Package ‘IdeoViz’

November 21, 2016

Type   Package
Title   Plots data (continuous/discrete) along chromosomal ideogram
Version 1.8.0
Date   2014-09-08
Author Shraddha Pai <shraddha.pai@utoronto.ca>, Jingliang Ren
Maintainer Shraddha Pai <shraddha.pai@utoronto.ca>
Depends Biobase, IRanges, GenomicRanges, RColorBrewer,
        rtracklayer,graphics,GenomeInfoDb
biocViews Visualization,Microarray
Description Plots data associated with arbitrary genomic intervals
           along chromosomal ideogram.
License GPL-2
NeedsCompilation no

R topics documented:

IdeoViz-package ......................................................... 2
avgByBin ................................................................. 2
binned_fullGenome ................................................. 4
binned_multiSeries ................................................. 5
binned_singleSeries ............................................... 5
getBins ................................................................. 6
getIdeo ................................................................. 7
GSM733664_broadPeaks ............................................. 7
hg18_ideo .............................................................. 8
plotChromValuePair ................................................. 8
plotOnIdeo ............................................................. 10
wins ................................................................. 11
wins_discrete ...................................................... 12
wins_entiregenome ................................................. 12

Index 13
Description

Plotting discrete or continuous dataseries in the context of chromosomal location has several useful applications in genomic analysis. Examples of possible metrics include RNA expression levels, densities of epigenetic marks or genomic variation, while applications could range from the analysis of a single variable in a single context, to multiple measurements in several biological contexts (e.g. age/sex/tissue/disease context). Visualization of metrics against the chromosome could identify:

1. could identify distinctive spatial distribution that could further hypotheses about the functional role of the metric (e.g. telocentric or pericentromeric enrichment)
2. could highlight distribution differences between different groups of samples, suggesting different regulatory mechanisms; in extreme cases, visualization may identify large genomic foci of differences
3. could confirm that a quantitative difference measured between groups of interest is consistent throughout the genome (i.e. that there are no foci, and that the change is global).

This package provides a method to plot one or several dataseries against the chromosomal ideogram. It provides some simple options (vertical/horizontal orientation, display in bars or linegraphs). Data are expected to be binned; IdeoViz provides a function for user-specified bin widths. Ideograms for the genome of choice can also be automatically downloaded from UCSC using the getIdeo() function.

Details

Package: IdeoViz
Type: Package
Title: Plots data (continuous/discrete) along chromosomal ideogram
Version: 0.99.1
Date: 2013-06-26
Author: Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren
Maintainer: Shraddha Pai <Shraddha.Pai@camh.ca>
Depends: Biobase, IRanges, GenomicRanges, RColorBrewer, rtracklayer
biocViews: Visualization,Microarray
License: GPL-2

Author(s)

Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren
Description
Computed mean value of binned data. This function assumes that all elements in `featureData` have identical width. If provided with elements of disparate widths, the respective widths are not weighted averaging. This behaviour may change in future versions of IdeoViz.

Usage
```
avgByBin(xpr, featureData, target_GR, justReturnBins = FALSE,
         getBinCountOnly = FALSE, FUN = mean, doSampleCor = FALSE,
         verbose = FALSE)
```

Arguments
- `xpr` (data.frame or matrix) Locus-wise values. Rows correspond to genomic intervals (probes, genes, etc.,) while columns correspond to individual samples
- `featureData` (data.frame or GRanges) Locus coordinates. Row order must match `xpr`. Column order should be: 1. chrom, 2. locus start, 3. locus end. All elements are assumed to be of identical width. Coordinates must be zero-based or one-based, but not half-open. Coordinate system must match that of `target_GR`.
- `target_GR` (GRanges) Target intervals, with coordinate system matching that of `featureData`.
- `justReturnBins` (logical) when TRUE, returns the coordinates of the bin to which each row belongs. Does not aggregate data in any way. This output can be used as input for more complex functions with data from each bin.
- `getBinCountOnly` (logical) when TRUE, does not aggregate or expect `xpr`. Only returns number of overlapping subject ranges per bin. Speeds up computation.
- `FUN` (function) function to aggregate data in bin
- `doSampleCor` (logical) set to TRUE to compute mean pairwise sample correlation (Pearson correlation) for each bin; when TRUE, this function overrides `FUN`.
- `verbose` (logical) print status messages

Details
This function allows the user to bin data if this hasn’t already been done, and is a step involved in preparing the data for `plotOnIdeo()`. This function computes binned within-sample average of probes overlapping the same range. Where a range overlaps multiple bins, it gets counted in all.

