

# Introduction for heatmap3 package

Shilin Zhao

April 6, 2015

## Contents

<a href="#">1 Example</a>	<a href="#">1</a>
<a href="#">2 Highlights</a>	<a href="#">4</a>
<a href="#">3 Usage</a>	<a href="#">5</a>

## 1 Example

Simulate a gene expression data set with 40 probes and 25 samples. These samples are divided into 3 groups, 5 in control group, 10 in treatment A and 10 in treatment B group. We will use the groups as categorical phenotype and we assume there is one continuous phenotype. Then we will annotate the two phenotypes in heatmap result.

Here we provided two samples: The first example is simple. It generated the color bar as legend, plotted row side color bars with two columns, didn't plot row dendrogram, and some cells at the bottom left were labeled by specified colors. The second example plotted color bars and its legend to display a continuous variable in row side, texted labels with specified colors, plotted column side phenotype annotation. Then we provided a cutoff height so that the samples will be cut into different clusters by the cutoff. And statistic tests for annotations in different groups will be performed and the result will be returned.

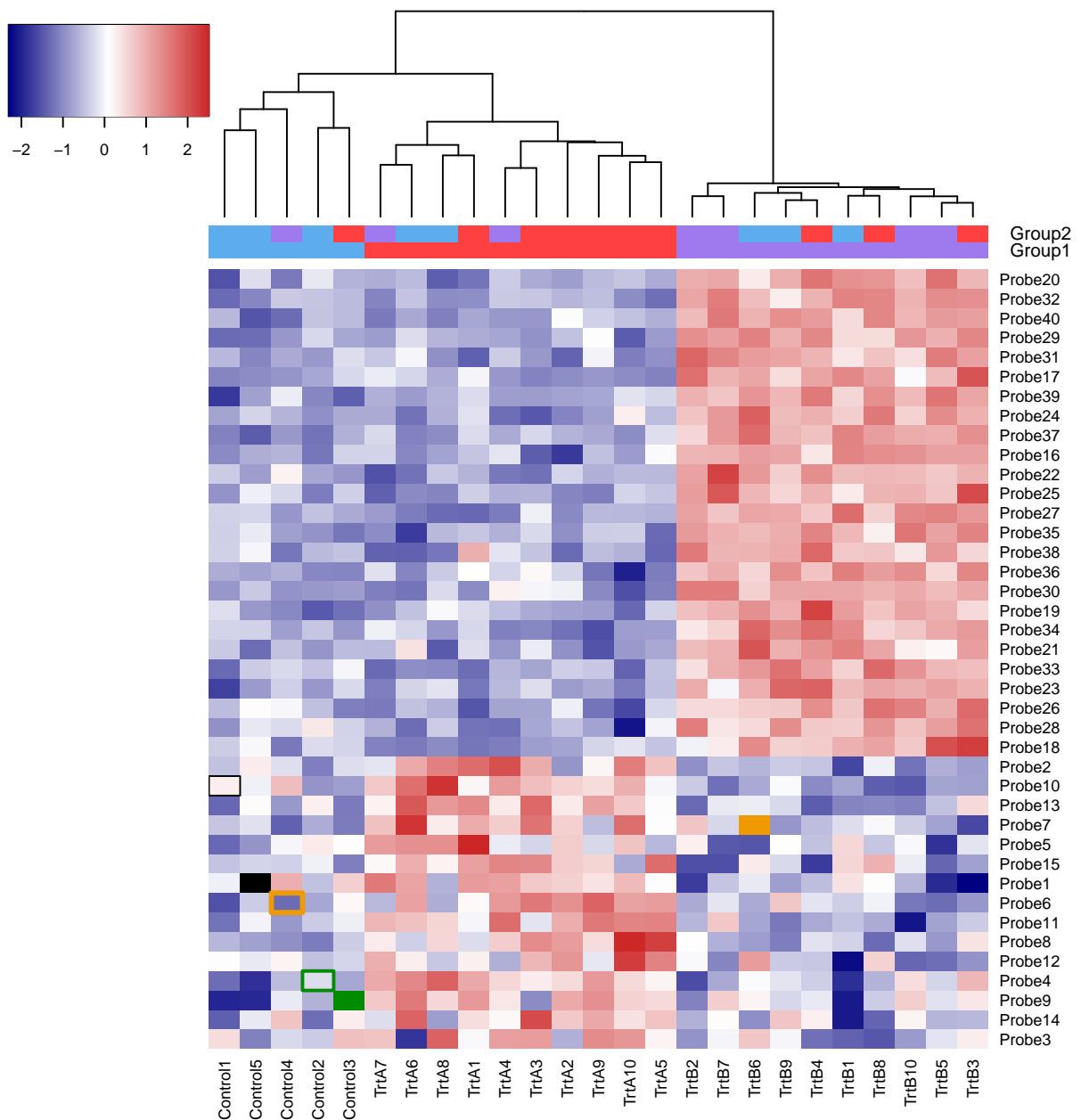
```
library(heatmap3)
example(heatmap3)

##
## hetmp3> #generate data
## hetmp3> set.seed(123456789)
##
## hetmp3> rnormData<-matrix(rnorm(1000), 40, 25)
##
## hetmp3> rnormData[1:15, seq(6, 25, 2)] = rnormData[1:15, seq(6, 25, 2)] + 2
##
## hetmp3> rnormData[16:40, seq(7, 25, 2)] = rnormData[16:40, seq(7, 25, 2)] + 4
##
## hetmp3> colnames(rnormData)<-c(paste("Control", 1:5, sep = ""), paste(c("TrtA", "TrtB"
```

```

## hetmp3+ rep(1:10,each=2), sep = "")
##
## hetmp3> rownames(rnormData)<-paste("Probe", 1:40, sep = "")
##
## hetmp3> ColSideColors<-cbind(Group1=c(rep("steelblue2",5), rep(c("brown1", "mediumpurp
##
## hetmp3> colorCell<-data.frame(row=c(1,3,5),col=c(2,4,6),color=c("green4","black","oran
##
## hetmp3> highlightCell<-data.frame(row=c(2,4,6),col=c(1,3,5),color=c("black","green4",'
