Simple karyotypes visualization using `chromDraw`

Jan Janecka
Research group Plant Cytogenomics
CEITEC, Masaryk University of Brno

This document shows the use of the `chromDraw` R package for linear and circular type of karyotype visualization. The linear type of visualization is usually used for demonstrating chromosomes structures in karyotype and the circular type of visualization is used for comparing of karyotypes between each other.

Main functionality of `chromDraw` was written in C++ language. BOARD library [3] was used for drawing graphic primitives. The integration of R and C++ is made by Rcpp package [1] and allows completely hiding C++ implementation for R user. BiocCheck [6] and BiocInstaller [8] R packages were used during development of package. In R is supported Genomic Ranges [2] and data frame as a input data and color data format. ChromDraw can visualize files in the BED file format, that is requirement the first nine of fields per each record.

1 Data format

`ChromDraw` has two own kinds of input files. The main input file contains description of karyotype(s) for drawing and the second input file contains description of colors.

1.1 The main input file

The main information about karyotype(s) is stored in this file. This input file includes karyotype definition, with definitions of each chromosome and blocks of that karyotype and definition of the marks. The file is in a plain text and is not case sensitive.

- **Karyotype definition:**
  The definition of whole karyotype is between two tags KARYOTYPE BEGIN and KARYOTYPE END. KARYOTYPE BEGIN requires karyotype name and alias in this order. Alias must be unique for each karyotype.

- **Chromosome definition:**
  The key word for chromosome definition is CHR, the chromosome name, alias and chromosome range (defined by start and stop value) go after this key word. All this chromosome information is mandatory and must follow this given order. The chromosome alias must be unique for each chromosome in the karyotype.

- **Chromosome parts definitions:**
  This part of file contains definitions of genomic blocks and centromeres. Genomic block is defined by key word BLOCK, name, alias, chromosome alias, start position, stop position and color. Block alias must be unique in the karyotype. Chromosome alias is alias of chromosome, which contains this block. Start and stop position is defined by the block

position at the chromosome. Color is a name of color in the second input file and it is optional parameter. Centromere is defined by key word CENTROMERE and alias of corresponding chromosome. It must follows block, which is directly before centromere.

- **Marks definitions:**
  Mark is defined by the keyword MARK and it follows the title, type of shape and size of the mark. Here is available only rectangle shape temporarily. This definition is finished by the alias and position of the participant chromosome. At the end is name of the color for drawing a mark. This symbols are plotted over the chromosome blocks.

Comments can put in any part of the file, assigned by # symbol at the beginning of new line. Example of input data file:

```
# Ancestral Crucifer Karyotype chromosome 1

# Karyotype definition
KARYOTYPE ACK ACK BEGIN

# Chromosome definition
CHR ACK1 al1 0 17000000

# Genomic blocks definitions
BLOCK A A al1 0 6700000 yellow
BLOCK B B al1 6700000 12400000 yellow
# Centromere definition
CENTROMERE al1
BLOCK C C al1 12400000 17000000 yellow

# Mark definition
MARK 35S_rDNA RECTANGLE 2 al1 11739990 white

KARYOTYPE END
```

1.2 The main input file using GenomicRanges

This is the other way how is it possible to specified input data for `chromDraw`. In this case, it was used R specific data structure GenomicRanges, but the idea of data structure is similar to definition before. Each karyotype is defined by one GenomicRanges structure.