Value
- (GRanges) Binned data or binning statistics; information returned for non-empty bins only. The default for this function is to return binned data; alternately, if `justReturnBins`=TRUE or `getBinCountOnly`=TRUE the function will return statistics on bin counts. The latter may be useful to plot spatial density of the input metric.

The flags and output types are presented in order of evaluation precedence:

1. If `getBinCountOnly`=TRUE, returns a list with a single entry: `bin.ID`: (data.frame) bin information: chrom, start, end, width, strand, index, and count. "index" is the row number of `target_GR` to which this bin corresponds
2. If `justReturnBins`=TRUE and `getBinCountOnly`=FALSE, returns a list with three entries:
   (a) `bin.ID`: same as `bin_ID` in output 1 above
binned_fullGenome

(b) xpr:(data.frame) B-by-n columns where B is total number of [target_GR,featureData] overlaps (see next entry, binmap_idx) and n is number of columns in xpr; column order matches xpr. Contains sample-wise data “flattened” so that each [target,subject] pair is presented. More formally, entry [i,j] contains expression for overlap of row i from binmap_idx for sample j (where 1 <= i <= B, 1 <= j <= n)

(c) binmap_idx:(matrix) two-column matrix: 1) target_GR row, 2) row of featureData which overlaps with index in column 1. (matrix output of GenomicRanges::findOverlaps())

3. Default: If justReturnBins=FALSE and getBinCountOnly=FALSE, returns a GRanges object. Results are contained in the elementMetadata slot. For a dataset with n samples, the table would have (n+1) columns; the first column is bin_count, and indicates number of units contained in that bin. Columns (2:(n+1)) contain binned values for each sample in column order corresponding to that of xpr.
For doSampleCor=TRUE, result is in a metadata column with name "mean_pairwise"cor". Bins with a single datapoint per sample get a value of NA.

Author(s)
Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren

See Also
getIdeo(), getBins()

Examples
ideo_hg19 <- getIdeo("hg19")
data(GSM733664_broadPeaks)
chrom_bins <- getBins(c("chr1","chr2","chrX"), ideo_hg19,stepSize=5*100*1000)
# default binning
mean_peak <- avgByBin(data.frame(value=GSM733664_broadPeaks[,7]), GSM733664_broadPeaks[,1:3], chrom_bins)
# custom function
median_peak <- avgByBin(data.frame(value=GSM733664_broadPeaks[,7]), GSM733664_broadPeaks[,1:3], chrom_bins, FUN=median)
# mean pairwise sample correlation
data(binned_multiSeries)

3. Data for example 3.

Description
Simulated data spanning all autosomes and X,Y chromosomes of the human genome (build hg18). Values consist of a single dataseries of random uniform distribution between -1 and +1. The chromosomes are tiled in 1Mb bins and coordinates are one-based.

Usage
data(binned_fullGenome)
### binned_multiSeries

**Source**
Simulated data, generated by Shraddha Pai

**Examples**
```r
data(binned_fullGenome)
head(binned_fullGenome)
seqlevels(binned_fullGenome)
```

**Description**
A simulated dataseries spanning chr1, chrX, chrY of the human genome (build hg18). Values consist of five series constructed to show mostly random behaviour with the exception of elevated signal in a few regions. The chromosomes are tiled in 1Mb bins and coordinates are one-based.

**Usage**
```r
data(binned_multiSeries)
```

**Source**
Simulated data, generated by Shraddha Pai

**Examples**
```r
data(binned_multiSeries)
head(binned_multiSeries)
```

### binned_singleSeries

**Data for example 2.**

**Description**
Simulated data spanning 3 human chromosomes and varying in a random uniform distribution between -1 and +1.

**Usage**
```r
data(binned_singleSeries)
```

**Source**
Simulated data by Shraddha Pai

**Examples**
```r
data(binned_singleSeries)
head(binned_singleSeries)
```
Description

Get uniformly-sized bins of specified width

Usage

getBins(chroms, ideo, binLim = NULL, stepSize)

Arguments

- **chrms**: (character) chromosomes to generate bins for
- **ideo**: (data.frame) ideogram table as generated by `getIdeo()`. See that function for details.
- **binLim**: (numeric, length 2) [start, end] of genomic range to generate bins for. A value of NULL results in binning of entire chromosome
- **stepSize**: (integer) bin size in bases

Details

This is a helper function used to generate binned data for `plotOnIdeo()`. It takes the chromosome-wide extents from `ideo`, which is essentially the `cytoBandIdeo` table from UCSC browser with the header as the first row. A use case is to generate bins using this function and supply the output to `avgByBin()` to bin the data.

