Journal of Statistical Software

October 2006, Volume 16, Code Snippet 3. *http://www.jstatsoft.org/*

LDheatmap: An R Function for Graphical Display of Pairwise Linkage Disequilibria between Single Nucleotide Polymorphisms

Ji-Hyung Shin
Simon Fraser University

Sigal Blay
Simon Fraser University

Brad McNeney
Simon Fraser University

Jinko Graham
Simon Fraser University

Abstract

We describe the R function `LDheatmap()` which produces a graphical display, as a heat map, of pairwise linkage disequilibrium measurements between single nucleotide polymorphisms within a genomic region. `LDheatmap()` uses the **grid** graphics system, an alternative to the traditional R graphics system. The features of the `LDheatmap()` function and the use of tools from the **grid** package to modify heat maps are illustrated by examples.

Keywords: single nucleotide polymorphisms, linkage disequilibrium, grid graphics.

1. Introduction

Single nucleotide polymorphisms (SNPs) are the most common form of genetic variation in the human genome. Due to their abundance and ease of genotyping, SNPs have become popular markers for genetic association studies. Although identifying as many SNPs as possible within a candidate gene is important in finding disease susceptibility alleles, typing them all can be expensive, and testing them for association with traits can lead to multiple-comparison issues. Moreover, due to genetic linkage, nearby SNPs within candidate genes are often highly correlated. Hence, it has become common practice to instead genotype only a subset of SNPs within a candidate gene.

Understanding the patterns of association or linkage disequilibrium (LD) between SNPs can aid in selecting SNP subsets. However, for a dense map of SNPs, it can be difficult to interpret results from tabular summaries of pairwise LD measurements since the number of measurements increases rapidly with the number of SNPs within a genomic region. As a

tool for interpretation of LD patterns, we developed an R (**?**) function `LDheatmap()` which provides a graphical summary of pairwise LD measurements as a heat map.

2. LDheatmap(): Overview

The function `LDheatmap()` takes an input data set that provides information on pairwise LD between SNPs in a genomic region, plots color-coded values of the pairwise LD measurements, and returns an object of class `"LDheatmap"` containing a number of components.

The input data set can be a data frame containing SNP genotypes, a matrix of pairwise LD measurements, or an `LDheatmap` object returned by the function `LDheatmap()`. SNP genotypes must be `genotype` objects created by the `genotype()` function from the **genetics** package (**?**). When genotypes are provided, LD measurements are computed using the function `LD()` from the **genetics** package. The user can specify either the squared allelic correlation r^2 (**?**) or Lewontin's $|D'|$ (**?**) as the measure of LD to be plotted. Users who have pre-computed pairwise r^2 or $|D'|$ measures, or who wish to plot some other measure of pairwise LD can provide a matrix of LD measurements instead of genotypes. In fact, any square matrix with values between 0 and 1 inclusive above the diagonal can be displayed by `LDheatmap()`. When `LDheatmap()` is passed an object of class `"LDheatmap"`, the function uses the object's `LDmatrix` component. `LDmatrix` is the matrix of LD measurements produced by or passed to the previous function call used to create the `LDheatmap` object. An optional diagonal line, drawn from the bottom left to the top right of the display, can be added to indicate physical or genetic map positions of the SNPs, along with text reporting the total length of the genetic region in either kilobases (kb; for physical distance) or centi-Morgans (cM; for genetic distance). In the display of LD, SNPs appear along this diagonal line in the order specified by the user, as the horizontal and vertical coordinates increase. The ordering is achieved by adopting the conventions of the `image()` function, in which horizontal coordinates of the display correspond to rows of the matrix and vertical coordinates correspond to columns, and vertical coordinates are indexed in increasing order from bottom to top (rather than top to bottom as in a matrix).

`LDheatmap()` depends on the **grid** package; the graphics functions of `LDheatmap()` are written in the *grid graphics* system (**?**) which provides more flexibility when manipulating a plot than the traditional graphics system. Consequently, traditional graphics functions such as `par()` do not have any effect on `LDheatmap()`. The next section provides examples of the use of `LDheatmap()` and shows how heat maps can be modified using functions from the **grid** package.

3. Illustration

The `hapmapCEU` data set from the **LDheatmap** package will be used in the next examples.

```
> library(LDheatmap)
> data("CEUData")
```

This will load the genetic data `CEUSNP` and associated distance vector `CEUDist` included in the package. The `CEUSNP` data-set is a data frame of genotypes for 15 SNPs on chromosome 7, obtained from 60 Utah residents with northern and western European ancestry. These

data are from release 7 of the International HapMap project (?); see the `hapmapCEU` help file for a more complete description. The `CEUDist` vector contains the physical map locations (base-pair positions) of the 15 SNPs.

