Seminar III: R/Bioconductor

August-December 2009

Leonardo Collado Torres

Bachelor in Genomic Sciences (LCG),
UNAM, Cuernavaca, Mexico

lcollado@lcg.unam.mx

http://www.lcg.unam.mx/~lcollado/

September 29, 2009

Assistants: Alejandro Reyes areyes@lcg.unam.mx, José Reyes jreyes@lcg.unam.mx and Víctor Moreno jmoreno@lcg.unam.mx

Note: Questions through the forum please. Those who are not from the sixth LCG generation send us an email so we can register you on the forum.

Abstract

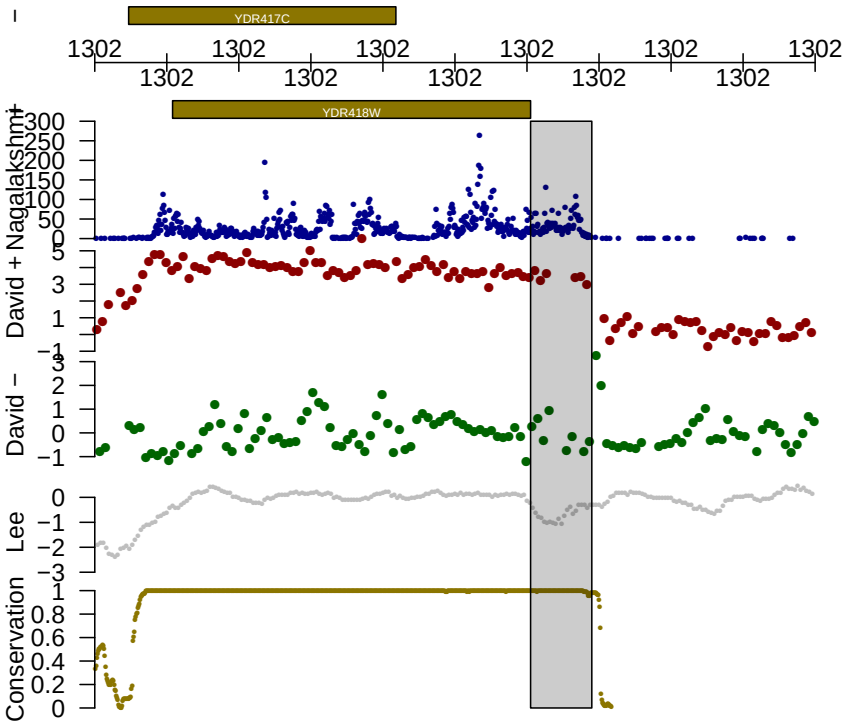
The following exercise will make sure that you can use the GenomeGraphs package.

1 GenomeGraphs

1. Download the following paper by Durinck, Bullard, Spellman and Dudoit: http://www.ncbi.nlm.nih.gov/pubmed/19123956
2. Reproduce figure 3 from the paper. Its just a matter of extracting the code from the text :)

```
> library(GenomeGraphs)
> data("seqDataEx", package = "GenomeGraphs")
> str = seqDataEx$david[, "strand"] == 1
> biomart = useMart("ensembl", "scerevisiae_gene_ensembl")
```

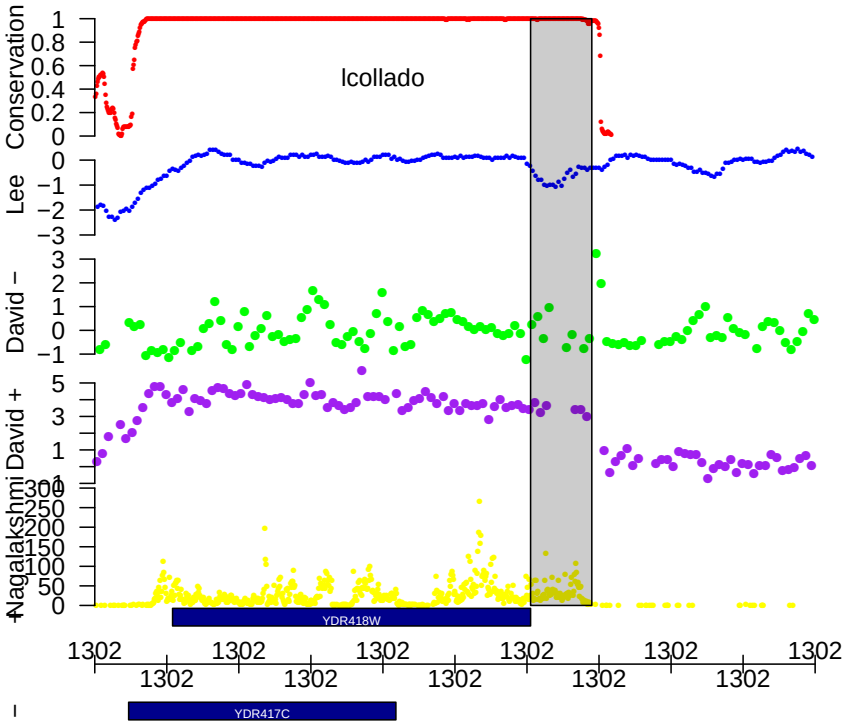
```
> a <- makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,
+     strand = "-", biomart = biomart, dp = DisplayPars(plotId = TRUE,
+         idRotation = 0, cex = 0.5))
> b <- makeGenomeAxis(dp = DisplayPars(byValue = 1000, size = 3))
> c <- makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,
+     strand = "+", biomart = biomart, dp = DisplayPars(plotId = TRUE,
+         idRotation = 0, cex = 0.5))
> d <- makeBaseTrack(base = seqDataEx$snyder[, "location"], value = seqDataEx$snyder[,
+     "counts"], dp = DisplayPars(lwd = 0.3, color = "darkblue",
+     ylim = c(0, 300)))
> e <- makeGenericArray(probeStart = seqDataEx$david[str, "location"],
+     intensity = seqDataEx$david[str, "expr", drop = FALSE], dp = DisplayPars(pointSize = 0.5))
> f <- makeGenericArray(probeStart = seqDataEx$david[!str, "location"],
+     intensity = seqDataEx$david[!str, "expr", drop = FALSE],
+     dp = DisplayPars(color = "darkgreen", pointSize = 0.5))
> g <- makeBaseTrack(base = seqDataEx$nislow[, "location"], value = seqDataEx$nislow[,
+     "evalue"], dp = DisplayPars(color = "grey", lwd = 0.25))
> h <- makeBaseTrack(base = seqDataEx$conservation[, "location"],
+     value = seqDataEx$conservation[, "score"], dp = DisplayPars(color = "gold4",
+         lwd = 0.25))
> pList <- list(`-` = a, b, `+` = c, Nagalakshmi = d, `David +` = e,
+     `David -` = f, Lee = g, Conservation = h)
> rOverlay <- makeRectangleOverlay(start = 1302105, end = 1302190,
+     region = c(4, 8), dp = DisplayPars(alpha = 0.2))
> gdPlot(pList, minBase = 1301500, maxBase = 1302500, overlay = rOverlay)
```