##
## hetmp3> #A simple example
## hetmp3> heatmap3(rnormData,ColSideColors=ColSideColors,showRowDendro=FALSE,colorCell=c

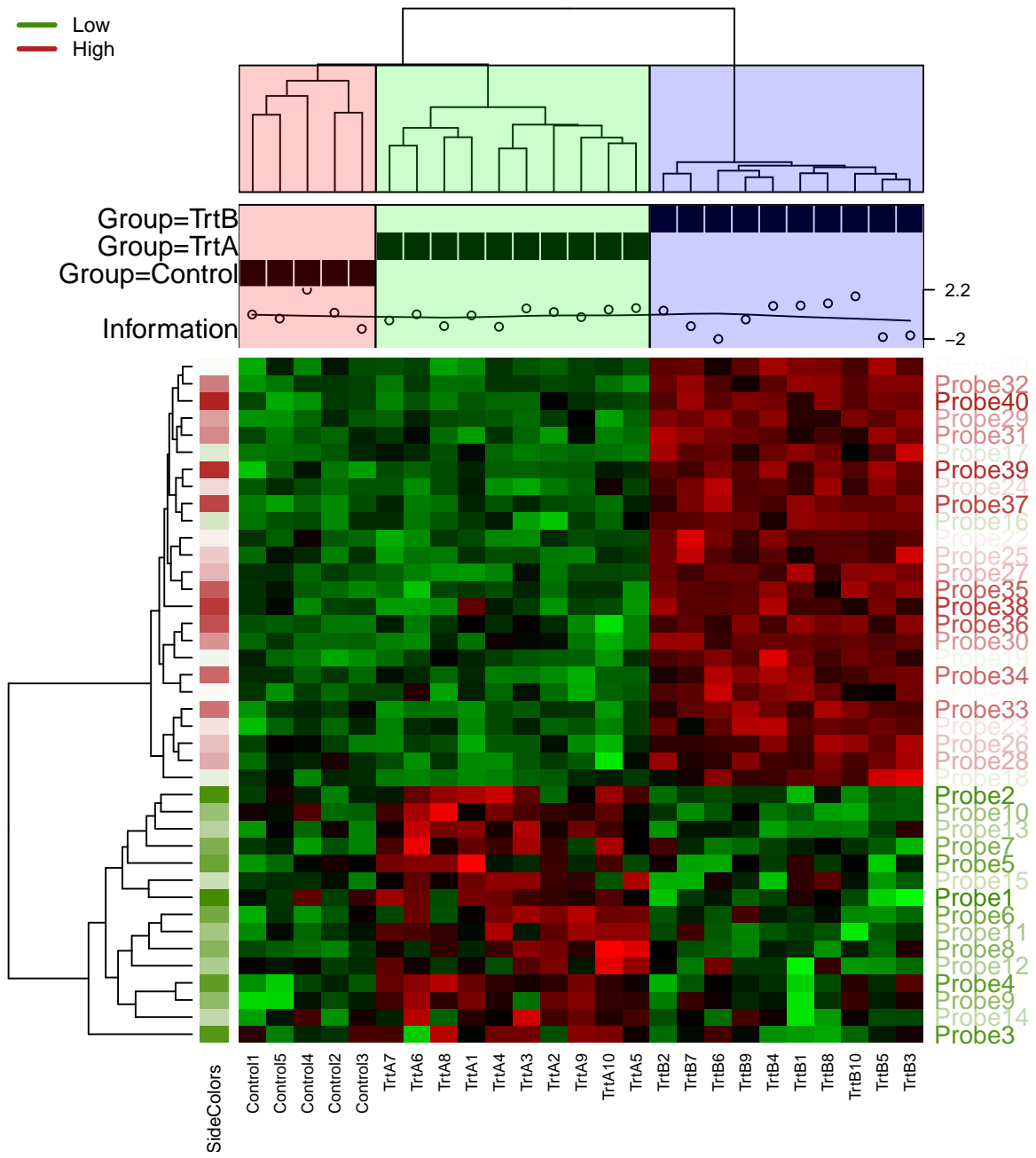
```



```

##
## hetmp3> #A more detail example
## hetmp3> ColSideAnn<-data.frame(Information=rnorm(25),Group=c(rep("Control",5), rep(c('
##
## hetmp3> row.names(ColSideAnn)<-colnames(rnormData)
##
## hetmp3> RowSideColors<-colorRampPalette(c("chartreuse4", "white", "firebrick"))(40)
##
## hetmp3> result<-heatmap3(rnormData,ColSideCut=1.2,ColSideAnn=ColSideAnn,ColSideFun=fun
## hetmp3+ showAnn(x),ColSideWidth=0.8,RowSideColors=RowSideColors,col=colorRampPalette(c
## hetmp3+ "black", "red"))(1024),RowAxisColors=1,legendfun=function() showLegend(legend=
## hetmp3+ "High"),col=c("chartreuse4","firebrick")),verbose=TRUE)
## The samples could be cut into 3 parts with height 1.2
## Differential distribution for Information, p value by ANOVA: 0.767
## Differential distribution for Group, p value by chi-squared test: 0

