

User manual of shinyChromosome

shinyChromosome is an R/Shiny application for interactive creation of non-circular plots of whole genomes.

Source code: <https://github.com/venyao/shinyChromosome>

Online use: <http://150.109.59.144:3838/shinyChromosome/>

<http://shinychromosome.ncpgr.cn/>

<https://yimingyu.shinyapps.io/shinychromosome/>

Contact: yimingyym@qq.com or yaowen@henau.edu.cn

1. Use shinyChromosome online	3
2. Interface of shinyChromosome	3
3. Installation of shinyChromosome on personal computers	7
Step 1: Install R and RStudio	7
Step 2: Install the R Shiny package and other packages required by shinyChromosome	7
Step 3: Launch the shinyChromosome application	8
4. Creation of non-circular single genome plots using shinyChromosome.....	10
4.1 Essential steps to create a non-circular single genome plot	10
Step 1. Prepare and upload the input file of the genome data	10
Step 2. Upload other input datasets to be displayed along all chromosomes of the input genome	11
Step 3. Set track index and plot type for each input dataset	12
Step 4. Click the “Go!” button to make the plot	12
4.2 Turn off an input dataset used to make a single genome plot	12
4.3 Replace an input dataset used to make a single genome plot	13
4.4 Download the created single genome plot in PDF or SVG format	14
4.5 Download the R scripts to reproduce the single genome plot	15
4.6 Create different types of single genome plot using shinyChromosome.....	17
4.6.1 Plot point	18
4.6.2 Plot line	19
4.6.3 Plot bar	20
4.6.4 Plot rect_gradual	21
4.6.5 Plot rect_discrete	22

4.6.6 Plot heatmap_gradual	23
4.6.7 Plot heatmap_discrete	24
4.6.8 Plot text.....	25
4.6.9 Plot segment.....	26
4.6.10 Plot vertical_line	27
4.6.11 Plot horizontal_line	28
4.6.12 Plot ideogram	29
4.7 Integration of multiple input datasets to create advanced single genome plot using shinyChromosome.....	30
4.8 Plotting options to decorate a single genome plot	31
4.8.1 Concatenated chromosomes v.s. Separated chromosomes	31
4.8.2 Horizontally aligned chromosomes v.s. vertically aligned chromosomes	33
4.8.3 Set point color, point size and point symbol.....	33
4.8.4 Set rect color for multiple datasets.....	36
5. Creation of Non-circular two genomes plot using shinyChromosome	39
5.1 Essential steps to create a non-circular two genomes plot	39
Step 1. Prepare and upload the input file of the genome data aligned along the horizontal axis	39
Step 2. Prepare and upload the input file of the genome data aligned along the vertical axis.....	39
Step 3. Prepare and upload the main dataset	39
Step 4. Set the plot type for the main dataset.....	40
Step 5. Click the “Go!” button to make the plot.....	40
5.2 Create different types of two genomes plot using shinyChromosome	40
5.2.1 Plot point_gradual	41
5.2.2 Plot point_discrete	41
5.2.3 Plot segment.....	42
5.2.4 Plot rect_gradual.....	43
5.2.5 Plot rect_discrete	44
6. Decorate the appearances of non-circular whole genome plot created by shinyChromosome.....	45
6.1 Figure size.....	45
6.2 Figure theme	46
6.3 Font size.....	47
6.4 Axis title.....	48
6.5 Legend	49
6.6 Height and width of different tracks	50

1. Use shinyChromosome online

shinyChromosome is deployed at <http://150.109.59.144:3838/shinyChromosome/>, <http://shinychromosome.ncpgr.cn/> and <https://yimingyu.shinyapps.io/shinychromosome/>, for online use. Users can choose to use shinyChromosome by accessing any of the three URLs based on the accessing speed. shinyChromosome is idle until you activate it by accessing the URL. So, it may take some time when you access the URL for the first time. Once it was activated, shinyChromosome could be used smoothly and easily.

2. Interface of shinyChromosome

The shinyChromosome application contains 5 main menus, “Single genome plot”, “Two genomes plot”, “Gallery”, “Help” and “About” (**Figure 1**). The “Help” menu includes three submenus as “Usage and installation”, “Input data format” and “User manual”. The “About” menu gives a brief overview of the shinyChromosome application, including tutorial messages and list of the R packages used by shinyChromosome.

shinyChromosome is a graphical user interface for interactive creation of non-circular whole genome diagrams developed using the R Shiny package.
To create single genome plot by aligning genome data along all chromosomes of a single genome, go to the [Single genome plot](#) menu.
To create two genomes plot for comparison of data across two genomes, go to the [Two genomes plot](#) menu.
For the detail format of input data, check the [Input data format](#) submenu of the [Help](#) menu.

Software references

1. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. R version 3.5.0 (2018)
2. RStudio and Inc. *shiny: Web Application Framework for R*. R package version 1.0.5 (2017)
3. Lionel Henry and Hadley Wickham. *rlang: Functions for Base Types and Core R and 'Tidyverse' Features*. R package version 0.2.1 (2018)
4. H. Wickham. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.0.0 (2018)
5. Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, et al. *grid: Various R Programming Tools for Plotting Data*. R package version 3.0.1 (2016)
6. Erich Neuwirth. *RColorBrewer: ColorBrewer palettes*. R package version 1.1-2 (2014)
7. Hadley Wickham. *plyr: Tools for Splitting, Applying and Combining Data*. R package version 1.8.4 (2016)
8. Jeffrey B. Arnold. *ggtthemes: Extra Themes, Scales and Geoms for 'ggplot2'*. R package version 3.4.0 (2017)
9. Christoph Barbour, Urs Tilmann Wolpert and Sebastian Kreutzer. *RLumShiny: "Shiny" Applications for the R Package "Luminescence"*. R package version 0.2.0 (2017)
10. Baptiste Auguie. *gridExtra: Miscellaneous Functions for "Grid" Graphics*. R package version 2.3 (2017)
11. Hadley Wickham. *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package*. R package version 1.4.3 (2017)
12. Matt Dowle and Arun Srinivasan. *data.table: Extension of 'data.frame'*. R package version 1.10.4-3 (2017)
13. JJ Allaire, Jeffrey Horner, Vincent Marti and Natacha Porte. *markdown: "Markdown" Rendering for R*. R package version 0.8 (2017)

Further references

This application was created by Yiming Yu and Wen Yao. Please send bugs and feature requests to Yiming Yu (yimingyu at gmail.com) or Wen Yao (venyaoy at qq.com). This application uses the shiny package from RStudio.

Figure 1. The “About” menu of the shinyChromosome application.

The “Single genome plot” menu allows uploading of input data to create non-circular plots along all chromosomes of a single genome (**Figure 2**). On the left of the “Single genome plot” menu is the options panel, which contains many widgets to accept user inputs. When suitable data are uploaded and plot options are properly set, the result plot can be created and displayed in the plot region of the main panel. The three “Download” buttons on top of the plot region of the main panel is provided for users to download the result plot in PDF or SVG format and the R scripts to

reproduce the plot.

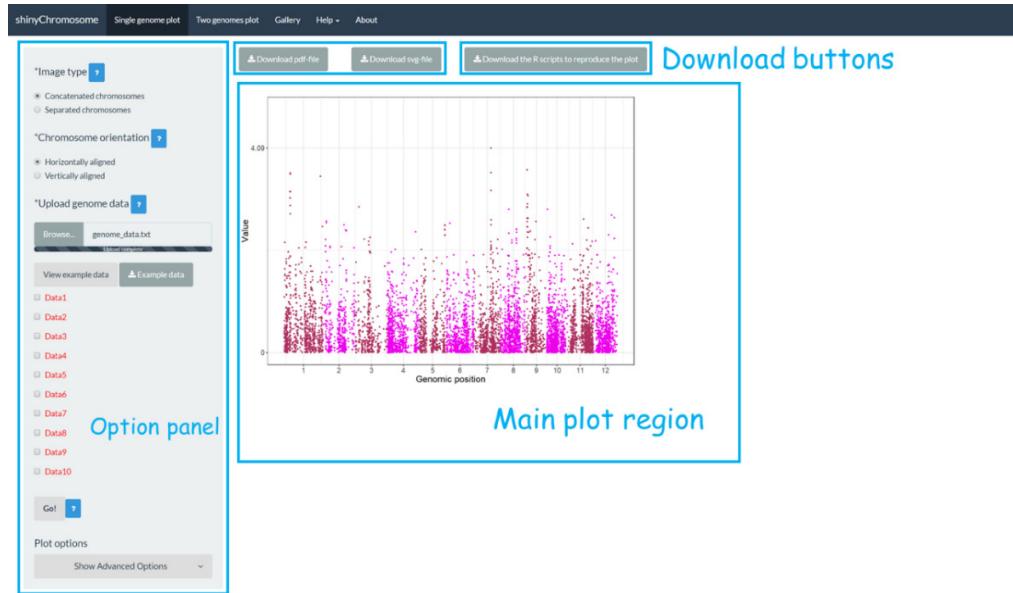


Figure 2. The “Single genome plot” menu of the shinyChromosome application.

The “Two genomes plot” menu allows uploading of input data to create two genomes plots for comparison of data across two genomes (**Figure 3**). The left panel of the “Two genomes plot” menu contains many widgets to accept user inputs. When suitable data are uploaded and plot options are properly set, the result plot could be created and displayed in the main plot region. The three “Download” buttons on top of the main plot region is provided for users to download the result plot and the R scripts to reproduce the plot.



Figure 3. The “Two genomes plot” menu of the shinyChromosome application.

A total of 65 example figures created using shinyChromosome are listed in the “Gallery” menu of the shinyChromosome application (**Figure 4**). The dataset used to generate each example figure is provided for downloading, which contains all the input files with proper file names indicating the track index and plot type corresponds to each file in the dataset.



Figure 4. The “Gallery” menu of the shinyChromosome application.

The “Usage and installation” submenu of shinyChromosome provides the usage of shinyChromosome through different approaches (**Figure 5**).

The screenshot shows the 'Usage and installation' tab of the shinyChromosome application. At the top, there is a navigation bar with tabs: 'shinyChromosome', 'Single genome plot', 'Two genomes plot', 'Gallery', 'Help', and 'About'. Below the navigation bar, there is a section titled 'installation of shinyChromosome' with a sub-section 'Input data format' highlighted with a blue box. The text in this section reads: 'This is the repository for the Shiny application presented in "shinyChromosome: an R/Shiny application for interactive creation of non-circular plots of whole genomes" (Yu et al. 2018).'. Below this, there is a section titled 'Use shinyChromosome online' with a note: 'shinyChromosome is deployed at <http://130.132.3.144:5000/shinyChromosome>, <http://shinychromosome.ncrgp.cn/> and <https://yinycg.uhry.app.io/shinychromosome/> for online use. So it may take some time when you access this URL for the first time. Once it was activated, shinyChromosome could be used smoothly and easily.' Below this, there is a section titled 'Launch shinyChromosome directly from R and GitHub' with a note: 'User can choose to run shinyChromosome installed on local computers (Windows, Mac or Linux) for a more preferable experience.' There are two steps listed: 'Step 1 Install R and RStudio' and 'Step 2 Install the R Shiny package and other packages required by shinyChromosome'. Step 2 includes an R session code block:

```

# try an http CRAN mirror if https CRAN mirror doesn't work
install.packages("shiny")
install.packages("gridExtra")
install.packages("grid")
install.packages("gridSVG")
install.packages("gridPlot2")
install.packages("plyr")
install.packages("dplyr")
install.packages("Rui3indiany")
install.packages("RColorBrewer")
install.packages("gridExtra")
install.packages("grid")
install.packages("data.table")
install.packages("shinythemes")
install.packages("shinyBS")
install.packages("shinyMatrix")
# install shinyjs
install.packages("devtools")
devtools::install_github("venyac/ShinySky", force=TRUE)

```

Figure 5. The “Usage and installation” tab of the shinyChromosome application.

The “Input data format” submenu provides the detailed format of input data to create different

types of plots using shinyChromosome (**Figure 6**).

The screenshot shows the shinyChromosome application interface. At the top, there is a navigation bar with tabs: shinyChromosome, Single genome plot, Two genomes plot, Gallery, Help ▾, and About. The Help ▾ tab is currently active, and its dropdown menu is open, showing three options: Usage and installation, Input data format (which is highlighted with a blue border), and User manual.

The main content area is titled "Input data format". It contains the following text:

The detailed format of input data for different types of plots are described in the following sections.

1. Single genome plot

1.1 Genome data

The dataset should contain only 2 columns with fixed order. Column names are optional.
1st column: chromosome ID.
2nd column: chromosome length.
Acceptable input data format can be

```
chr    size
1 43268879
2 35930381
3 36406689
```

or

```
1 43268879
2 35930381
3 36406689
```

1.2 Point

The dataset should contain >=3 columns.
In the simplest situation, the dataset should contain 3 columns with fixed order. In this case, column names are optional.
1st column: chromosome ID.
2nd column: chromosome position.
3rd column: data value.

Figure 6. The “Input data format” tab of the shinyChromosome application.

The “User manual” submenu of shinyChromosome provides this user manual in PDF format (**Figure 7**).

The screenshot shows the shinyChromosome application interface. At the top, there is a navigation bar with tabs: shinyChromosome, Single genome plot, Two genomes plot, Gallery, Help ▾, and About. The Help ▾ tab is currently active, and its dropdown menu is open, showing three options: Usage and installation, Input data format, and User manual (which is highlighted with a blue border).

The main content area displays a PDF document titled "shinyChromosome: an R/Shiny application for interactive creation of non-circular plots of whole genomes". The document is presented in a dark-themed viewer window. The title page includes the authors' names: Yiming Yu^{2,†}, Wen Yao^{1,‡,§}, Yuping Wang¹, Fangfang Huang¹. Below the title, there are two footnotes: ¹National Key Laboratory of Wheat and Maize Crop Science, College of Life Sciences, Henan Agricultural University, Zhengzhou 450002, China and ²National Key Laboratory of Crop Genetic Improvement, National Center of Plant Gene Research, Huazhong Agricultural University, Wuhan 430070, China.

Figure 7. The “User manual” tab of the shinyChromosome application.

3. Installation of shinyChromosome on personal computers

User can choose to install and run shinyChromosome on personal computers (Windows, Mac or Linux) without uploading data to online servers. Installation of shinyChromosome is platform independent, i.e., shinyChromosome can be installed on any platform with the R environment available. Installation of shinyChromosome includes 3 steps.

Step 1: Install R and RStudio

Please check CRAN (<https://cran.r-project.org/>) for the installation of R.

Please check <https://www.rstudio.com/> for the installation of RStudio.

Step 2: Install the R Shiny package and other packages required by shinyChromosome

Start an R session using RStudio and run these lines (**Figure 8**):

```
# try an http CRAN mirror if https CRAN mirror doesn't work
install.packages("shiny")
install.packages("rlang")
install.packages("gplots")
install.packages("ggplot2")
install.packages("plyr")
install.packages("ggthemes")
install.packages("RLumShiny")
install.packages("RColorBrewer")
install.packages("gridExtra")
install.packages("reshape2")
install.packages("data.table")
install.packages("shinythemes")
install.packages("shinyBS")
install.packages("markdown")
# install shinysky
install.packages("devtools")
devtools::install_github("venyao/ShinySky", force=TRUE)
```

```

> install.packages("shiny")
Installing package into 'C:/Users/venya/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/shiny_1.2.0.zip'
Content type 'application/zip' length 4529161 bytes (4.3 MB)
downloaded 4.3 MB

package 'shiny' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/venya/AppData/Local/Temp/RtmpuGcyA2/downLoaded_packages
> install.packages("rlang")
Installing package into 'C:/Users/venya/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/rlang_0.3.0.1.zip'
Content type 'application/zip' length 1031403 bytes (1.0 MB)
downloaded 1.0 MB

package 'rlang' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/venya/AppData/Local/Temp/RtmpuGcyA2/downLoaded_packages
> install.packages("gplots")
Installing package into 'C:/Users/venya/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/gplots_3.0.1.zip'
Content type 'application/zip' length 636406 bytes (641 KB)
downloaded 641 KB

package 'gplots' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/venya/AppData/Local/Temp/RtmpuGcyA2/downLoaded_packages
> install.packages("plyr")
Installing package into 'C:/Users/venya/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/plyr_1.8.4.zip'
Content type 'application/zip' length 1298938 bytes (1.2 MB)
downloaded 1.2 MB

package 'plyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/venya/AppData/Local/Temp/RtmpuGcyA2/downLoaded_packages

```

Figure 8. Installation of R packages used by shinyChromosome in RStudio on a Windows PC.

Step 3: Launch the shinyChromosome application

Start an R session using RStudio and run these lines:

```
shiny::runGitHub("shinyChromosome", "venyao")
```

This command will download the source code of shinyChromosome from GitHub to a temporary directory of your computer and then launch the shinyChromosome app in the web browser. Once the web browser was closed, the downloaded code of shinyChromosome would be deleted from your computer. Next time when you run this command in RStudio, it will download the source code of shinyChromosome from GitHub to a temporary directory again. This process is frustrating since it takes some time to download the code of shinyChromosome from GitHub.

Users are suggested to download the source code of shinyChromosome from GitHub to a fixed directory of your computer, such as “E:\apps” on Windows (**Figure 9**). Following the procedure illustrated in **Figure 9**, a zip file named “shinyChromosome-master.zip” would be downloaded into your computer. Move this file to “E:\apps” and unzip this file. Then a directory named “shinyChromosome-master” would be generated in “E:\apps”. The scripts “server.R” and “ui.R” could be found in “E:\apps\shinyChromosome-master”. Then you can start the shinyChromosome

app by running these lines in RStudio.

```
library(shiny)  
runApp("E:/apps/shinyChromosome-master", launch.browser = TRUE)
```

Then the shinyChromosome application would be opened in the default browser of your computer

(Figure 10).

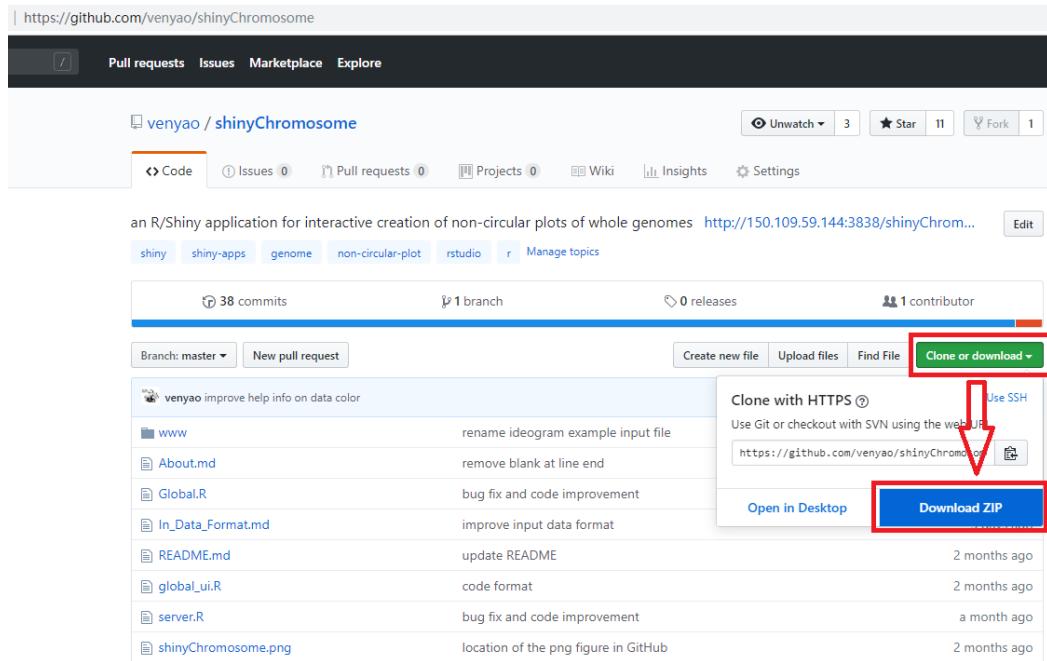


Figure 9. Download the source code of shinyChromosome from GitHub (<https://github.com/venyao/shinyChromosome>).

