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 superimposed on the chromosomal ideogram could provide varied insights into the metric of interest:

1. It could identify distinctive spatial distribution that could further hypotheses about the functional role of the metric (e.g. telocentric or pericentromeric enrichment)

2. It 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. It 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.

1 Setup

> require(IdeoViz)
> require(RColorBrewer) ### nice colours
> data(binned_multiSeries)
2 Example 1: Plotting several trendlines along one ideogram

The ideogram table containing cytogenetic band information is used to render chromosomes. This table corresponds directly to the `cytoBandIdeo` table from the UCSC genome browser. There are two ways to supply an ideogram table to `plotOnIdeo()`:

1. First, it can be automatically downloaded from UCSC for your genome of choice, using the `getIdeo()` function.

2. Alternately, a pre-downloaded `cytoBandIdeo` table can be provided to downstream functions such as `plotOnIdeo()`. In this case, the table must be provided as a data.frame object with a header row and the column order matching that of the `cytoBandIdeo()` table at UCSC.

```r
> ideo <- getIdeo("hg18")
> head(ideo)

  chrom chromStart chromEnd name gieStain
1  chr1     0    2300000   p36.33    gneg
2  chr1  2300000   5300000   p36.32  gpos25
3  chr1  5300000   7100000   p36.31    gneg
4  chr1  7100000  92000000   p36.23  gpos25
5  chr1  9200000 126000000   p36.22    gneg
6  chr1 12600000 161000000   p36.21  gpos50

> plotOnIdeo(chrom=seqlevels(binned_multiSeries), # which chrom to plot?
+ ideoTable=ideo, # ideogram name
+ values_GR=binned_multiSeries, # data goes here
+ value_cols=colnames(mcols(binned_multiSeries)), # col to plot
+ col=brewer.pal(n=5, 'Spectral'), # colours
+ val_range=c(0,10), # set y-axis range
+ ylab="array intensities",
+ plot_title="Trendline example")
```

![](image.png)
Example 2: Plotting a single series in bar format

For this example, we specify a local file to obtain the chromosome ideograms, rather than having IdeoViz download it from UCSC.

```r
> data(binned_singleSeries)
> data(hg18_ideo)  # cytoBandIdeo table downloaded previously and stored as a data.frame.
> plotOnIdeo(chrom=seqlevels(binned_singleSeries),
+    ideo=hg18_ideo,
+    values_GR=binned_singleSeries,
+    value_cols=colnames(mcols(binned_singleSeries)),
+    plotType='rect',  # plot as bars
+    col='blue', vertical=T,
+    val_range=c(-1,1), ylab="dummy score",
+    plot_title="Discretized example")
```

Plot chromosome done
Plot chromosome done
Plot chromosome done
Example 3: Plotting a single series in bar format along entire genome

```r
> data(binned_fullGenome)
> plotOnIdeo(chrom=seqlevels(binned_fullGenome),
+     ideo=ideo,
+     values_GR=binned_fullGenome,
+     value_cols=colnames(mcols(binned_fullGenome)),
+     plotType='rect',
+     col='orange', addScale=F, # hide scale to remove visual clutter
+     plot_title="Whole genome view",
+     val_range=c(-1,1),cex.axis=0.5,chromName_cex=0.6)
```
3 Example 4: Binning data using IdeoViz functions

