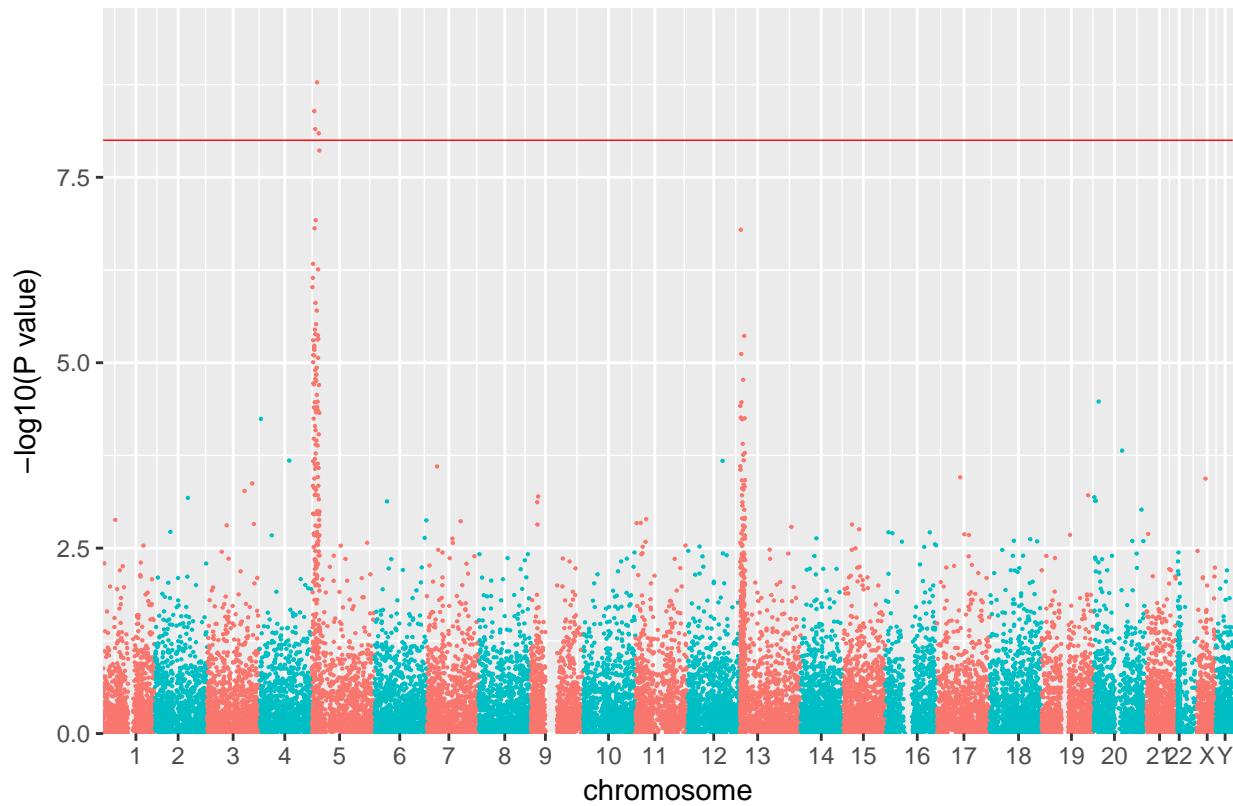


## Use relative positioning

By enabling the relative positioning, the base pair positions will be scaled in proportion to the real genome positions. Hence, the gaps with no SNPs can be visualized. Be default this is not enabled. To use the relative positions, use the option `relative.positions = TRUE`

```
ggman(toy.gwas, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "pvalue",
       relative.positions = TRUE)
```

## Manhattan Plot



### Add labels

Specific set of points in the plot can be annotated by providing a data.frame with **only the SNPs those need to be labelled**. Let's take a subset of the main data frame `toy.gwas`.

```
#subset only the SNPs with -log10(pvalue) > 8
toy.gwas.sig <- toy.gwas[-log10(toy.gwas$pvalue)>8,]

# dimensions
dim(toy.gwas.sig)
```

```
## [1] 4 7
```

```
#head
head(toy.gwas.sig)
```

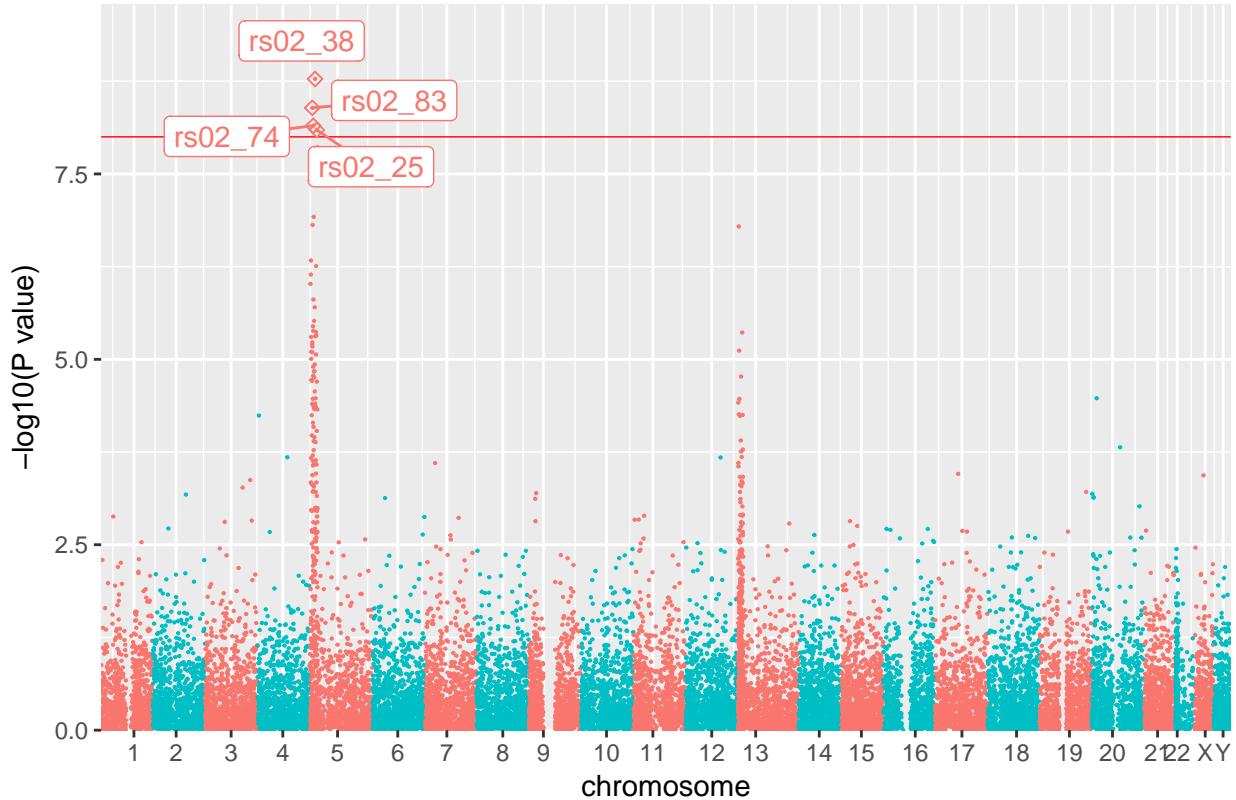
```
##          snp chrom      bp   pvalue      beta      or     gene
## 3873 rs02_83      5 6887869 4.057e-09 0.5988365 1.820  GENE97
## 3918 rs02_74      5 9657902 7.084e-09 0.6119371 1.844  GENE98
## 4000 rs02_38      5 14907898 1.658e-09 0.61195006 1.858 GENE100
## 4065 rs02_25      5 19843813 8.075e-09 0.5641768 1.758 GENE102
```