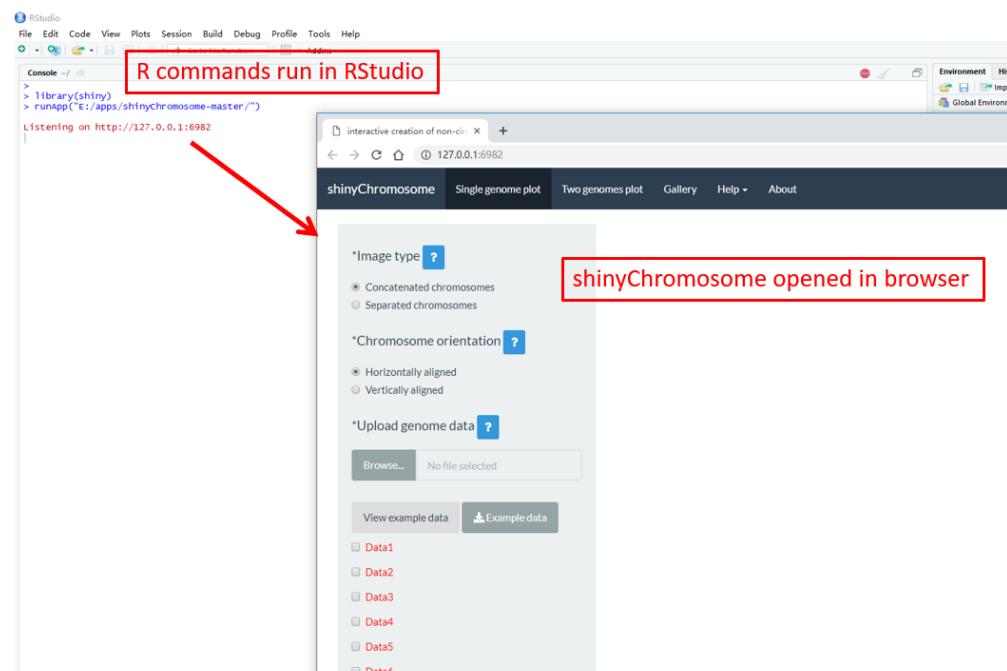


Figure 10. shinyChromosome is launched in the web browser on local personal computer.

4. Creation of non-circular single genome plots using shinyChromosome

To create a non-circular single genome plot, you need to use the “Single genome plot” menu of the shinyChromosome application. A dataset to define the genome used in the single genome plot and the other 1-10 datasets to be displayed along all the chromosomes of the genome are the compulsory input data to make a single genome plot. In the following section, we demonstrate all the essential steps to create a non-circular single genome plot using shinyChromosome with example datasets.

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

Step 1. Prepare and upload the input file of the genome data

The genome data is compulsory and defines the frame of a non-circular plot (An example dataset is available at https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt). The genome data is basically a text file with **2 columns** (**Figure 11**). 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. The content of the “Input data format” menu is also available in GitHub (https://github.com/venyao/shinyChromosome/blob/master/In_Data_Format.md).

chr	size
1	43268879
2	35930381
3	36406689
4	35278225
5	29894789
6	31246789
7	29696629
8	28439308
9	23011239
10	23134759
11	28512666
12	27497214

Figure 11. The format of input file for genome data.

Here, we have prepared this file and stored this file on the disk (For example “E:/” on Windows). Next, we need to upload this file to the shinyChromosome application using the “Browse...” widget

below the “Upload genome data” indicator in the left panel of the “Single genome plot” menu of the shinyChromosome application (**Figure 12**).

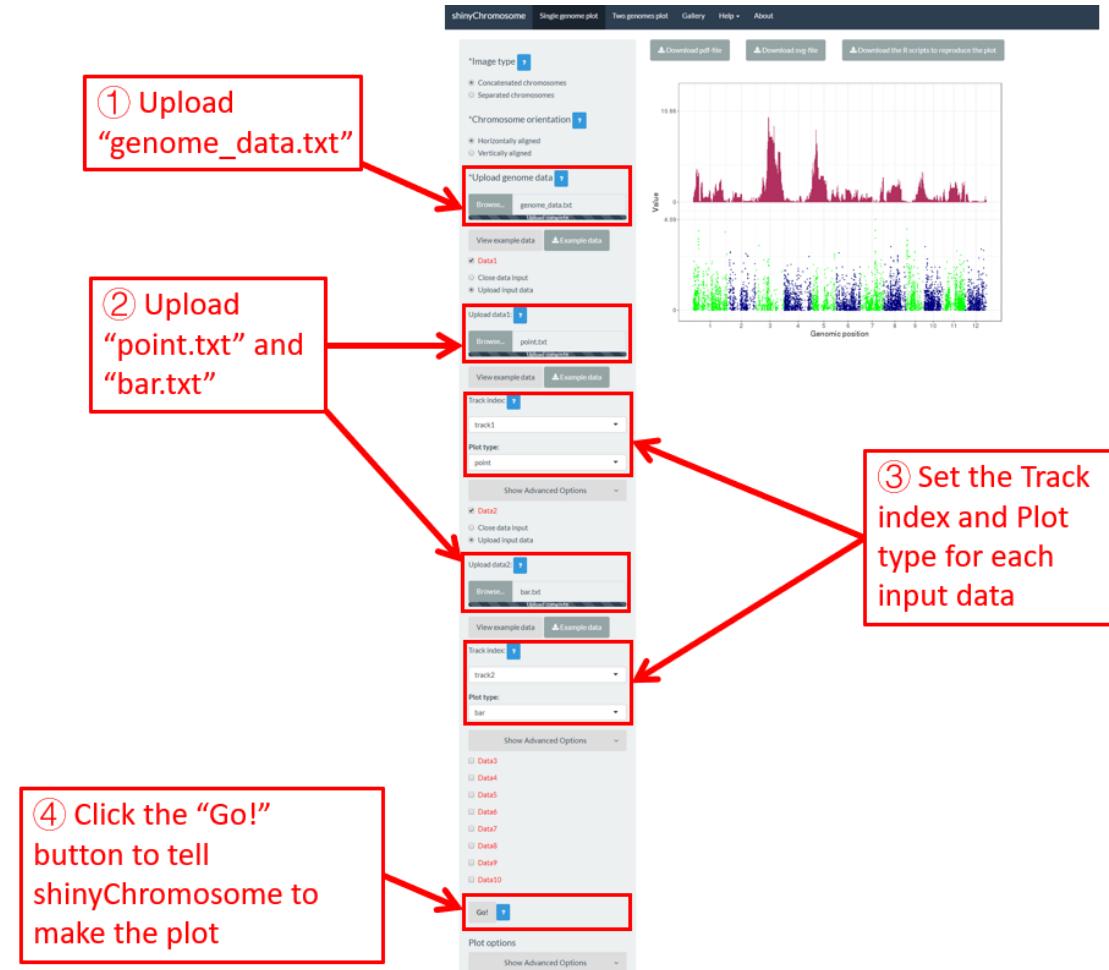


Figure 12. Essential steps to create a single genome plot using shinyChromosome. The file “genome_data.txt” is uploaded to the genome data widget. The file “point.txt” is uploaded to the “Data1” track while the file “bar.txt” is uploaded to the “Data2” track.

Step 2. Upload other input datasets to be displayed along all chromosomes of the input genome

1-10 datasets could be then uploaded to be displayed along all chromosomes of the genome data uploaded in **Step 1**. The ten “DataX” (Data1 to Data10) checkbox on the left panel of the “Single genome plot” menu are provided for this purpose (**Figure 12**). Here, we use two input datasets

(https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/point.txt and https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/bar.txt)

[gle_genome/bar.txt](#)) to demonstrate this process. The detailed format of input file to create different types of plots are illustrated in the “Input data format” submenu of the shinyChromosome application (**Figure 6**). Here, we have prepared the two files and stored them on the disk (For example “E:/” on Windows). To upload the file “point.txt” to the Data1 track, check the “Data1” checkbox, choose the “Upload input data” radio button and then use the “Browse...” widget to upload the “point.txt” from the disk (**Figure 12**). To upload the file “bar.txt” to the Data2 track, check the “Data2” checkbox, choose the “Upload input data” radio button and then use the “Browse...” widget to upload the “bar.txt” from the disk.

Step 3. Set track index and plot type for each input dataset

By default, the track index for each input dataset is “track1” and the plot type for each input dataset is “point”. We need to set the track index as “track2” and the plot type as “bar” for the input file “bar.txt” as we want to use this file to create a bar plot (**Figure 12**).

Step 4. Click the “Go!” button to make the plot

After all the input datasets has been successfully uploaded to the shinyChromosome application and the track index and plot type have been set properly, we need to click the “Go!” button at the bottom of the left panel of the “Single genome plot” menu to tell shinyChromosome to make the plot (**Figure 12**). The plot shown in the main panel of **Figure 12** is the plot generated using the input datasets uploaded in Step 1 and Step 2. **By default, random color or predefined colors would be used by shinyChromosome when generating the plot.**

4.2 Turn off an input dataset used to make a single genome plot

For each “DataX” checkbox on the left panel of the “Single genome plot” menu, we provide two radio button options: “Close data input” and “Upload input data”. By default, the “Close data input” radio button is checked. The “Upload input data” is used to upload an input dataset while the “Close data input” is used to turn off an input dataset already uploaded.

In section 4.1, we uploaded the input files “point.txt” and “bar.txt” to “Data1” and “Data2” respectively. The “point.txt” file was used to create the scatter plot in **Figure 12** while the “bar.txt”

was used to create the bar plot in **Figure 12**. If we want to remove the scatter plot from **Figure 12** and only keep the bar plot, we can check the “Close data input” radio button under the “Data1” checkbox and then click the “Go!” button on the bottom of the left panel to tell shinyChromosome to update the plot. This process is demonstrated in **Figure 13**. In this way, the “point.txt” file would be not be used by shinyChromosome and only the bar plot would be created as is shown in **Figure 13**.

13. Remember to click the “Go!” button to update the plot whenever you modify any option or input file through the diverse widgets provided in the left panel.

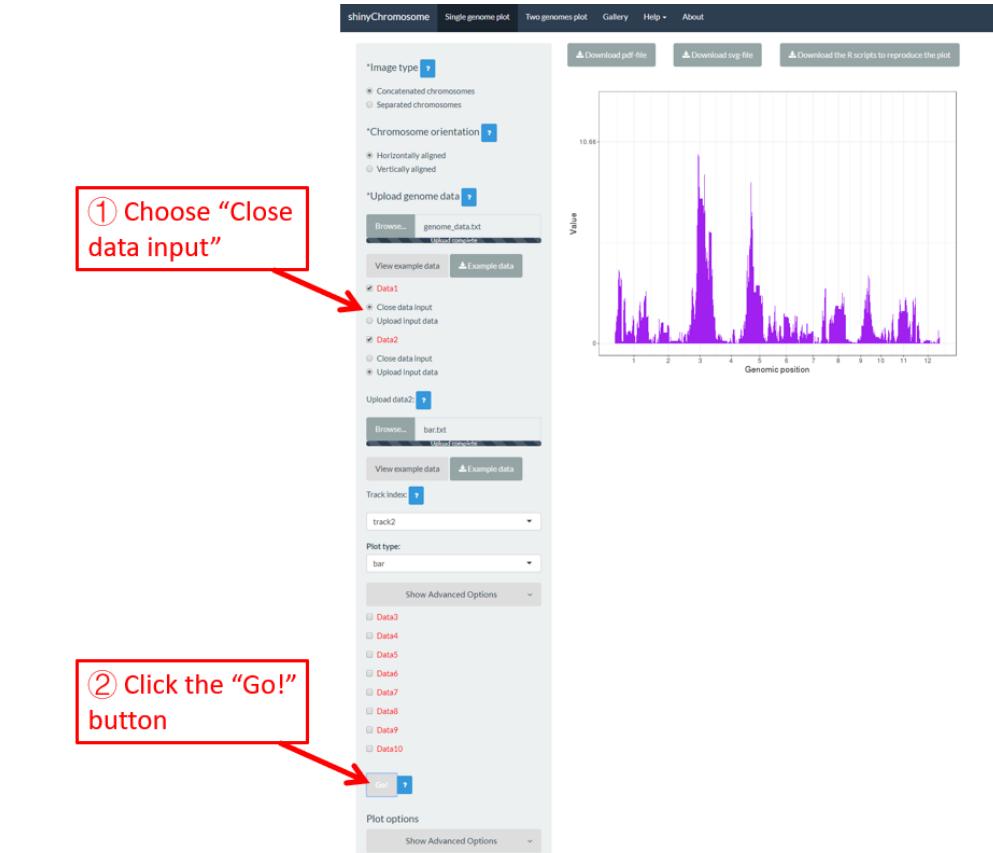


Figure 13. The procedure to turn off an input dataset used to make a single genome plot.

4.3 Replace an input dataset used to make a single genome plot

A single genome plot is usually composed of several basic type of plots distributed in different tracks. Each plot is created using an uploaded input dataset. Sometimes, we may want to replace one or more input files so that we can update some components of the single genome plot without creating the whole plot all over again. For example, we want to replace the “bar.txt” uploaded to “Data2” using a new input file “rect_discrete.txt” to create discrete rectangles. To achieve this

purpose, we can use the “Browse...” widget under the “Upload input data” radio button in “Data2” to upload the “rect_discrete.txt” to “Data2”. Then “bar.txt” will be replaced by “rect_discrete.txt” in “Data2”. At the same time, we need to set the plot type as “rect_discrete” for “Data2”. Finally, we need to click the “Go!” button on the bottom of the left panel to tell shinyChromosome to update the corresponding plot. This process is demonstrated in **Figure 14**.



Figure 14. The procedure to replace an input dataset used to make a single genome plot.

4.4 Download the created single genome plot in PDF or SVG format

After a single genome plot was generated, the user can use the widgets “Download PDF-file” and “Download SVG-file” on top of the generated plot in the main panel of the “Single genome plot” menu to download the generated plot in PDF or SVG format (**Figure 15**). By default, the two downloaded files are named as “Visualization_1.pdf” and “Visualization_1.svg” respectively.

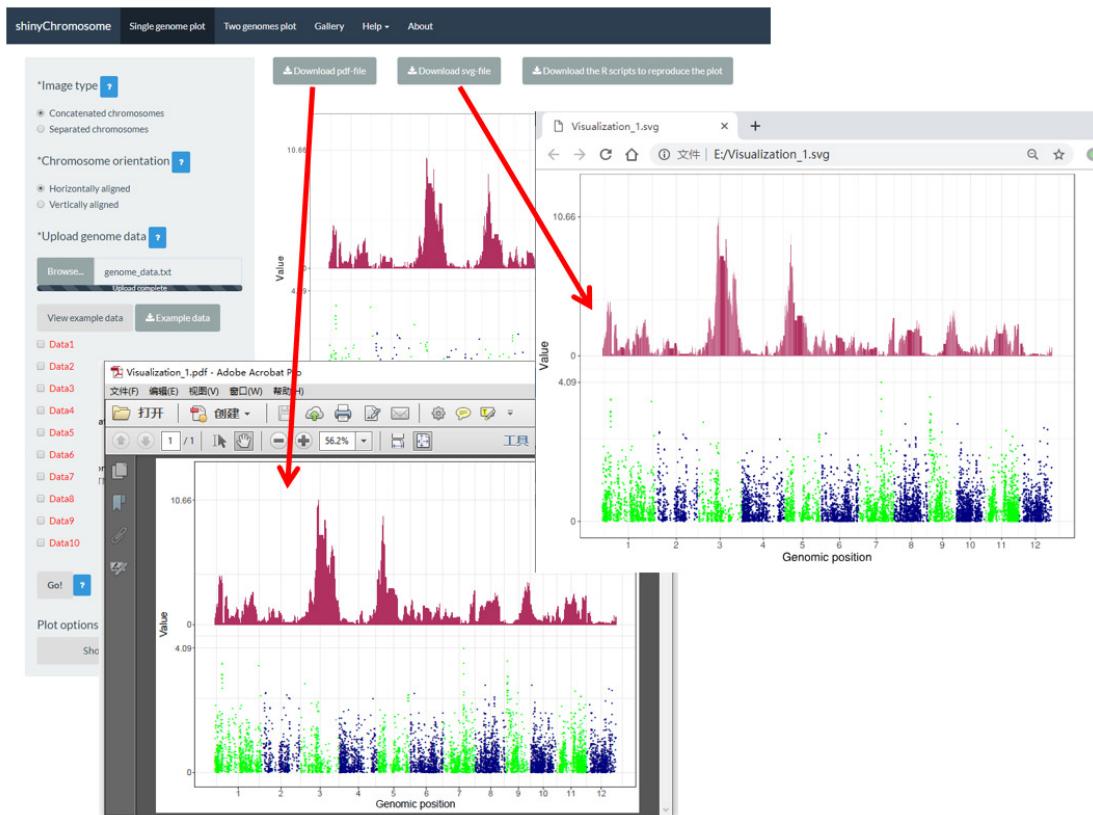


Figure 15. The downloaded PDF file “Visualization_1.pdf” is opened in Adobe Acrobat and the downloaded SVG file “Visualization_1.svg” is opened in Google Chrome browser.

4.5 Download the R scripts to reproduce the single genome plot

Some users may have noticed that a download widget named “Download the R scripts to reproduce the plot” is provided on top of the generated plot in the main panel of the “Single genome plot” menu when a single genome plot has been created (**Figure 16**). The downloaded R scripts can be used outside the shinyChromosome application to reproduce the plot generated using the graphical interface of shinyChromosome. The downloaded R scripts should be used in the R environment. The downloaded R scripts can be used with other scripts of the users in an analysis pipeline. The downloaded R scripts can also be cycled to generate hundreds of similar plots using different input datasets and the same set of parameters. By default, the downloaded R script is named as “Script_1.R”.



Figure 16. The widget to download the R script to reproduce the plot.