Value

(GRanges) bin ranges in 1-base coordinates

Author(s)

Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren

See Also

`getIdeo()`, `avgByBin()`

Examples

```r
ideo_hg19 <- getIdeo("hg19")
chrom_bins <- getBins(c("chr1","chr2","chrX"), ideo_hg19, stepSize=5*100*1000)
```
getIdeo

**Description**

Download ideogram table containing chromosomal extent and band locations from the UCSC genome browser.

**Usage**

getIdeo(ideoSource)

**Arguments**

- **ideoSource** (character) Genome build for data (e.g. mm10).

**Details**

Uses `rtracklayer` to retrieve the `cytoBandIdeo` table from the UCSC genome browser. The `cytoBandIdeo` table contains chromosomal ideogram information and is used to graph the chromosomal bands in `plotOnIdeo()`. This table is provided as input to `plotOnIdeo()`. In the case where the user bins the data, the output of this function can also be used as input to generate bin coordinates for binning the data (see `avgByBin()`).

**Value**

(data.frame) ideogram table

**Author(s)**

Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren

**See Also**

`avgByBin()`, `getBins()`

**Examples**

getIdeo("mm9")

---

GSM733664_broadPeaks

**Description**

Data for vignette example 4.

**Usage**

data(GSM733664_broadPeaks)
Details

GEO accession GSM733664, subset containing chr1, chr2, and chrX peaks.

References


Examples

data(GSM733664_broadPeaks)
head(GSM733664_broadPeaks)

---

**hg18_ideo**

*Ideogram table for hg18*

Description

Cytoband information for all chromosomes in human genome build hg18. Used for vignette examples.

Usage

data(hg18_ideo)

Source

UCSC genome browser.

Examples

data(hg18_ideo)
head(hg18_ideo)

---

**plotChromValuePair**

*Plot a chromosome-value pair*

Description

Base function which plots the ideogram and superimposed data for a single chromosome. `plotOnIdeo()` calls this function and stacks the resulting output.

Usage

plotChromValuePair(chrom, cytoTable, bpLim, vertical, values_GR, val_range, col, value_cols = "values", default_margins, addScale, ablines_y, smoothVals, span=0.03, verbose = FALSE, ...)
**Arguments**

- **chrom** (character): chromosome(s) to create ideograms for
- **cytoTable** (data.frame): loaded ideogram table. (see ideoTable argument to plotOnIdeo())
- **bpLim** (numeric): (aka xlim); display only a section of the chromosome and the corresponding values
- **vertical** (logical): if TRUE, chromosomes will be plotted vertically
- **values_GR** (GenomicRanges): data to be plotted must be in metadata columns
- **val_range** (numeric): (aka ylim); y-axis scale for data series
- **col** (character): colour for series
- **value_cols** (character): column name for series to plot
- **default_margins** (numeric): page inner margins (in inches)
- **addScale** (logical): if FALSE, bp positions will be hidden
- **ablines_y** (numeric): when specified, will draw reference lines on the y-axis
- **smoothVals** (logical): when T applies loess() to each series
- **span** (numeric): span argument for loess function
- **verbose** (logical): print messages
- **...** arguments to axis(), line(), and rect()

**Details**

Plots one unit of chromosome ideogram with dataseries superimposed. Usually, the user can avoid this function and directly call plotOnIdeo(). However, this function may be used in cases where further plot customization is required.

**Author(s)**

Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren

**See Also**

plotOnIdeo()

**Examples**

data(hg18_ideo)
data(binned_multiSeries)
layout(matrix(1:2, byrow=TRUE,ncol=1),heights=c(2.5,1))
plotChromValuePair("chr1",hg18_ideo,
values_GR=binned_multiSeries,value_cols=colnames(mcols(binned_multiSeries)),plotType='lines',
col=1:5,val_range=c(0,10),bpLim=NULL,vertical=FALSE,addScale=TRUE,ablines_y=NULL,
smoothVals=FALSE,default_margins=c(0.5,.5,.1,.1))
plotOnIdeo

Plot data superimposed on chromosomal ideogram

Description

Main function to plot binned data alongside chromosomal ideogram.

Usage

plotOnIdeo(chrom = stop("enter chromosome(s) to plot"), ideoTable, values_GR, value_cols = "values", plotType = "lines", col = "orange", bpLim = NULL, val_range = NULL, addScale = TRUE, scaleChrom = TRUE, vertical = FALSE, addOnetoStart = TRUE, smoothVals = FALSE, cex.axis = 1, plot_title = NULL, ablines_y = NULL, cex.main=1, ...)