The following example shows a typical call to the `LDheatmap()` function. The heat map generated by this call is shown in Figure 1.

```
> MyHeatmap <- LDheatmap(CEUSNP, CEUDist, LDmeasure = "r",
+     title = "Pairwise LD in r^2", add.map = TRUE,
+     SNP.name = c("rs2283092", "rs6979287"), color = grey.colors(20),
+     name = "myLDgrob", add.key = TRUE)
```

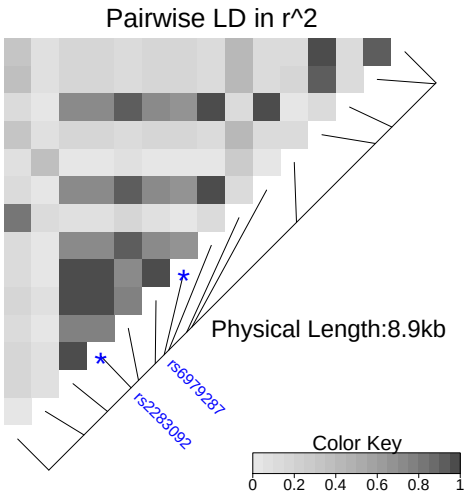

Figure 1: Heat map of pairwise LD measurements for the 15 SNPs in `CEUSNP` produced by `LDheatmap()`.

Each colored rectangle represents the squared correlation r^2 between a pair of SNPs (specified by `LD.measure="r"`). The vector `CEUDist` of physical map locations of the 15 SNPs provides the information on their relative positions, which are indicated on the diagonal line by line segments, and the total length of the genetic region indicated by the text `"Physical Length:8.9kb"` (all added by `add.map=TRUE`). Two of the SNPs are labeled by `SNP.name = c("rs2283092", "rs6979287")`. It is also possible to label selected SNPs without showing the other genetic information by specifying `add.map=FALSE`. The default grey-scale color-scheme is specified by `color=grey.colors(20)` and is indicated by the 'Color Key' on the bottom right corner of the plot (`add.key=TRUE`).

When the function is called, a ***grid*** *graphical object* (`grob`) named `LDheatmapGrob`, representing the heat map, is created and the graphical output is produced from this `grob`. The `grob` is also one of the components of the `LDheatmap` object returned by the `LDheatmap()` function. In this example, the returned `LDheatmap` object is stored as `MyHeatmap`, and its `LDheatmapGrob` component has name `"myLDgrob"`.

`LDheatmapGrob` is a `gTree` object (?) and has a hierarchical structure as shown in Figure 2. The children of `LDheatmapGrob` represent the heat map (`"heatMap"`), optional line parallel

to the diagonal of the image indicating the physical or genetic map positions of the SNPs (`"geneMap"`), and color-scale (`"Key"`).

The children of `"heatMap"` represent the region of colored rectangles (`"heatmap"`) and main title (`"title"`) of the heat map. When `add.map=TRUE`, `"geneMap"` is created with children representing the diagonal line (`"diagonal"`), line segments (`"segments"`) and text reporting the total length of the candidate region (`"title"`). When the parameter `SNP.name` is specified to label one or more SNPs, as in our example, two additional children are created, representing the labels (`"SNPnames"`) and the symbols plotted at the tips of the corresponding line segments (`"symbols"`). When `add.map=FALSE` and `SNP.name` is specified, only `"SNPnames"` is created. When `add.key=TRUE`, `"Key"` is created with children which represent the colored rectangles (`"colorKey"`), title (`"title"`), numeric labels (`"labels"`), ticks (`"ticks"`) and box frame (`"box"`) of the color legend. These `grob`s can be used to modify a heat map produced by the