```



```
##
## hetmp3> #annotations distribution in different clusters and the result of statistic te
## hetmp3> result$cutTable
## $Information
##      Cluster 1 Cluster 2 Cluster 3   pValue
## Min.      -1.1140 -0.929200 -1.97300 0.7672268
## 1st Qu.    -0.2307 -0.319700 -1.47000      NA
## Median     0.1175  0.087170  0.06997      NA
## Mean       0.2541  0.002873 -0.17270      NA
## 3rd Qu.    0.2711  0.483100  0.87300      NA
## Max.       2.2270  0.662200  1.67000      NA
##
```

```
## $Group
##           Cluster 1 Cluster 2 Cluster 3      pValue
## Control           5         0         0 3.610865e-10
## TrtA              0         10         0          NA
## TrtB              0         0         10          NA
## Control_Percent   1         0         0          NA
```

## 2 Highlights

- Completely compatible with the original R function heatmap. You don't need to learn anything new or change your old commands to use it.
- Provides highly customizable function interface so that the users can use their own functions to generate legend or side annotation. And two convenient example functions were also provided.
- Provides a height cutoff so that the samples will be cut into different clusters by the cutoff and labeled by different colors. And then statistic tests for the distribution of annotations in different clusters will be performed.
- More convenient coloring features: Provides color legend for the input matrix automatically. A more fancy color series is set as default color. You can balance the colors in color legend so that you can ensure the median color will represent the 0 value.
- More powerful labeling features: The labels in axis could be labeled with colors. The side color bars support more than one column of colors.
- The color legend, column and row side color bars can exist in the same figure, which can't be done in other heatmap compatible packages.
- Improvement in parameters: Pearson correlation is set as default method; the agglomeration method for clustering now can be specified; the input values can be transformed into matrix automatically if it is a data.frame.

## 3 Usage

The main function is heatmap3, which was generate form the R function heatmap. So it is completely compatible with the original R function heatmap. You can use your commands for heatmap in heatmap3 as well. And you can use ?heatmap3 to get help for its new parameters. Here I just listed some of the new parameters for your information.

- legendfun: function used to generate legend in top left of the figure. More details will be discussed below.
- ColSideFun and ColSideAnn: function used to generate annotation and labeling figure in column side. The users can use any plot functions to generate their own figure. More details will be discussed below.

- ColSideCut: the value to be used in cutting column dendrogram. The dendrogram and annotation will be divided into different parts and labeled respectively.
- method: the agglomeration method to be used by hclust function.
- ColAxisColors, RowAxisColors: integer indicating which column of ColSideColors or RowSideColors will be used as colors for labels in axis.
- showColDendro, showRowDendro: logical indicating if the column or row dendrogram should be plotted.

A very important new feature in heatmap3 is the legendfun and ColSideFun parameter. You can generate your own legend in the top left of the figure by legendfun. And generate your own sample annotation in column side. Here we provided a function called showLegend as an example.

