

# Package ‘chromDraw’

November 16, 2019

**Type** Package

**Title** chromDraw is a R package for drawing the schemes of karyotypes in the linear and circular fashion.

**Version** 2.17.0

**Date** 2016-06-02

**Author** Jan Janecka, Ing., Mgr. CEITEC Masaryk University

**Maintainer** Jan Janecka <jan.janecka@ceitec.muni.cz>

**biocViews** Software

**Depends** R (>= 3.0.0)

**SystemRequirements** Rtools (>= 3.1)

**Description** ChromDraw is a R package for drawing the schemes of karyotype(s) in the linear and circular fashion. It is possible to visualize cytogenetic marks on the chromosomes. This tool has its own input data format. Input data can be imported from the GenomicRanges data structure. This package can visualize the data in the BED file format. Here is a requirement on the first nine fields of the BED format. Output file formats are \*.eps and \*.svg.

**License** GPL-3

**Imports** Rcpp (>= 0.11.1), GenomicRanges (>= 1.17.46)

**LinkingTo** Rcpp

**URL** [www.plantcytogenomics.org/chromDraw](http://www.plantcytogenomics.org/chromDraw)

**git\_url** <https://git.bioconductor.org/packages/chromDraw>

**git\_branch** master

**git\_last\_commit** e690d0b

**git\_last\_commit\_date** 2019-10-29

**Date/Publication** 2019-11-15

## R topics documented:

chromDraw-package	2
chromDraw	3
chromDrawGR	3
convertInputColors	4
convertInputData	5
main_chromDraw	6

<b>Index</b>	<b>7</b>
--------------	----------

chromDraw-package      *chromDraw - simple karyotype visualization tool.*

---

## Description

ChromDraw is a R package for drawing the schemes of karyotype(s) in the linear and circular fashion. It is possible to visualize cytogenetic marks on the chromosomes. This tool has its own input data format. Input data can be imported from the GenomicRanges data structure. This package can visualize the data in the BED file format. Here is a requirement on the first nine fields of the BED format. Output file formats are \*.eps and \*.svg.

## Details

Package: chromDraw  
Type: Package  
Date: 2015-12-12  
License: GPL-3  
License: LGPL | library Board

This package has only one function with two parameters, such as the main function in C. The first parameter ARGC is a count of input parameters and the second parameter ARGV is a vector containing calling parameters. Example: `chromDraw(argc=5, argv=c("chromDraw", "-c", "/home/user/Documents/colors.txt", "-d", "/home/user/Documents/input_data.txt"))`

The first string in the vector with arguments must be the package name. The other strings in the vector are parameter strings and strings with parameter values. As shown in the example.

## Author(s)

Jan Janečka, Ing. Mgr. CEITEC MU Masaryk University Kamenice 5, Building A26 CZ-625 00, Brno Czech Republic tel.: 549 49 8190

Email: [jan.janecka@ceitec.muni.cz](mailto:jan.janecka@ceitec.muni.cz)

## References

LibBoard: A vector graphics C++ library (Version 0.9.0). GREYC laboratory. [Software]. <http://libboard.sourceforge.net/>. [accessed Sept. 2014].

## See Also

[chromDraw](#) [main\\_chromDraw](#)

## Examples

```
OUTPUTPATH = file.path(getwd());  
INPUTPATH = system.file('extdata', 'Ack_and_Stenopetalum_nutans.txt', package = 'chromDraw')  
COLORPATH = system.file('extdata', 'default_colors.txt', package = 'chromDraw')  
chromDraw(argc=7, argv=c("chromDraw", "-c", COLORPATH, "-d", INPUTPATH, "-o", OUTPUTPATH));
```

---

chromDraw	<i>Main chromDraw function. This R function calls the main C++ function.</i>
-----------	--

---

**Description**

Main chromDraw function. This R function calls the main C++ function.

**Usage**

```
chromDraw(argc, argv)
```

**Arguments**

argc	count of the input parameters.
argv	vector with the input parameters strings.

**Value**

return exit state.

**Examples**

```
OUTPUTPATH = file.path(getwd());
INPUTPATH = system.file('extdata', 'Ack_and_Stenopetalum_nutans.txt', package = 'chromDraw')
COLORPATH = system.file('extdata', 'default_colors.txt', package = 'chromDraw')
chromDraw(argc=7, argv=c("chromDraw", "-c", COLORPATH, "-d", INPUTPATH, "-o",
OUTPUTPATH));
```

---

chromDrawGR	<i>Function chromDrawGR uses Genomic Ranges as input data format. This R function call the main C++ function.</i>
-------------	---

---

**Description**

Function chromDrawGR uses Genomic Ranges as input data format. This R function call the main C++ function. If the output directory is not set, then it is use working directory for the outputs.

**Usage**

```
chromDrawGR(karyotypes, colors)
```

**Arguments**

karyotypes	vector of the Genomic Ranges structures per karyotype.
colors	data frame definition of coloros, containing color name and RGB of the color values.

**Value**

return exit state.

