Overview of shinyChromosomeR

Yiming Yu and Wen Yao

2019-06-04

Contents

1 Introduction

shinyChromosome is an Shiny application for interactive creation of non-circular plots of whole genomes within the web browser. shinyChromosome is deployed at http://150.109.59.144:3838/shinyChromosome/, http://shinychromosome.ncpgr.cn/ and https://yimingyu.shinyapps.io/shinychromosome/, for online use.

shinyChromosomeR wraps the core script of shinyChromosome as an R package, allowing the creation of non-circular whole genome diagram from the R command line.

In this vignette, we will rely on several simple, illustrative example datasets to demonstrate the usage of **shinyChromosomeR**.

To use the **shinyChromosomeR** package, we need to load it into R first.

```r
library(shinyChromosomeR)
```

2 Creation of single-genome plot using shinyChromosomeR

2.1 Essential steps to create a non-circular single genome plot

To create a non-circular single genome plot, we need a dataset to define the genome used in the single genome plot and other datasets to be displayed along all the chromosomes of the genome.

2.1.1 Read in the genome dataset

The genome dataset is compulsory and defines the frame of a non-circular plot. The genome dataset is basically a text file with 2 columns. The 1st column is the chromosome ID. The 2nd column is the chromosome length. The detailed format of the genome data is illustrated in the 'Input data format' menu (Under the 'Help' menu) of the shinyChromosome application (http://150.109.59.144:3838/shinyChromosome/).

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
dim(data.chr)
```

```
## [1] 12  2
```

```
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

2.1.2 Read in other input datasets to be displayed along all chromosomes of the input genome

One or more datasets could be then read into R, which would be displayed along all chromosomes of the genome dataset in Step 2.1.1. These datasets can be used to create different types of plot, including 'point', 'line', 'bar', 'rect_gradual', 'rect_discrete', 'heatmap_gradual', 'heatmap_discrete', 'text', 'segment', 'vertical_line', 'horizontal_line' and 'ideogram'. Please check the 'Input data format' menu (Under the 'Help' menu) of the shinyChromosome application (http://150.109.59.144:3838/shinyChromosome/) for more details.

Please be noted that each dataset should be read into R as a data frame. Then each of the data frames should be stored as an element of a list. Please check the following example.

```
data.track.file <- system.file("examples/single_genome/point.txt",
                               package="shinyChromosomeR")
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 10000     4
```

```
head(data.track[[1]], 2)
```

```
##   chr position value color
## 1   1   202360 0.315     a
## 2   1   213775 1.113     a
```

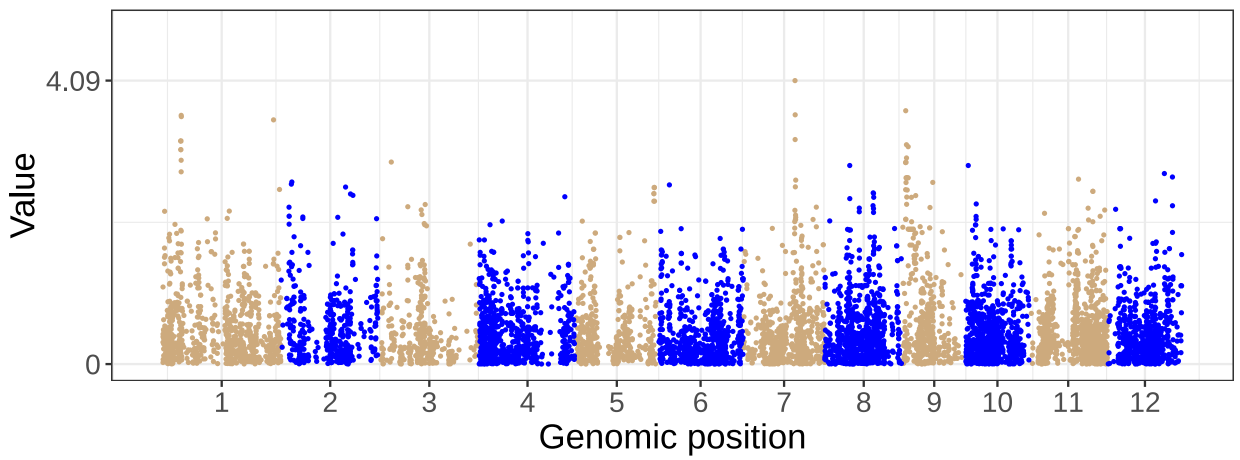

Figure 1: Plot point using single_genome_plot.

2.1.3 Make the plot using the `single_genome_plot` function

After all the input datasets has been prepared and read into R, we can call the `single_genome_plot` function to make the plot. By default, different datasets in step 2.1.2 would be displayed in different tracks. We can set the track of each dataset using the `layer_index` parameter. We also need to set the plot type for each dataset in step 2.1.2.

```
single_genome_plot(data.chr=data.chr, data.track=data.track, plot_type="point")
```