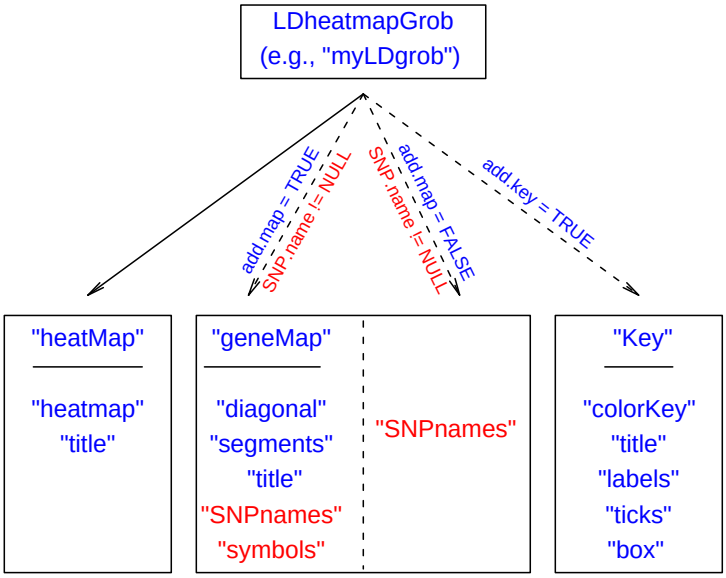

Figure 2: Hierarchical structure of `LDheatmapGrob`, the `grob` created by `LDheatmap()` to produce an LD heat map.

`LDheatmap()`. In the next section, we will show how to do this by examples.

4. Modifying a heat map

Modifying an LD heat map produced by `LDheatmap()` can be done *interactively* or *statically* using the functions `grid.edit()` or `editGrob()`, respectively (`?`). Interactive editing requires that a `grob` be drawn on the current device, such as the `grob` named `"myLDgrob"` on the current display in our example. Static editing requires that a `grob` be saved in the user's workspace, such as the `LDheatmapGrob` component of the `MyHeatmap` object saved in our

example. Suppose we wish to modify the font sizes and colors of the main title, text indicating the genetic region length and title of the color key. We can do so interactively with

```
> grid.edit(gPath("myLDgrob", "heatMap", "title"),
+     gp = gpar(cex = 1.25, col = "blue"))
> grid.edit(gPath("myLDgrob", "geneMap", "title"),
+     gp = gpar(cex = 0.8, col = "orange"))
> grid.edit(gPath("myLDgrob", "Key", "title"), gp = gpar(cex = 1.25,
+     col = "red"))
```

or we can do so statically with

```
> LD.grob1 <- editGrob(MyHeatmap$LDheatmapGrob,
+     gPath("heatMap", "title"), gp = gpar(cex = 1.25,
+         col = "blue"))
> LD.grob2 <- editGrob(LD.grob1, gPath("geneMap",
+     "title"), gp = gpar(cex = 0.8, col = "orange"))
> LD.grob3 <- editGrob(LD.grob2, gPath("Key", "title"),
+     gp = gpar(cex = 1.25, col = "red"))
```

The final `grob`, `LD.grob3`, can be drawn with

```
> grid.newpage()
> grid.draw(LD.grob3)
```

For more information on the functions `grid.edit()`, `editGrob()`, `grid.newpage()` and `grid.draw()` from the **grid** package, see their respective help files or `?`. Figure 3 shows the resulting modified heat map.

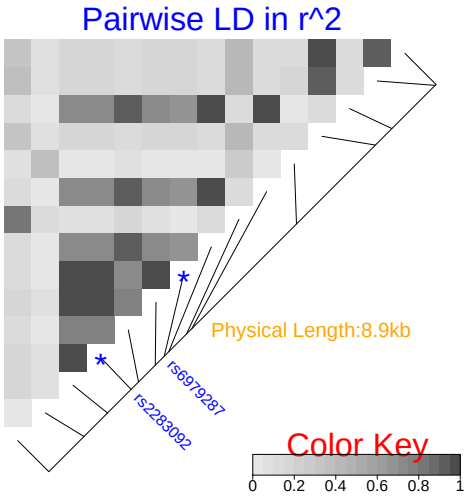

Figure 3: Heat map with modified colors and font sizes for `"Pairwise LD in r^2"` (main title), `"Physical Length:8.9kb"` (genetic length text) and `"Color Key"` (color-scale title).