3. Make a new plot with some re-ordering: invert the order of tracks. Meaning that you'll have conservation on top, followed by the Lee data, then David -, David +, Nagalakshmi, + gene region, genome axis, and finally - gene region. Change the colors of all the tracks to any ones you like (without repeating

them). Finally, add a text overlay with your username on the conservation track around positions 1301700 to 1301900. You might prefer to build each `gdObject` like in the class (a, b, c, ...) and then create the list when you use `gdPlot`.

```
> a <- makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,
+     strand = "-", biomart = biomart, dp = DisplayPars(protein_coding = "darkblue",
+         plotId = TRUE, idRotation = 0, cex = 0.5))
> b <- makeGenomeAxis(dp = DisplayPars(byValue = 1000, size = 3))
> c <- makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,
+     strand = "+", biomart = biomart, dp = DisplayPars(protein_coding = "darkblue",
+         plotId = TRUE, idRotation = 0, cex = 0.5))
> d <- makeBaseTrack(base = seqDataEx$snyder[, "location"], value = seqDataEx$snyder[,
+     "counts"], dp = DisplayPars(lwd = 0.3, color = "yellow",
+     ylim = c(0, 300)))
> e <- makeGenericArray(probeStart = seqDataEx$david[str, "location"],
+     intensity = seqDataEx$david[str, "expr", drop = FALSE], dp = DisplayPars(color = "purple",
+         pointSize = 0.5))
> f <- makeGenericArray(probeStart = seqDataEx$david[!str, "location"],
+     intensity = seqDataEx$david[!str, "expr", drop = FALSE],
+     dp = DisplayPars(color = "green", pointSize = 0.5))
> g <- makeBaseTrack(base = seqDataEx$nislow[, "location"], value = seqDataEx$nislow[,
+     "evalue"], dp = DisplayPars(color = "blue", lwd = 0.25))
> h <- makeBaseTrack(base = seqDataEx$conservation[, "location"],
+     value = seqDataEx$conservation[, "score"], dp = DisplayPars(color = "red",
+         lwd = 0.25))
> pList <- list(Conservation = h, Lee = g, `David -` = f, `David +` = e,
+     Nagalakshmi = d, `+` = c, b, `-` = a)
> rOverlay <- makeRectangleOverlay(start = 1302105, end = 1302190,
+     region = c(1, 5), dp = DisplayPars(alpha = 0.2))
> tOverlay <- makeTextOverlay("lcollado", 1301900, 0.5, region = c(1,
+     1), dp = DisplayPars(color = "black"))
> gdPlot(pList, minBase = 1301500, maxBase = 1302500, overlays = c(rOverlay,
+     tOverlay))
```



4. Explain every "make..." command :)

- To create object `a` I used `makeGeneRegion` which uses biomaRt to find the protein coding genes on the negative strand on yeast chromosome IV from bases 1300000 to 1310000. It adds the names in white and makes dark blue boxes for the protein coding regions.
- To create object `b` I used `makeGenomeAxis` which simply creates the genome axis with a tick every 1000 base pairs.
- Object `c` is very similar to object `a` except that it looks for protein coding regions on the plus strand.
- Object `d` uses the snyder data frame (Nagalakshmi data) and plots the points in yellow. The points plotted are those within 0 and 300.
- Object `e` plots the data for the plus strand using the david data frame. The points are in purple with a smaller point size. As this in array data, I used `makeGenericArray`.
- Object `f` is very similar to object `e` except that it plots the points in green and it uses the data for the negative strand.
- Object `g` plots the Lee data in blue using the nislow data frame.

- Object `h` plots in red the conservation information using the data frame with the same name.
- `rOverlay` and `tOverlay` are rectangle and text overlays. The rectangle one covers tracks 1 to 5 and helps highlight a region of interest. I used the text overlay to add my username to the plot.