2.2 Create different types of single genome plot using shinyChromosome

2.2.1 Plot line

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- system.file("examples/single_genome/line.txt",
                                package="shinyChromosomeR")
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/line.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 1619    3
```

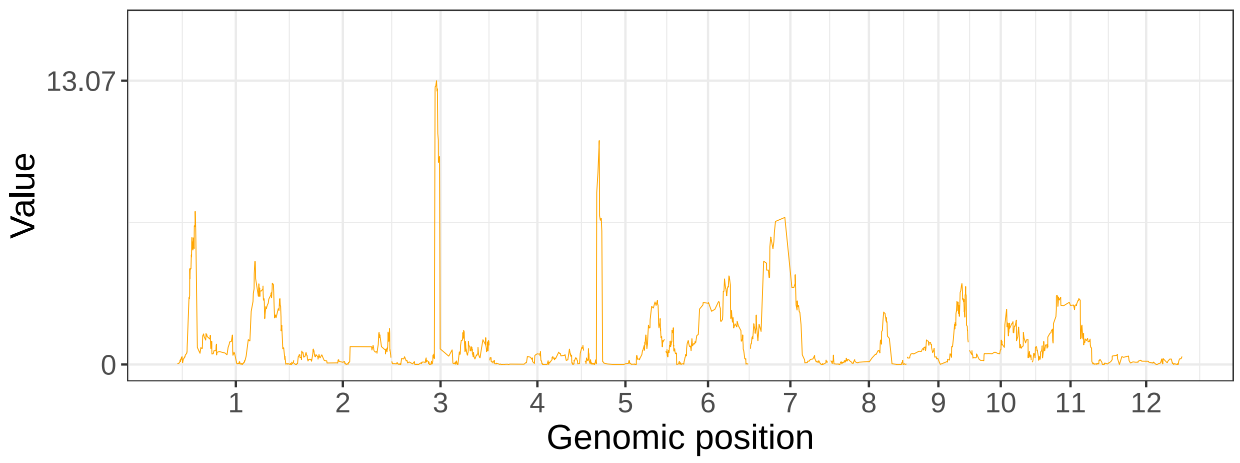

Figure 2: Plot line using single_genome_plot.

```
head(data.track[[1]], 2)
```

```
##   chr position  value
## 1   1        0 0.0428
## 2   1   565000 0.0522
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track, plot_type="line")
```

2.2.2 Plot point + line

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
## 3   3 36406689
## 4   4 35278225
## 5   5 29894789
## 6   6 31246789
```

```
data.track.file <- c(system.file("examples/single_genome/point.txt",
                                 package="shinyChromosomeR"),
                     system.file("examples/single_genome/line.txt",
                                 package="shinyChromosomeR"))
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/point.txt"
## [2] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/line.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

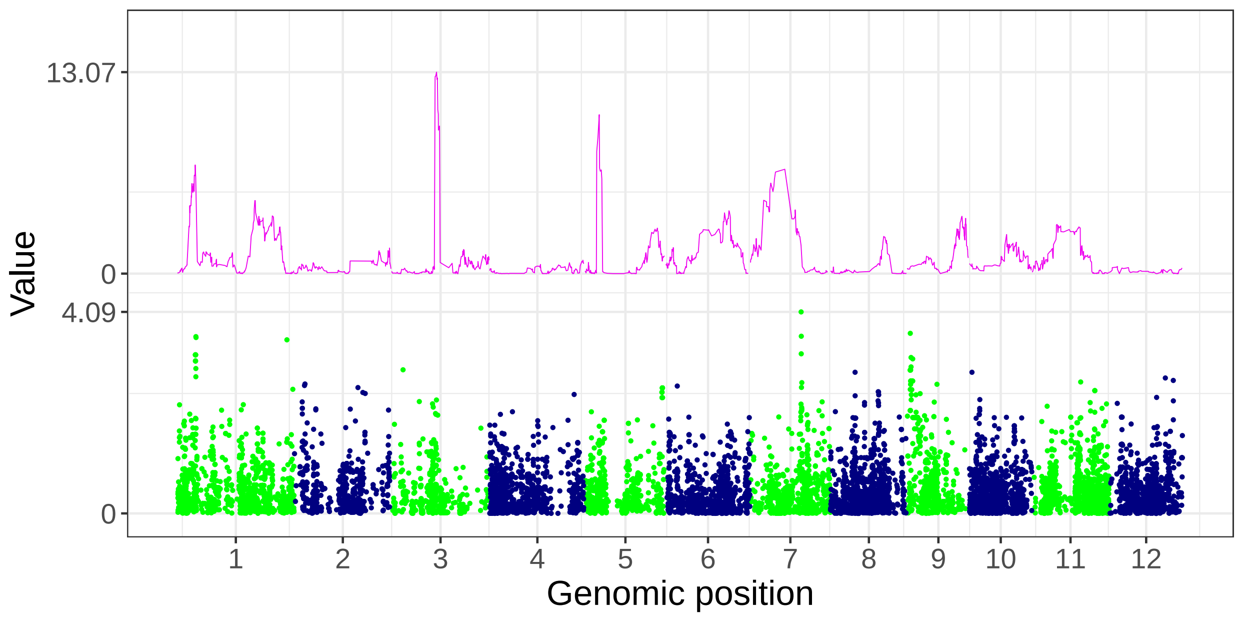

Figure 3: Plot point + line using single_genome_plot.