The main layer of Manhattan plot should be saved in a variable and provided subsequently to `ggmanLabel` function. The name of the columns with snps and labels has to be supplied. In this case, we will label with SNP identifiers.

```
## save the main layer in a variable
p1 <- ggman(toy.gwas,.snp = "snp",.bp = "bp",.chrom = "chrom",.pvalue = "pvalue",
             relative.positions = TRUE)

##add label
ggmanLabel(p1, labelDfm = toy.gwas.sig, .snp = "snp", .label = "snp")
```

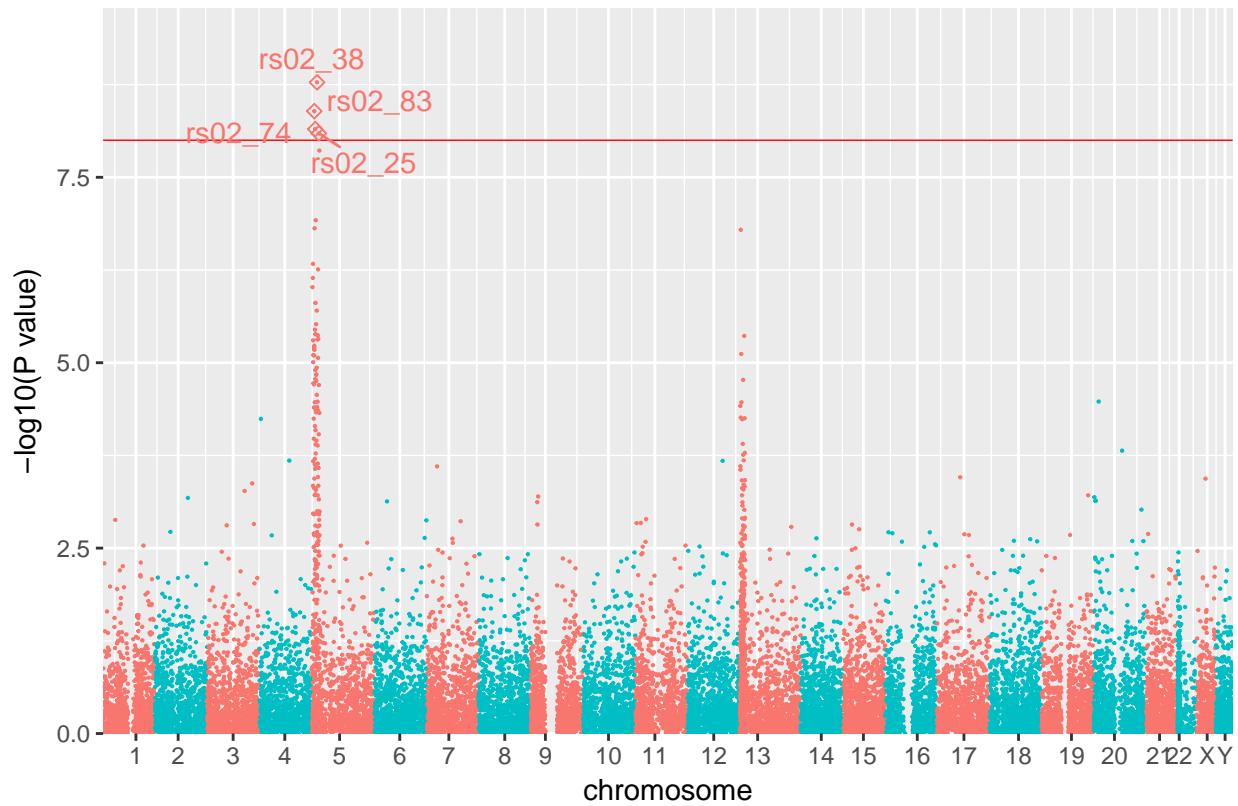
## Manhattan Plot



Annotations can be just text instead of labels. Use the `type =` argument.

```
#add text
ggmanLabel(p1, labelDfm = toy.gwas.sig, .snp = "snp", .label = "snp", type = "text")
```

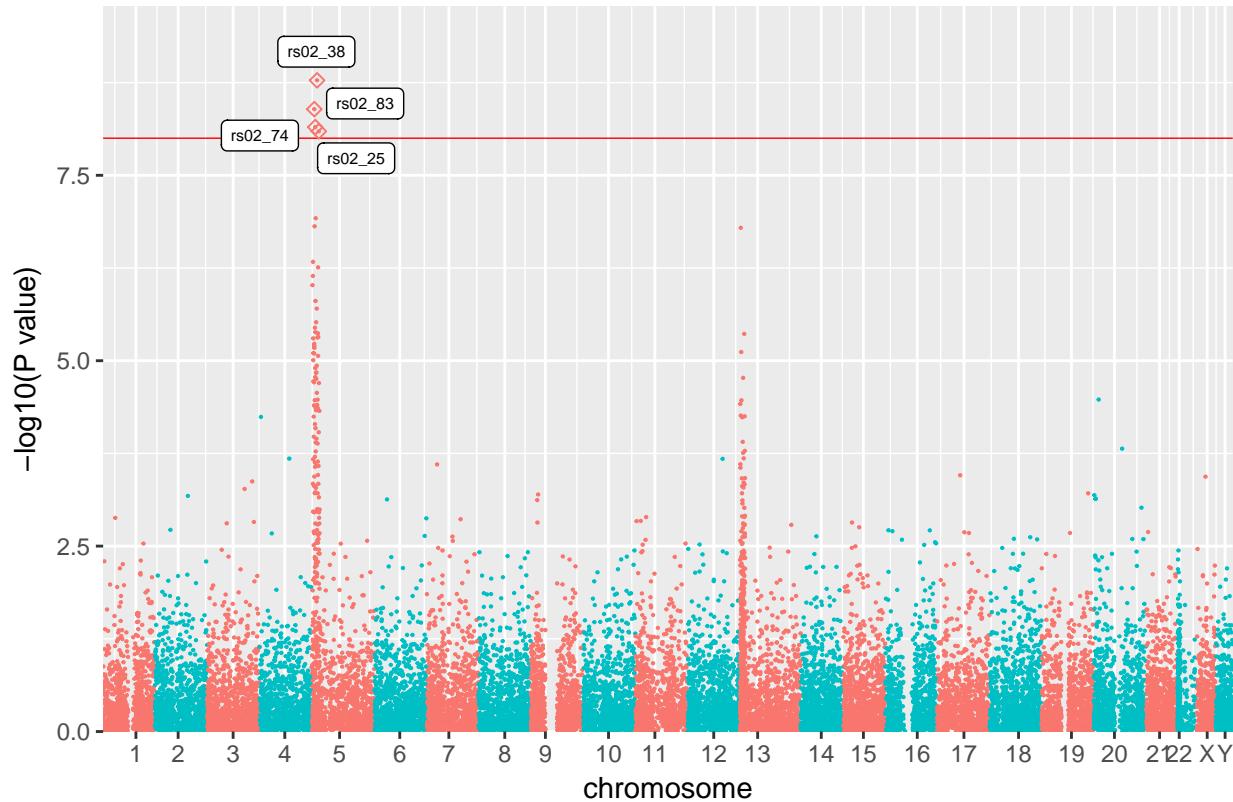
## Manhattan Plot



The R package `ggrepel` is used for annotations. All the arguments that are applicable to `geom_text_repel` and `geom_label_repel` can be passed on to `ggmanLabel`. Lets change the size and colour of the labels.

```
ggmanLabel(p1, labelDfm = toy.gwas.sig, snp = "snp", label = "snp", colour = "black", size = 2)
```