Arguments

- **chrom** (character): chromosome(s) to create ideograms for
- **ideoTable** (data.frame): ideogram table. See getIdeo()
- **values_GR** (GenomicRanges): data to be plotted must be in metadata columns
- **value_cols** (character): which series to plot. Should be column names of the mcols() slot of values_GR
- **plotType** (character): Plot type for each series. Values can be "lines" or "rect" to plot lines or barplots respectively. The latter is not recommended when several series are to be plotted on the same axis.
- **col** (character): vector of colors for data series
- **bpLim** (numeric): (xlim); display only a section of the chromosome and the corresponding values
- **val_range** (numeric): (ylim); y-axis scale for data series
- **addScale** (logical): if TRUE, bp positions will be shown along the chromosomes. This feature should be turned off if numerous chromosomes’ worth of data are being plotted and all objects don’t fit on the final graphics device.
- **scaleChrom** (logical): if FALSE, all chroms will display as the same size. scaleChrom will be ignored if bpLim is not NULL
- **vertical** (logical): if TRUE, chromosomes will be plotted vertically
- **addOnetoStart** (logical): if TRUE, adds 1 to chromStart. Useful to convert data in half-open coordinates - which is all data from the UCSC genome browser, including cytobandIdeo, into 1-base.
- **smoothVals** (logical): if T, smoothes each trendline. Currently hard-coded to lowess smoothing with span=0.03
- **cex.axis** (integer): axis font size
- **plot_title** (character): title for overall graph
- **ablines_y** (numeric): when supplied, draws reference lines on the y-axis
- **cex.main** (numeric): font size for plot title
- **...** other graphing options for barplot (i.e. main="Values", to title bar plot "Values")
Details

`plotOnIdeo()` is the main function of this package. It is the one the end-user is expected to call to generate plots. Input is provided as a `GRanges` object (`values_GR`), with data to be plotted contained in its metadata slot. The user is responsible for providing pre-binned data, if binning is required. Data can also be binned using the `avgByBin()` function in this package. The ideogram table (`ideoTable`) is the same as the `cytoBandIdeo` table available from the UCSC genome browser database for a given genome and can be either automatically downloaded from UCSC (see `getIdeo()` or read in from a local file and passed to this function.

There are numerous arguments which control the appearance of the plot. The main decision points are:

1. `vertical`: Whether the entire plot should have a horizontal or vertical orientation
2. `plotType`: Whether each dataseries should be shown as a trendline or as a barplot. The latter is not recommended for cases where multiple series are to be plotted on the same axis.

Other considerations:

- The size of the graphics device limits the number of chromosomes that can be plotted. A simple solution may be to set `addScale=FALSE`. However, it is recommended to call `plotOnIdeo()` multiple times, and plotting a fewer number of chromosomes on each page.
- The code expects coordinates of `values_GR` to be in 1-base. Set `addOneToStart=TRUE` if supplied coordinates are in 0-base.

Author(s)

Shraddha Pai <Shraddha.Pai@camh.ca>, Jingliang Ren

Examples

```r
data(binned_multiSeries)
data(hg18_ideo)
plotOnIdeo(chrom=seqlevels(binned_multiSeries), ideoTable=hg18_ideo, values_GR=binned_multiSeries, value_cols=colnames(mcols(binned_multiSeries)), col=1:5)
```

Description

A simulated dataseries spanning three chromosomes, and containing five series. The chromosomes are tiled in 1Mb windows.

Usage

```r
data(wins)
```

Source

Simulation by Shraddha Pai

Examples

```r
data(wins)
head(wins)
```
### Data for example 2.

**Description**

Simulated data spanning 3 human chromosomes and varying in a random uniform distribution between -1 and +1.

**Usage**

```r
data(wins_discrete)
```

**Source**

Simulated data by Shraddha Pai

**Examples**

```r
data(wins_discrete)
head(wins_discrete)
```

### Data for example 3.

**Description**

Simulated data spanning all human chromosomes. Values follow random uniform distribution between -1 and +1.

**Usage**

```r
data(wins_entiregenome)
```

**Source**

Simulated data, generated by Shraddha Pai

**Examples**

```r
data(wins_entiregenome)
head(wins_entiregenome)
seqlevels(wins_entiregenome)
```
Index

∗Topic datasets
  binned_fullGenome, 4
  binned_multiSeries, 5
  binned_singleSeries, 5
  GSM733664_broadPeaks, 7
  hg18_ideo, 8
  wins, 11
  wins_discrete, 12
  wins_entiregenome, 12

∗Topic package
  IdeoViz-package, 2

avgByBin, 2

binned_fullGenome, 4
binned_multiSeries, 5
binned_singleSeries, 5

getBins, 6
getIdeo, 7
GSM733664_broadPeaks, 7

hg18_ideo, 8

IdeoViz-package, 2

plotChromValuePair, 8
plotOnIdeo, 10

wins, 11
wins_discrete, 12
wins_entiregenome, 12