

# ChromHeatMap

Tim F. Rayner

April 22, 2010

Cambridge Institute of Medical Research

## 1 Introduction

The **ChromHeatMap** package provides functions for visualising expression data in a genomic context, by generating heat map images in which data is plotted along a given chromosome for all the samples in a data matrix.

These functions rely on the existence of a suitable **AnnotationDbi** package which provides annotation for the probes on your array, and either an **ExpressionSet** or a data matrix with row names corresponding to probe identifiers and columns corresponding to samples.

The output heatmap can include sample clustering, and data can either be plotted for each strand separately, or both strands combined onto a single heat map. An idiogram showing the cytogenetic banding pattern of the chromosome will be plotted for supported organisms (at the time of writing: *Homo sapiens*, *Mus musculus* and *Rattus norvegicus*).

Once a heat map has been plotted, probes of interest can be identified interactively. These identifiers may then be mapped back to gene symbols and other annotation via the **AnnotationDbi** package.

## 2 Data preparation

Expression data in the form of a data matrix must initially be mapped onto its corresponding chromosome coordinates. This is done using the **makeChrStrandData**:

```
> library("ALL")
> data("ALL")
> selSamples <- ALL$mol.biol %in% c("ALL1/AF4", "E2A/PBX1")
> ALLs <- ALL[, selSamples]
> library("ChromHeatMap")
> chrdata <- makeChrStrandData(exprs(ALLs), lib = "hgu95av2")
```

The output *chrdata* object here contains the expression data indexed by coordinate. Note that the **makeChrStrandData** function is based on the **Makesense** function in the **geneplotter** package, removing the internal call to **lowess** to avoid smoothing the data (which is undesirable in this case). The **makeChrStrandData** function is used specifically because it incorporates information on both

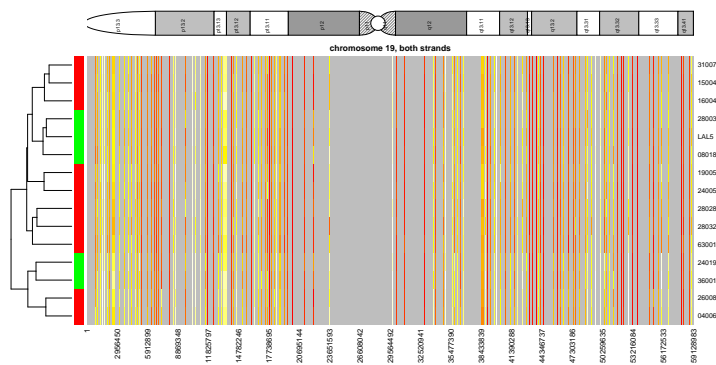
the start and end chromosome coordinates for each probe target. This allows the `plotChrMap` function to accurately represent target widths on the chromosome plot.

### 3 Plotting the heat map

Once the data has been prepared, a single call to `plotChrMap` will generate the chromosome heat map. There are many options available for this plot, and only a couple of them are illustrated here. Here we generate a whole-chromosome plot (chromosome 19), with both strands combined into a single heat map:

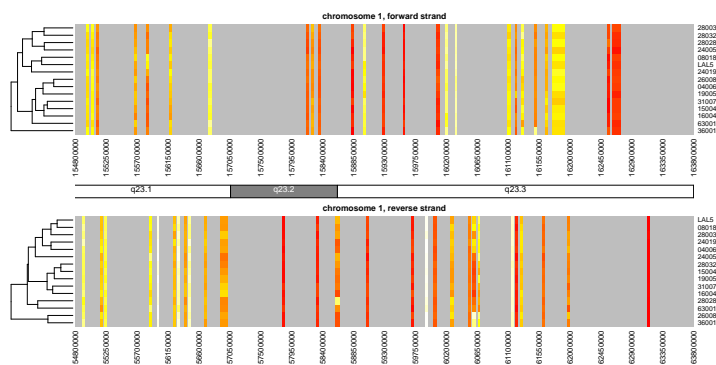
```
> groupcol <- ifelse(ALLs$mol.biol == "ALL1/AF4", "red", "green")
> plotChrMap(chrdata, 19, strands = "both", RowSideColors = groupcol)
```

ChrMapPlot  
Number of features plotted: 202



Chromosomes can be subsetted by cytoband or start/end coordinates along the chromosome. The following illustrates how one might plot the strands separately (this is the default behavior):

```
> plotmap <- plotChrMap(chrdata, 1, cytoband = "q23", interval = 5000,
+   srtCyto = 0, cexCyto = 1.2)
```



Other options include subsetting of samples, adding a color key to indicate sample subsets, deactivating the sample-based clustering and so on. See the help pages for `plotChrMap` and `drawMapDendro` for details.

Note that the default colors provided by the `heat.colors` function are not especially attractive or informative; consider using custom-defined colors, for example by using the **RColorBrewer** package.

The output of the `plotChrMap` function can be subsequently used with the `grabChrMapProbes` function which enables the user to identify the probes responsible for heatmap bands of interest.

Note that the `layout` and `par` options for the current graphics device are *not* reset following generation of the image. This is so that the `grabChrMapProbes` function can accurately identify the region of interest when the user interactively clicks on the diagram.

## 4 Interactive probe identification

Often it will be of interest to determine exactly which probes are shown to be up- or down-regulated by the `plotChrMap` heat map. This can be done using the `grabChrMapProbes` function. This takes the output of the `plotChrMap` function, asks the user to mouse-click the heatmap on either side of the bands of interest and returns a character vector of the probe identifiers in that region. These can then be passed to the **AnnotationDbi** function `mget` to identify which genes are being differentially expressed.

```
> probes <- grabChrMapProbes(plotmap)
> genes <- unlist(mget(probes, envir = hgu95av2SYMBOL, ifnotfound = NA))
```

Note that due to the way the expression values are plotted, genes which lie very close to each other on the chromosome may have been averaged to give a signal that could be usefully plotted at screen resolution. In such cases the probe identifiers will be returned concatenated, separated by semicolons (e.g. “37687\_i\_at;37688\_f\_at;37689\_s\_at”). Typically this is easily solved by zooming in on a region of interest, using either the “cytoband” or “start” and “end” options to `plotChrMap`. See also the “interval” option for another approach to this problem.

## 5 Session information

The version number of R and packages loaded for generating the vignette were:

```
R version 2.11.0 (2010-04-22)
x86_64-unknown-linux-gnu
```

```
locale:
```

```
[1] LC_CTYPE=en_US      LC_NUMERIC=C         LC_TIME=en_US
[4] LC_COLLATE=en_US    LC_MONETARY=C        LC_MESSAGES=en_US
[7] LC_PAPER=en_US      LC_NAME=C            LC_ADDRESS=C
[10] LC_TELEPHONE=C      LC_MEASUREMENT=en_US LC_IDENTIFICATION=C
```

attached base packages:

```
[1] tools      stats      graphics  grDevices  utils      datasets  methods
[8] base
```

other attached packages:

```
[1] ChromHeatMap_1.2.0  hgu95av2.db_2.4.1  org.Hs.eg.db_2.4.1
[4] RSQLite_0.8-4      DBI_0.2-5          annotate_1.26.0
[7] AnnotationDbi_1.10.0 ALL_1.4.7          Biobase_2.8.0
```

loaded via a namespace (and not attached):

```
[1] Biostrings_2.16.0  BSgenome_1.16.0    GenomicRanges_1.0.0
[4] IRanges_1.6.0      RCurl_1.4-1        rtracklayer_1.8.0
[7] XML_2.8-1          xtable_1.5-6
```