## Manhattan Plot



Caution: providing the whole main data frame as `labelDfm` will fill the entire plot with text or might crash the R if the data frame is too big

### Highlight a single group of points

The function `ggmanHighlight` can be used to highlight a single group of points. By default, while highlighting specific points, the main layer of Manhattan plot is greyed out. We need to supply a vector object with SNP names to highlight. The example file `toy.highlights` comes along with package.

```
class(toy.highlights)

## [1] "character"

length(toy.highlights)

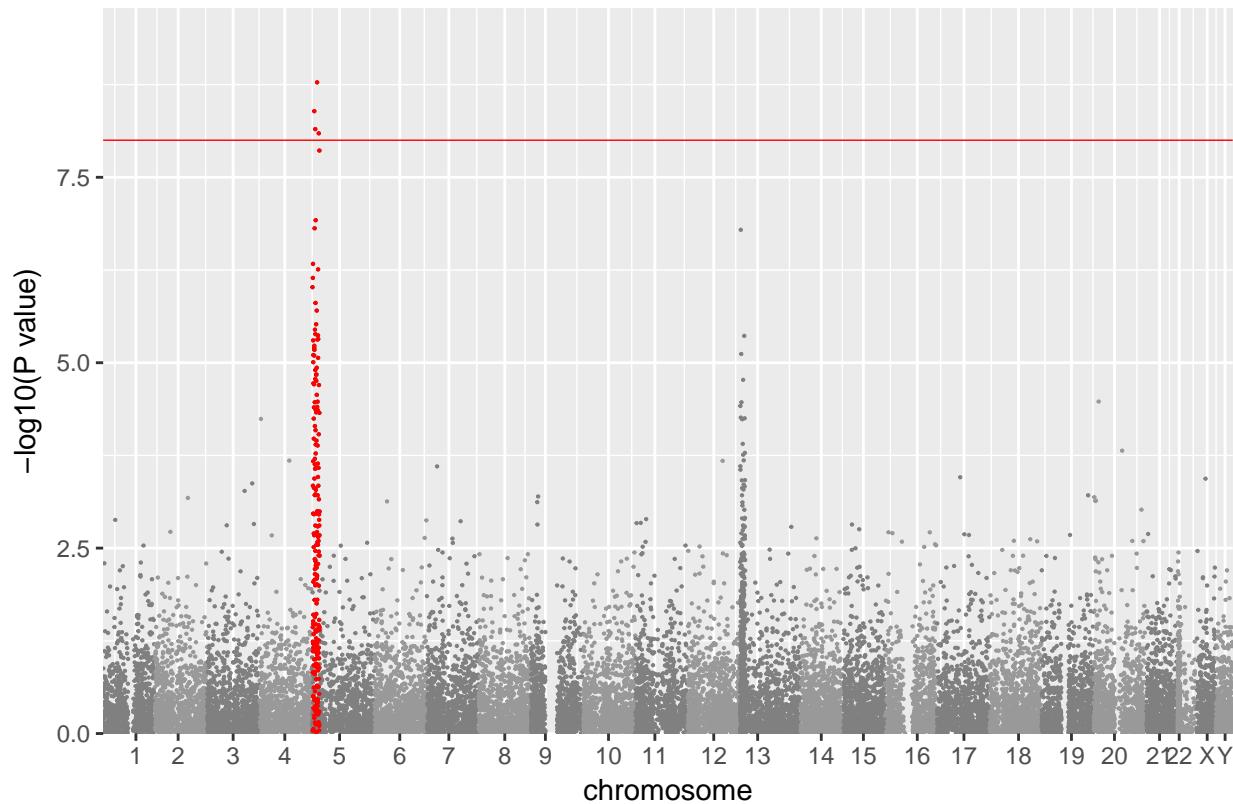
## [1] 209

head(toy.highlights)

## [1] "rs02_2"  "rs02_7"  "rs02_12" "rs02_17" "rs02_22" "rs02_27"

ggmanHighlight(p1, highlight = toy.highlights)
```

## Manhattan Plot



### Highlight multiple groups of points with a legend

The function `ggmanHighlightGroup` can be used to highlight multiple groups of points and a legend can be added. Let's look at the example file `toy.highlights.group`.

```
class(toy.highlights.group)

## [1] "data.table" "data.frame"

dim(toy.highlights.group)

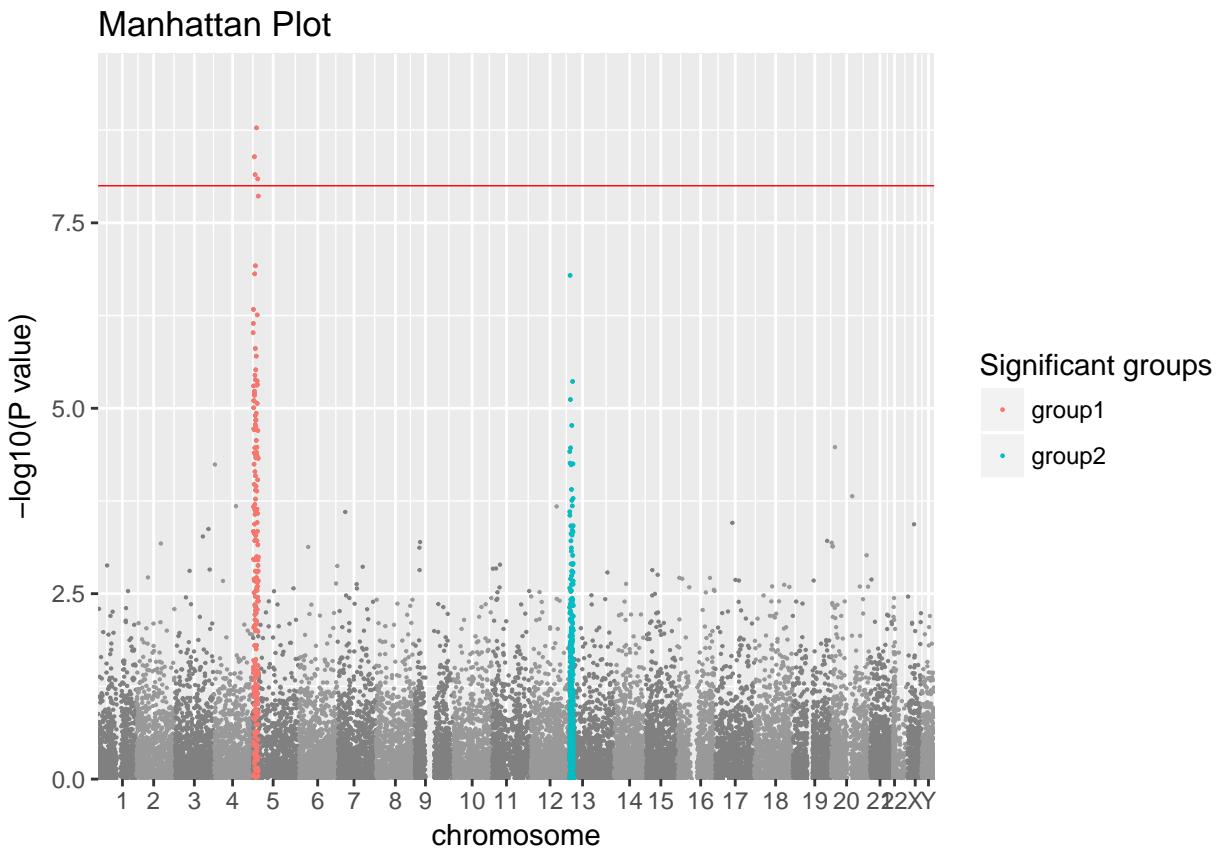
## [1] 609 8

head(toy.highlights.group)

##   chrom      snp      bp  pvalue      beta      or      gene group
## 1    13 rs06_2_M 24226825 0.0794900 0.18148788 1.199 GENE2180 group2
## 2    13 rs06_7_M 23664350 0.0005127 0.36325326 1.438 GENE2180 group2
## 3    13 rs06_12_M 19042292 0.2111000 0.13627762 1.146 GENE2180 group2
## 4    13 rs06_17_M 24586858 0.0193900 0.27459683 1.316 GENE2180 group2
## 5    13 rs06_22_M 20332216 0.4479000 0.09621886 1.101 GENE2180 group2
## 6    13 rs06_27_M 24855237 1.0000000 0.00000000 1.000 GENE2180 group2
```