Chromosomes are defined by same seqnames. Blocks are described by ranges and chromosome names. Names of chromosomes are stored in array called name. When you define centromere, insert to this array label CENTROMERE and set the ranges [0,0]. Colors of each blocks are defined by string in array color. There is some example of GenomicRandges input data, which contains the same information like a example above:

```
> library(GenomicRanges)
> exampleData <- GRanges(seqnames =Rle(c("ACK1"), c(4)),ranges =
```

```
+ IRanges(start = c(0, 6700000,0,12400000),
+          end = c(6700000,12400000,0,17000000),
+          names = c("A","B","CENTROMERE","C")),
+          color = c("yellow","yellow","","yellow")
+        );
> exampleData;

GRanges object with 4 ranges and 1 metadata column:
             seqnames            ranges strand |       color
                <Rle>         <IRanges>  <Rle> | <character>
           A     ACK1         0-6700000      * |      yellow
           B     ACK1  6700000-12400000      * |      yellow
  CENTROMERE     ACK1                 0      * |
           C     ACK1 12400000-17000000      * |      yellow
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

1.3 Colors

The color input file contains colors definitions in a plain text. Colors are used for coloration of chromosomes blocks. Each color is defined by key word COLOR, name and red, green, blue (RGB) value. Name of each color must be unique. RGB values are in range 0 to 255. You can also put comments in any part of this file, assigned by # symbol at the beginning of new line. Example of the color input file:

```
#Definition yellow color for AK1
COLOR yellow    255    255    0
COLOR red       255    255    0
```

1.4 Colors using data frame

In R is supported other way, how define input colors. Structure of color is similar, like was said above. In this case, it was used data frame for storing colors. Each colors are defined by name and RGB values, each item is defined at separated vector. There is some example of data frame of colors, which contains the same information like a example above:

```
> name <- c("yellow", "red");
> r <- c(255, 255);
> g <- c(255, 0);
> b <- c(0, 0);
> exampleColor <- data.frame(name,r,g,b);
> exampleColor;

    name   r   g b
1 yellow 255 255 0
2    red 255   0 0
```

2 Input parameters

In chromDraw is possible to use short or long type of parameters.

- -h , –help Show help.
- -o , –outputpath Path to output directory. Current working directory is set as default.
- -d , –datainputpath Path to the input file with chromosome matrix or BED file.
- -c, –colorinputpath The file with path to color definitions.
- -s , –scale Use same scale for the linear visualization outputs.
- -f , –format Type of the input data format BED or CHROMDRAW. Default setting is CHROMDRAW.

3 Visualization

After preparation of all necessary input files, the using of `chromDraw` is very simple. There are only two functions in package `chromDraw`. First function has parameter structure just like main function in C/C++. The first parameter is ARGC with number of strings in ARGV. ARGV is a vector containing strings with arguments for program. First string of this vector must be a package name. Here is an example, how to use this package:

```
> library(chromDraw)
> 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));

[1] 0
```

The second function supporting GenomicRanges, has two parameters. First parameter is list of GenomicRanges structure per karyotype. The second one is data frame of colors, this parameter is optional. Here is example of this function, which is using examples of data:

```
> library(chromDraw)
> chromDrawGR(list(exampleData), exampleColor);