**Examples**

```

#load package
library(GenomicRanges)

#data generating
karyotype1 <- GRanges(seqnames =Rle(c("Ack1", "Ack2"), c(5, 5)),ranges =
IRanges(start = c(0, 400000,0,3300000,6000000,0,2500000,0,3800000,6400000),
end = c(400000,3300000,0,6000000,10400000,2500000,3800000,0,6400000,14800000),
names = c("A","B","CENTROMERE","C","D","E","F","CENTROMERE","G","H")),
color = c("orange","yellow","","orange","red","green","green","","red","orange"));

karyotype2 <- GRanges(seqnames =Rle(c("Ack3", "Ack4"), c(4, 4)),ranges =
IRanges(start = c(0,0, 2400000,3500000,0,0,2400000,6700000),
end = c(2400000,0,3500000,12200000,2400000,0,6700000,9200000),
names = c("I","CENTROMERE","J","K","L","CENTROMERE","M","N")),
color = c("light_blue","","orange","red","pink","","red","light_blue"));

inputData <- list(karyotype1,karyotype2);

#colors generating
name <- c("yellow", "red", "blue", "violet", "orange", "green", "light_blue", "pink");
r <- c(255, 255, 0, 255, 247, 0, 0, 230);
g <- c(255, 0, 0, 0, 148, 255, 255, 170);
b <- c(0, 0, 255, 255, 29, 0, 255, 160);
inputColors <- data.frame(name,r,g,b);

#run the function with generated data and colors
chromDrawGR(inputData,inputColors);

```

---

convertInputColors	<i>The R function for converting the data frame with colors to chromDraw color data format.</i>
--------------------	---

---

**Description**

The R function for converting the data frame with colors to chromDraw color data format. Output of this function is file "colors.txt" in the working directory.

**Usage**

```
convertInputColors(colors);
```

**Arguments**

colors                    data frame definition of coloros, containing color name and RGB of color values.

**Value**

return file with colors in working directory.

**Examples**

```
#colors generating
name <- c("yellow", "red", "blue", "violet", "orange", "green", "light_blue", "pink");
r <- c(255, 255, 0, 255, 247, 0, 0, 230);
g <- c(255, 0, 0, 0, 148, 255, 255, 170);
b <- c(0, 0, 255, 255, 29, 0, 255, 160);
inputColors <- data.frame(name,r,g,b);

#run the function for generate chromDraw color file.
convertInputColors(inputColors);
```

---

convertInputData	<i>The R function for converting Genomic Ranges data structure to the chromDraw dat format.</i>
------------------	---

---

**Description**

The R function for converting Genomic Ranges data structure to the chromDraw dat format. Output of this function is file "data.txt" in the working directory.

**Usage**

```
convertInputData(karyotypes)
```

**Arguments**

karyotypes      vector of the Genomic Ranges structures per karyotype.

**Value**

return exit state.

**Examples**

```
#load package
library(GenomicRanges)

#data generating
karyotype1 <- GRanges(seqnames =Rle(c("Ack1", "Ack2"), c(5, 5)),ranges =
IRanges(start = c(0, 400000,0,3300000,6000000,0,2500000,0,3800000,6400000),
end = c(400000,3300000,0,6000000,10400000,2500000,3800000,0,6400000,14800000),
names = c("A","B","CENTROMERE","C","D","E","F","CENTROMERE","G","H")),
color = c("orange","yellow","", "orange","red","green","green","", "red","orange"));

karyotype2 <- GRanges(seqnames =Rle(c("Ack3", "Ack4"), c(4, 4)),ranges =
IRanges(start = c(0,0, 2400000,3500000,0,0,2400000,6700000),
end = c(2400000,0,3500000,12200000,2400000,0,6700000,9200000),
names = c("I","CENTROMERE","J","K","L","CENTROMERE","M","N")),
color = c("light_blue","", "orange","red","pink","", "red","light_blue"));

inputData <- list(karyotype1,karyotype2);

#run the function for generate chromDraw data file.
convertInputData(inputData);
```

---

main_chromDraw	<i>The main chromDraw function, that controls all other C++ functions.</i>
----------------	--

---

**Description**

The main chromDraw function, that controls all other C++ functions.

**Usage**

```
main_chromDraw(argc, argv)
```

**Arguments**

argc	Count of the input parameters.
argv	Vector with the input parameters strings.

**Value**

Return exit state.

**Examples**

```
OUTPUTPATH = file.path(getwd());  
INPUTPATH = system.file('extdata', 'Ack_and_Stenopetalum_nutans.txt', package = 'chromDraw')  
COLORPATH = system.file('extdata', 'default_colors.txt', package = 'chromDraw')  
chromDraw(argc=7, argv=c("chromDraw", "-c", COLORPATH, "-d", INPUTPATH, "-o",  
OUTPUTPATH));
```

# Index

\* **package, libBoard, getopt, Rcpp,**  
**chromosome visualization,**  
**karyotype visualization,**  
**comparing karyotypes**  
chromDraw-package, [2](#)

chromDraw, [2](#), [3](#)

chromDraw-package, [2](#)

chromDrawGR, [3](#)

convertInputColors, [4](#)

convertInputData, [5](#)

main\_chromDraw, [2](#), [6](#)