Unlike `ggmanHighlight`, the function `ggmanHighlightGroup` requires `data.frame` as an input. One of the column names should be supplied as a grouping variable. The size of the highlighted points can be changed with `size` argument. The legend title can be specified with `legend.title` argument.

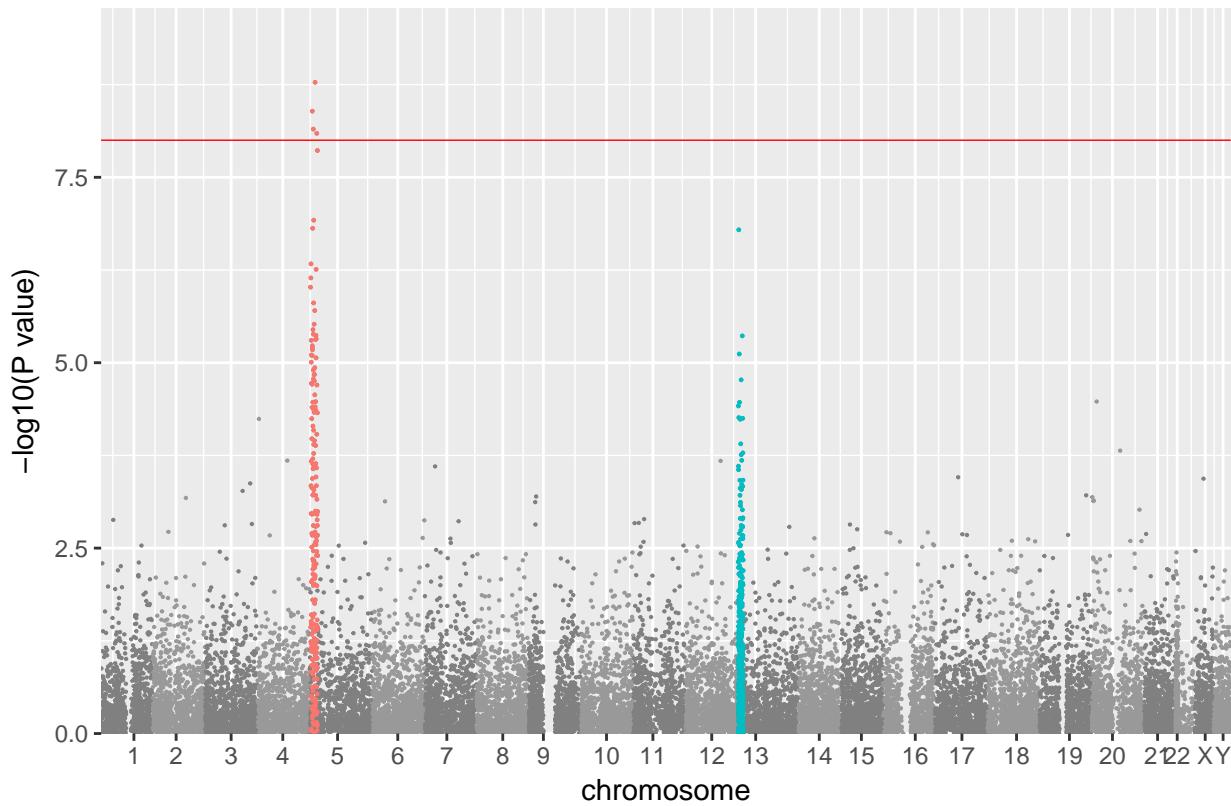
```
ggmanHighlightGroup(p1, highlightDfm = toy.highlights.group, snp = "snp", group = "group",
                     size = 0.5, legend.title = "Significant groups")
```



It is also possible to remove the legend using `legend.remove` argument.

```
ggmanHighlightGroup(p1, highlightDfm = toy.highlights.group, snp = "snp", group = "group",
                     size = 0.5, legend.remove = TRUE)
```

## Manhattan Plot



### Add SNP clumps

In a typical genome wide association study, it is a standard practice to display SNPs in linkage disequilibrium with the index SNP as clumps. The plink software has clumping procedure, which outputs clump file with `.clumped` extension.

Adding clumps to Manhattan plot involves four steps.

1. Perform clumping using Plink `--clump` function
2. Read the output file in to a data.frame. Suppose the name of the output file is `plink.clumped` then

```
gwas.clump <- read.table("plink.clumped", header = TRUE)
```

Here, the example file `toy.clumped` is a data.frame, which is created by reading the `plink.clumped` file and subsetting only the columns 'SNP' and 'SP2'.