To use the “Script_1.R” script, open the R environment using RStudio. The path of the “Script_1.R” script in the disk is usually not the same as the default working directory of RStudio. We need to set them as the same directory. Here, we copy the “Script_1.R” script to “E:/” and set the working directory as “E:/” by editing the “Script_1.R” script in RStudio (**Figure 17**). In addition, we need to copy the input files (the “point.txt” and “bar.txt”) used to make the single genome plot to the directory “E:/”. The “Script_1.R” script depends on another R script “writeCmd-1_function.R”. This R script is provided in the source code of shinyChromosome in GitHub (https://github.com/venyao/shinyChromosome/blob/master/writeCmd-1_function.R). We need to download this R script and copy it to the directory “E:/”. Finally, run all the code in the edited “Script_1.R” script and a PDF file named “Visualization_1.pdf” would be generated in the directory “E:/”. The content of the “Visualization_1.pdf” is the same as the plot generated using the graphical interface of shinyChromosome in **Figure 16**.

```

## setwd("absolute path of a directory containing the input data files")
options(scipen=5)
## copy the R script "writeCmd-1_function.R" to the directory set by the "setwd" function.
source("writeCmd-1_function.R")
library(gplots)
library(ggplot2)
library(plotly)
library(plyr)
library(shinyBS)
library(ggthemes)
library(RLumShiny)
library(RColorBrewer)
library(gridExtra)
library(reshape2)
library(data.table)
library(grid)
Height <- 550
width <- 750
chr_plotype <- 1

```



```

## setwd("absolute path of a directory containing the input data files")
setwd("E:/")
options(scipen=5)
## copy the R script "writeCmd-1_function.R" to the directory set by the "setwd" function.
source("writeCmd-1_function.R")
library(gplots)
library(ggplot2)
library(plotly)
library(plyr)
library(shinyBS)
library(ggthemes)
library(RLumShiny)
library(RColorBrewer)
library(gridExtra)
library(reshape2)
library(data.table)
library(grid)
Height <- 550
width <- 750
chr_plotype <- 1
plot_direct <- 1

```

Figure 17. Open and edit the “Script_1.R” script as indicated in red box. Then run all the code of the edited “Script_1.R” script in RStudio to reproduce the single genome plot.

4.6 Create different types of single genome plot using shinyChromosome

A total of 12 different types of plot can be created using shinyChromosome, including point, line, bar, rect_gradual, rect_discrete, heatmap_gradual, heatmap_discrete, text, segment, vertical_line, horizontal_line and ideogram. To create a single genome plot, at least two input data files are needed, the genome data file which defines the genome used in the plot and other datasets to be displayed along all chromosomes of the genome. The format of genome data is illustrated in section 4.1. The detailed format of input files to make different types of single genome plot is demonstrated in the “Input data format” menu (under the “Help” menu) of the shinyChromosome application. In this section, we will show the key parameters to make different types of single genome plot using the graphical interface of shinyChromosome with example input datasets. The example genome data files used in this section is the same as the file used in section 4.1 (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/si

ngle_genome/genome_data.txt).

4.6.1 Plot point

To make point plot using shinyChromosome, we need two input files, the genome data file and the input file defining the genomic position and the value of each point. The simplest dataset to plot point should contain **3 columns** including the chromosome IDs, genomic positions and numeric values. Each genomic position would be represented as a point along the defined genome. Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “point.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/point.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot point using shinyChromosome (**Figure 18**).

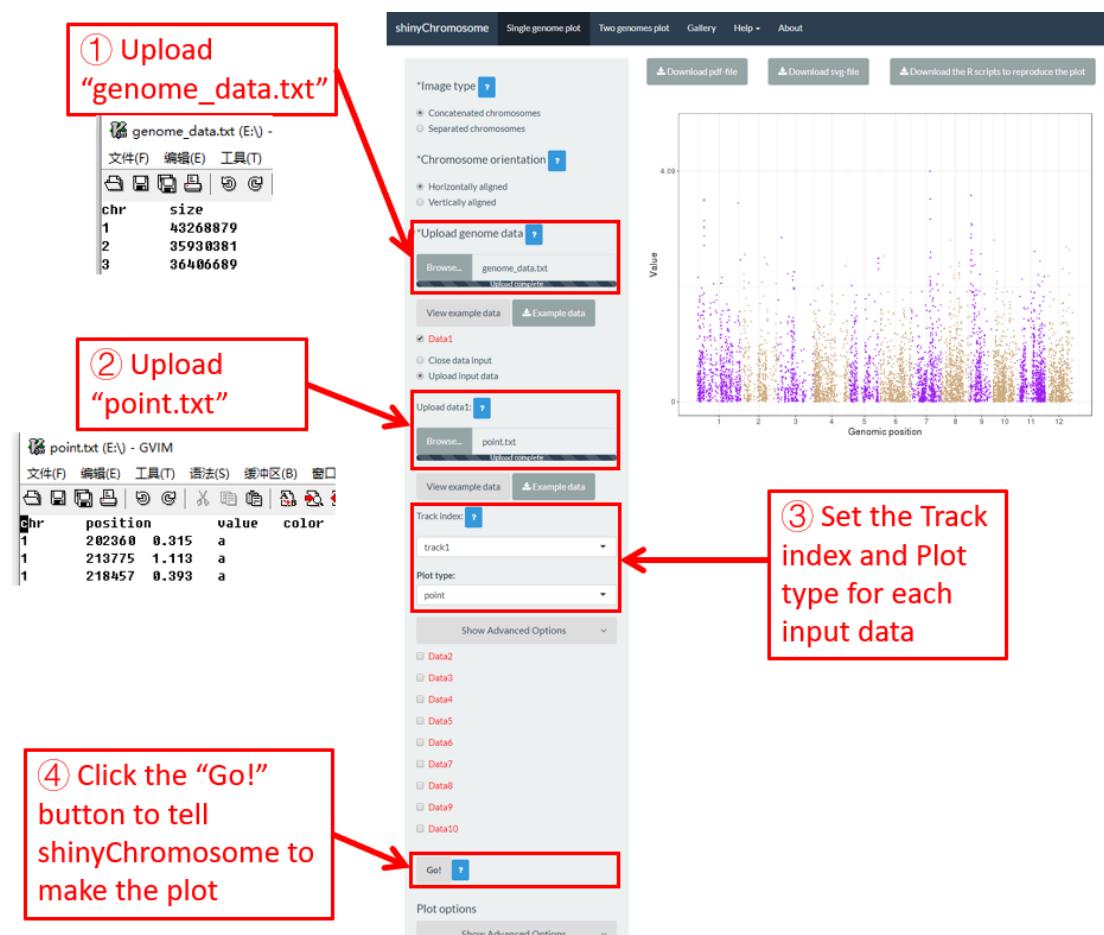


Figure 18. The procedure to plot point using shinyChromosome.

4.6.2 Plot line

To make line plot using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the value of each point to be connected in a line. The simplest dataset to plot line should contain **3 columns** including the chromosome IDs, genomic positions and numeric values. Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “line.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/line.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot line using shinyChromosome (**Figure 19**).

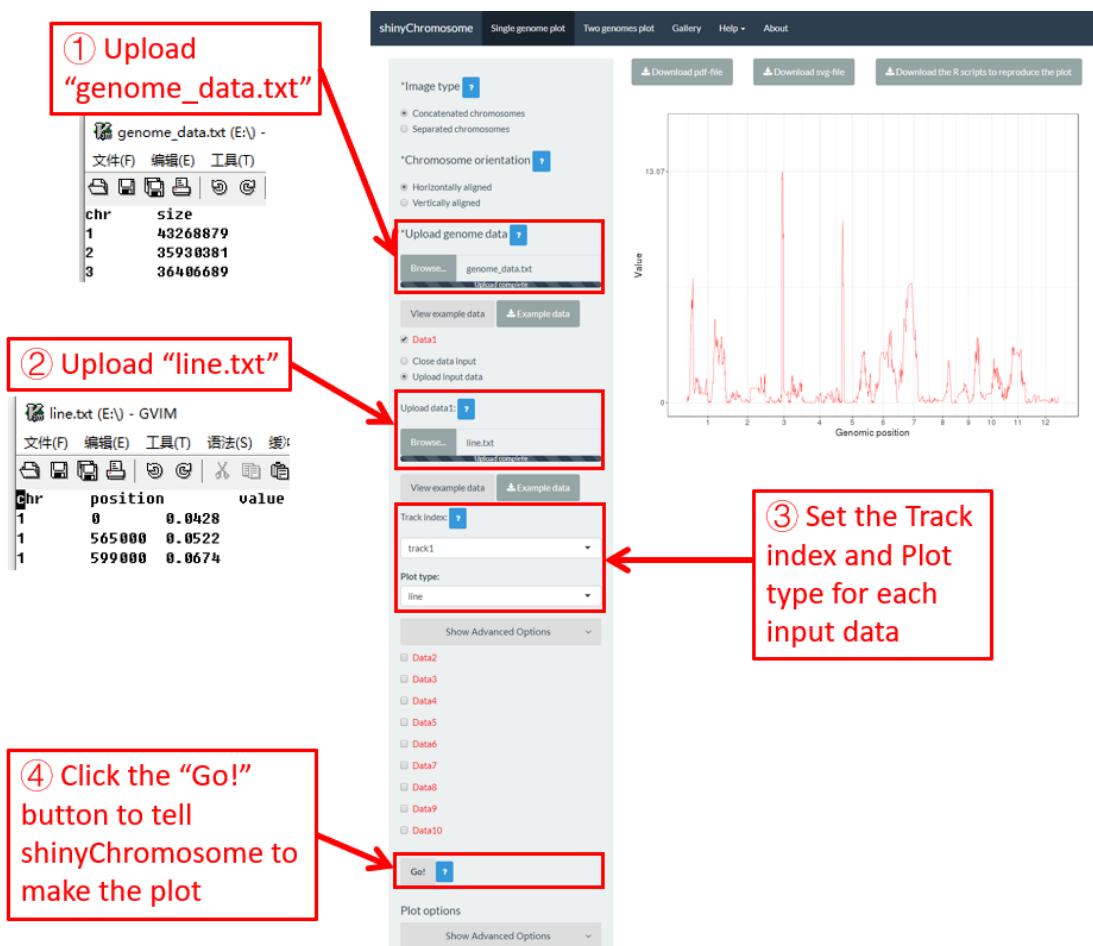


Figure 19. The procedure to plot line using shinyChromosome.

4.6.3 Plot bar

To make bar plot using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the value of each genomic region to be displayed as a bar. The simplest dataset to plot bar should contain **4 columns** including the chromosome IDs, start coordinates of genomic regions, end coordinates of genomic regions and the heights of different bars. Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “bar.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/bar.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot bar using shinyChromosome (Figure 20).

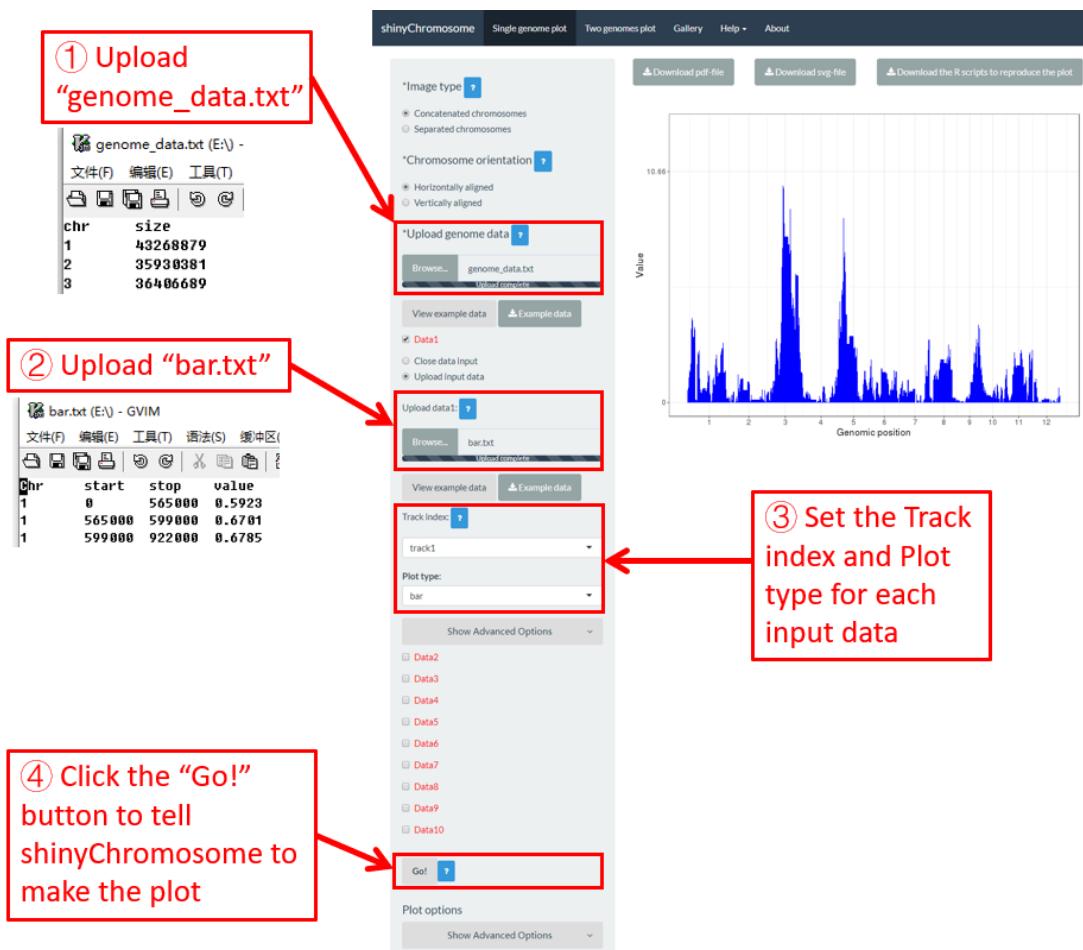


Figure 20. The procedure to plot bar using shinyChromosome.

4.6.4 Plot rect_gradual

To make gradual rectangle plot using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the value of each genomic region to be displayed as a rectangle. The simplest dataset to plot gradual rectangle should contain **4 columns** including the chromosome IDs, start coordinates of genomic regions, end coordinates of genomic regions and the value of each rectangle. The 4th column should be a numeric vector representing continuous variables. Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “rect_gradual.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/rect_gradual.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot rect_gradual using shinyChromosome (**Figure 21**). Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

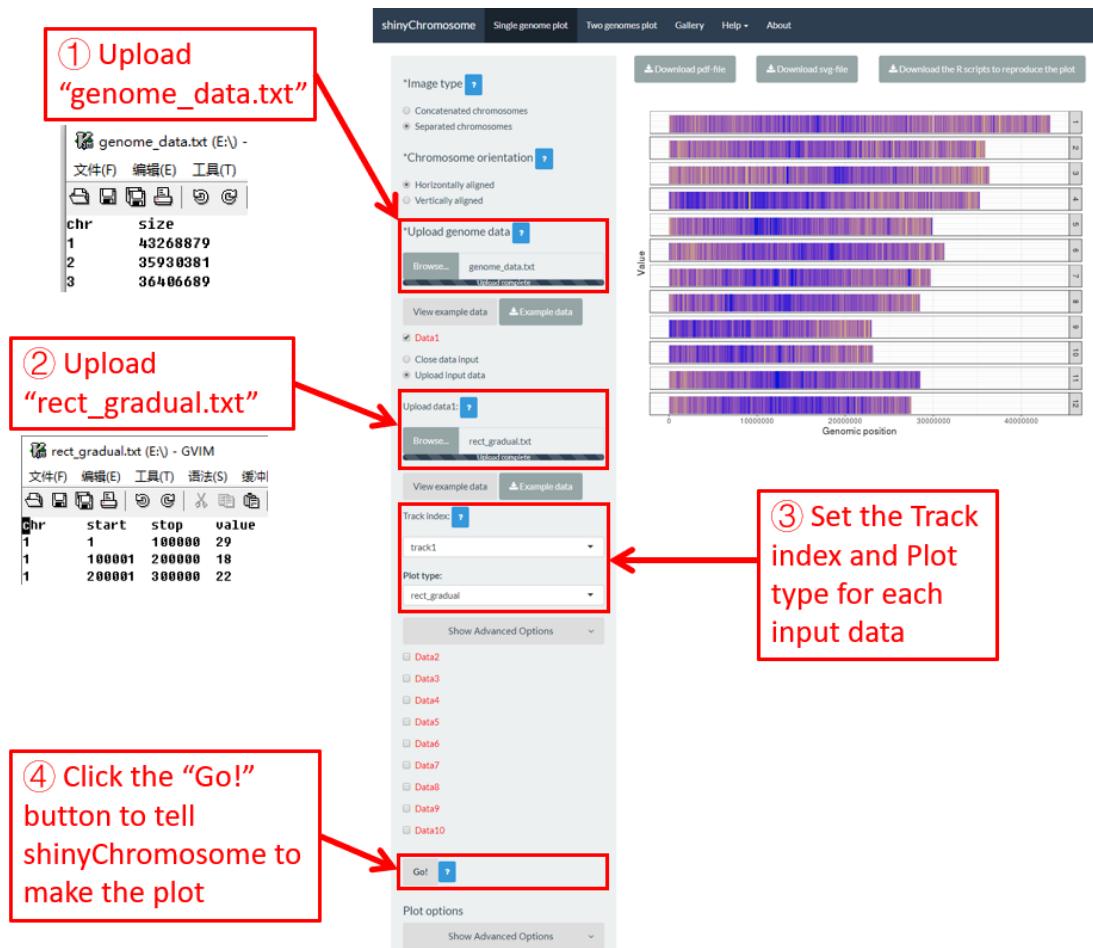


Figure 21. The procedure to plot rect_gradual using shinyChromosome.

4.6.5 Plot rect_discrete

To make discrete rectangle plot using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the value of each genomic region to be displayed as a rectangle. The simplest dataset to plot discrete rectangle should contain **4 columns** including the chromosome IDs, start coordinates of genomic regions, end coordinates of genomic regions and the value of each rectangle. □ The 4th column should be a character vector representing discrete variables. Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “rect_discrete.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/rect_discrete.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot rect_discrete using shinyChromosome (**Figure 22**). Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

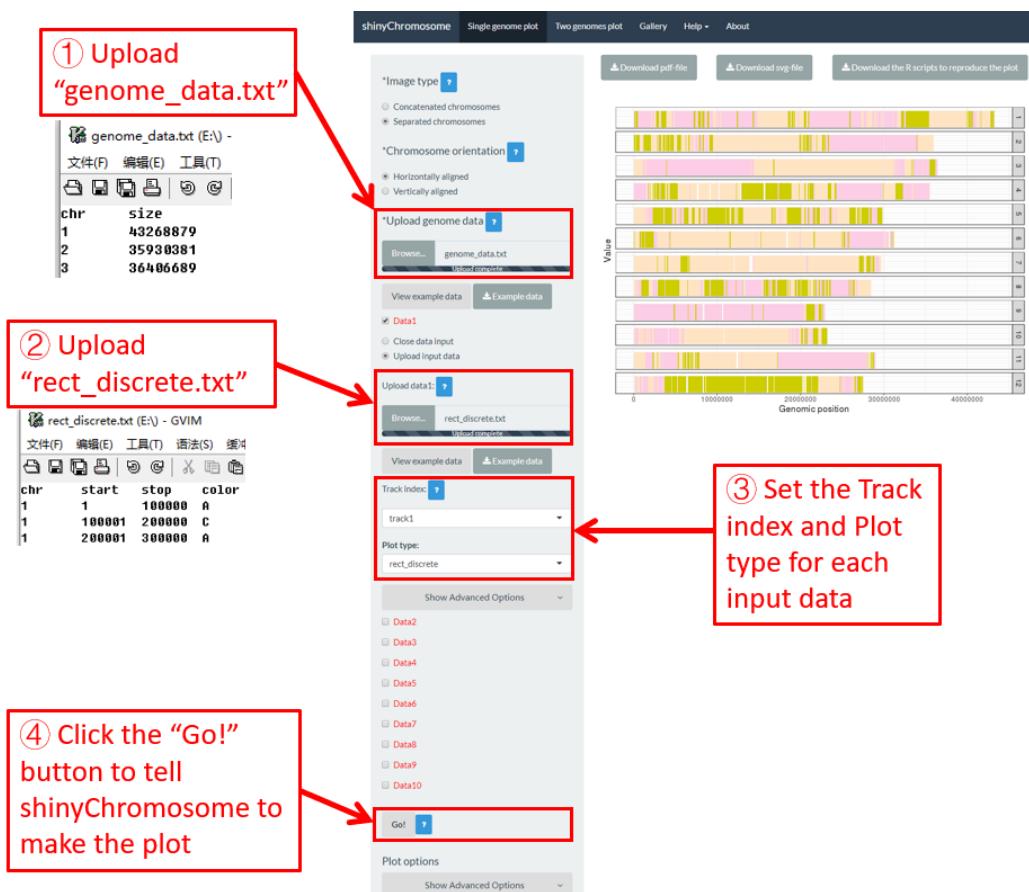


Figure 22. The procedure to plot rect_discrete using shinyChromosome.