```
## [1] 10000     4
```

```
head(data.track[[1]], 2)
```

```
##   chr position value color
## 1   1   202360 0.315     a
## 2   1   213775 1.113     a
```

```
dim(data.track[[2]])
```

```
## [1] 1619    3
```

```
head(data.track[[2]], 2)
```

```
##   chr position  value
## 1   1        0 0.0428
## 2   1   565000 0.0522
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track, plot_type=c("point", "line"))
```

2.2.3 Plot bar

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- system.file("examples/single_genome/bar.txt",
                               package="shinyChromosomeR")
```

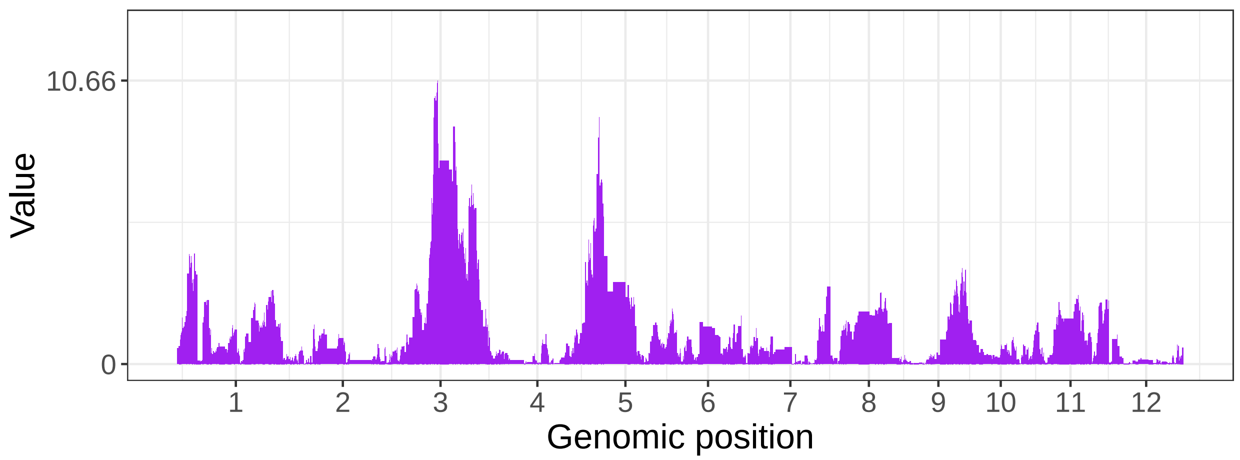

Figure 4: Plot bar using single_genome_plot.

```
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/bar.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 1619    4
```

```
head(data.track[[1]], 2)
```

```
##   Chr  start   stop  value
## 1   1      0 565000 0.5923
## 2   1 565000 599000 0.6701
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track, plot_type="bar")
```

2.2.4 Plot heatmap_discrete

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- system.file("examples/single_genome/heatmap_discrete.txt",
                               package="shinyChromosomeR")
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/heatmap_discrete.txt
```

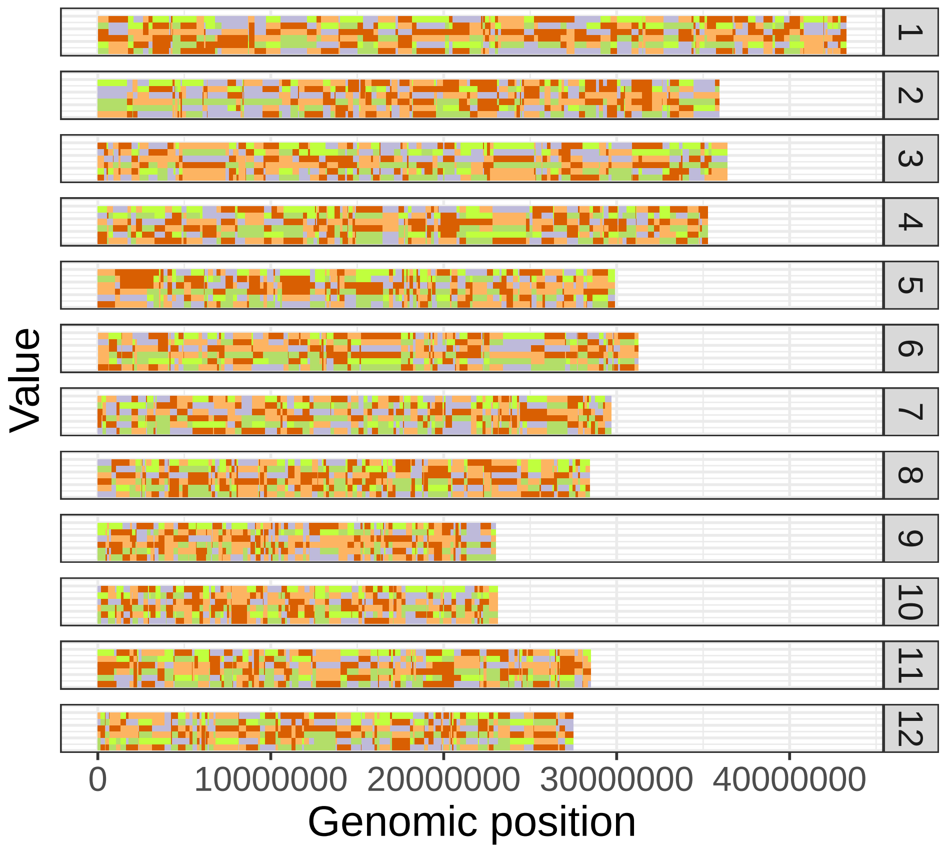

Figure 5: Plot heatmap_discrete using single_genome_plot.

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 1200    9
```

```
head(data.track[[1]], 2)
```

```
##   chr   start    stop val1 val2 val3 val4 val5 val6
## 1   1       0  631164    a    e    c    c    a    b
## 2   1  631165 1749192    b    b    c    d    d    c
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track,
                   plot_type="heatmap_discrete", chr_plotype=2,
                   margin_layer=0.01)
```

2.2.5 Plot heatmap_gradual

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
```

```
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- system.file("examples/single_genome/heatmap_gradual.txt",
                                package="shinyChromosomeR")
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/heatmap_gradual.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 3729    7
```

