LDheatmap (Version 0.99-4): Example of Adding Tracks

Jinko Graham and Brad McNeney

March 12, 2018

1 Introduction

As of version 0.9, `LDheatmap` allows users to "flip" the heatmap below a horizontal line in the style of Haploview. This extension was made so that "tracks" of genomic annotation information could be placed above the genetic map. Facilities for adding tracks are illustrated in this vignette. Limitations of the current implementation are discussed briefly in Section 6.2

2 Getting Started

First load the package, the snpStats package (to use its `SnpMatrix` data structure), the example dataset `CIMAP5.CEU`, and an R data file that contains two LDheatmap objects and one picture that will be used in the vignette.

```
> library(LDheatmap)
> library(snpStats)
> data(GIMAP5.CEU)
> load(system.file("extdata/addTracks.RData",package="LDheatmap"))
```

The object `GIMAP5.CEU` is is a list with two elements: `snp.data` and `snp.support`. `snp.data` is a SnpMatrix object from the `snpStats` package, containing the SNP genotypes. Rows correspond to subjects and columns correspond to SNPs. `snp.support` is a data frame whose rows correspond to SNPs and whose columns give information on the SNPs, such as their alleles and genomic location. The help file `help("GIMAP5.CEU")` gives full details. In addition to `GIMAP5.CEU`, you should have the LDheatmap objects `llGenes` and `llGenesRecomb` in your workspace. These objects are the heatmap with tracks for genes and recombination rates added incrementally. Adding these tracks requires fetching information from the UCSC genome browser, which requires an internet connection and can be time-consuming. These objects are provided so that users can see the track-annotated heatmaps without having to wait. The object `GIMAP5ideo` should also have been loaded. This is an array that represents a bitmap image of a chromosome ideogram used in Section 5 of the vignette.

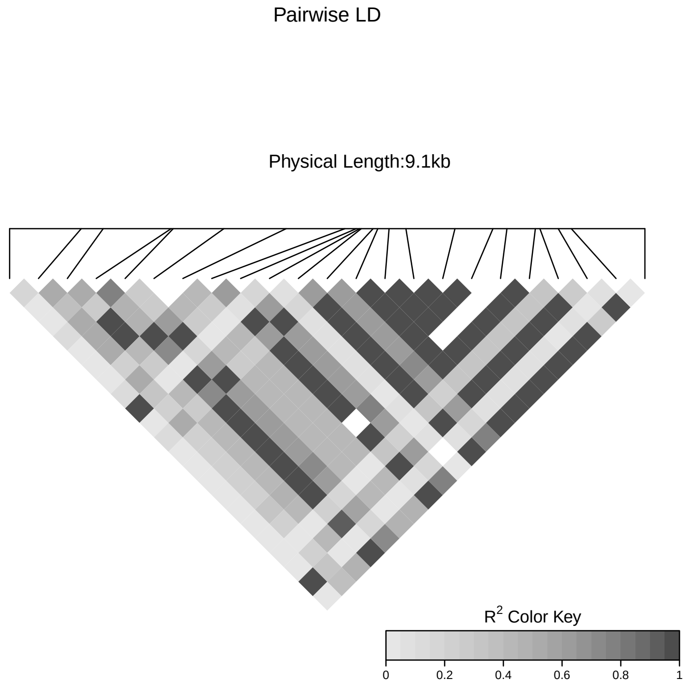

Figure 1: LDheatmap of GIMAP5.CEU data.

The basic LDheatmap for the `GIMAP5.CEU` data to which tracks will be added can be created with the following R commands:

```
> ll<-LDheatmap(GIMAP5.CEU$snp.data,GIMAP5.CEU$snp.support$Position,flip=TRUE)
```

Figure 1 displays the results.

3 Adding tracks from the UCSC genome browser

We will add "UCSC Gene" and "Recomb Rate" tracks to Figure 1. Data for these tracks is downloaded from the UCSC genome browser using the `rtracklayer` package. The genes track is added by the `LDheatmap.addGenes` function:

```
> llGenes <- LDheatmap.addGenes(ll, chr="chr7", genome="hg18")
```

It takes several minutes for this function to fetch the necessary data from the UCSC genome browser. To avoid this download time, users can draw the graphical object (grob) stored in `llGenes` directly (Figure 2) with the commands:

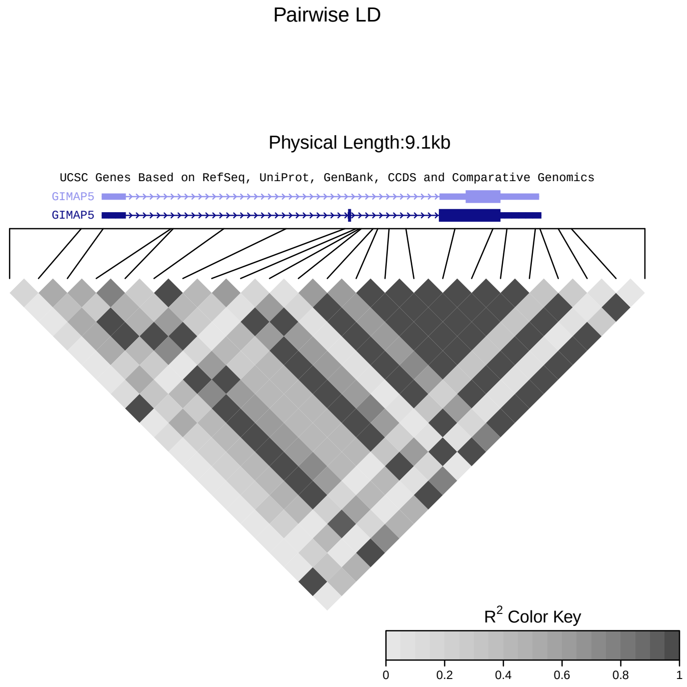

Figure 2: LDheatmap of GIMAP5.CEU data with UCSC Genes track.

```
> grid.newpage()
> grid.draw(llGenes$LDheatmapGrob)
```

We now add a track that illustrates recombination rates:

```
> llGenesRecomb <- LDheatmap.addRecombRate(llGenes, chr="chr7", genome="hg18")
```

The result is shown in Figure 3. Users who wish to avoid the download time may draw the grob from the `llGenesRecomb` object with the commands:

```
> grid.newpage()
> grid.draw(llGenesRecomb$LDheatmapGrob)
```

4 Adding a scatterplot above the heatmap

Above the two tracks fetched from the UCSC genome browser we will superpose a scatterplot depicting the results of association tests between each SNP in the *GIMAP5* gene and a hypothetical disease. We create a set of hypothetical p-values that includes one very small value.

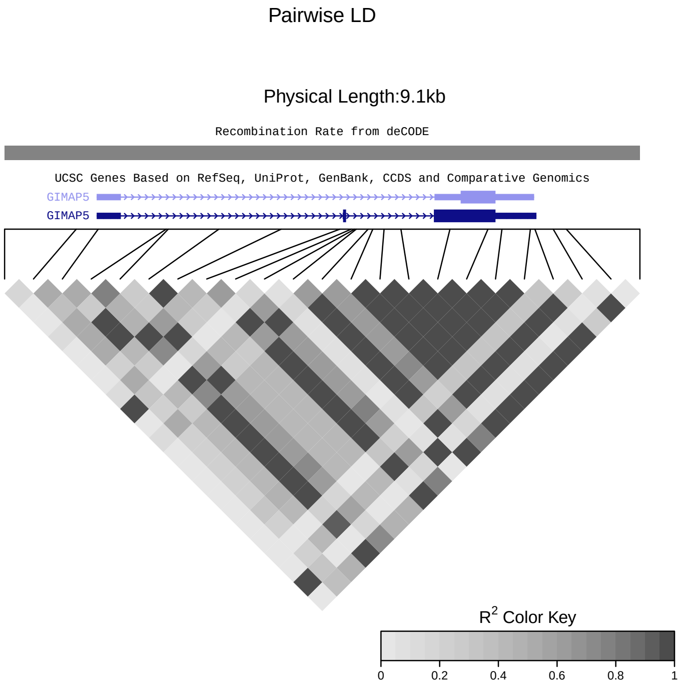

Figure 3: LDheatmap of GIMAP5.CEU data with UCSC Genes and Recomb Rate tracks.

