Plotting using Genominator and GenomeGraphs (Beta)

James Bullard  Kasper Daniel Hansen


This vignette is preliminary, and should be viewed as subject to change. A number of the functions are not directly exported by the package – there is a reason for that.

In this vignette we demonstrate how to visualize data using the GenomeGraphs package. The main idea is that we want to build a plotting function which we can use to plot regions. The simplest case is the following:

First, we make a database:

```r
> require(Genominator)
> options(verbos = FALSE)
> N <- 100000  # the number of observations.
> K <- 100     # the number of annotation regions, not less than 10
> df <- data.frame(chr = sample(1:16, size = N, replace = TRUE),
+                   location = sample(1:1000, size = N, replace = TRUE),
+                   strand = sample(c(1L,-1L), size = N, replace = TRUE))
> eData <- aggregateExpData(importToExpData(df, dbFilename = "pmy.db", overwrite = TRUE, tablename = "ex")
> annoData <- data.frame(chr = sample(1:16, size = K, replace = TRUE),
+                         location = sample(1:1000, size = K, replace = TRUE),
+                         strand = sample(c(1L,-1L), size = K, replace = TRUE),
+                         start = (st <- sample(1:1000, size = K, replace = TRUE)),
+                         end = st + rpois(K, 75),
+                         feature = c("gene", "intergenic") [sample(1:2, size = K, replace = TRUE)])
> rownames(annoData) <- paste("elt", 1:K, sep = ".")

> rp <- Genominator::makeRegionPlotter(list("track.1" = list(expData = eData, what = "counts")))
> args(rp)

function (chr, start, end, overlays = NULL, title = NULL, ...)

This constructs a function which can be called to view particular pieces of data.

> rp(1, 10, 1000)
GenomeGraphs provides a wealth of customization options and means of plotting which for the most part are transferable using the list.

```r
> rp <- Genominator:::makeRegionPlotter(
list("track.1" = list(expData = eData, what = "counts",
        dp = DisplayPars(lwd = .45, color = "grey")))
> rp(1, 400, 500)
```
Here we can plot our annotation using the annotation factory construct. This is probably a little advanced. An easier thing is to use Ensembl to do the plotting of the annotation. Often, however, you will want to augment the annotation produced by Ensembl.

```r
> annoFactory <- Genominator:::makeAnnoFactory(annoData, featureColumnName = "feature",
+ groupColumnName = NULL, idColumnName = NULL,
+ dp = DisplayPars("gene" = "blue",
+ "intergenic" = "green"))
> rp <- Genominator:::makeRegionPlotter(list("track.1" = list(expData = eData, what = "counts",
+ dp = DisplayPars(lwd=.2, color = "grey")),
+ "track.2" = list(expData = eData, what = "counts",
+ fx = log2, DisplayPars(lwd=.3, color = "black"))),
+ annoFactory = annoFactory)
> rp(annoData[,1,"chr"], annoData[,1, "start"] - 100, annoData[,1, "end"] + 100)
```
\textbf{GenomeGraphs} also offers a nice way to plot annotation for a given region using data from Ensembl or other sources of annotation - in some cases you have to do a little work because of the way that Biomart indexes the annotation and the way the \textit{Genominator} package works (in this case yeast annotation is stored with Roman numerals denoting the chromosomes).

\begin{verbatim}
> require("biomaRt")
> mart <- useMart("ensembl", dataset = "scerevisiae_gene_ensembl")
> annoFactory <- Genominator:::makeAnnoFactory(mart, chrFunction = function(chr) as.roman(chr))
> mart <- useMart("ensembl", dataset = "scerevisiae_gene_ensembl")
> load(system.file("data", "chr1_yeast.rda", package = "Genominator"))
> head(chr1_yeast)

chr location strand mRNA_1 mRNA_2
1  1 1   -1  9.038919  8.614710
2  1 1   -1  9.172428  8.558421
3  1 2   -1  9.422065  9.131857
4  1 2   -1  8.679480  8.442943
5  1 2   -1  8.546894  8.794416
6  1 2   -1  8.784635  8.918863

> yData <- importToExpData(chr1_yeast, dbFilename = "my.db", tablename = "yeast",
+                        overwrite = TRUE)
> rp <- Genominator:::makeRegionPlotter(list("track.-" = list(expData = yData, what = c("mRNA_1", "mRNA_2"))
+                            fx = rowMeans, strand = -1,
\end{verbatim}
\[\begin{align*}
\text{dp} &= \text{DisplayPars(lwd=.3, color = "grey"))}, \\
\text{annoFactory} &= \text{annoFactory}
\end{align*}\]

\[> \text{rp}(1, 20000, 50000)\]

\begin{figure}
\centering
\includegraphics[width=\textwidth]{plot.png}
\end{figure}

\section*{SessionInfo}
\begin{itemize}
\item R version 3.3.1 (2016-06-21), x86_64-pc-linux-gnu
\item Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_US.UTF-8, LC\_COLLATE=C, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=en\_US.UTF-8, LC\_PAPER=en\_US.UTF-8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C
\item Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, stats4, utils
\item Other packages: BiocGenerics 0.20.0, DBI 0.5-1, GenomeGraphs 1.34.0, Genominator 1.28.0, IRanges 2.8.0, RSQLite 1.0.0, S4Vectors 0.12.0, biomaRt 2.30.0
\item Loaded via a namespace (and not attached): AnnotationDbi 1.36.0, Biobase 2.34.0, RCurl 1.95-4.8, XML 3.98-1.4, bitops 1.0-6, tools 3.3.1
\end{itemize}