```
head(data.track[[1]], 2)
```

```
##   chr    start     stop TE NTE TR NR
## 1   1        1   100000  4  29 17 45
## 2   1 10000001 10100000  9  14 20 28
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track,
                   plot_type="heatmap_gradual", chr_plotype=2,
                   margin_layer=0.01)
```

2.2.6 Plot ideogram

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- system.file("examples/single_genome/ideogram.txt",
                                package="shinyChromosomeR")
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/ideogram.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 573   5
```

```
head(data.track[[1]], 2)
```

```
##   chr  start    end value1 value2
## 1   1      1 399271 p36.33   gneg
```

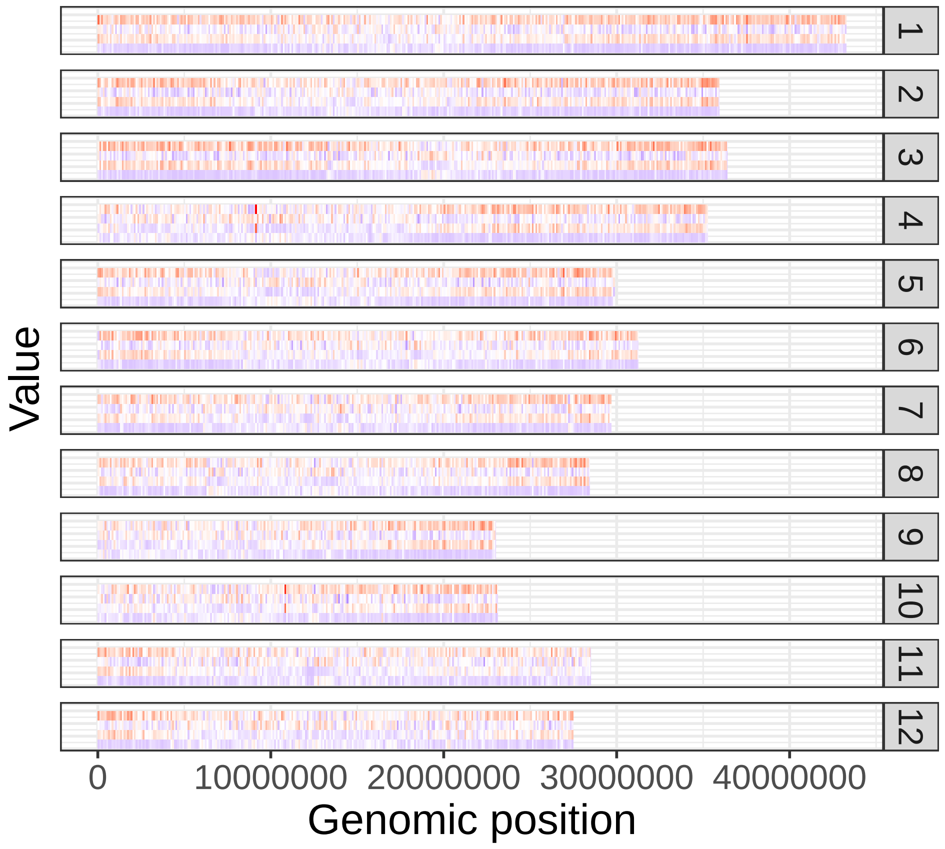

Figure 6: Plot heatmap_gradual using single_genome_plot.

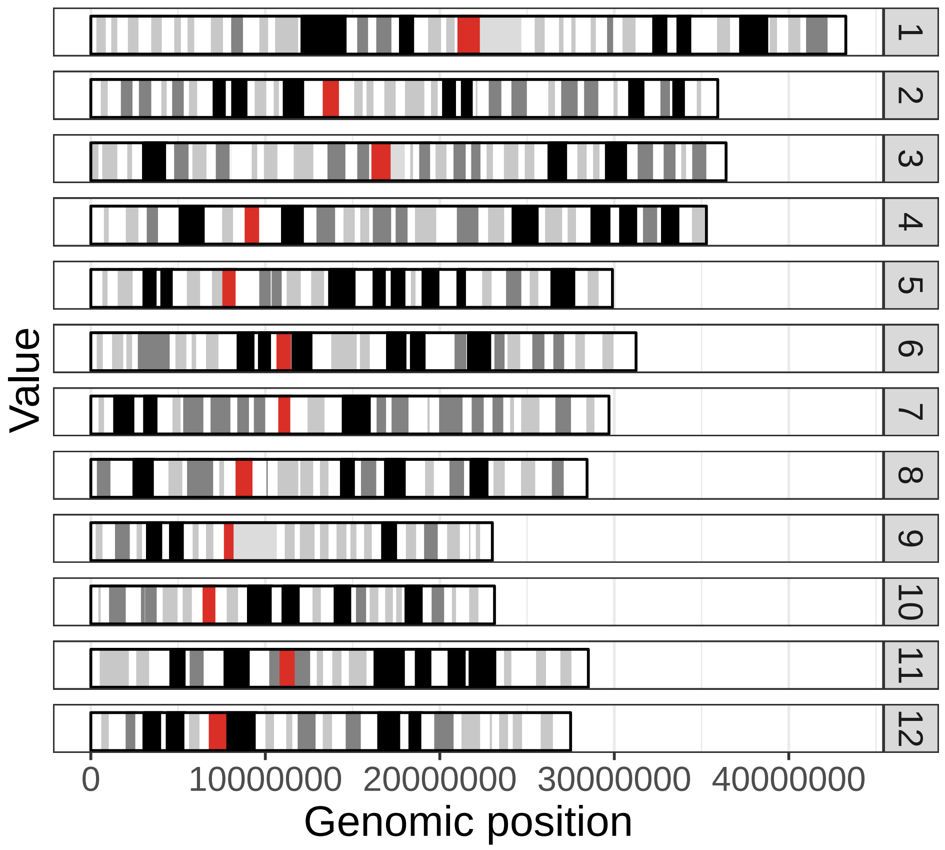

Figure 7: Plot ideogram using single_genome_plot.

```
## 2   1 399271 937418 p36.32 gpos25
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track,
                   plot_type="ideogram", chr_plotype=2,
                   margin_layer=0.01)
```