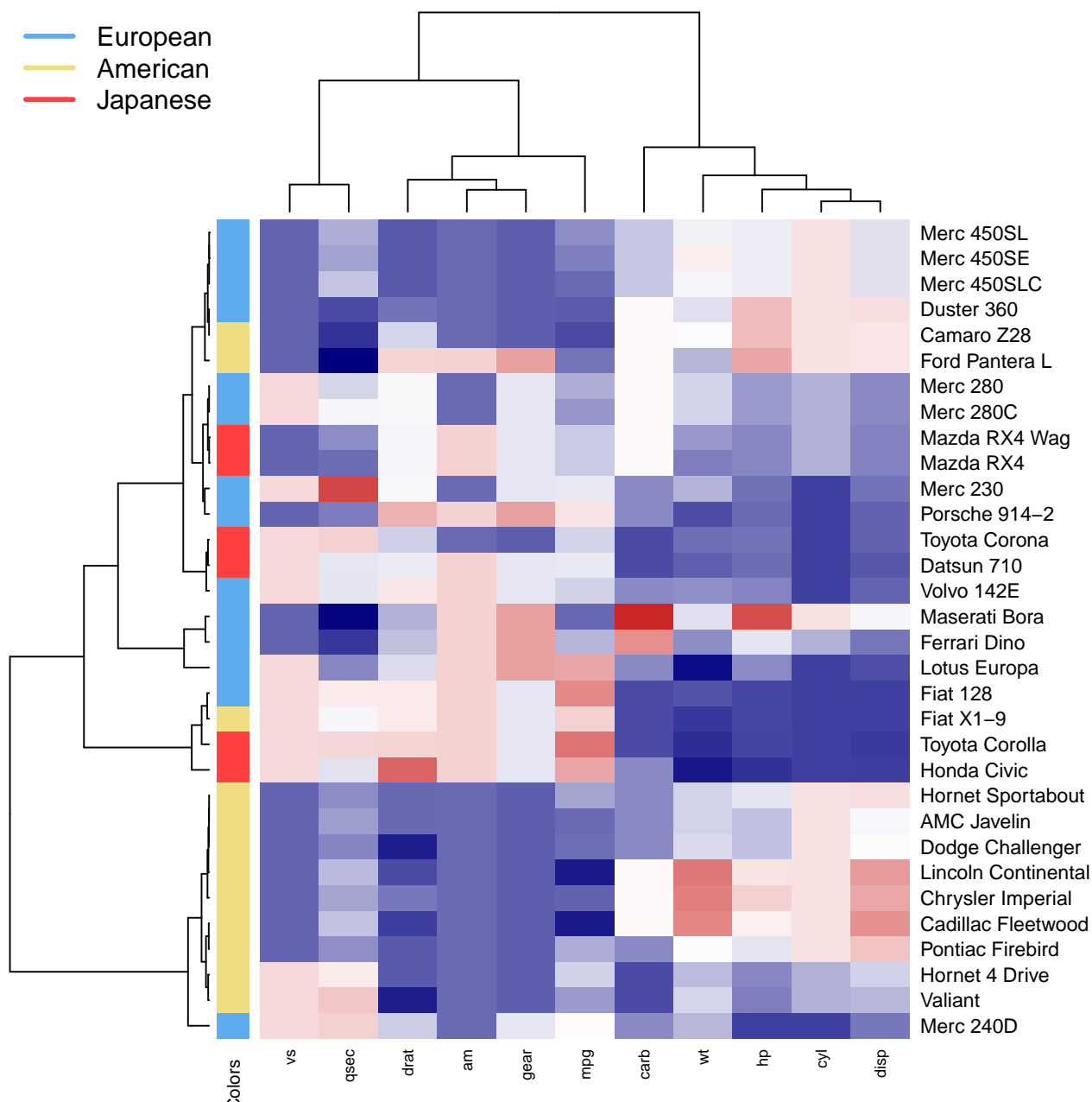
```
library(heatmap3)
showLegend

## function (legend = c("Group A", "Group B"), lwd = 3, cex = 1.1,
##   col = c("red", "blue"), ...)
## {
##   plot(0, xaxt = "n", bty = "n", yaxt = "n", type = "n", xlab = "",
##     ylab = "")
##   legend("topleft", legend = legend, lwd = lwd, col = col,
##     bty = "n", cex = cex, ...)
## }
## <environment: namespace:heatmap3>
```

This function is very simple. It first generates a empty figure and then uses the R function legend to generate legend. So you can simply write your own function to show legend or something else in the top left of the figure. Here is an example for showLegend function.

```
example(showLegend)

##
## shwLgn> RowSideColors<-rep("steelblue2",nrow(mtcars))
##
## shwLgn> RowSideColors[c(4:6,15:17,22:26,29)]<-"lightgoldenrod"
##
## shwLgn> RowSideColors[c(1:3,19:21)]<-"brown1"
##
## shwLgn> heatmap3(mtcars,scale="col",margins=c(2,10),RowSideColors=RowSideColors,legendfun=
## shwLgn+ showLegend(legend=c("European","American","Japanese"),col=c("steelblue2","lightgoldenrod",
## shwLgn+ "brown1"),cex=1.5))
```



We also provided a showAnn function as an example to show column side annotation.

```
example(showAnn)

##
## shwAnn> annData<-data.frame(mtcars[,c("mpg","am","wt","gear")])
##
## shwAnn> annData[,2]<-as.factor(annData[,2])
##
## shwAnn> annData[,4]<-as.factor(annData[,4])
##
## shwAnn> #Display annotation
## shwAnn> ## Not run:
## shwAnn> ##D showAnn(annData)
```

```
## shwAnn> ## End(Not run)
## shwAnn> #Heatmap with annotation
## shwAnn> heatmap3(t(mtcars), ColSideAnn=annData, ColSideFun=function(x)
## shwAnn+ showAnn(x), ColSideWidth=1.2, balanceColor=TRUE)
```