5. Multiple heat maps on a single device

Displaying multiple heat maps on one graphical device may be useful for making comparisons. However, as mentioned earlier, changing `par()` settings does not affect **grid** graphics; hence modifying settings of `mfrow()` or `mfcol()` for arranging multiple plots in the traditional graphics system are not compatible with `LDheatmap()`. In the **grid** graphics system, multiple regions on a device can be defined and plotted by controlling ***grid*** *viewports* (**?**). One possible way to display and arrange multiple heat maps produced by `LDheatmap()` is with a *layout* (**?**). Alternately, users may manually position heat maps. Either approach involves controlling *grid viewports* and navigating the *viewport tree* managed by **grid** (**?**). In the following example two heat maps of different color scales (grey and white-to-red) are displayed side-by-side by manual positioning.

```
> VP1 <- viewport(x = 0, y = 0, width = 0.5, height = 1,
+     just = c("left", "bottom"), name = "vp1")
> pushViewport(VP1)
> LD1 <- LDheatmap(MyHeatmap, color = grey.colors(20),
+     title = "Pairwise LD in grey.colors(20)",
+     SNP.name = "rs6979572", name = "ld1", newpage = FALSE)
> upViewport()
> VP2 <- viewport(x = 0.5, y = 0, width = 0.5, height = 1,
+     just = c("left", "bottom"), name = "vp2")
> pushViewport(VP2)
> LD2 <- LDheatmap(MyHeatmap, color = heat.colors(20),
+     title = "Pairwise LD in heat.colors(20)",
+     SNP.name = "rs6979572", name = "ld2", newpage = FALSE)
> upViewport()
```

The argument `newpage=FALSE` tells `LDheatmap()` not to erase previously defined viewports: in the first call to `LDheatmap()` do not erase viewport `VP1`, and in the second function call do not erase `VP1` or `VP2`.

Our next example shows how to use the `grid.edit()` function to modify the `LDheatmapGrob`s `"ld1"` and `"ld2"`, created in the previous example, so that white lines separate each pixel in the heat map displayed on the left and so that the color of the `"geneMap"` title along the diagonal is changed from black (the default) to blue in the heat map displayed on the right.

```
> grid.edit(gPath("ld1", "heatMap", "heatmap"),
+     gp = gpar(col = "white", lwd = 2))
> grid.edit(gPath("ld2", "geneMap", "title"), gp = gpar(col = "blue"))
```

Note that the `gPath`s in the two calls to `grid.edit()` name the top-level `grob`s `"ld1"` and `"ld2"`, respectively, to specify which heat map is going be modified. Figure 4 shows the two modified heat maps displayed together.

```
> data("CHBJPTData")
> pop <- factor(c(rep("chinese", 45), rep("japanese",
+     45)))
```

Our final example shows how to produce a lattice-like plot with LDheatmaps in the panels, which is useful for viewing patterns of LD in different populations. The command

```
> data("CHBJPTData")
```

loads the genotypes `CHBJPTSNP` and associated distance vector `CHBJPTDist`. The data frame `CHBJPTSNP` contains genotypes for 13 SNPs on chromosome 7, from 45 Chinese and 45 Japanese individuals. The Chinese individuals were unrelated residents of the community at Beijing Normal University with at least 3 Han Chinese grandparents. The Japanese individuals were unrelated residents of the Tokyo metropolitan area with all grandparents from Japan. The data are from release 21 of the International HapMap project (The International HapMap Consortium 2005). We first create a factor variable describing the population:

```
> pop <- factor(c(rep("chinese", 45), rep("japanese",
+     45)))
```

The population variable may then be used to stratify the heat maps as follows.

```
> library(lattice)
> xyplot(1:nrow(CHBJPTSNP) ~ 1:nrow(CHBJPTSNP) |
+     pop, type = "n", scales = list(draw = FALSE),
+     xlab = "", ylab = "", panel = function(x,
+         y, subscripts, ...) {
+         LDheatmap(CHBJPTSNP[subscripts, ], CHBJPTDist,
+             newpage = FALSE)
+     })
```

The resulting heat maps are shown in Figure 5.

6. Further Notes

The package **LDheatmap** contains two other functions written in the **grid** graphics system. `LDheatmap.highlight()` and `LDheatmap.marks()` can be used to highlight or mark with a symbol, respectively, pairwise LD measures on an LD heat map. For more details, see the documentation for the **LDheatmap** package.

7. Acknowledgments

We would like to thank Nicholas Lewin-Koh for his suggestion to modify our original implementation of `LDheatmap()` to use the **grid** graphics system, and for his work to develop the `LDheatmap.highlight()` and `LDheatmap.marks()` functions in the **LDheatmap** package. We also would like to thank the anonymous reviewers for their helpful comments regarding the package and manuscript. This research was supported by Natural Sciences and Engineering Research Council of Canada grants 227972-00 and 213124-03, by Juvenile Diabetes Research Foundation International grant 1-2001-873, by Canadian Institutes of Health Research grants NPG-64869 and ATF-66667, and in part by the Mathematics of Information Technology and

Complex Systems, Canadian National Centres of Excellence. JG is a Scholar of the BC Michael Smith Foundation for Health Research.