2.2.7 Plot bar + vertical_line

The user can tune the appearance of the generated plot by setting the values of diverse parameters of the `single_genome_plot` function.

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
                       header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```
data.track.file <- c(system.file("examples/single_genome/bar.txt",
                                 package="shinyChromosomeR"),
                     system.file("examples/single_genome/vertical_line.txt",
                                 package="shinyChromosomeR"))
```

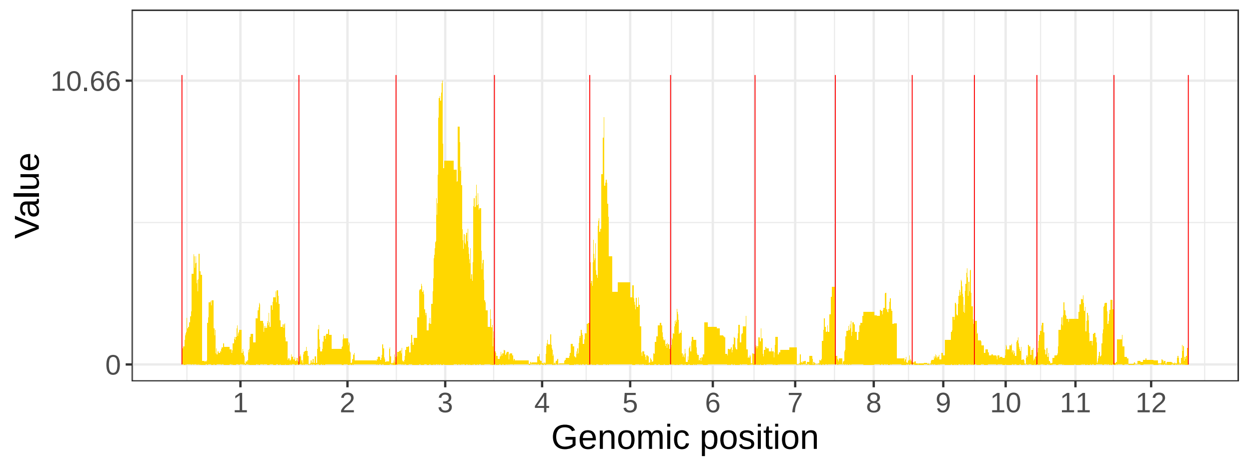

Figure 8: Plot bar + vertical_line using single_genome_plot.

```
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/bar.txt"
## [2] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/vertical_line.txt"
```

```
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 1619    4
```

```
head(data.track[[1]], 2)
```

```
##   Chr  start   stop  value
## 1   1      0 565000 0.5923
## 2   1 565000 599000 0.6701
```

```
dim(data.track[[2]])
```

```
## [1] 13  2
```

```
head(data.track[[2]], 2)
```

```
##   chr position
## 1   1        0
## 2   1 43268879
```

```
single_genome_plot(data.chr=data.chr, data.track=data.track,
                   plot_type=c("bar", "vertical_line"), chr_plotype=1,
                   margin_layer=0.01, layer_index=c(1, 1),
                   col_type=c(2, 2), color_cus=c("gold", "grey50"))
```

2.2.8 Plot ideogram + bar

```
data.chr <- read.table(system.file("examples/single_genome/genome_data.txt",
                                   package="shinyChromosomeR"),
```

```r
                        header=TRUE, as.is=TRUE, sep="\t")
head(data.chr, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```r
data.track.file <- c(system.file("examples/single_genome/ideogram.txt",
                                 package="shinyChromosomeR"),
                     system.file("examples/single_genome/bar.txt",
                                 package="shinyChromosomeR"))
data.track.file
```

```
## [1] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/ideogram.txt"
## [2] "C:/Program Files/R/R-3.5.3/library/shinyChromosomeR/examples/single_genome/bar.txt"
```

```r
data.track <- lapply(data.track.file, function(x){
   dt <- read.table(x, header=TRUE, as.is=TRUE, sep="\t")
   return(dt)
})
dim(data.track[[1]])
```

```
## [1] 573   5
```

```r
head(data.track[[1]], 2)
```

```
##   chr  start    end value1 value2
## 1   1      1 399271 p36.33   gneg
## 2   1 399271 937418 p36.32 gpos25
```

```r
dim(data.track[[2]])
```

```
## [1] 1619    4
```

```r
head(data.track[[2]], 2)
```

```
##   Chr  start   stop  value
## 1   1      0 565000 0.5923
## 2   1 565000 599000 0.6701
```

```r
single_genome_plot(data.chr=data.chr, data.track=data.track,
                   plot_type=c("ideogram", "bar"), chr_plotype=1,
                   layer_index=c(1, 2),
                   col_type=c(2, 2), height_layer=c(0.006, 0.08),
                   margin_layer=c(0.001, 0.01))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

3 Create two genomes plot using shinyChromosomeR

3.1 Essential steps to create a non-circular two-genomes plot

To create a non-circular two genomes plot, we need three datasets. The first dataset defines the genome aligned along the horizontal axis. The second dataset defines the genome aligned along the vertical axis. The third dataset is the main dataset used to create the two genomes plot.