4.6.6 Plot heatmap_gradual

To make gradual heatmap using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the values of each genomic region to be displayed as a cell of a heatmap. The simplest dataset to plot gradual heatmap should contain **at least 4 columns**.

- The 1-3 columns of data for heatmap_gradual plot are the chromosome IDs, start coordinates of genomic regions and end coordinates of genomic regions.
- Apart from the first three columns, other columns should be **numeric vectors** representing continuous variables.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “heatmap_gradual.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/heatmap_gradual.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot heatmap_gradual using shinyChromosome (**Figure 23**). Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

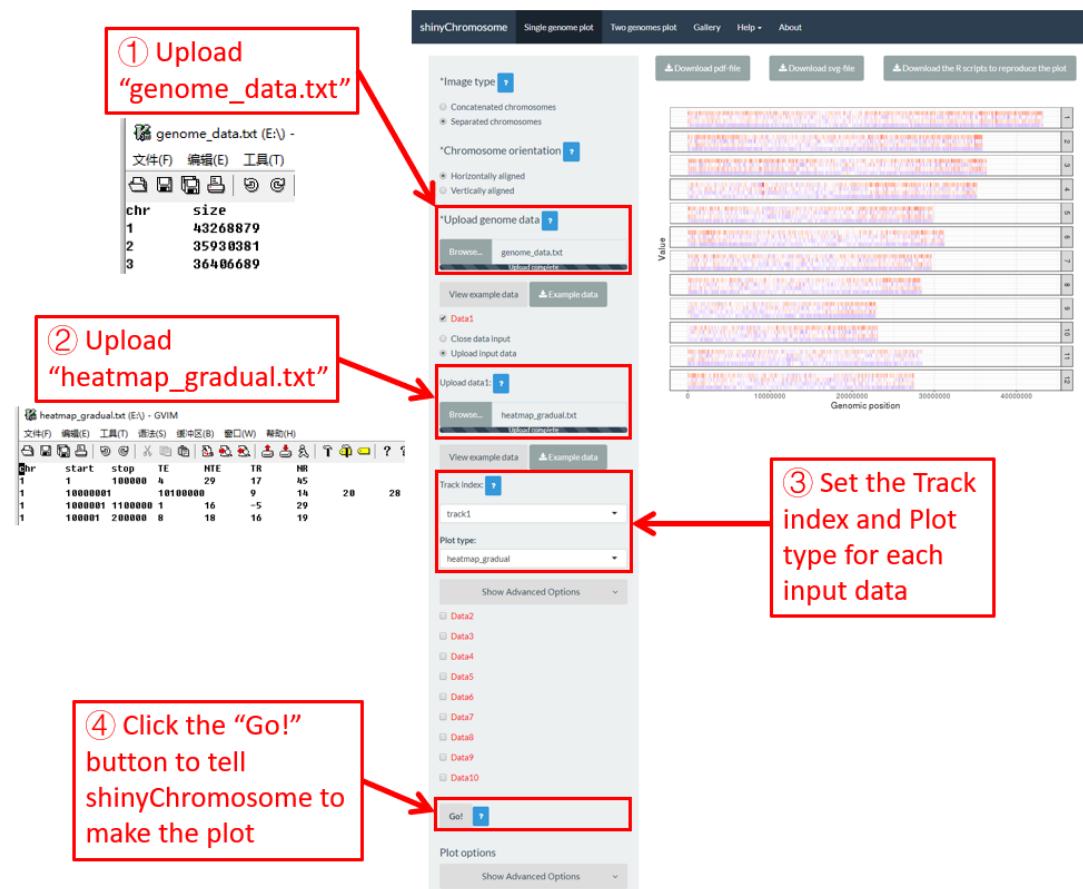


Figure 23. The procedure to plot heatmap_gradual using shinyChromosome.

4.6.7 Plot heatmap_discrete

To make discrete heatmap using shinyChromosome, we need two input files, the genome data file and the input file defining the position and the values of each genomic region to be displayed as a cell of a heatmap. The simplest dataset to plot discrete heatmap should contain **at least 4 columns**.

- The 1-3 columns of data for heatmap_gradual plot are the chromosome IDs, start coordinates of genomic regions and end coordinates of genomic regions.
- Apart from the first three columns, other columns should be **character vectors** representing different categories.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “heatmap_discrete.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/heatmap_discrete.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot heatmap_discrete using shinyChromosome (**Figure 24**). Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

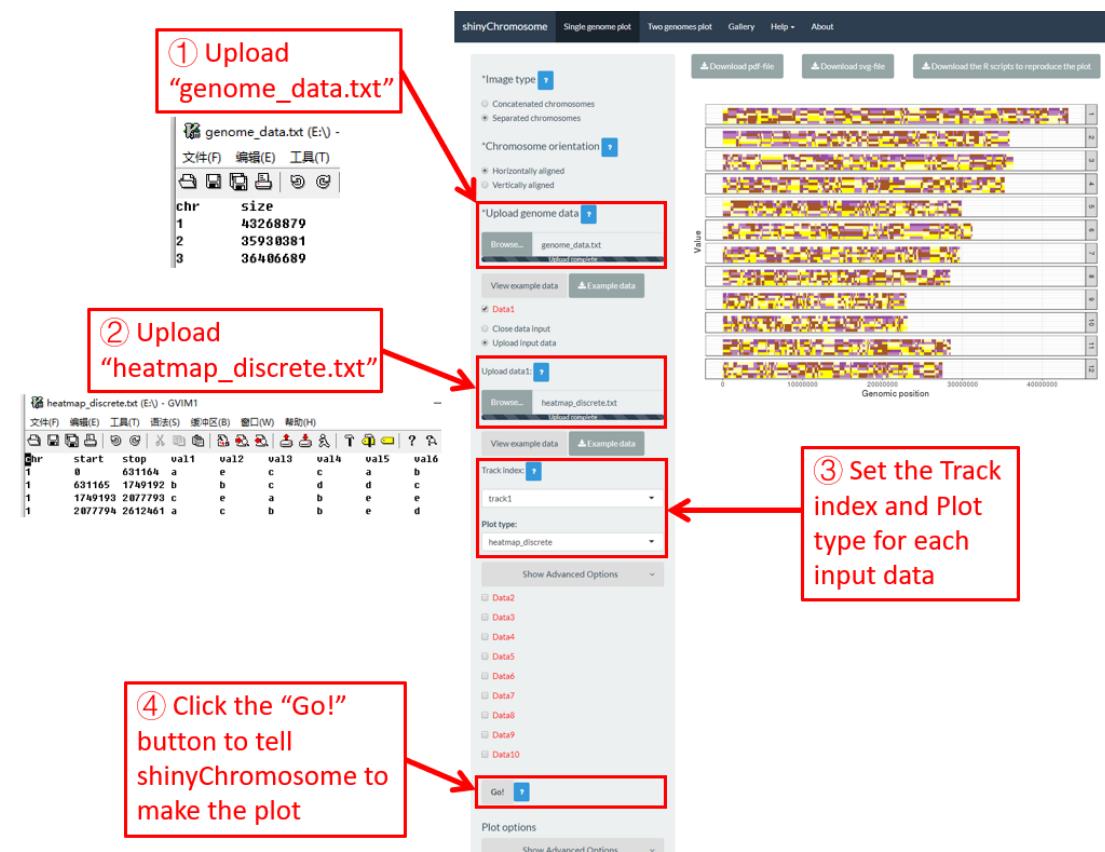


Figure 24. The procedure to plot heatmap_discrete using shinyChromosome.

4.6.8 Plot text

To plot text using shinyChromosome, we need two input files, the genome data file and the input file defining the position of the text to be displayed along the genome. The simplest dataset to plot point should contain **4 columns**.

- The 1-3 columns of data for text plot are the chromosome IDs, X-axis coordinates and the Y-axis coordinates of texts.
- The last column should be a **character vector** representing texts.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “text.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/text.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot text using shinyChromosome (**Figure 25**). Please be noted that some of the advanced options has been modified as is shown in **Figure 25**. Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

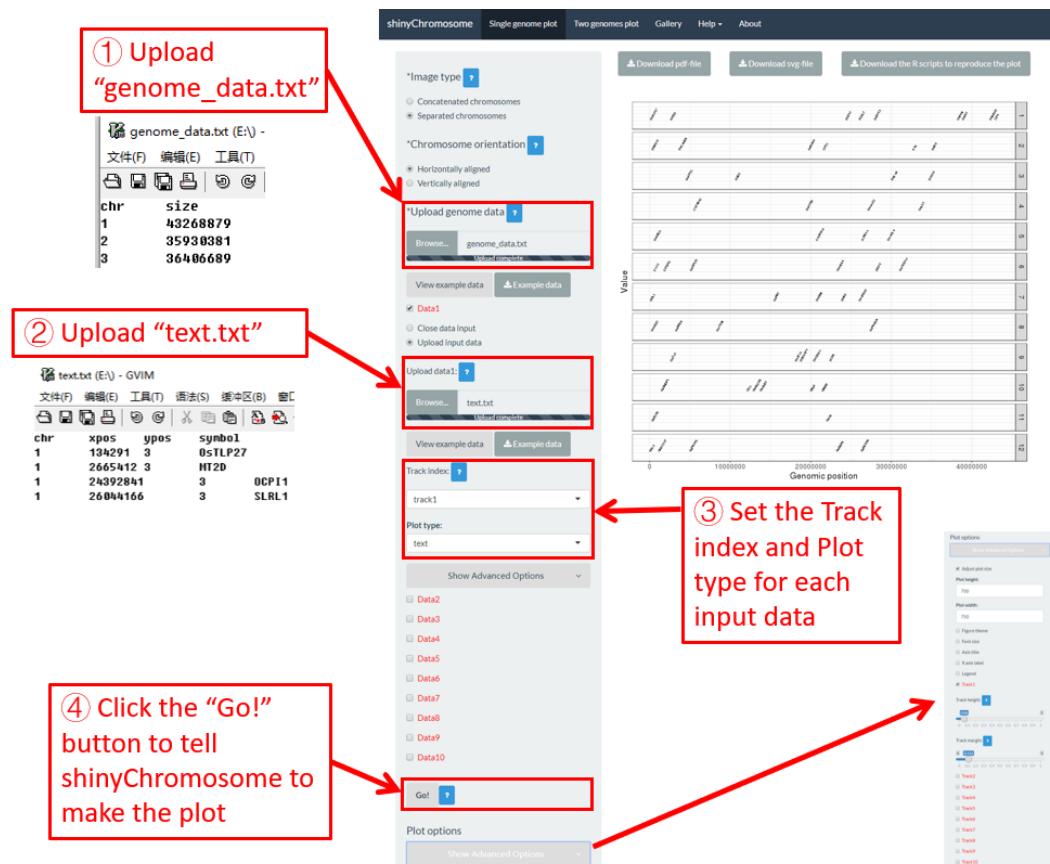


Figure 25. The procedure to plot text using shinyChromosome.

4.6.9 Plot segment

To plot segment using shinyChromosome, we need two input files, the genome data file and the input file defining the start and end positions of the segment to be displayed along the genome.

The simplest dataset to plot point should contain **5 columns**.

- The 1st column contains the chromosome IDs of each segment.
- Columns 2-3 and columns 4-5 represent the positions of the two ends of segment respectively.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “segment.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/segment.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot segment using shinyChromosome (**Figure 26**).

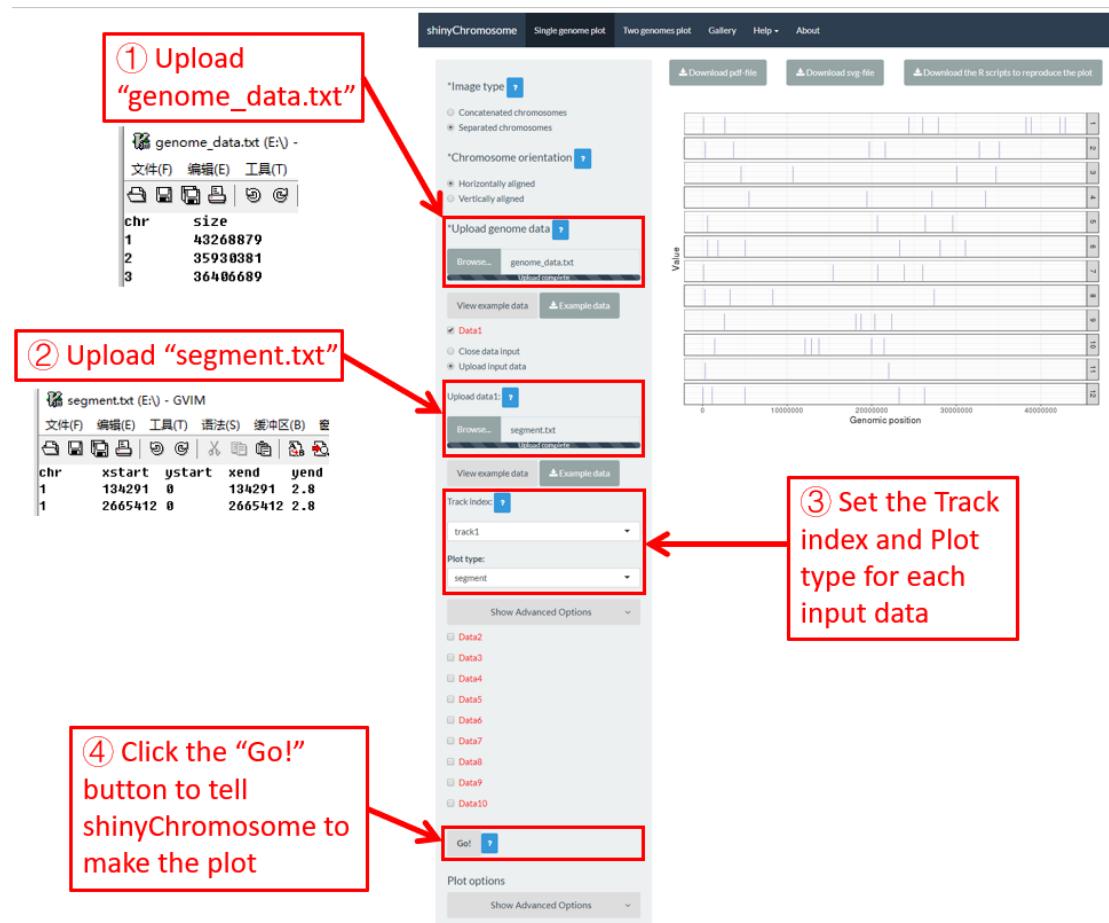


Figure 26. The procedure to plot segment using shinyChromosome.

4.6.10 Plot vertical_line

Vertical lines are usually mixed with other types of plot. The input data to create vertical lines should include **two columns**. The first column is the chromosome IDs and the second column is the X-axis coordinate of each vertical line.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt), “vertical_line.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/vertical_line.txt) and “bar.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/bar.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot vertical line using shinyChromosome (Figure 27).

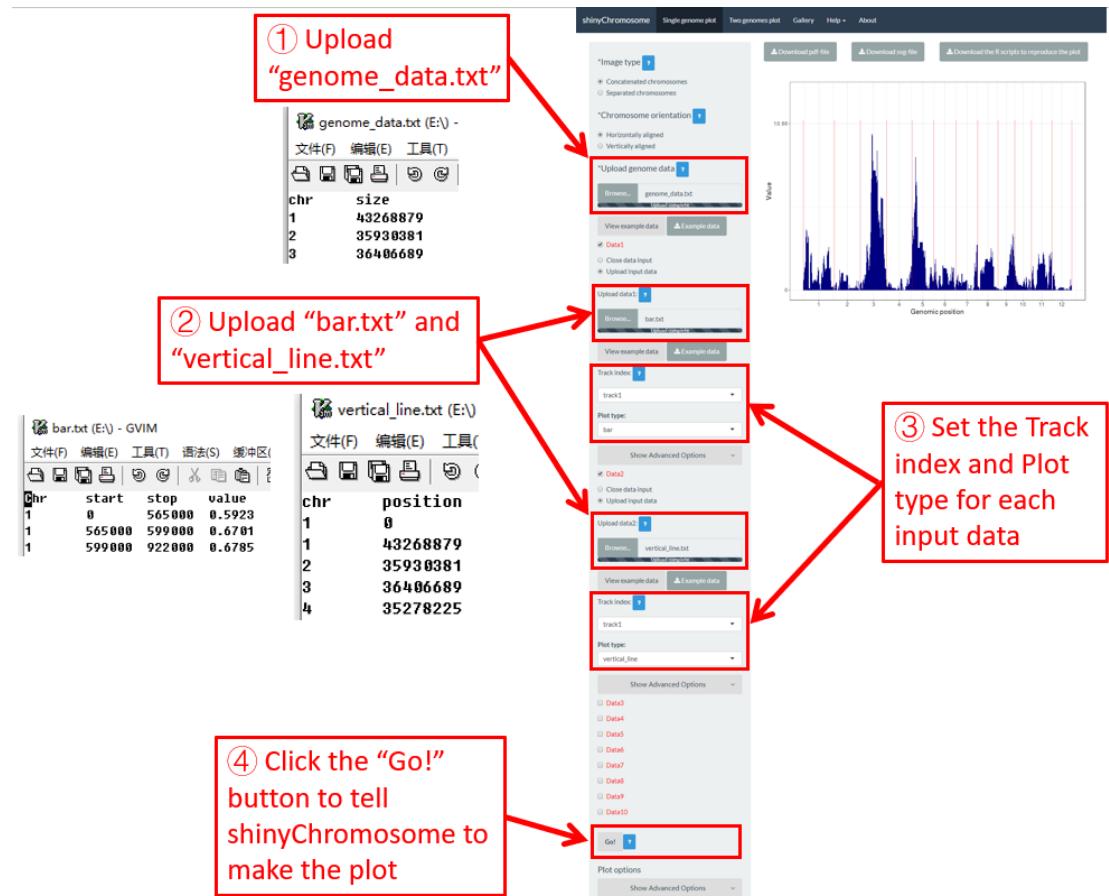


Figure 27. The procedure to plot vertical_line using shinyChromosome.

4.6.11 Plot horizontal_line

Horizontal lines are usually mixed with other types of plot. The input data to create horizontal lines should include **one column** representing the Y-axis coordinate of each horizontal line.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt), “horizontal_line.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/horizontal_line.txt) and “bar.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/bar.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot horizontal line using shinyChromosome (Figure 28).