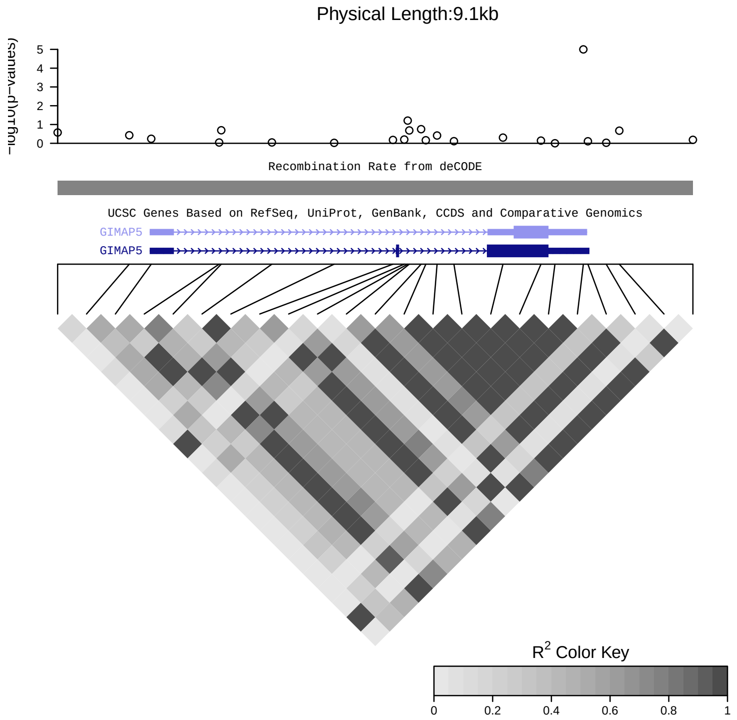

Figure 4: LDheatmap of GIMAP5.CEU data with gene, recombination and Manhattan plot tracks.

```
> set.seed(1)
> atests<-runif(nrow(GIMAP5.CEU$snp.support))
> names(atests)<-rownames(GIMAP5.CEU$snp.support)
> atests["rs6598"]<-1e-5
```

A Manhattan plot (scatterplot of the $-\log_{10}$ p-values versus genomic location of the SNPs) can be added to the heatmap as follows:

```
> llGenesRecombScatter<-LDheatmap.addScatterplot(llGenesRecomb,-log10(atests),
+ ylab="-log10(p-values)")
```

The result is shown in Figure 4.

5 Adding other graphical objects

Users may add an arbitrary grob above the heatmap with the `LDheatmap.addGrob` function. This function aligns the grob so that its horizontal dimension matches the gene map of the heatmap. The idea is to allows users to create tracks from grobs returned by other R packages that use grid graphics (e.g., ggplot2), or to add images using functions such as rasterGrob or pictureGrob. Development of LDheatmap.addGrob is in its early stages.

The first illustration of LDheatmap.addGrob is to add a Manhattan plot created with the qplot (quick plot) function of ggplot2. We convert the output of qplot to a grob that can be added to our heatmap.

```
> require("ggplot2")
> posn<-GIMAP5.CEU$snp.support$Position
> manhattan2<-ggplotGrob(
+ {
+ qplot(posn,-log10(atests),xlab="", xlim=range(posn),asp=1/10)
+ last_plot() + theme(axis.text.x=element_blank(),
+               axis.title.y = element_text(size = rel(0.75)))
+ }
+ )
> llQplot<-LDheatmap.addGrob(ll,manhattan2,height=.7)
```

The resulting heatmap is given in Figure 5

In Figure 5 the entire width of the manhattan2 grob has been aligned to the genetic map, but the x-axis within this grob does not align. As a kludge, the Manhattan plot can be aligned manually by creating space on the display and using grid graphics functionality to place the grob:

1. Use LDheatmap.addGrob to make space for manhattan2 grob, by adding a track comprised of a rectangle with white lines.

2. Use grid graphics to create a viewport on the display to hold the manhattan2 grob. This step takes some trial and error with the placement and size of the viewport.

3. Draw the manhattan2 grob.

4. (optional) Clean up by popping the viewport created to hold the manhattan2 grob off of the viewport tree.

Alignment of the x-axis of the manhattan2 is device-dependent. The following commands align the grob when displayed on the pdf() device. The resulting plot does not render well when the vignette is generated from its Sweave source, and so the plot is excluded from the vignette.

```
> manhattan2<-ggplotGrob(
+ {
+ qplot(posn,-log10(atests),xlab="", xlim=range(posn),asp=1/10)
+ last_plot() + theme(axis.text.x=element_blank(),
+               axis.title.y = element_text(size = rel(0.75)))
+ }
+ )
> llQplot2<-LDheatmap.addGrob(ll,rectGrob(gp=gpar(col="white")),height=.2)
```

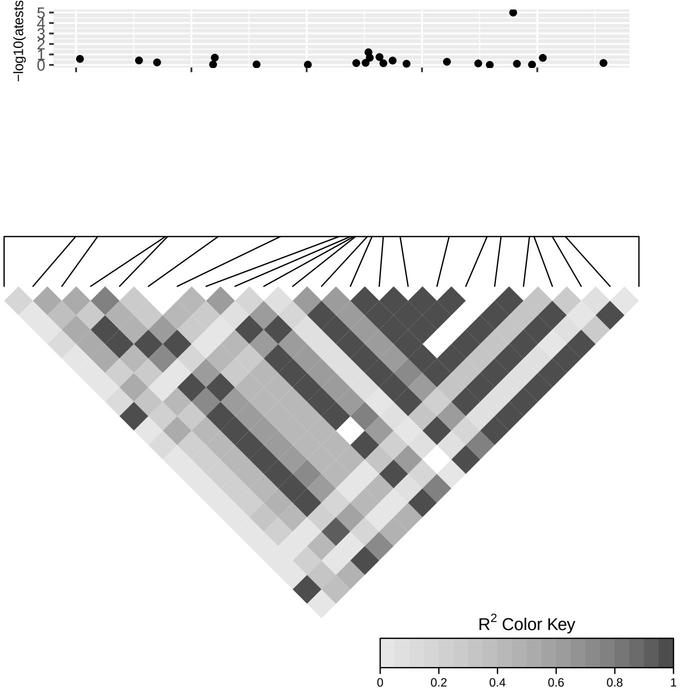

Figure 5: LDheatmap of GIMAP5.CEU data with Manhattan plot created with the `qplot` function.

```
> pushViewport(viewport(x=.48,y=.76,width=.99,height=.2))
> grid.draw(manhattan2)
> popViewport(1)
> dev.off()
```

```
null device
          1
```