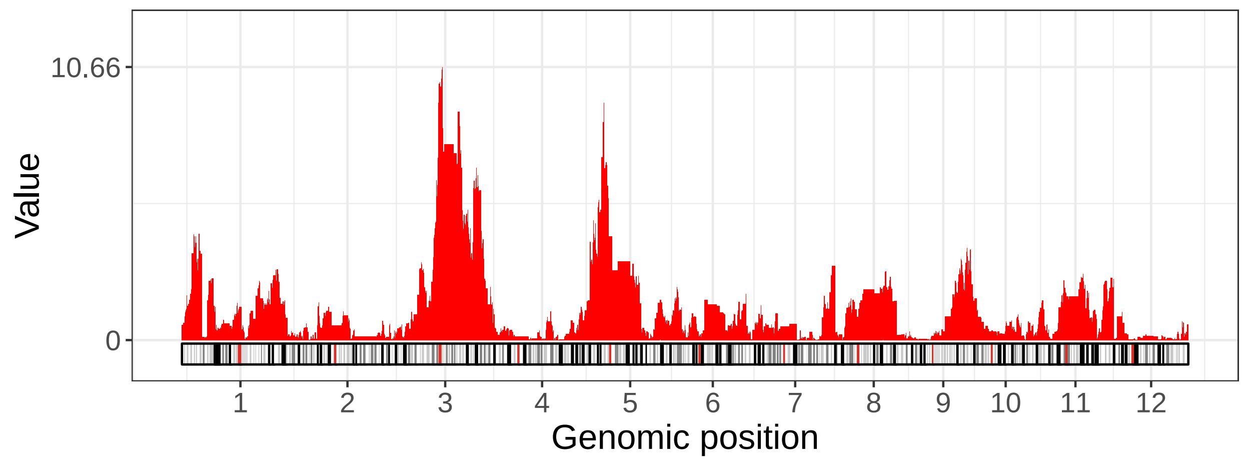

Figure 9: Plot ideogram + bar using single_genome_plot.

3.1.1 Read in the genome dataset aligned along the horizontal axis

The format of the genome dataset aligned along the horizontal axis should be the same as the genome dataset illustrated in section 2.1.1.

```
data.chr1 <- read.table(system.file("examples/two_genome/genome1_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
dim(data.chr1)
```

```
## [1] 12  2
```

```
head(data.chr1, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

3.1.2 Read in the genome dataset aligned along the vertical axis

The format of the genome dataset aligned along the vertical axis should be the same as the genome dataset illustrated in section 2.1.1.

```
data.chr2 <- read.table(system.file("examples/two_genome/genome2_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
dim(data.chr2)
```

```
## [1] 12  2
```

```
head(data.chr2, 2)
```

```
##     chr     size
## 1 Chr01 41185095
## 2 Chr02 34608401
```

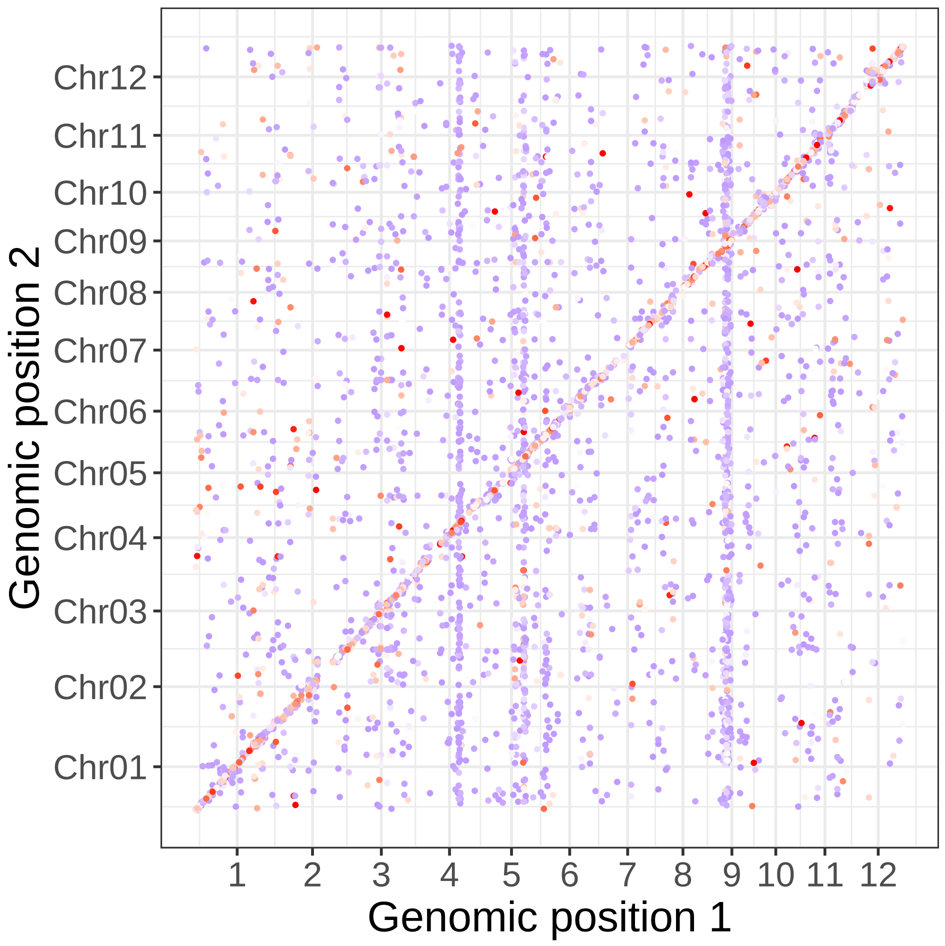

Figure 10: Plot point_gradual using two_genomes_plot.

3.1.3 Read in the main dataset

The main dataset can be used to create different types of plot, including 'point_discrete', 'point_gradual', 'rect_gradual', 'rect_discrete' and 'segment'. Please check the 'Input data format' menu (Under the 'Help' menu) of the shinyChromosome application (http://150.109.59.144:3838/shinyChromosome/) for more details.