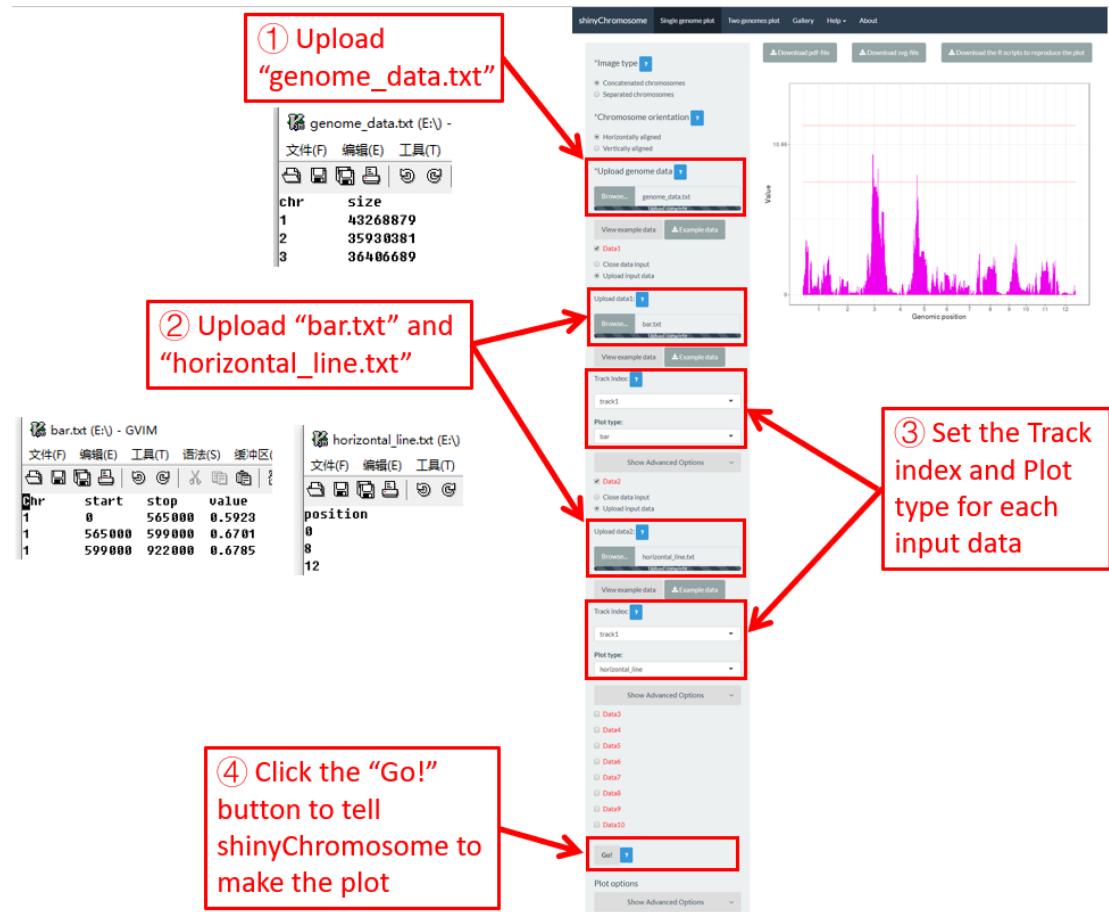


Figure 28. The procedure to plot horizontal_line using shinyChromosome.

4.6.12 Plot ideogram

Ideogram is a schematic representation of chromosomes. The input data to create ideogram should contain 5 **columns**. Please check <https://www.nature.com/scitable/topicpage/chromosome-mapping-idiograms-302> for more information. To plot ideogram using shinyChromosome, we need two input files, the genome data file and the input file to create ideogram.

Here, we use the example files “genome_data.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/genome_data.txt) and “ideogram.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/single_genome/ideogram.txt) provided in the source code of shinyChromosome to demonstrate the procedure to plot ideogram using shinyChromosome (**Figure 29**). Please be noted that the “Image type” is set as “Separated chromosomes” to split the 12 chromosomes.

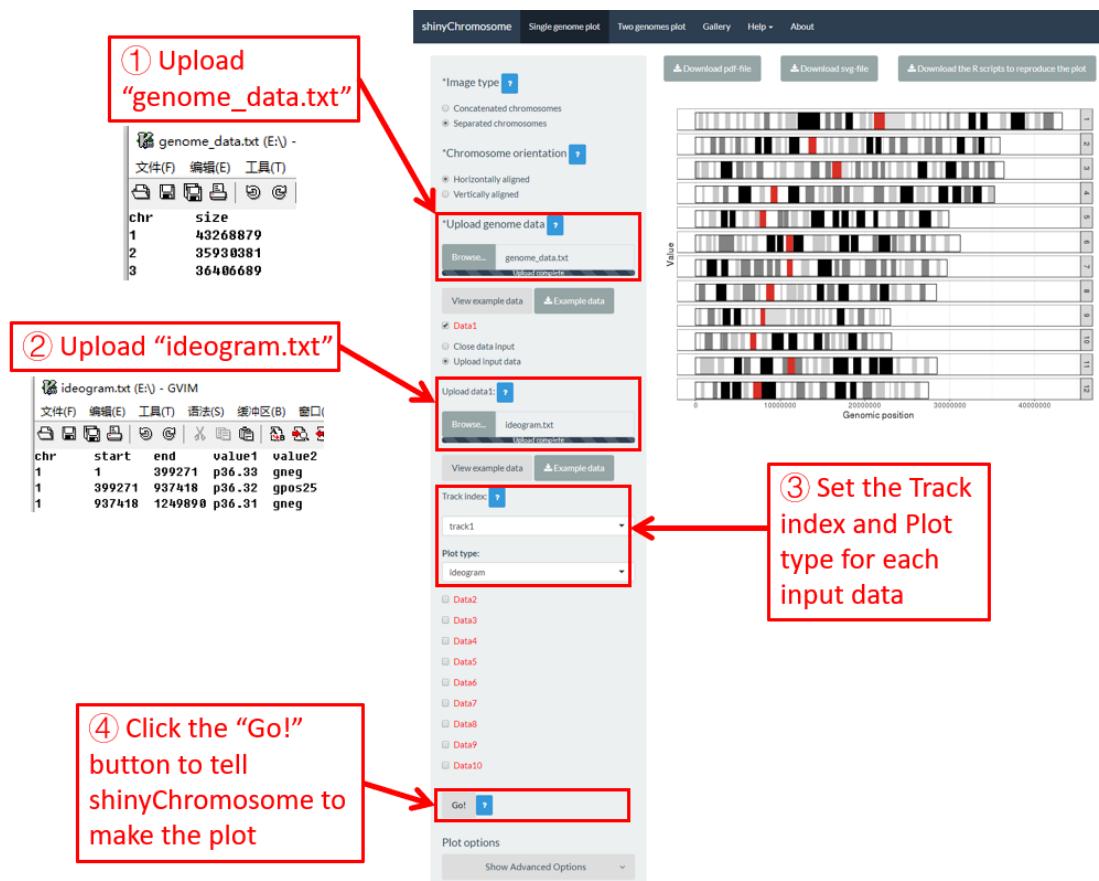


Figure 29. The procedure to plot ideogram using shinyChromosome.

4.7 Integration of multiple input datasets to create advanced single genome plot using shinyChromosome

In section 4.6, we demonstrated the procedure to create different types of single genome plot using shinyChromosome. To make things simple, we create a single type of plot in each example at a time in section 4.6. Actually, shinyChromosome accepts as many as 10 input datasets to create a single genome plot. Each input dataset can be used to create any of the 12 different types of plot. Here, we use the example dataset “Example 1” provided in the “Gallery” menu of shinyChromosome to demonstrate the procedure to create advanced single genome plot using shinyChromosome (**Figure 30**).

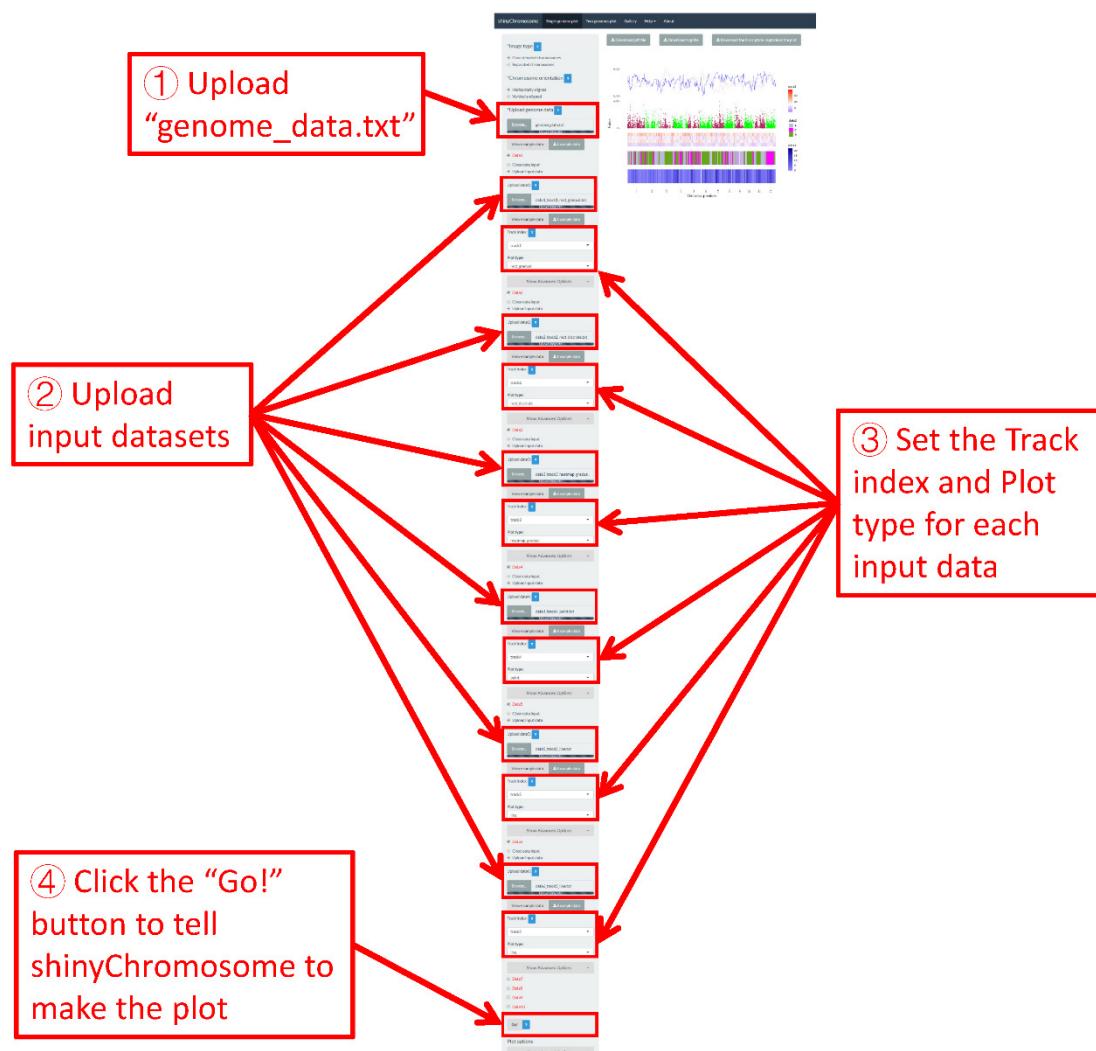


Figure 30. The procedure to create advanced single genome plot with multiple input datasets using shinyChromosome.

A total of 6 input datasets are distributed in 5 tracks of the generated plot. The uploading of the 6 input datasets and the setting of the track index and plot type for each dataset are shown in **Figure 30**. Except for the procedure demonstrated in **Figure 30**, other options were tuned to create the plot, including the display of figure legends, setting of figure theme and the size of each track. Setting of these options are shown in **Figure 31**.

The figure consists of four panels. The first three panels show the configuration for three tracks:

- Data1:** Shows 'Upload data1' (data1_track1_rect_gradual.txt), 'Track index' (track1), and 'Plot type' (rect_gradual). A red box highlights the 'Color legend' section for 'data1'.
- Data2:** Shows 'Upload data2' (data2_track2_rect_discrete.txt), 'Track index' (track2), and 'Plot type' (rect_discrete). A red box highlights the 'Color legend' section for 'data2'.
- Data3:** Shows 'Upload data3' (data3_track3_heatmap_gradual.txt), 'Track index' (track3), and 'Plot type' (heatmap_gradual). A red box highlights the 'Color legend' section for 'data3'.

The fourth panel, titled 'Plot options', contains global settings for all tracks:

- Figure theme:** Set to 'theme6' (highlighted by a red box).
- Adjust plot size:** Includes sliders for 'Track height' (0.01 to 1.0) and 'Track margin' (0.01 to 1.0) for each track, with sections for 'Track2', 'Track3', and 'Track4' through 'Track10'.
- Font size:** Sliders for 'Axis title', 'X axis label', and 'Legend'.
- Track1:** Toggled on.

Figure 31. Settings of various options to decorate the plot created in Figure 30 using shinyChromosome.

4.8 Plotting options to decorate a single genome plot

Various widgets are provided in the left panel of the “Single genome plot” menu under each “DataX” checkbox to decorate the appearance of the generated single genome plot. The following section will demonstrate the setting of some of these options.

4.8.1 Concatenated chromosomes v.s. Separated chromosomes

All the chromosomes in a single genome plot can either be concatenated in sequential order or can be separated in different panels by setting the “Image type” widget at the top of the left panel of the “Single genome plot” menu. The input datasets are the same for “Example 10” and “Example 12” displayed in the “Gallery” menu of the shinyChromosome application. All the chromosomes are concatenated in sequential order in the plot shown in “Example 10” while all the chromosome are separated in different panels in the plot shown in “Example 12”. The procedure and the setting of options to create the plot in “Example 10” and “Example 12” are demonstrated in **Figure 32** and **Figure 33**, respectively.

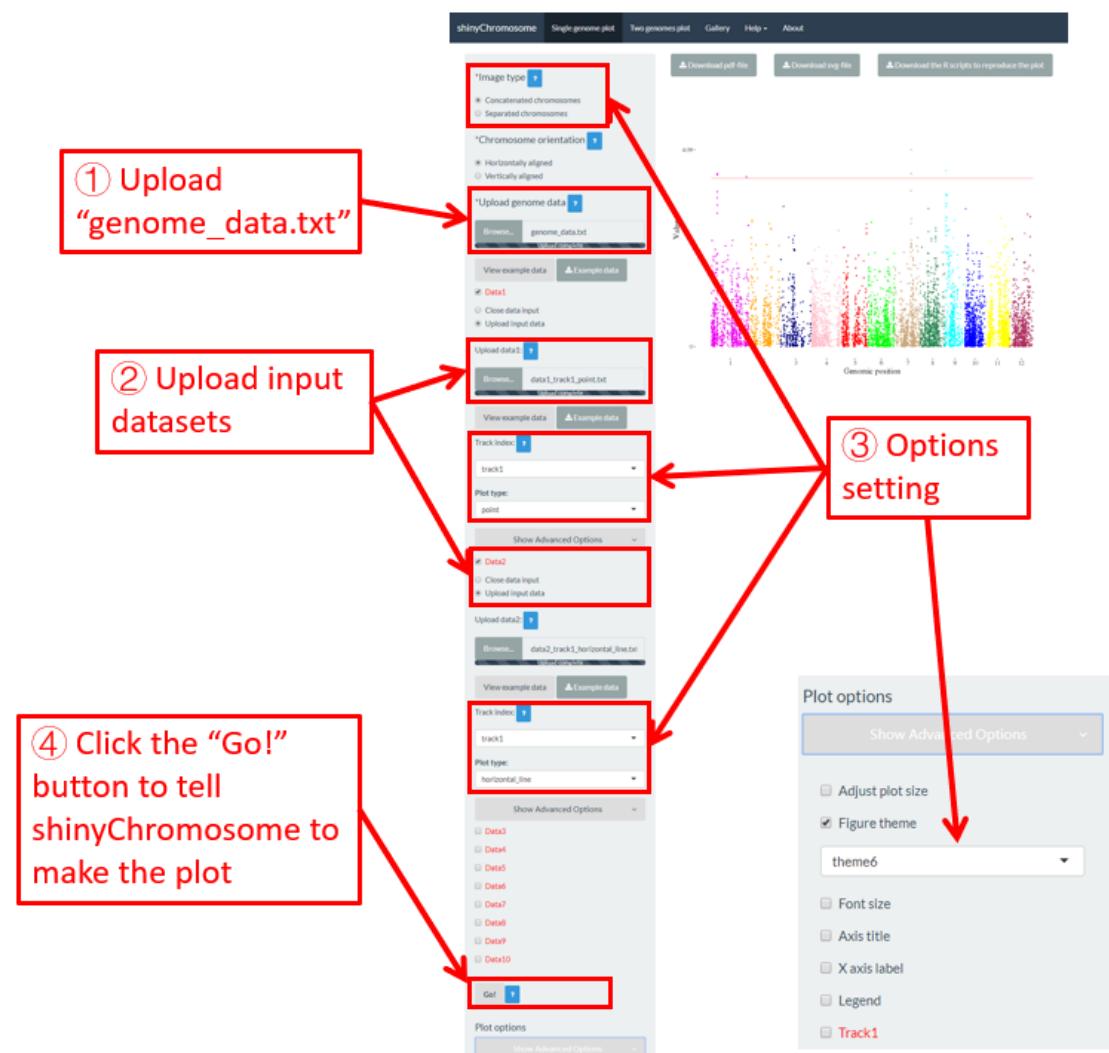


Figure 32. A single genome plot with concatenated chromosomes created using the input dataset of “Example 10” in the “Gallery” menu.

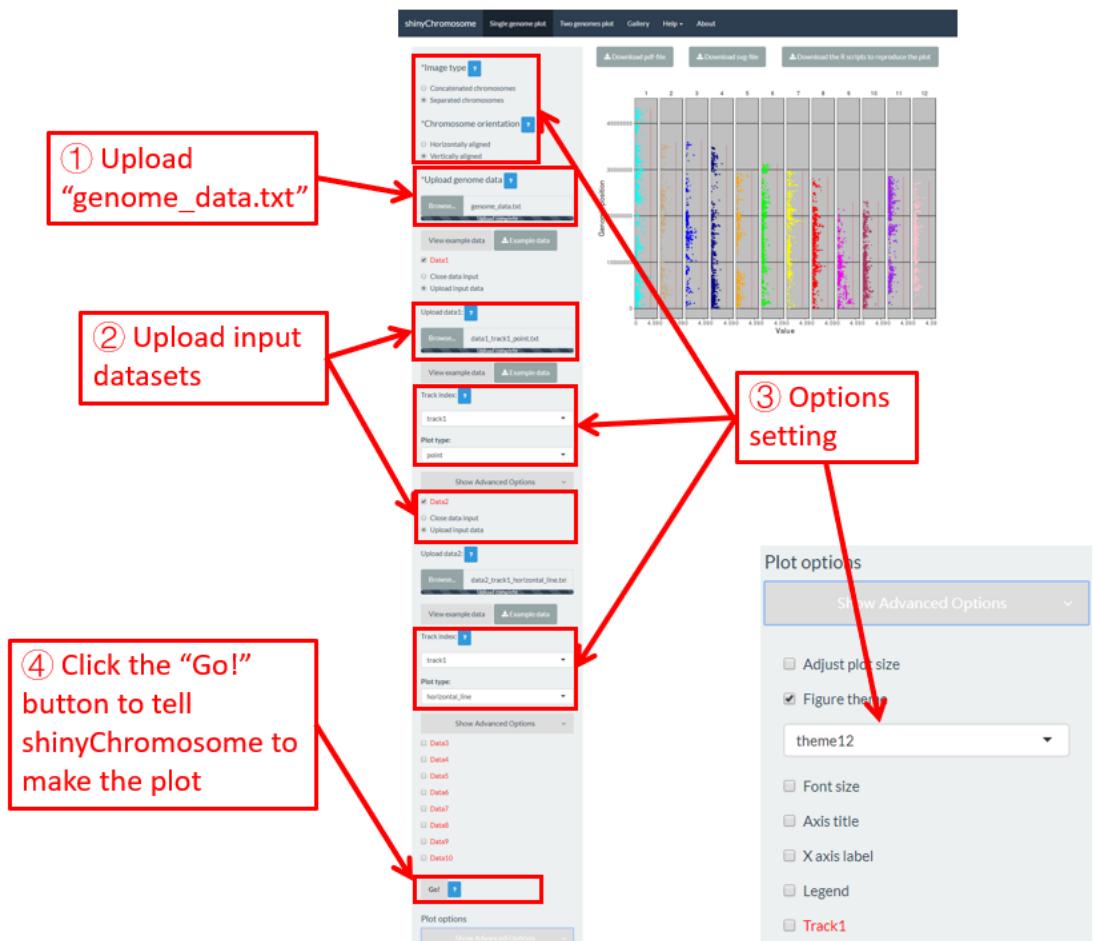


Figure 33. A single genome plot with separated chromosomes created using the input dataset of “Example 12” in the “Gallery” menu.

