Using the RCircos Package

Hongen Zhang, Ph.D.
Genetics Branch, Center for Cancer Research,
National Cancer Institute, NIH

December 18, 2015

Contents

1 Introduction 1

2 Input Data Format 2

3 Plot Track Layout 3

4 Getting Started: Initialize RCircos core components first 4

4.1 Initialize RCircos core components 4

4.2 Modifying RCircos core components 8

5 Making a Plot with RCircos 8

5.1 Initialize Graphic Device 8

5.2 Plot Chromosome Ideogram 9

5.3 Gene Labels and connectors on RCircos Plot 9

5.4 Heatmap, Histogram, Line, Scatter, and Tile Plot 9

5.5 Links: A Special Plot 10

6 More Information 11

7 sessionInfo 12

1 Introduction

The RCircos package provides a set of graphic functions which implement basic Circos 2D track plot [1] for visualizing similarities and differences of genome structure and positional relationships between genomic intervals. The package is implemented with R graphics package that comes with R base installation and aimed to reduce the complexity of usage and increase the flexibility in integrating into other R pipelines of genomic data processing.

Currently, following graphic functions are provided:
Chromosome ideogram plots for human, mouse, and rat

Data plots include:
- heatmap
- histogram
- lines
- scatterplot
- tiles

Plot items for further decoration include:
- connectors
- links (lines and ribbons)
- text (gene) labels

After successful installation of RCircos, one needs to load the library to get started using it:

```r
> library(RCircos)
```

## 2 Input Data Format

RCircos takes the input data in the form of a data frame that could be an object returned from `read.table()` or generated with other pipelines in the current R session. The first three columns of the data frame, except for input to the link plot, must be genomic position information in the order of chromosome names, chromosome start, and chromosome end positions.

```r
> data(RCircos.Histogram.Data)
> head(RCircos.Histogram.Data)
```

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>chromStart</th>
<th>chromEnd</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>45000000</td>
<td>49999999</td>
<td>0.070859</td>
</tr>
<tr>
<td>chr1</td>
<td>55000000</td>
<td>59999999</td>
<td>0.300460</td>
</tr>
<tr>
<td>chr1</td>
<td>60000000</td>
<td>64999999</td>
<td>0.125421</td>
</tr>
<tr>
<td>chr1</td>
<td>70000000</td>
<td>74999999</td>
<td>0.158156</td>
</tr>
<tr>
<td>chr1</td>
<td>75000000</td>
<td>79999999</td>
<td>0.163540</td>
</tr>
<tr>
<td>chr1</td>
<td>80000000</td>
<td>84999999</td>
<td>0.342921</td>
</tr>
</tbody>
</table>

For gene labels and heatmap plots, the gene/probe names must be provided in the fourth column. For other plots, this column could be optional.

```r
> data(RCircos.Heatmap.Data)
> head(RCircos.Heatmap.Data)
```

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>chromStart</th>
<th>chromEnd</th>
<th>Gene/ Probe Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>45000000</td>
<td>49999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chr1</td>
<td>55000000</td>
<td>59999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chr1</td>
<td>60000000</td>
<td>64999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chr1</td>
<td>70000000</td>
<td>74999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chr1</td>
<td>75000000</td>
<td>79999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chr1</td>
<td>80000000</td>
<td>84999999</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Different from other plot data, the input data for link line plot has only
paired genomic position information for each row in the order of chromosome
name A, chromStart A, chromEnd A, chromosome name B, chromStart B, and
chromEnd B.

> data(RCircos.Link.Data)
> head(RCircos.Link.Data)

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>chromStart</th>
<th>chromEnd</th>
<th>Chromosome.1</th>
<th>chromStart.1</th>
<th>chromEnd.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>8284703</td>
<td>8285399</td>
<td>chr1</td>
<td>8285752</td>
<td>8286389</td>
</tr>
<tr>
<td>chr1</td>
<td>85980143</td>
<td>85980624</td>
<td>chr7</td>
<td>123161313</td>
<td>123161687</td>
</tr>
<tr>
<td>chr1</td>
<td>118069850</td>
<td>118070319</td>
<td>chr1</td>
<td>118070329</td>
<td>118070689</td>
</tr>
<tr>
<td>chr1</td>
<td>167077258</td>
<td>167077658</td>
<td>chr1</td>
<td>169764630</td>
<td>169764965</td>
</tr>
<tr>
<td>chr1</td>
<td>171671272</td>
<td>171671550</td>
<td>chr1</td>
<td>179790879</td>
<td>179791292</td>
</tr>
<tr>
<td>chr1</td>
<td>174333479</td>
<td>174333875</td>
<td>chr6</td>
<td>101861516</td>
<td>101861840</td>
</tr>
</tbody>
</table>

Note: RCircos will convert the input data to RCircos plot data but it does
not provide functionality for general data processing. If the data frame does not
have genomic position information, you have to add the information to the data
frame before passing it to RCircos functions. Sample datasets are included in the
package for demo purpose and they could be easily explored with data() method.