[1] 0
```

See example of the linear visualization output from `chromDraw` in the first picture 1 with Ancestral Crucifer Karyotype [4, 7]. The second visualization of four ancestral or extant karyotypes from the mustard family (Brassicaceae): *Stenopetalum nutans* (Sn, n = 4), *Arabidopsis thaliana* (At, n = 5), *Boechera stricta* (Bs, n = 7) and Ancestral Crucifer Karyotype (ACK, n = 8). Data matrices are based on [5] and [7]. 5S rDNA and 35S rDNA loci are visualized as black and white rectangles, respectively.

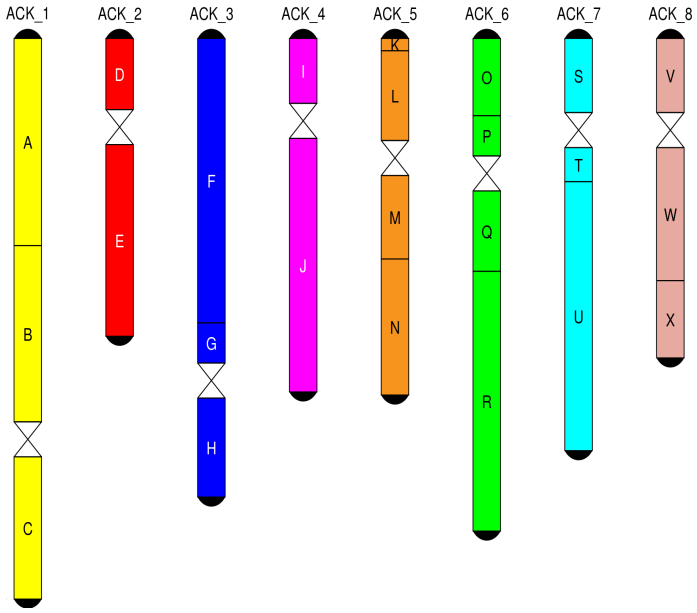

Figure 1: Linear visualization of Ancestral Crucifer Karyotype

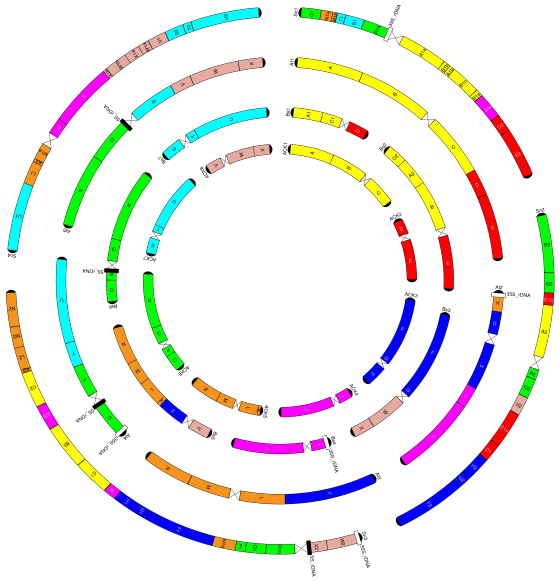

Figure 2: Circular visualization of four ancestral or extant karyotypes from the mustard family (Brassicaceae)

The BED file is visualized by the same function chromDraw, where is necessary to set the parameter format to the value BED. Here is an example how to use this feature:

```
> library(chromDraw)
> OUTPUTPATH = file.path(getwd());
> INPUTPATH = system.file('extdata',
+                         'bed.bed',
+                         package ='chromDraw')
> chromDraw(argc=7,
+           argv=c("chromDraw", "-f", "bed", "-d",INPUTPATH, "-o",
+ OUTPUTPATH));

[1] 0
```

4 Acknowledgements

I would like to thank to: M. A. Lysak for constructive comments, Matej Lexa for advices on bioinformatics and to Jiri Hon for introdution to R package creating.
Funding: Czech Science Foundation (P501/12/G090) and
European Social Fund (CZ.1.07/2.3.00/20.0189)

References

[1] Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[2] Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin Morgan, and Vincent Carey. Software for computing and annotating genomic ranges. *PLoS Computational Biology*, 9, 2013.

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

[4] Martin A. Lysak, Alexandre Berr, Ales Pecinka, Renate Schmidt, Kim McBreen, and Ingo Schubert. Mechanisms of chromosome number reduction in *Arabidopsis thaliana* and related Brassicaceae species. *Proceedings of the National Academy of Sciences of the United States of America*, 103(13):5224–5229, 2006.

[5] Terezie Mandakova, Simon Joly, Martin Krzywinski, Klaus Mummenhoff, and Martin A. Lysak. Fast diploidization in close mesopolyploid relatives of *Arabidopsis*. *The Plant Cell Online*, 22(7):2277–2290, 2010.

[6] Bioconductor Package Maintainer. *BiocCheck: Bioconductor-specific package checks*. R package version 1.1.9.

[7] M. Eric Schranz, Martin A. Lysak, and Thomas Mitchell-Olds. The ABC's of comparative genomics in the Brassicaceae: building blocks of crucifer genomes. *Trends in Plant Science*, 11(11):535 – 542, 2006.

[8] Dan Tenenbaum and Biocore Team. *BiocInstaller: Install/Update Bioconductor and CRAN Packages*. R package version 1.15.5.