4.8.2 Horizontally aligned chromosomes v.s. vertically aligned chromosomes

All the chromosomes in a single genome plot can either be aligned along the horizontal axis or be aligned along the vertical axis by setting the “Chromosome orientation” widget at the top of the left panel of the “Single genome plot” menu. The input datasets are the same for “Example 10” and “Example 12” displayed in the “Gallery” menu of the shinyChromosome application. All the chromosomes are aligned along the horizontal axis in the plot shown in “Example 10” while all the chromosomes are aligned along the vertical axis in the plot shown in “Example 12”. The procedure and the setting of options to create the plot in “Example 10” and “Example 12” are demonstrated in **Figure 32** and **Figure 33**, respectively.

4.8.3 Set point color, point size and point symbol

For a point plot, we can modify the point color, point size and point symbol using the widgets provided in the left panel of the “Single genome plot” menu. Here, we use the input datasets of “Example 17” displayed in the “Gallery” menu to demonstrate these widgets.

By default, random color and predefined size and symbol would be assigned to the points as is shown in **Figure 34**. If we want to change the point color, point size or point symbol, we can edit the default values of these widgets under the “Color”, “Symbol” and “Size” checkbox, as is shown in **Figure 35**.

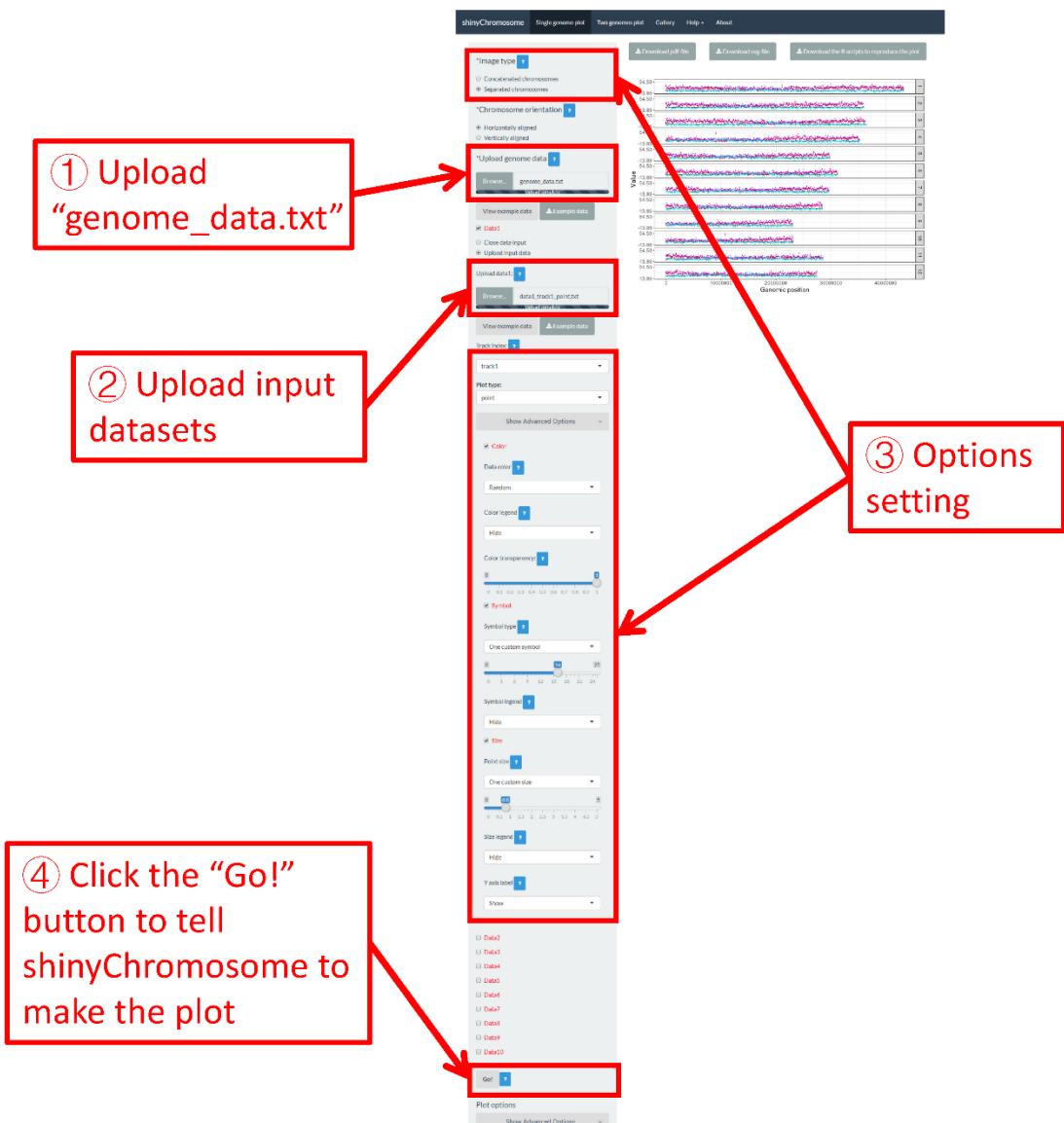


Figure 34. Default settings of point color, point shape and point size in shinyChromosome.

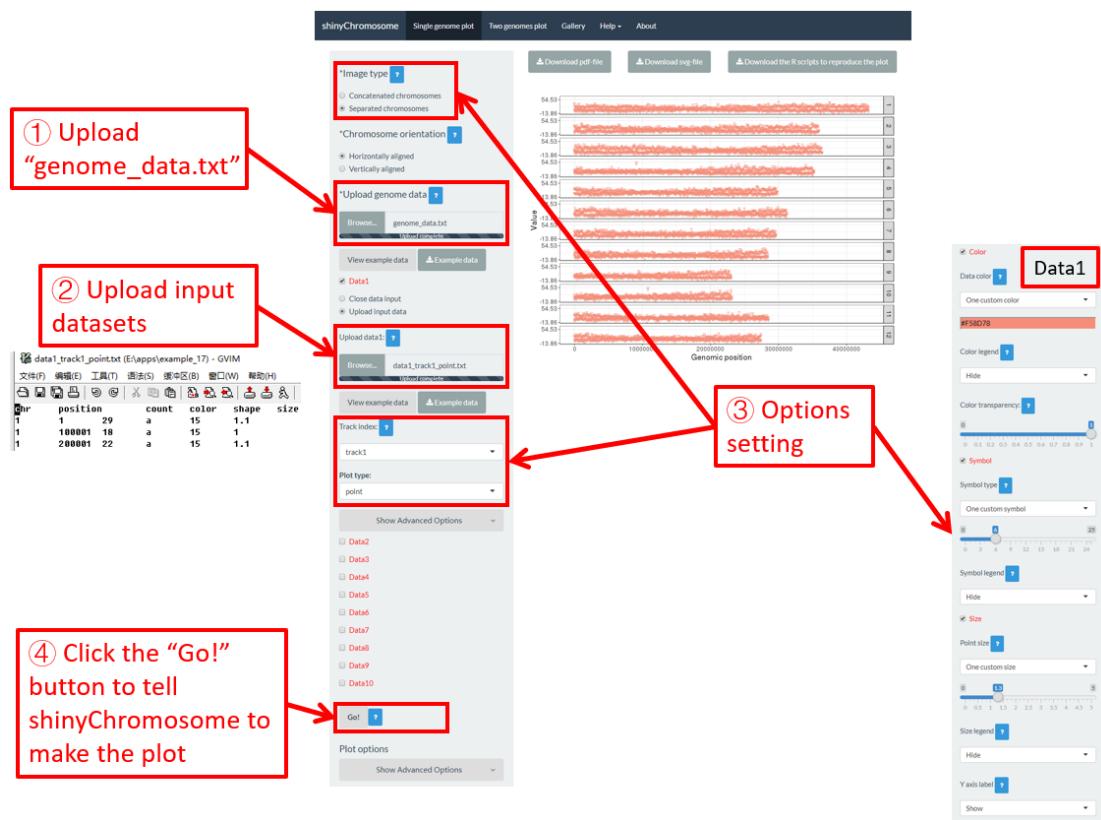


Figure 35. Settings of point color, point shape and point size using different widgets in shinyChromosome.

Actually, the input dataset of “Example 17” contains a “color” column, a “shape” column and a “size” column to assign the color, symbol and the size of the points. To set the point color, point size and point symbol using the “color” column, the “shape” column and the “size” column inside the input dataset, we need to set the values of widgets under the “Color”, “Symbol” and “Size” checkbox, as is shown in **Figure 36**.

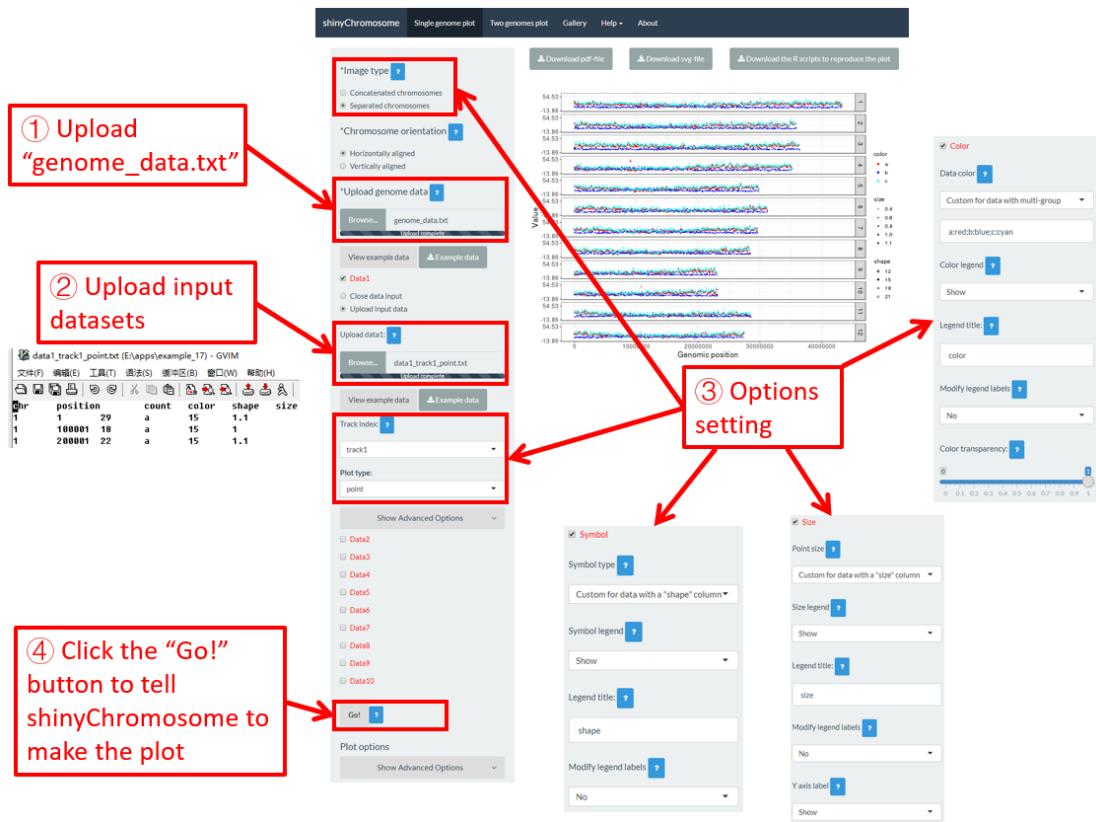


Figure 36. Settings of point color, point shape and point size using the “color”, “symbol” and “size” columns of input dataset in shinyChromosome.

4.8.4 Set rect color for multiple datasets

For a rect_discrete plot, we can set the color of rectangles belonging to different groups using the widgets provided in the left panel of the “Single genome plot” menu. Here, we use the input datasets of “Example 37” displayed in the “Gallery” menu to demonstrate these widgets. Three input datasets are uploaded to make rect_discrete plot. The fourth column of each dataset is a “color” column to define the group of each genomic region, which will be assigned different colors. By default, random colors would be assigned as is shown in **Figure 37**. If we want to set the colors of different data group, we can use the “color” widget of each dataset. The procedure is shown in **Figure 38**, including the settings of various options.

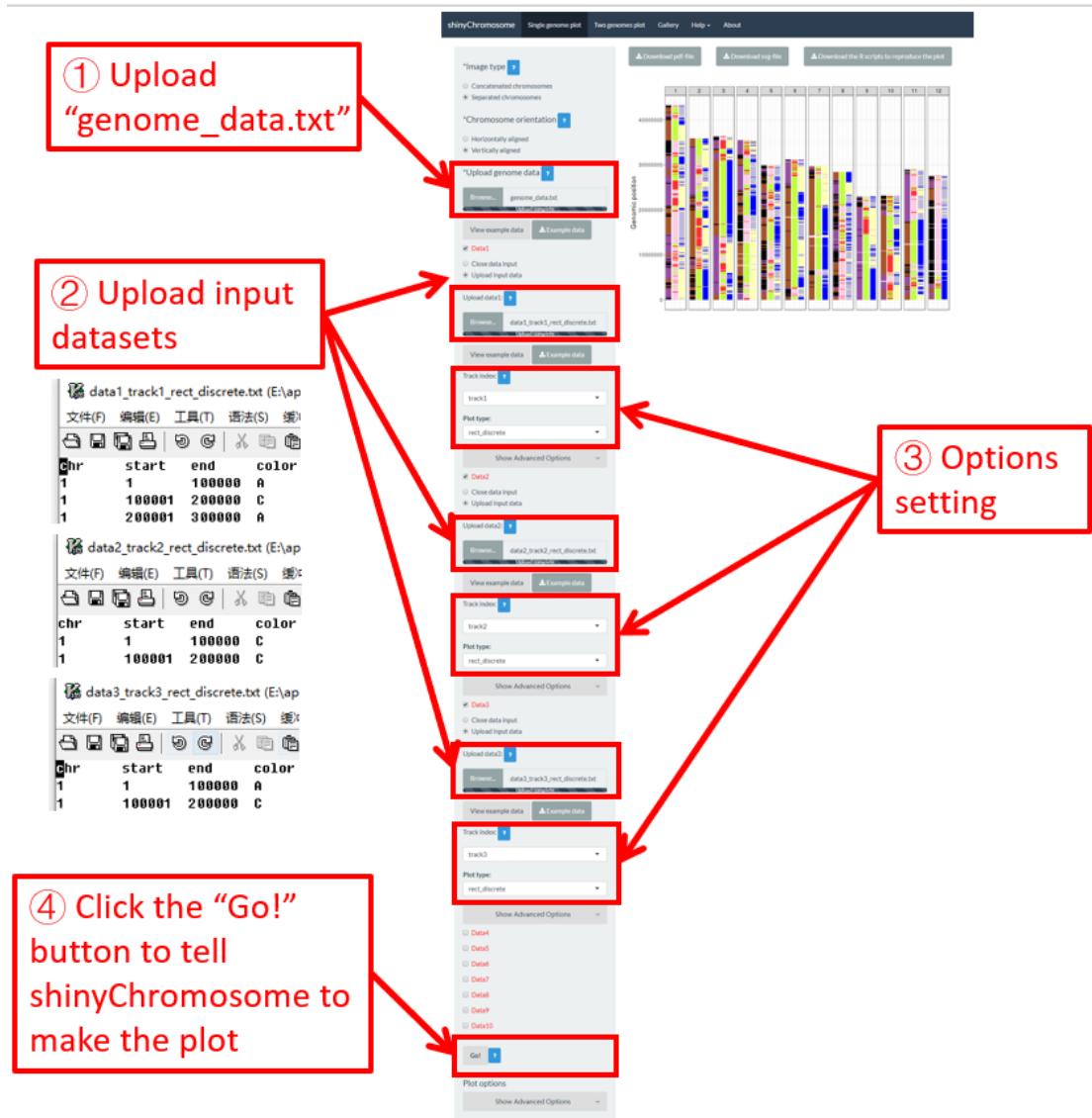


Figure 37. Default settings of rect colors in shinyChromosome.

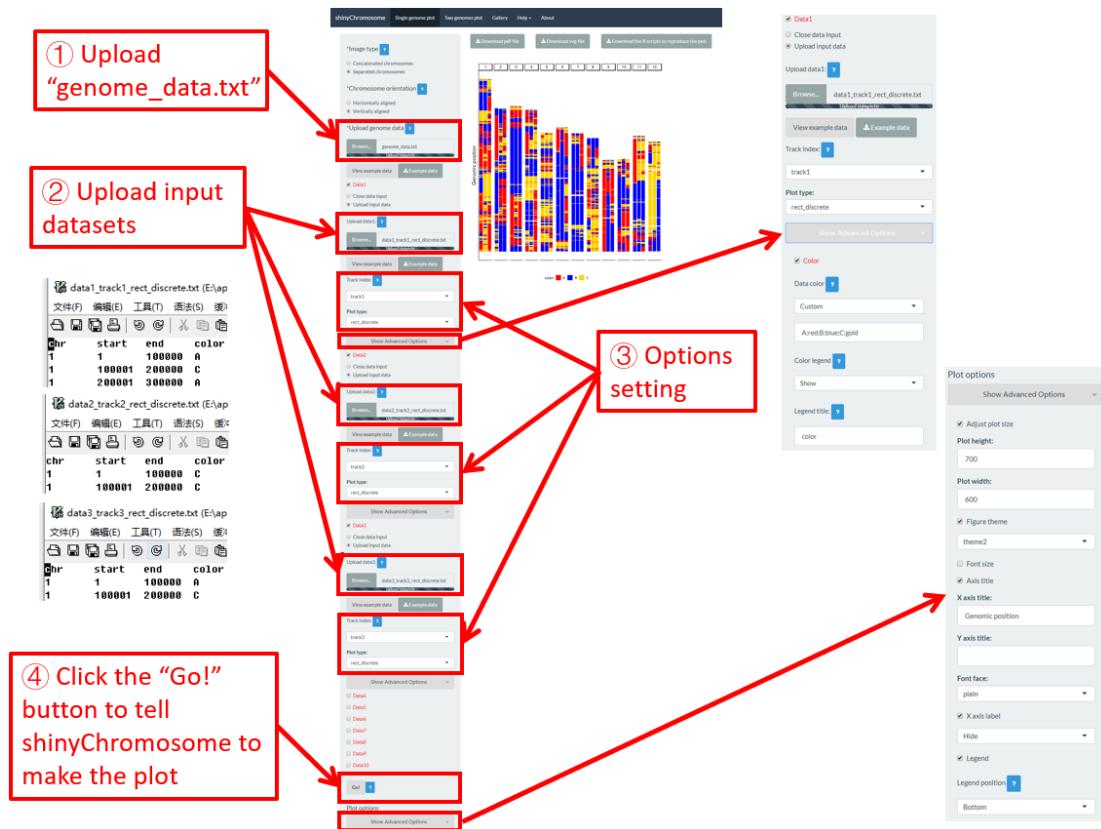


Figure 38. Settings of rect colors using the “color” column of input dataset in shinyChromosome.