```
data.2geno.plot <- read.table(system.file("examples/two_genome/point_gradual.txt",
                                          package="shinyChromosomeR"),
                              header=TRUE, as.is=TRUE, sep="\t")
head(data.2geno.plot, 2)
```

```
##   chrX     posX  chrY     posY  color
## 1    1        0 Chr01   412394 21.041
## 2    5 26841000 Chr05 26330003 38.726
```

3.1.4 Make the plot using the `two_genomes_plot` function

```
two_genomes_plot(data.chr1=data.chr1, data.chr2=data.chr2,
                 data.2geno.plot=data.2geno.plot, plot_type="point_gradual")
```

3.2 Create different types of two-genomes plot using shinyChromosome

3.2.1 Plot rect_discrete

```r
data.chr1 <- read.table(system.file("examples/two_genome/genome1_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
head(data.chr1, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```r
data.chr2 <- read.table(system.file("examples/two_genome/genome2_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
head(data.chr2, 2)
```

```
##     chr     size
## 1 Chr01 41185095
## 2 Chr02 34608401
```

```r
data.2geno.plot <- read.table(system.file("examples/two_genome/rect_discrete.txt",
                                          package="shinyChromosomeR"),
                              header=TRUE, as.is=TRUE, sep="\t")
head(data.2geno.plot, 2)
```

```
##   chrX   startX    stopX  chrY   startY    stopY color
## 1    2 11000001 12000000 Chr06 12000001 13000000     c
## 2    1 26000001 27000000 Chr02  6000001  7000000     c
```

```r
two_genomes_plot(data.chr1=data.chr1, data.chr2=data.chr2,
                 data.2geno.plot=data.2geno.plot, plot_type="rect_discrete",
                 theme_sty="theme6", vertical=1, horizontal=1)
```

3.2.2 Plot segment

```r
data.chr1 <- read.table(system.file("examples/two_genome/genome1_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
head(data.chr1, 2)
```

```
##   chr     size
## 1   1 43268879
## 2   2 35930381
```

```r
data.chr2 <- read.table(system.file("examples/two_genome/genome2_data.txt",
                                    package="shinyChromosomeR"),
                        header=TRUE, as.is=TRUE, sep="\t")
head(data.chr2, 2)
```