In this example, we do everything in IdeoViz: download the ideogram from UCSC, bin the data, and finally, plot along chromosomes. For the example, we use histone H3K9me3 peak intensities mapped in the human lymphoblastoid cell line.
GM12878 (GEO accession GSM733664, only 3 chromosomes shown for simplicity). Here, average peak signal is plotted in 500Kb bins along the chromosome. The ideogram plots show high signal in pericentromeric and telomeric regions, consistent with the association of this histone mark with heterochromatin.


```r
> ideo_hg19 <- getIdeo("hg19")
> chroms <- c("chr1","chr2","chrX")
> data(GSM733664_broadPeaks)
> head(GSM733664_broadPeaks)

<table>
<thead>
<tr>
<th>chrom</th>
<th>chromStart</th>
<th>chromEnd</th>
<th>name</th>
<th>score</th>
<th>strand</th>
<th>signalValue</th>
<th>pValue</th>
<th>qValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>10141</td>
<td>10374</td>
<td>.</td>
<td>993</td>
<td>.</td>
<td>10.796883</td>
<td>10.3</td>
<td>-1</td>
</tr>
<tr>
<td>chr1</td>
<td>567457</td>
<td>567702</td>
<td>.</td>
<td>1000</td>
<td>.</td>
<td>16.590333</td>
<td>100.0</td>
<td>-1</td>
</tr>
<tr>
<td>chr1</td>
<td>569826</td>
<td>570047</td>
<td>.</td>
<td>1000</td>
<td>.</td>
<td>15.757614</td>
<td>100.0</td>
<td>-1</td>
</tr>
<tr>
<td>chr1</td>
<td>723167</td>
<td>727602</td>
<td>.</td>
<td>808</td>
<td>.</td>
<td>8.389733</td>
<td>14.2</td>
<td>-1</td>
</tr>
<tr>
<td>chr1</td>
<td>816959</td>
<td>817136</td>
<td>.</td>
<td>793</td>
<td>.</td>
<td>8.188648</td>
<td>1.7</td>
<td>-1</td>
</tr>
<tr>
<td>chr1</td>
<td>821181</td>
<td>821421</td>
<td>.</td>
<td>753</td>
<td>.</td>
<td>7.660859</td>
<td>3.4</td>
<td>-1</td>
</tr>
</tbody>
</table>

> chrom_bins <- getBins(chroms, ideo_hg19,stepSize=5*100*1000)
> avg_peak <- avgByBin(data.frame(value=GSM733664_broadPeaks[,7]),
+ GSM733664_broadPeaks[,1:3], chrom_bins)
> plotOnIdeo(chrom=seqlevels(chrom_bins),
+ ideoTable=ideo_hg19,
+ values_GR=avg_peak, value_cols='value',
+ val_range=c(0,50),
+ plotType='rect',
+ col='blue', vertical=T
+ )

Plot chromosome done
Plot chromosome done
Plot chromosome done

6
4 Example 5: Plotting a set of coordinates as tracks

Here we plot multiple GRanges(), each as its own track.

```r
ideo_hg19 <- getIdeo("hg19")
x <- GSM733664_broadPeaks
gr <- GRanges(x[,1], IRanges(x[,2], x[,3]))
pal <- brewer.pal(n=5, name="Dark2")
chroms <- c("chr1", "chr2", "chrX")
chrom_bins <- getBins(chroms, ideo_hg19,
                      stepSize=5*100*1000)
grList <- list(gr, gr, gr, gr, gr)
plotOnIdeo(chrom=seqlevels(chrom_bins),
          ideoTable=ideo_hg19,
          values_GR=grList, value_cols="value",
          plotType="seg_tracks",
          col=pal, vertical=F)
```
Session info

> sessionInfo()

R version 3.4.0 (2017-04-21)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.2 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.5-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.5-bioc/R/lib/libRlapack.so

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8 LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8 LC_NAME=C
[9] LC_ADDRESS=C LC_TELEPHONE=C

attached base packages:
[1] stats4 parallel stats graphics grDevices utils datasets
[8] methods base

other attached packages:
[1] IdeoViz_1.10.0 rtracklayer_1.36.0 RColorBrewer_1.1-2
[4] GenomicRanges_1.28.0 GenomeInfoDb_1.12.0 IRanges_2.10.0
[7] S4Vectors_0.14.0 Biobase_2.36.0 BiocGenerics_0.22.0
loaded via a namespace (and not attached):

[1] XVector_0.16.0 zlibbioc_1.22.0
[3] GenomicAlignments_1.12.0 BiocParallel_1.10.0
[5] lattice_0.20-35 tools_3.4.0
[7] SummarizedExperiment_1.6.0 grid_3.4.0
[9] matrixStats_0.52.2 Matrix_1.2-9
[11] GenomeInfoDbData_0.99.0 bitops_1.0-6
[13] RCurl_1.95-4.8 DelayedArray_0.2.0
[15] compiler_3.4.0 Biostrings_2.44.0
[17] Rsamtools_1.28.0 XML_3.98-1.6