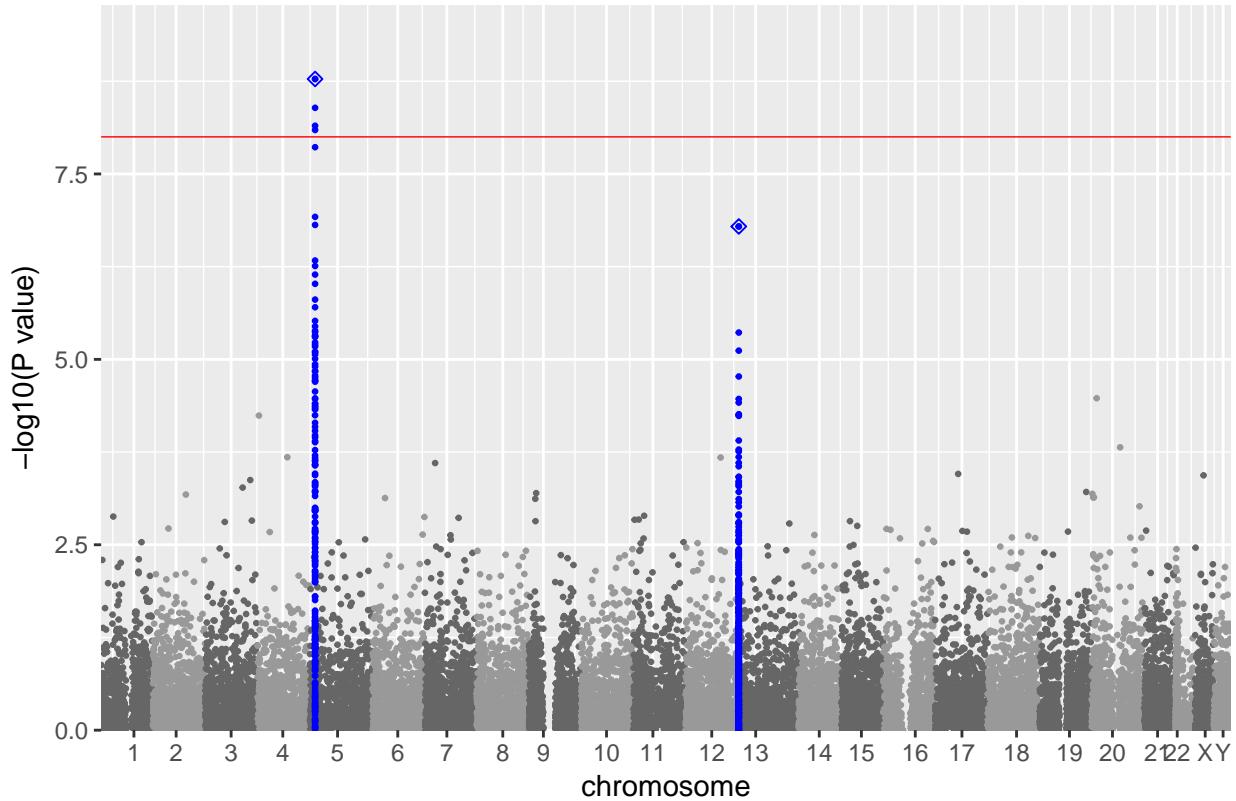
3. Convert the `toy.clumped` data.frame to a `ggclumps` object using the `ggmanClumps` function. The arguments `index.snp.column` and `clumps.column` are mandatory. The name of the column containing index SNPs ('SNP') should be passed to argument `index.snp.column` and the name of the column containing the clumps should be passed to argument `clumps.column`.

```
toy.clumps <- ggmanClumps(toy.clumped, index.snp.column = "SNP", clumps.column = "SP2")
```

4. Pass the `toy.clumped` object to `clumps=` argument of `ggman` function.

```
ggman(toy.gwas, clumps = toy.clumps, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "pvalue",
       relative.positions = TRUE, pointSize = 0.5)
```

Manhattan Plot

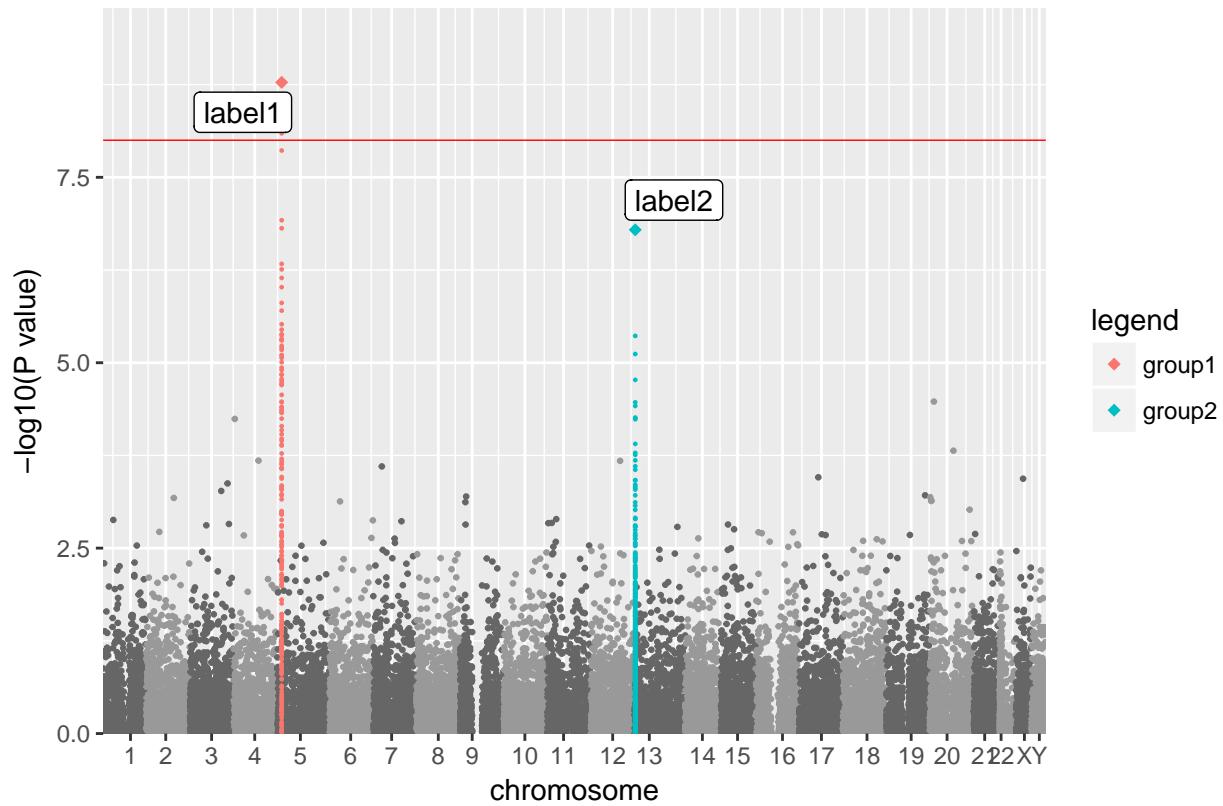


#### highlight and label the clumps

The clumps can be grouped using a grouping variable and the index SNPs can be labelled with user preferred labels. All you need to do is to add additional columns in the plink.clumped file and specify them in the ggmanClumps function. Here in the example toy.clumped file, there are 2 extra columns with names 'group' and 'label'.

```
toy.clumps <- ggmanClumps(toy.clumped, index.snp.column = "SNP", clumps.column = "SP2",
                           group.column = "group", label.column = "label")
ggman(toy.gwas, clumps = toy.clumps, snp = "snp", bp = "bp", chrom = "chrom",
       pvalue = "pvalue", relative.positions = TRUE, pointSize = 0.5)
```