```
##     chr     size
## 1 Chr01 41185095
## 2 Chr02 34608401
```

```r
data.2geno.plot <- read.table(system.file("examples/two_genome/segment.txt",
                                          package="shinyChromosomeR"),
```

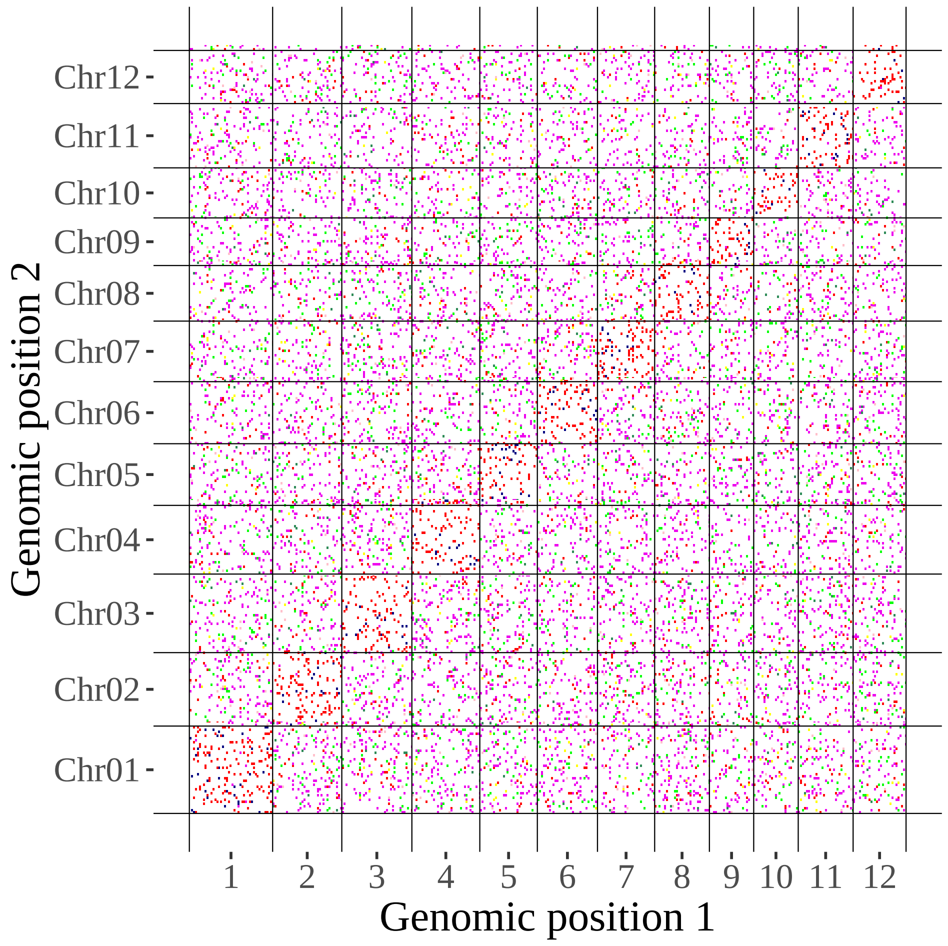

Figure 11: Plot rect_discrete using two_genomes_plot.

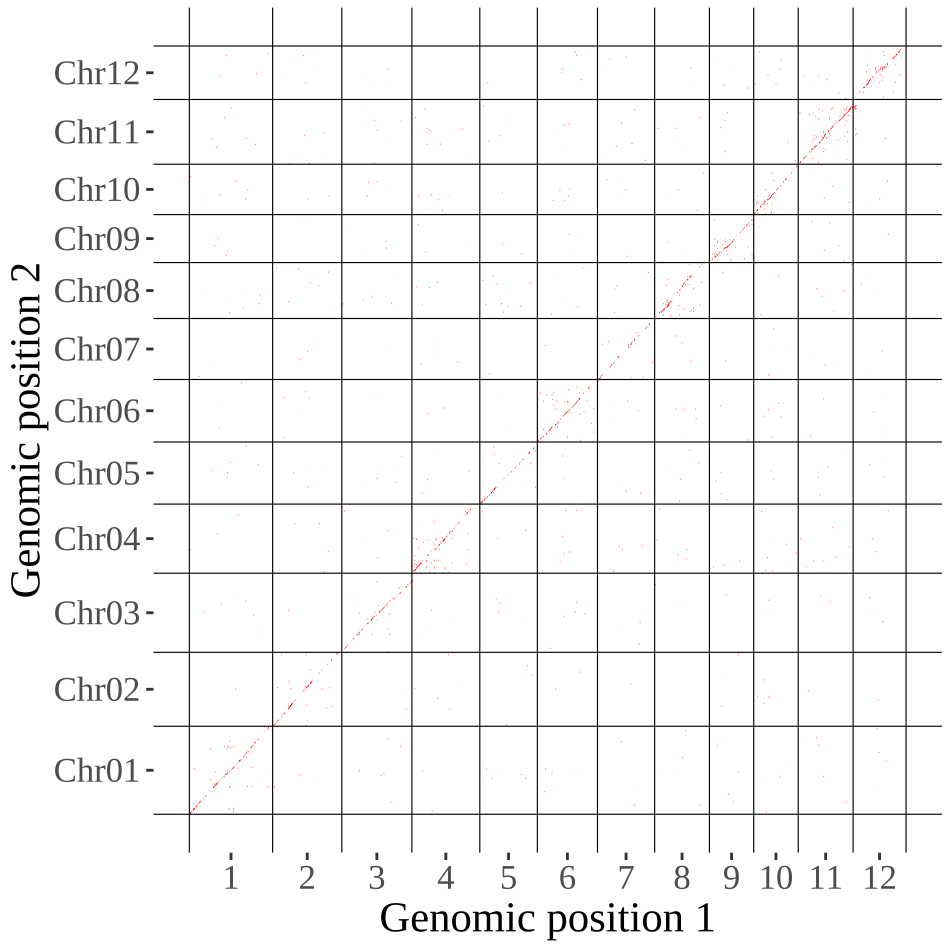

Figure 12: Plot segment using two_genomes_plot.

```
                                   header=TRUE, as.is=TRUE, sep="\t")
head(data.2geno.plot, 2)
```

```
##   chrX  startX   stopX  chrY  startY   stopY
## 1    1 3571629 3648277 Chr01 3631006 3707623
## 2   10 8626250 8630087 Chr10 8061782 8065621
```

```
two_genomes_plot(data.chr1=data.chr1, data.chr2=data.chr2,
                 data.2geno.plot=data.2geno.plot, plot_type="segment",
                 theme_sty="theme6", vertical=1, horizontal=1)
```

4 Session Information

The version number of R and packages loaded for generating the vignette were:

```
sessionInfo()
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] shinyChromosomeR_1.0.0 ggthemes_4.2.0         RColorBrewer_1.1-2
## [4] ggplot2_3.1.1          plyr_1.8.4
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.1       knitr_1.22       magrittr_1.5     munsell_0.5.0
##  [5] colorspace_1.4-1 rlang_0.3.4      highr_0.8        stringr_1.4.0
##  [9] tools_3.5.3      grid_3.5.3       gtable_0.3.0     xfun_0.6
## [13] withr_2.1.2      htmltools_0.3.6  yaml_2.2.0       lazyeval_0.2.2
## [17] digest_0.6.18    tibble_2.1.1     crayon_1.3.4     reshape2_1.4.3
## [21] purrr_0.3.2      evaluate_0.13    rmarkdown_1.12   labeling_0.3
## [25] stringi_1.4.3    compiler_3.5.3   pillar_1.4.1     scales_1.0.0
## [29] pkgconfig_2.0.2
```