Affiliation:

Ji-Hyung Shin
Department of Statistics & Actuarial Science
Simon Fraser University
8888 University Drive,
Burnaby, British Columbia, V5A 1S6
E-mail: shin@sfu.ca

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 16, Code Snippet 3 *Submitted:* 2006-05-15
October 2006 *Accepted:* 2006-09-20

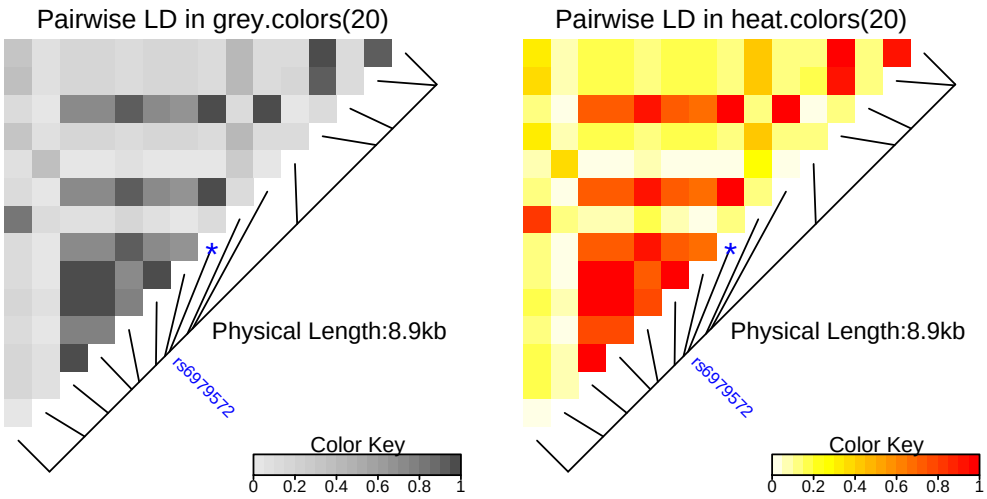

Figure 4: Modified heat maps displayed together. The heat map on the left uses a 'grey' (`grey.colors(20)`) color-scale. The heat map on the right uses a 'white-to-red' (`heat.colors(20)`) color-scale. White grid-lines were added to the heat map on the left and the color of the text `"Physical Length:8.9kb"` was changed from black to blue in the heat map on the right.

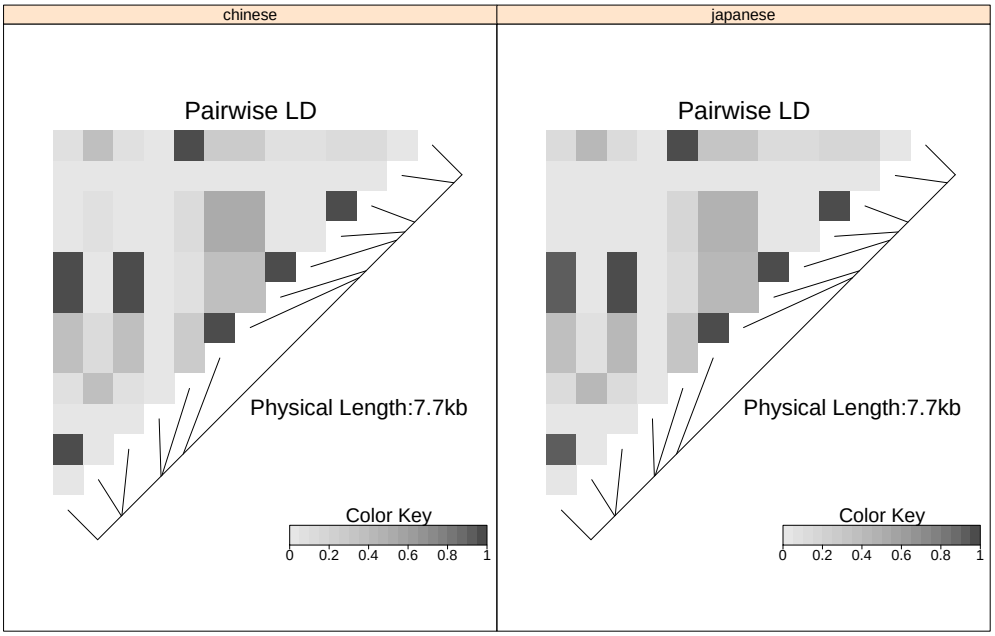

Figure 5: Lattice-like plot with LD heat maps in the panels.