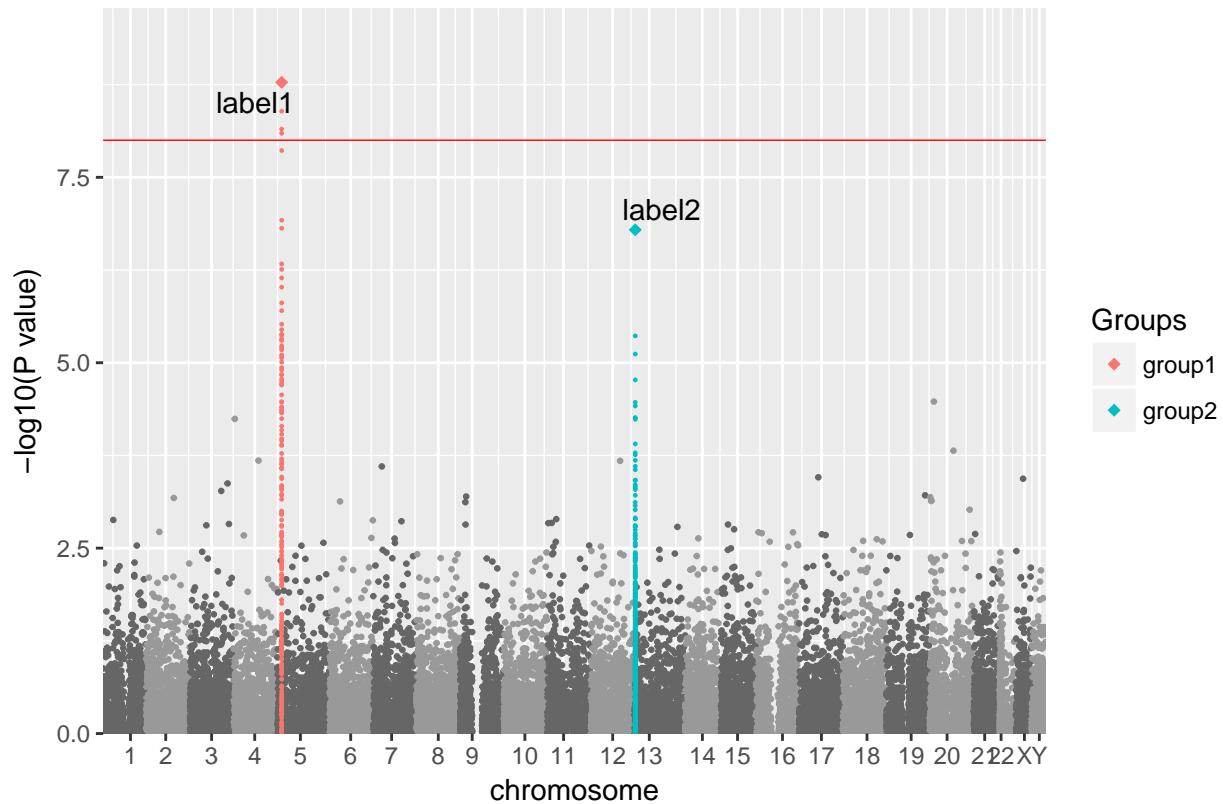
## Manhattan Plot



Use `legend.title` to change the legend title. If you prefer plain text without box for labels, use `clumps.label.type = 'text'`

```
ggman(toy.gwas,clumps = toy.clumps, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "pvalue",
       relative.positions = TRUE, pointSize = 0.5, legend.title = "Groups", clumps.label.type = 'text')
```

## Manhattan Plot

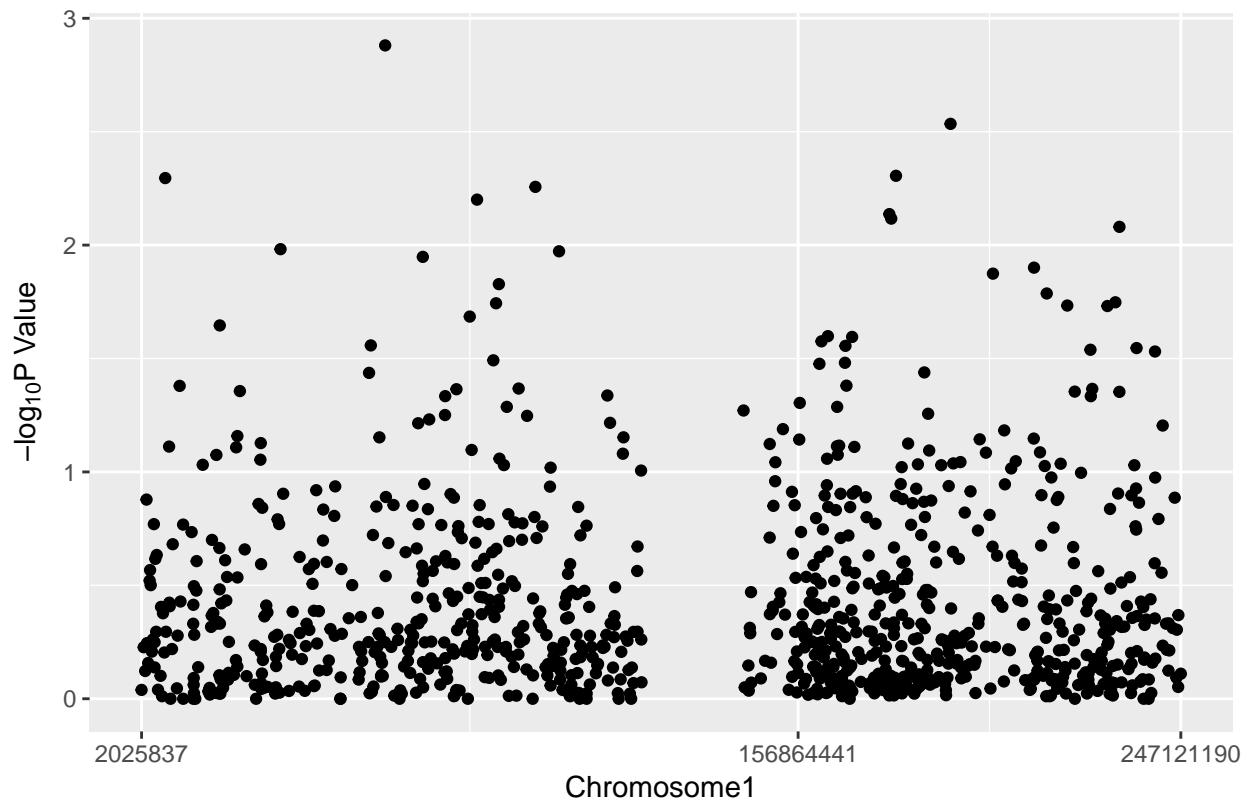


### Zoom in to a specific chromosome

The function `ggmanZoom` can be used to create regional association plot. Plotting a single chromosome is very simple.

```
ggmanZoom(p1, chromosome = 1)
```