5. Creation of Non-circular two genomes plot using shinyChromosome

To create a non-circular two genomes plot, you need to use the “Two genomes plot” menu of the shinyChromosome application. **Three datasets are required to create a two genomes plot.** 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. In the following section, we demonstrate all the essential steps to create a non-circular two genomes plot using shinyChromosome with example datasets.

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

Step 1. Prepare and upload the input file of the genome data aligned along the horizontal axis

The format of the genome data is the same as the genome data illustrated in section 4.1. An example dataset is available at https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/two_genome/genome1_data.txt. This example dataset is used to create the plot in **Figure 39**. This input file should be uploaded using the “Upload genome1 data” widget in the left panel of the “Two genomes plot” menu.

Step 2. Prepare and upload the input file of the genome data aligned along the vertical axis

The format of the genome data is the same as the genome data illustrated in section 4.1. An example dataset is available at https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/two_genome/genome2_data.txt. This example dataset is used to create the plot in **Figure 39**. The input files used in Step 1 and Step 2 can either be the same file or different files. This input file should be uploaded using the “Upload genome2 data” widget in the left panel of the “Two genomes plot” menu.

Step 3. Prepare and upload the main dataset

The detailed file format of the main dataset used to create a “Two genomes plot” is described in the “Input data format” menu of the shinyChromosome application. Here, we use the example dataset “point_gradual.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/two_genome/point_gradual.txt) to create the plot in **Figure 39**.

Step 4. Set the plot type for the main dataset

Here, the plot type is set as “point_gradual” (Figure 39).

Step 5. Click the “Go!” button to make the plot

After all the input datasets has been successfully uploaded to the shinyChromosome application, we need to click the “Go!” button at the bottom of the left panel of the “two genomes plot” menu to tell shinyChromosome to make the plot (Figure 39). The plot shown in the main panel of Figure 39 is the plot generated using the input datasets uploaded in Step 1, Step 2 and Step 3. **By default, random color or predefined colors would be used by shinyChromosome when generating the plot. Remember to click the “Go!” button to update the plot whenever you modify any option or input file through the diverse widgets provided in the left panel.**

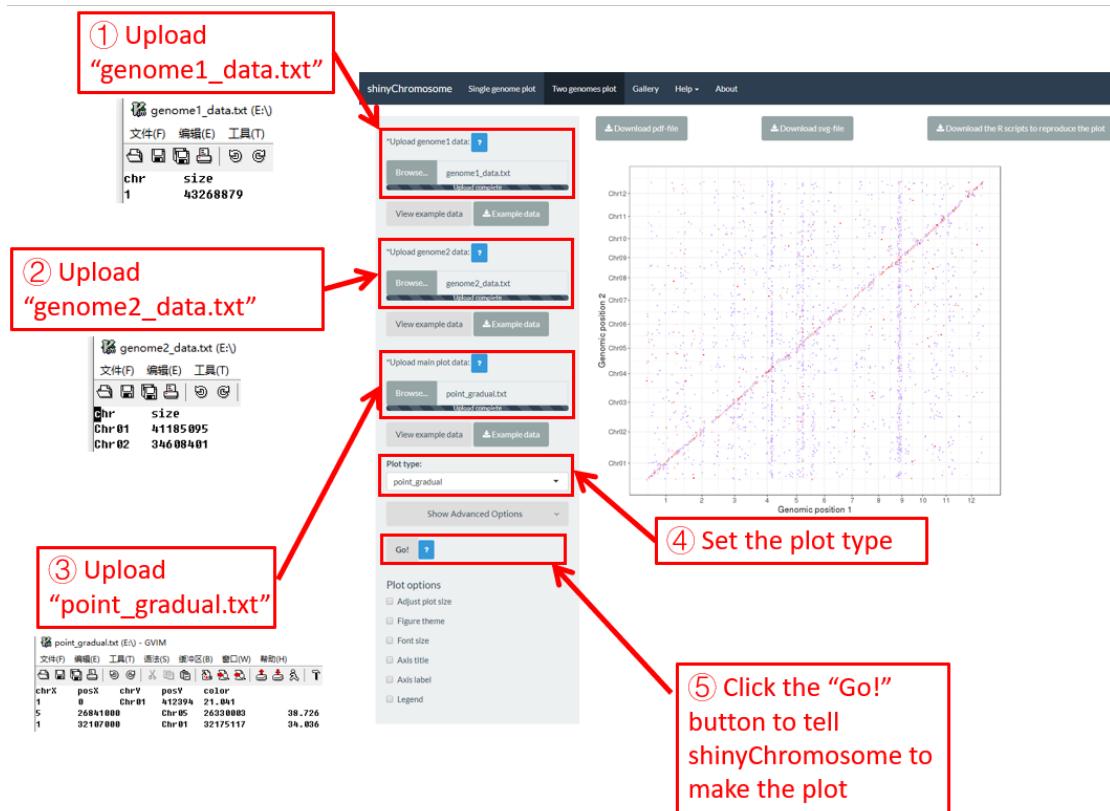


Figure 39. Essential steps to create a two genomes plot using shinyChromosome.

5.2 Create different types of two genomes plot using shinyChromosome

A total of 5 different types of plot can be created using shinyChromosome, including point_gradual, point_discrete, segment, rect_gradual and rect_discrete. To create a two genomes plot, at least three input data files are needed. The detailed format of input files to make a two

genomes plot is demonstrated in the “Input data format” menu (under the “Help” menu) of the shinyChromosome application. In this section, we will show the key parameters to make different types of two genomes plot using the graphical interface of shinyChromosome with example input datasets. The two example genome data files used in this section is the same as the file used in section

5.1

(https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/wo_genome/genome1_data.txt,

https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/two_genome/genome2_data.txt).

5.2.1 Plot point_gradual

The input dataset should contain 5 columns.

1st column: chromosome ID of genome along the horizontal axis.

2nd column: chromosome position in genome along the horizontal axis.

3rd column: chromosome ID of genome along the horizontal axis.

4th column: chromosome position in genome along the vertical axis.

5th column: a numeric vector defining the value of each point.

The procedure to plot point_gradual using shinyChromosome is demonstrated in **Figure 39**.

5.2.2 Plot point_discrete

The input dataset should contain 5 columns.

1st column: chromosome ID of genome along the horizontal axis.

2nd column: chromosome position in genome along the horizontal axis.

3rd column: chromosome ID of genome along the horizontal axis.

4th column: chromosome position in genome along the vertical axis.

5th column: a character vector defining the category of each point.

Here, we use the example dataset “point_discrete.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/wo_genome/point_discrete.txt) provided in the source code of the shinyChromosome application.

The procedure to plot point_discrete using shinyChromosome is demonstrated in **Figure 40**.

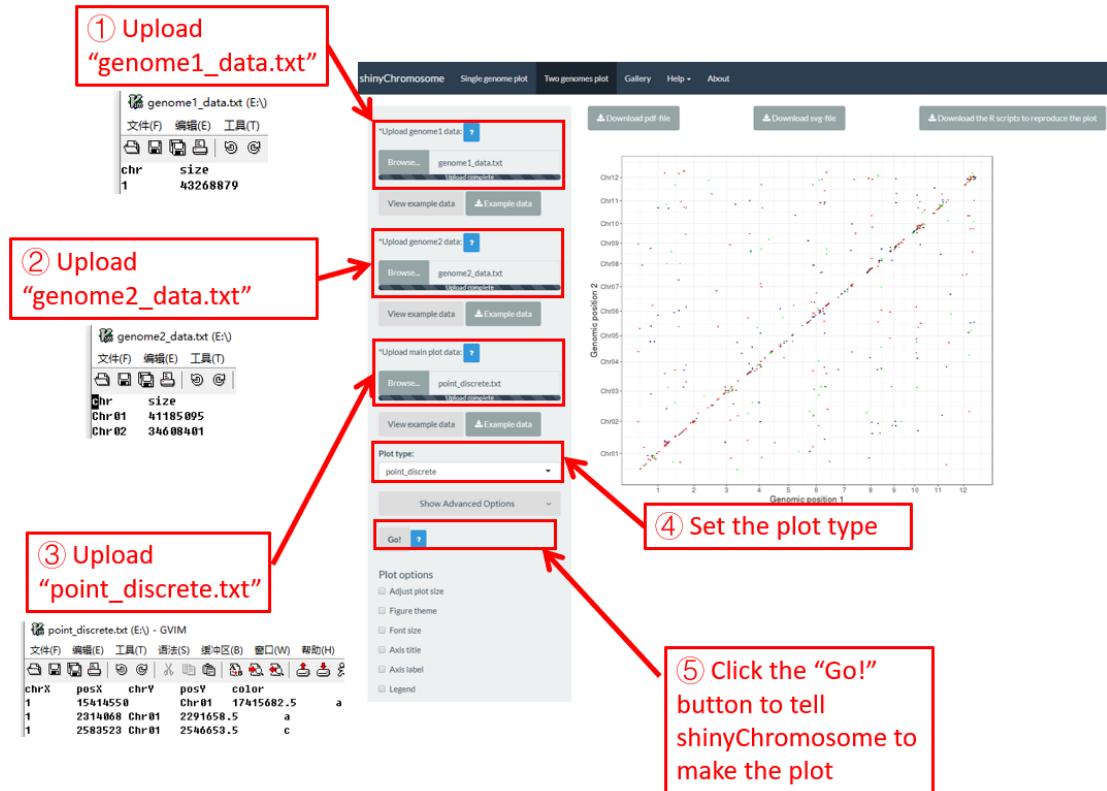


Figure 40. The procedure to plot point_discrete using shinyChromosome.

5.2.3 Plot segment

The dataset should contain ≥ 6 columns. In the simplest situation, the dataset should contain 6 columns with fixed order.

1st column: chromosome ID of genome along the horizontal axis.

2nd column: X-axis start coordinate of segments.

3rd column: X-axis end coordinate of segments.

4th column: chromosome ID of genome along the vertical axis.

5th column: Y-axis start coordinate of segments.

6th column: Y-axis end coordinate of segments.

Here, we use the example dataset “segment.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/wo_genome/segment.txt) provided in the source code of the shinyChromosome application. The procedure to plot segment using shinyChromosome is demonstrated in **Figure 41**.

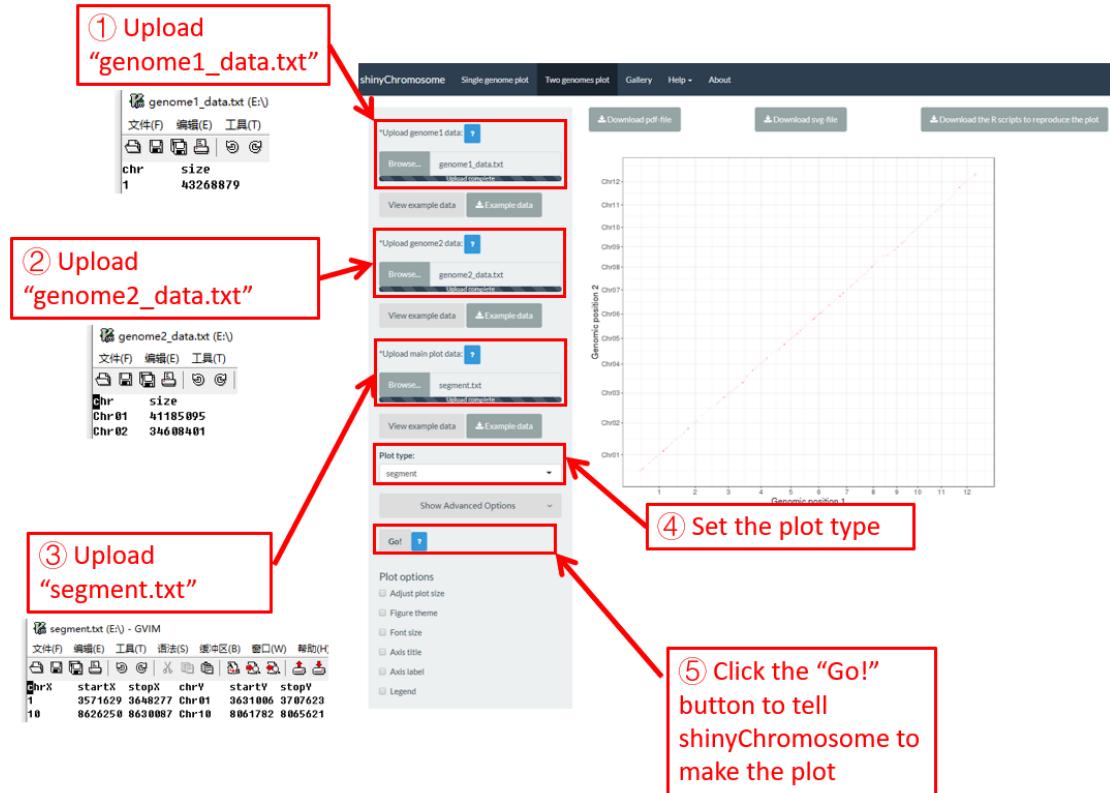


Figure 41. The procedure to plot segment using shinyChromosome.

5.2.4 Plot rect_gradual

The dataset should contain 7 columns with fixed order.

1st column: chromosome ID of genome along the horizontal axis.

2nd column: X-axis start coordinate of rects.

3rd column: X-axis end coordinate of rects.

4th column: chromosome ID of genome along the vertical axis.

5th column: Y-axis start coordinate of rects.

6th column: Y-axis end coordinate of rects.

7th column: a numeric vector defining the value of each rectangle.

Here, we use the example dataset “rect_gradual.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/wo_genome/rect_gradual.txt) provided in the source code of the shinyChromosome application.

The procedure to plot rect_gradual using shinyChromosome is demonstrated in **Figure 42**.

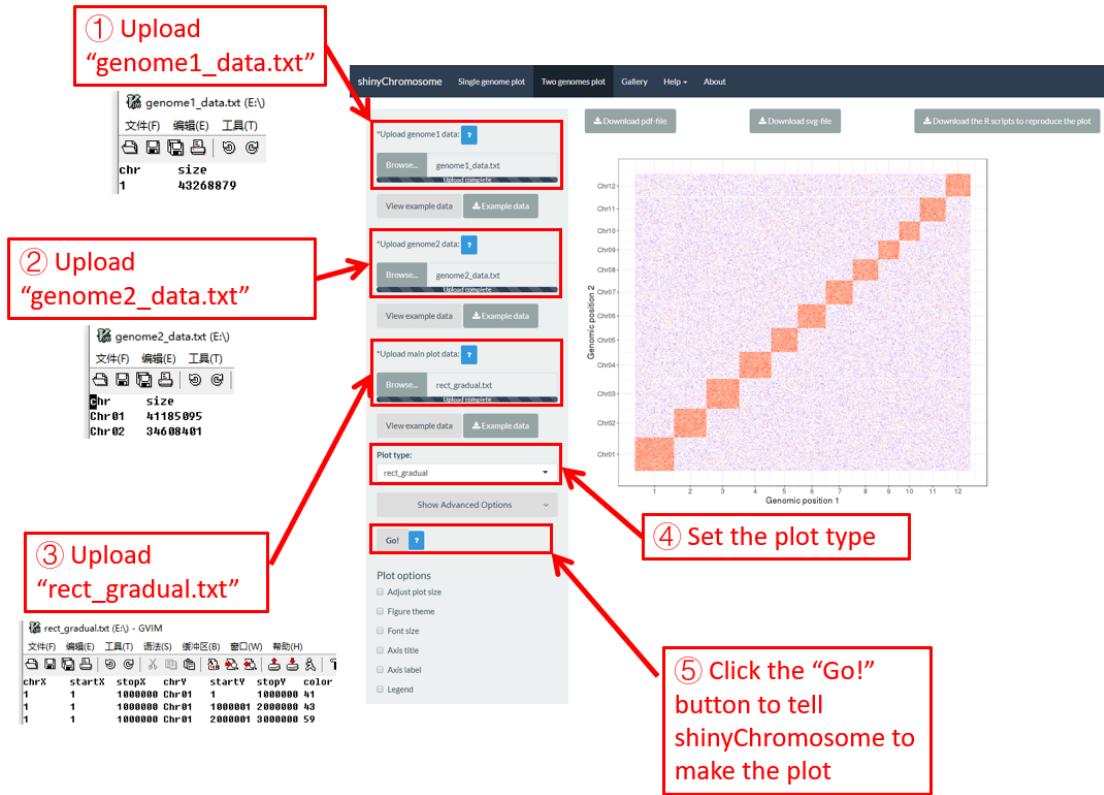


Figure 42. The procedure to plot rect_gradual using shinyChromosome.

5.2.5 Plot rect_discrete

The dataset should contain 7 columns with fixed order.

1st column: chromosome ID of genome along the horizontal axis.

2nd column: X-axis start coordinate of rects.

3rd column: X-axis end coordinate of rects.

4th column: chromosome ID of genome along the vertical axis.

5th column: Y-axis start coordinate of rects.

6th column: Y-axis end coordinate of rects.

7th column: a character vector defining the category of each rectangle.

Here, we use the example dataset “rect_discrete.txt” (https://github.com/venyao/shinyChromosome/blob/master/www/data/download_example_data/wo_genome/rect_discrete.txt) provided in the source code of the shinyChromosome application.

The procedure to plot rect_discrete using shinyChromosome is demonstrated in **Figure 43**.