Starting from version 1.1.2, user can append to input data a column of plot
color names with header of "PlotColor" to control the colors for each data point
except of heatmap plot.

3 Plot Track Layout

RCircos follows the same algorithm of Circos plot and arranges data plots in
tracks. A track could be placed either inside or outside of chromosome ideogram
and the detailed position for a track could be easily manipulated by changing of the track width and track numbers.

The figure below shows a human chromosome ideogram plus three empty tracks arranged in both inside and outside of chromosome ideogram.

R Circos Layout Demo

4 Getting Started: Initialize RCircos core components first

4.1 Initialize RCircos core components

The first step of making RCircos plot is to initialize RCircos core components. To setup RCircos core components, user needs load the chromosome ideogram data into current R session. The RCircos package have three build-in datasets for human, mouse, and rat chromosome ideograms which can be loaded with data() command. Ideogram data in text files with same format can also be
loaded with read.table() function in R.

```r
> data(UCSC.HG19.Human.CytoBandIdeogram);
> head(UCSC.HG19.Human.CytoBandIdeogram);

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>ChromStart</th>
<th>ChromEnd</th>
<th>Band</th>
<th>Stain</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>0</td>
<td>2300000</td>
<td>p36.33</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>2300000</td>
<td>5400000</td>
<td>p36.32</td>
<td>gpos25</td>
</tr>
<tr>
<td>chr1</td>
<td>5400000</td>
<td>7200000</td>
<td>p36.31</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>7200000</td>
<td>9200000</td>
<td>p36.23</td>
<td>gpos25</td>
</tr>
<tr>
<td>chr1</td>
<td>9200000</td>
<td>12700000</td>
<td>p36.22</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>12700000</td>
<td>16200000</td>
<td>p36.21</td>
<td>gpos50</td>
</tr>
</tbody>
</table>

> data(UCSC.Mouse.GRCm38.CytoBandIdeogram);
> head(UCSC.Mouse.GRCm38.CytoBandIdeogram);

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>ChromStart</th>
<th>ChromEnd</th>
<th>Band</th>
<th>Stain</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>0</td>
<td>8840440</td>
<td>qA1</td>
<td>gpos100</td>
</tr>
<tr>
<td>chr1</td>
<td>8840440</td>
<td>12278390</td>
<td>qA2</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>12278390</td>
<td>20136559</td>
<td>qA3</td>
<td>gpos33</td>
</tr>
<tr>
<td>chr1</td>
<td>20136559</td>
<td>22101102</td>
<td>qA4</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>22101102</td>
<td>30941543</td>
<td>qA5</td>
<td>gpos100</td>
</tr>
<tr>
<td>chr1</td>
<td>30941543</td>
<td>43219933</td>
<td>qB</td>
<td>gneg</td>
</tr>
</tbody>
</table>

> data(UCSC.Baylor.3.4.Rat.cytoBandIdeogram);
> head(UCSC.Baylor.3.4.Rat.cytoBandIdeogram);

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>ChromStart</th>
<th>ChromEnd</th>
<th>Band</th>
<th>Stain</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>0</td>
<td>10142096</td>
<td>p13</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>10142096</td>
<td>24272657</td>
<td>p12</td>
<td>gvar</td>
</tr>
<tr>
<td>chr1</td>
<td>24272657</td>
<td>38517175</td>
<td>p11</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>38517175</td>
<td>48659271</td>
<td>q11</td>
<td>gpos</td>
</tr>
<tr>
<td>chr1</td>
<td>48659271</td>
<td>69741157</td>
<td>q12</td>
<td>gneg</td>
</tr>
<tr>
<td>chr1</td>
<td>69741157</td>
<td>90025350</td>
<td>q21</td>
<td>gpos</td>
</tr>
</tbody>
</table>

After the chromosome ideogram data is loaded, RCircos core components can be initialized with function of RCircos.Set.Core.Components(). This function needs four arguments:

cytinfo the chromosome idegram data loaded

chr.exclude chromosomes should be excluded from ideogram, e.g., chr.exclude <- c("chrX", "chrY"); If it is set to NULL, no chromosome will be excluded.

tracks.inside how many tracks will be plotted inside chromosome ideogram

tracks.outside how many data tracks will be plotted outside chromosome ideogram
Following code initialize RCircos core components with all human chromosome ideogram and 10 data track space inside of chromosome ideogram.