As another illustration of `LDheatmap.addGrob`, we add to a heatmap an ideogram of human chromosome 7 with the GIMAP5 region highlighted. The ideogram was manually saved as a PDF file from the UCSC genome browser and was then exported from PDF to a PNG file using Adobe Acrobat. The PNG file was read into R with the `readPNG` function from the `png` package and saved as the object `GIMAP5ideo`, which was loaded at the beginning of the vignette. We package the image as a grob with the `rasterGrob` function and use `LDheatmap.addGrob` to add it to the heatmap:

```
> llImage<-LDheatmap.addGrob(ll,rasterGrob(GIMAP5ideo))
```

The results are shown in Figure 6. The resolution of the ideogram image is rather poor, and the ideogram is only readable when one zooms in on the plot.

6 Details and Limitations

6.1 Details

Grobs representing the components of an `LDheatmap` object are stored as children of the main `LDheatmapGrob` grob:

```
> names(ll$LDheatmapGrob$children)
```

```
[1] "heatMap" "geneMap" "Key"
```

Grobs that represent tracks are added to the list of children:

```
> names(llGenesRecombScatter$LDheatmapGrob$children)
```

```
[1] "heatMap"     "geneMap"     "Key"         "transcripts"
[5] "recombRate"  "association"
```

The `transcripts` grob is the genes track added by `LDheatmap.addGenes`, `recombRate` is the graphic depicting recombination rates added by `LDheatmap.addRecombRate`, and `association` is the Manhattan plot added by `LDheatmap.addScatter`. Tracks added by `LDheatmap.addGrob` have names assigned by the grid graphics system:

```
> names(llQplot$LDheatmapGrob$children)
```

```
[1] "heatMap"       "geneMap"       "Key"
[4] "GRID.gTree.51"
```

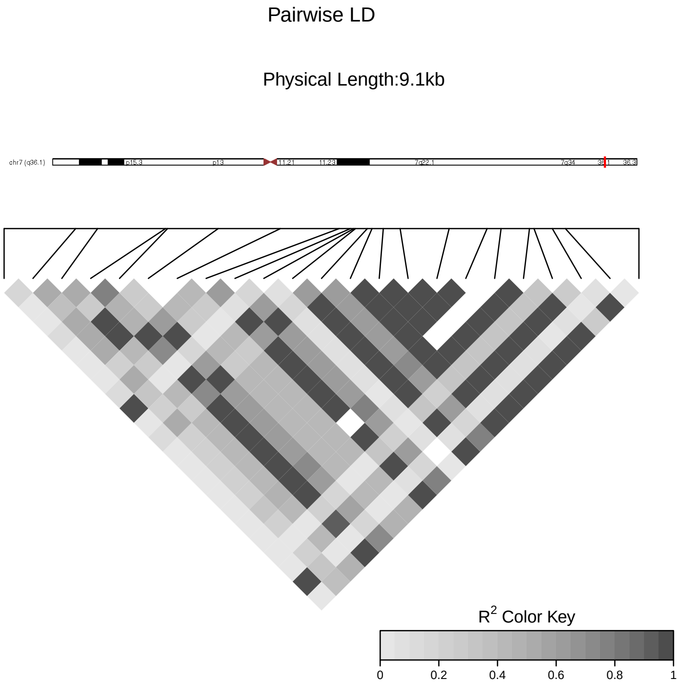

Figure 6: LDheatmap of GIMAP5.CEU data with chromosome 7 ideogram showing the GIMAP5 region as a small vertical red line.

```
> names(llImage$LDheatmapGrob$children)

[1] "heatMap"       "geneMap"       "Key"
[4] "GRID.gTree.94"
```

6.2 Limitations

Adding tracks requires creating space for the track and moving the gene map title (the one that reports the map length) and the main title of the graph. Currently the calculation of space needed to insert a track only accounts for tracks previously added by LDheatmap.addGenes, LDheatmap.addRecombRate, or LDheatmap.addScatter. Thus, a track added by LDheatmap.addGrob will be over-written by subsequent tracks. Another limitation is that SNP names added with the SNP.name argument of LDheatmap are over-written by any tracks added to the heatmap. We intend to fix these limitations in a future release of the package.