## Regional association plot

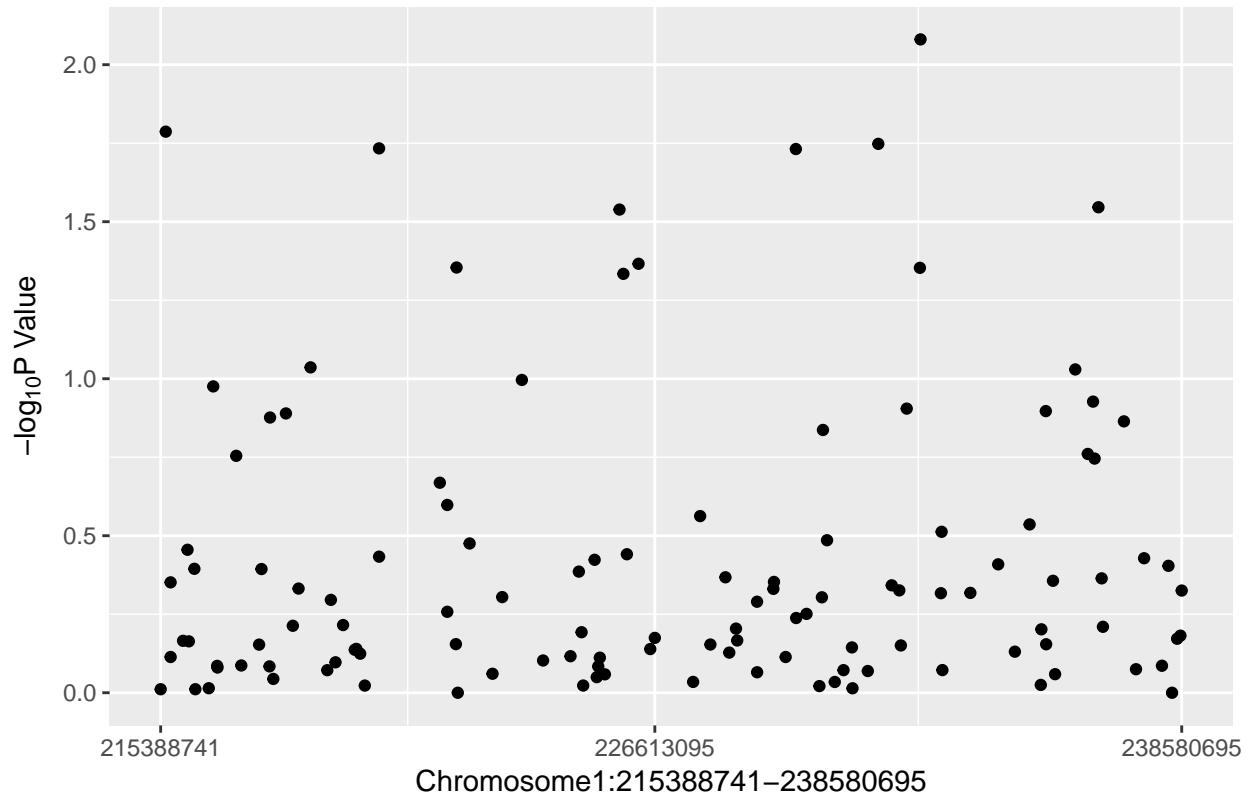


### Zoom in to a specific region of a chromosome

To plot a specific region, the starting and the ending basepair positions have to be specified. Let's zoom in to the chromosome 1 region containing genes: GENE21, GENE22 and GENE23.

```
ggmanZoom(p1, chromosome = 1, start.position = 215388741, end.position = 238580695)
```

## Regional association plot

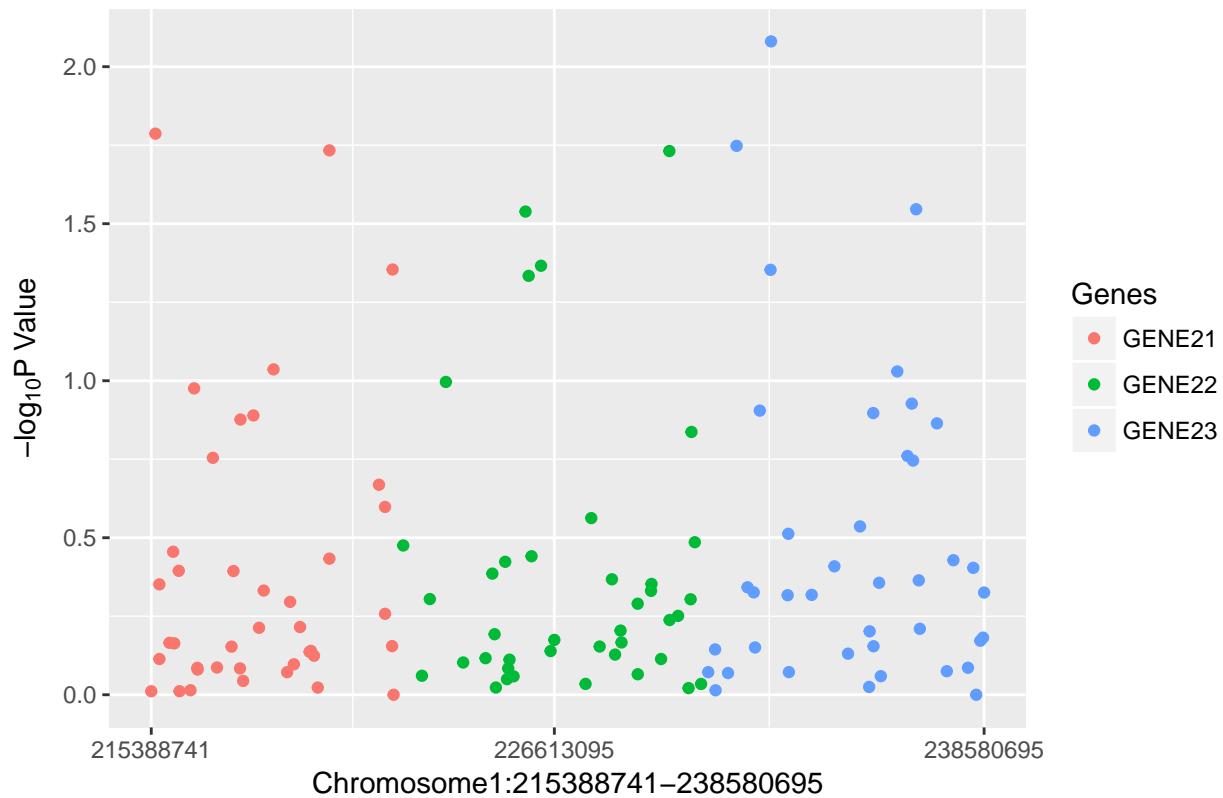


### Highlight points in the zoomed region

It's also possible to highlight using specific grouping variable. Here we have a column named `gene` in the main data frame `toy.gwas` that was used to construct the main layer `p1`.

```
ggmanZoom(p1, chromosome = 1, start.position = 215388741, end.position = 238580695,  
highlight.group = "gene", legend.title = "Genes")
```