```r
> chr.exclude <- NULL;
> tracks.inside <- 10;
> tracks.outside <- 0;
> RCircos.Set.Core.Components(cyto.info, chr.exclude,
+    tracks.inside, tracks.outside);
```

RCircos use three core components to perform data transformation and data plot:

**RCircos cytoband data** RCircos cytoband data is derived from the input chromosome ideogram data. Except of the chromosome name, start and end positions, band name and stain intensity for each band, chromosome highlight colors, band colors, band length in base pairs and chromosome units as well as the relative location on the circular layout are also included. These data are used to calculate the plot location of each genomic data.

**RCircos plot positions** RCircos plot positions are x and y coordinates for a circular line of radius 1.0 and the total number of points for the circular line are decided by the total number of chromosome units. One chromosome units is a plot point which covers a defined number of base pairs and total units for chromosome ideogram include units of each band plus chromosome padding area, both of them are defined in the list of plot parameters.

**RCircos plot parameters** RCircos plot parameters are only one core components open to users. With the get and reset methods, users can modify the parameters for updating other two core components. Following are RCircos plot parameters and their default values:

- `radiu.len` The radius of a circular line which serves as baseline for calculation of plot items, default: 1
- `chr.ideog.pos` Radius of chromosome ideogram position, default: 1.1
- `highlight.pos` Radius of chromosome ideogram highlights, default: 1.25
- `chr.name.po` Radius of chromosome name position, default: 1.4
- `plot.radius` Radius of plot area, default: 1.5
- `track.in.start` Radius of start position of the first track inside of chromosome ideogram, default: 1.05
- `track.out.start` Radius of start position of the first track outside of chromosome ideogram, default: 1.6
- `chrom.width` Width of chromosomes of the ideogram, default: 0.1
- `track.padding` Width of padding between two plot tracks, default: 0.02
track.height  Height of data plot track, default: 0.1 (Note: Parameters above are all relative to the radius.len and will be updated automatically when reset plot parameters with new radiu.len).

base.per.unit  Number of base pairs a chromosome unit (a plot point) will cover, default: 3000

chrom.paddings  Width of padding between two chromosomes in chromosome unit, default: 3000 (Note: chrom.paddings is binded to base.per.unit. It will be automatically updated if the base.per.unit is changed, unless be set to zero).

heatmap.width  Width of heatmap cells in chromosome unit, default: 1000

hist.width  Width of histogram column in chromosome unit, default: 1000

color.map  Color scales for heatmap plot, default: BlueWhiteRed

text.size  Character size (same as cex in R graphics package) for text plot, default: 0.4

highlight.width  Line type (same as lty in R graphics package) for chromosome highlight, default: 2

point.type  Point type (same as pch in R graphics package) for scatter plot, default: "."

point.size  Point size (same as cex in R graphics package) for scatter plot, default: 1

hist.color  Color for histogram plot, default: red

tile.color  Color for tile plot, default: black

track.background  Color of track background, default: wheat

grid.line.color  Color for track grids, default: gray

Bezier.point  Total number of points for a link(Bezier) line default: 1000

max.layers  Maximum number of layers for tile plot, default: 5

sub.tracks  Number of sub tracks in a data track, default: 5

The core components are stored in RCircos session and each component is supplied with one Get method for advanced usage. In addition, simply call the function RCircos.List.Parameters() will list all current plot parameters.

> rcircos.params <- RCircos.Get.Plot.Parameters();
> rcircos.cyto <- RCircos.Get.Plot.Ideogram();
> RCircos.List.Parameters()
4.2 Modifying RCircos core components

Among the three RCircos core components, RCircos cytoband data and RCircos plot positions are calculated based on plot parameter setting. Users can modify RCircos core components by changing plot parameters. Once the plot parameter(s) is changed, call and pass the new parameters to the function of RCircos.Reset.Plot.Parameters(), other two components will be checked for update.

```r
> rcircos.params <- RCircos.Get.Plot.Parameters();
> rcircos.params$base.per.unit <- 30000;
> RCircos.Reset.Plot.Parameters(rcircos.params);
> RCircos.List.Parameters();
```

5 Making a Plot with RCircos

Plotting with RCircos is a stepwise process. First, an initialization step is needed. Then, tracks and other aspects of the plot are added sequentially. The result is available after the plot has been entirely constructed. The next subsections walk through the process in detail.

5.1 Initialize Graphic Device

RCircos provides a set of graphic plot functions but does not handle graphic devices. To make RCircos plots, a graphic device has to be opened first. Currently, RCircos works with files supported by R graphics package such as tiff, png, pdf images as well as GUI windows. For example, to make a pdf file with RCircos plot image:

```r
> out.file <- "RCircosDemoHumanGenome.pdf";
> pdf(file=out.file, height=8, width=8, compress=TRUE);
> RCircos.Set.Plot.Area();
```

Note: RCircos.Set.Plot.Area() will setup plot area base on total number of tracks inside and outside of chromosome ideogram. User can also setup plot area by summit the R plot commands for user defined plot area, for example:

```r
> par(mai=c(0.25, 0.25, 0.25, 0.25));
> plot.new();
> plot.window(c(-2.5,2.5), c(-2.5, 2.5));
```

Note: After everything is done, the graphic device needs to be closed with dev.off().
5.2 Plot Chromosome Ideogram

For RCircos plot, a common first step is to draw chromosome ideograms and label chromosomes with names and highlights. After the RCircos core components were initialized and graphic device was open, simply call the function of RCircos.Chromosome.Ideogram.Plot() will add the chromosome ideogram to the current plot.

> RCircos.Chromosome.Ideogram.Plot();

5.3 Gene Labels and connectors on RCircos Plot

Due to the resolution issues, only limited number of gene names can be labeled. For best visualization, cex should be no less than 0.4 when draw gene labels. When cex is set to 0.4, width of character will be 5000 chromosome units when each unit covers 3000 base pairs. If the gene name list supplied is too long, it will be truncated to fit the chromosome length. Also the long gene name will span more than one track so one or more tracks may be needed to skip for next track.

Connectors are used to mark a genomic position with their names or variant status. Currently, RCircos only provide connector plot between genes and their genomic positions. The following code draw connectors on the first track inside chromosome ideogram and plot gene names on the next track.

> data(RCircos.Gene.Label.Data);
> name.col <- 4;
> side <- "in";
> track.num <- 1;
+ track.num, side);
> track.num <- 2;
+ name.col,track.num, side);

5.4 Heatmap, Histogram, Line, Scatter, and Tile Plot

Heatmap, histogram, line, scatter, and tile plot with RCircos require that the first three columns of input data are genomic position information in the order of chromosome name, start, and end position. RCircos provides one function for each type of plots and each function will draw one data track. User can simply call each function with appropriate arguments such as plot location (which track and which side of chromosome ideogram). No more data processing needed.

The code below will draw data tracks of heatmap, scatter, line, histogram, and tile plots.

> data(RCircos.Heatmap.Data);
> data.col <- 6;
> track.num <- 5;
> side <- "in";
> RCircos.Heatmap.Plot(RCircos.Heatmap.Data, data.col, +
    track.num, side);

> data(RCircos.Scatter.Data);
> data.col <- 5;
> track.num <- 6;
> side <- "in";
> by.fold <- 1;
> RCircos.Scatter.Plot(RCircos.Scatter.Data, data.col, +
    track.num, side, by.fold);

> data(RCircos.Line.Data);
> data.col <- 5;
> track.num <- 7;
> side <- "in";
> RCircos.Line.Plot(RCircos.Line.Data, data.col, +
    track.num, side);

> data(RCircos.Histogram.Data);
> data.col <- 4;
> track.num <- 8;
> side <- "in";
> RCircos.Histogram.Plot(RCircos.Histogram.Data, +
    data.col, track.num, side);

> data(RCircos.Tile.Data);
> track.num <- 9;
> side <- "in";
> RCircos.Tile.Plot(RCircos.Tile.Data, track.num, side);

### 5.5 Links: A Special Plot

Links presents relationship of two genomic positions and it is always the last track inside chromosome ideogram. Different from other data plots, input data for links plot is a data frame with paired genomic positions in the order of chromosome, start, and end position for each one genomic position. RCircos supports two types of link plot: lines and ribbons. Link lines are used for presenting relationship of two small genomic regions and ribbons are plotted for bigger genomic regions. Colors for links between chromosomes or same chromosomes could be modified by defining by.chromosome=TRUE (or FALSE).

Following code draw link lines and ribbons in the center of plot area.

> data(RCircos.Link.Data);
> track.num <- 11;
> RCircos.Link.Plot(RCircos.Link.Data, track.num, TRUE);
> data(RCircos.Ribbon.Data);
> RCircos.Ribbon.Plot(ribbon.data=RCircos.Ribbon.Data,
+  track.num=11, by.chromosome=FALSE, twist=FALSE);
> dev.off();

Run all code above will generate an image like below.

RCircos Demo with Human Genome

6 More Information

Several demo samples are included in the package. Simply run following demos to see how the RCircos works for simple and complex RCircos plot.

> library(RCircos);
> demo("RCircos.Demo.Human");
> demo("RCircos.Demo.Mouse.And.Rat");
7 sessionInfo

> sessionInfo()

R version 3.2.2 (2015-08-14)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Red Hat Enterprise Linux Server release 6.7 (Santiago)

locale:
[1] C

attached base packages:
[1] stats  graphics  grDevices  utils  datasets  methods  base

other attached packages:
[1] RCircos_1.1.3

loaded via a namespace (and not attached):
[1] tools_3.2.2

References