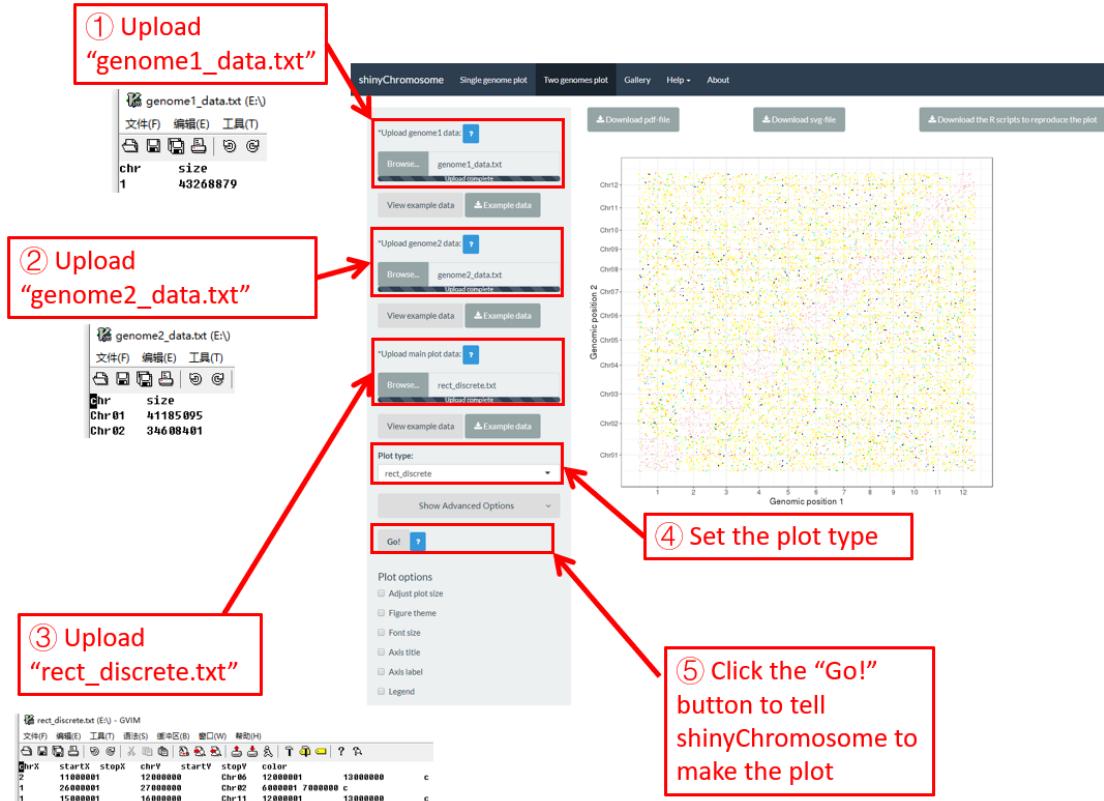


Figure 43. The procedure to plot rect_discrete using shinyChromosome.

6. Decorate the appearances of non-circular whole genome plot created by shinyChromosome

Several widgets are provided in the left panel of the “Single genome plot” and “Two genomes plot” menus to decorate the appearances of the plot generated using shinyChromosome, including figure size, figure theme, font size, axis title, axis label and legend, etc.

6.1 Figure size

Users can adjust the height and width of a single genome plot using the “Adjust plot size” widget under the “Show advanced options” widget on the bottom of the left panel of the “Single genome plot” menu. The figure size in both the browser and the downloaded PDF/SVG files would be affected. Here, we use the example datasets demonstrated in section 4.6.1 to show this function. The default height and width of a single genome plot are 550 and 750, which is modified as 250 and 800 in **Figure 44**. Finally, we need to click the “Go!” button to tell shinyChromosome to update the plot.

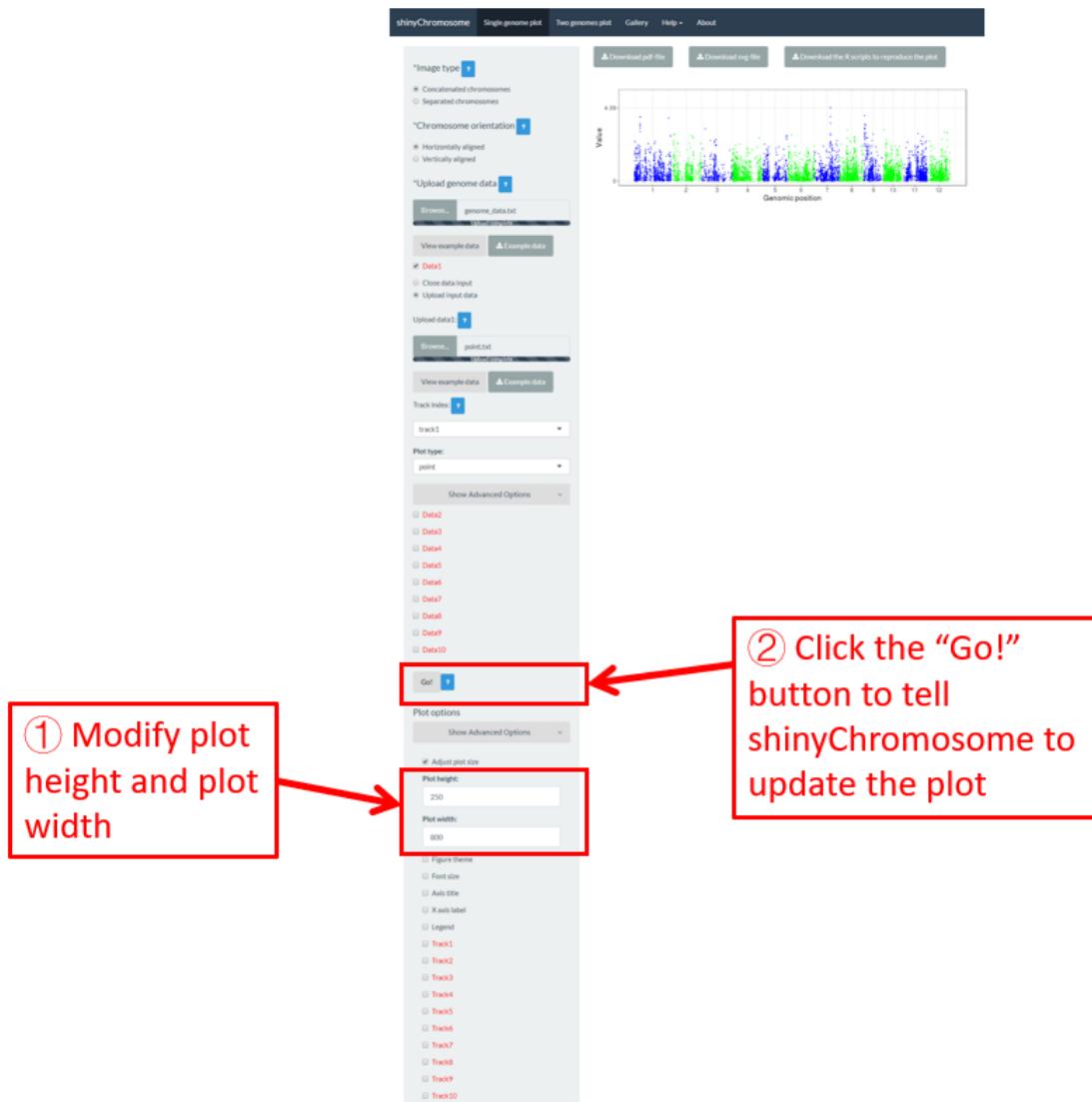


Figure 44. The procedure to modify the height and width of a single genome plot using shinyChromosome.

6.2 Figure theme

shinyChromosome use the ggplot2 graphics system as the engine to create single genome plot and two genomes plot. The ggplot2 package provides lots of options to tune the appearances of the plot created using ggplot2. A set of options with predefined values is called a figure theme in ggplot2. This allows changing the overall appearance of a plot generated using ggplot2 with a single command. The ggthemes package is an R package with tens of different themes used to tune the appearance of a plot created using ggplot2. All the themes provided by ggthemes is available at <https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/>. The ggthemes package

is used in shinyChromosome to decorate the appearance of the single genome plot and two genomes plot generated using shinyChromosome. Here, we use the example datasets demonstrated in section 4.6.1 to show this function. The default figure theme of a single genome plot is “theme1” in shinyChromosome, which is modified as “theme18” in **Figure 45**. Finally, we need to click the “Go!” button to tell shinyChromosome to update the plot.

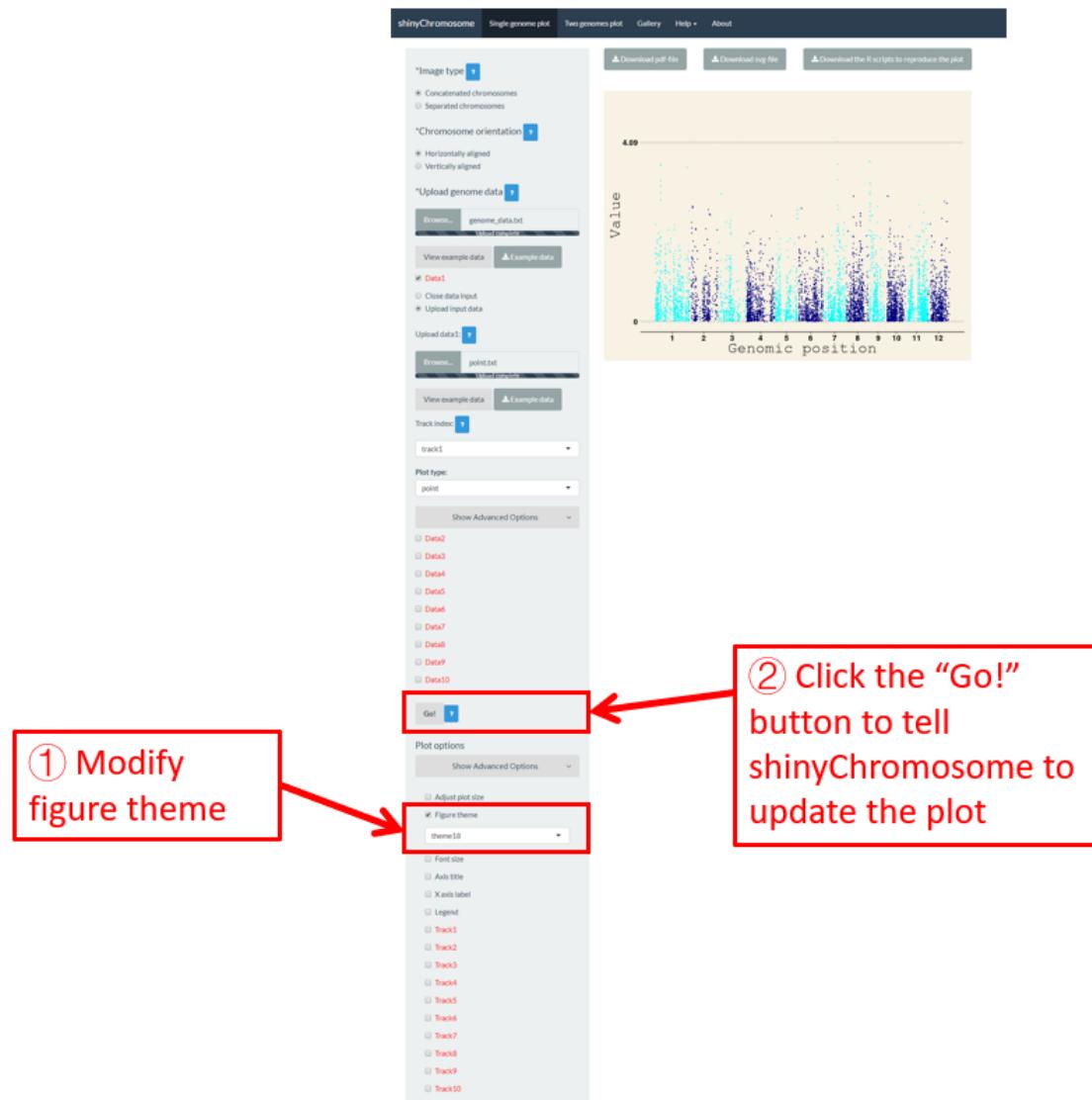


Figure 45. The procedure to modify the theme of a single genome plot using shinyChromosome.

6.3 Font size

The “Font size” widget on the bottom of the left panel of the “Single genome plot” and “Two genomes plot” menus can be used to tune the font size of the plot created using shinyChromosome, including the font size of axis titles and axis tick labels. Here, we use the example datasets

demonstrated in section 4.6.1 to show this function. The default font size is 16, which is modified as 30 in **Figure 46**. Finally, we need to click the “Go!” button to tell shinyChromosome to update the plot.



Figure 46. The procedure to modify the font size of a single genome plot using shinyChromosome.

6.4 Axis title

The “Axis title” widget on the bottom of the left panel of the “Single genome plot” and “Two genomes plot” menus can be used to tune the axis titles of the plot created using shinyChromosome, including the X-axis title, Y-axis title and the font face of axis titles. Here, we use the example datasets demonstrated in section 4.6.1 to show this function. The default axis titles are modified in

Figure 47. Finally, we need to click the “Go!” button to tell shinyChromosome to update the plot.



Figure 47. The procedure to modify the axis titles of a single genome plot using shinyChromosome.

6.5 Legend

A legend can be added to the right or the bottom of a non-circular whole genome plot using shinyChromosome to indicate the meanings of different colors or different symbols in the plot. The user can choose to show or hide the legend. A total of 7 widgets are provided at the bottom of the left panel of the “Single genome plot” menu to add a legend to the generated plot. The meaning of these widgets is shown as follows.

Legend position: The position to place the legend.

Legend region size: Percent of legend size relative to the main plotting region. Applicable values

are numbers in [0-1].

Intra-spacing: Intra-spacing between different legends.

Title font size: The font size of legend title.

Title font face: The font face of legend title.

Label font size: The font size of legend tick label.

Label font face: The font face of legend tick label.

Here, we use the input datasets of “Example 17” displayed in the “Gallery” menu to demonstrate these widgets (**Figure 48**).

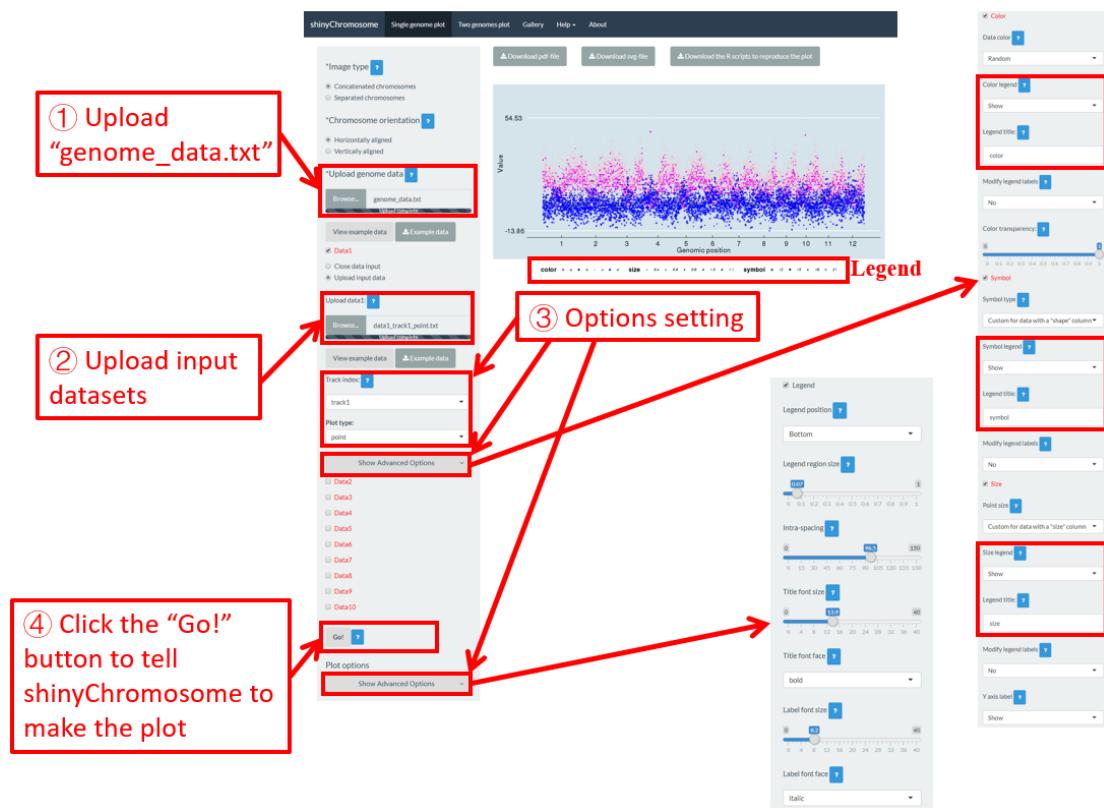


Figure 48. The procedure to add legend to the bottom of a single genome plot using shinyChromosome.

6.6 Height and width of different tracks

For a single genome plot, we can modify the height and margin size of each track to tune the size of each track in the generated plot. Here, we use the input datasets in section 4.1 to demonstrate the setting of these options (**Figure 49**).

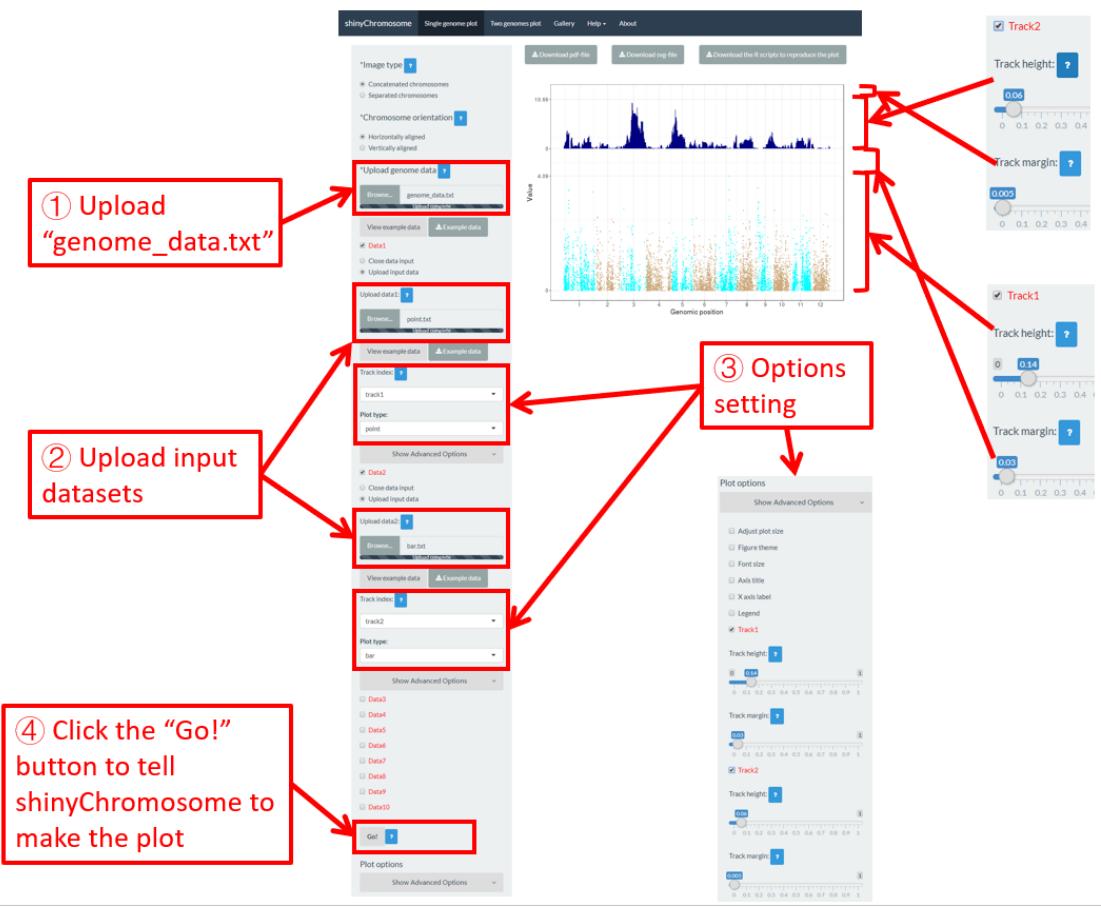


Figure 49. The procedure to modify the height and margin size of each track of a single genome plot using shinyChromosome.