## Regional association plot

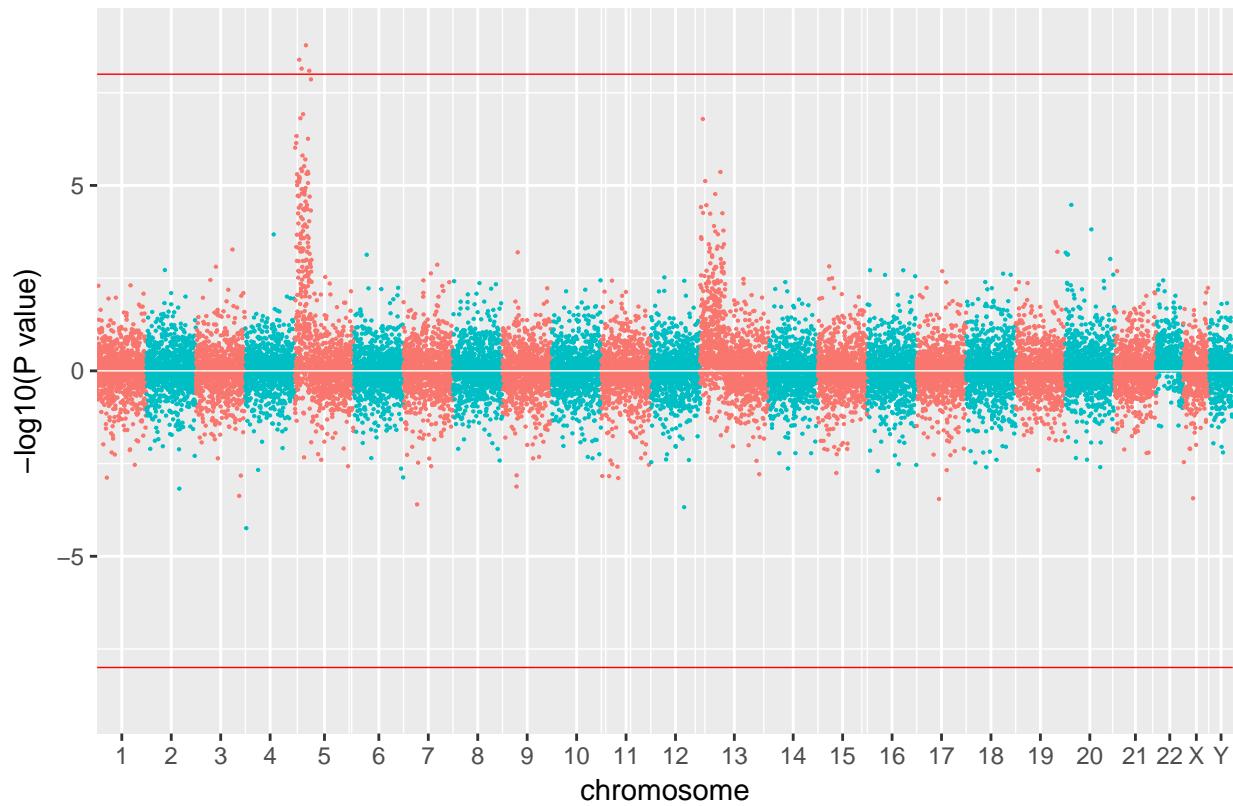


## Create an inverted Manhattan plot

An inverted Manhattan plot can be created by inverting the direction of p values of variants with negative beta values (or odds ratio < 1). Set the argument `invert` to `TRUE` to get an inverted Manhattan plot. If `invert=TRUE`, then `invert.method` and `invert.var` should be specified. The `invert.method` can be either `or` or `beta`. The `invert.var` is the name of the column containing the beta or odds ratio according to the value passed to `invert.method`

```
ggman(toy.gwas, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "pvalue", invert = TRUE,  
      invert.method = 'or', invert.var = "or")
```

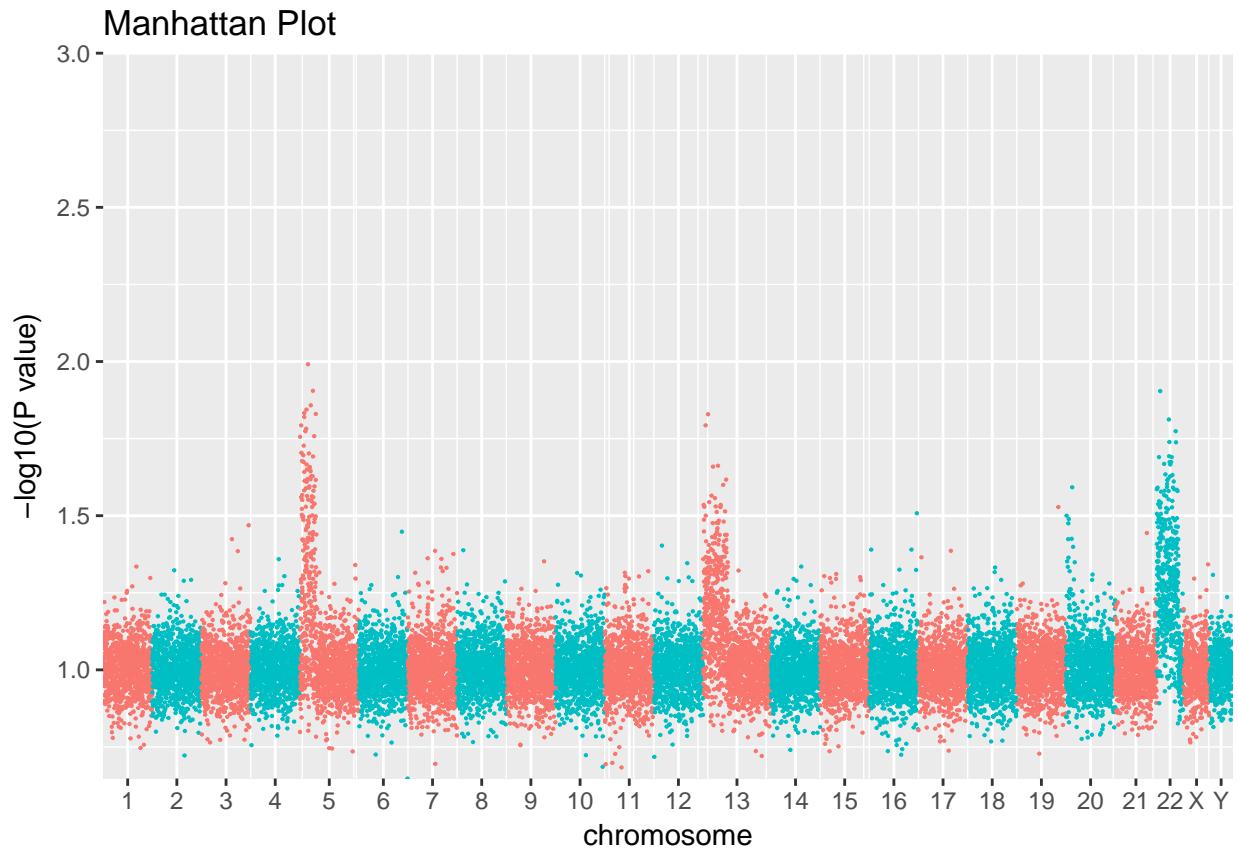
## Manhattan Plot



## Plot Odds ratio

```
ggman(toy.gwas, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "or",
logTransform = FALSE, ymax = 3)
```

```
## Warning: Removed 21751 rows containing missing values (geom_hline).
```



Plot beta

```
ggman(toy.gwas, snp = "snp", bp = "bp", chrom = "chrom", pvalue = "beta",
      logTransform = FALSE, ymin = -2, ymax = 2)
```

```
## Warning: Removed 21751 rows containing missing values (geom_hline).
```

