Effects of TCR and T-bet on early transcription

Gretchen Harms Pritchard

July 2015

Experimental Design

Sorted naive (CD62L^{hi}CD44^{lo}) CD8^{+} T cells from WT and T-bet KO mice were cultured in vitro under neutral conditions (aIFNg and aIL-12) for 40hours with 0ug, 1ug, or 10ug plate bound aCD3 and soluble aCD28. RNA was isolated and analyzed using Illumina Bead Arrays.

Analysis

R packages used in this analysis and markdown report

```
library(lumi)
library(lumiMouseIDMapping)
library(lumiMouseAll.db)
library(RColorBrewer)
library(gplots)
library(ggplot2)
library(genefilter)
library(limma)
library(annotate)
library(reshape2)
library(Biobase)
library(knitcitations)
library(rmarkdown)
library(knitr)
library(devtools)
```

Set-up and QC

Begin by reading in the study design text file
Use this file to set treatment groups, sample labels, etc.

```
targets <- read.delim("GHPStudyDesign_alt.txt", sep="\t", stringsAsFactors = FALSE)
```

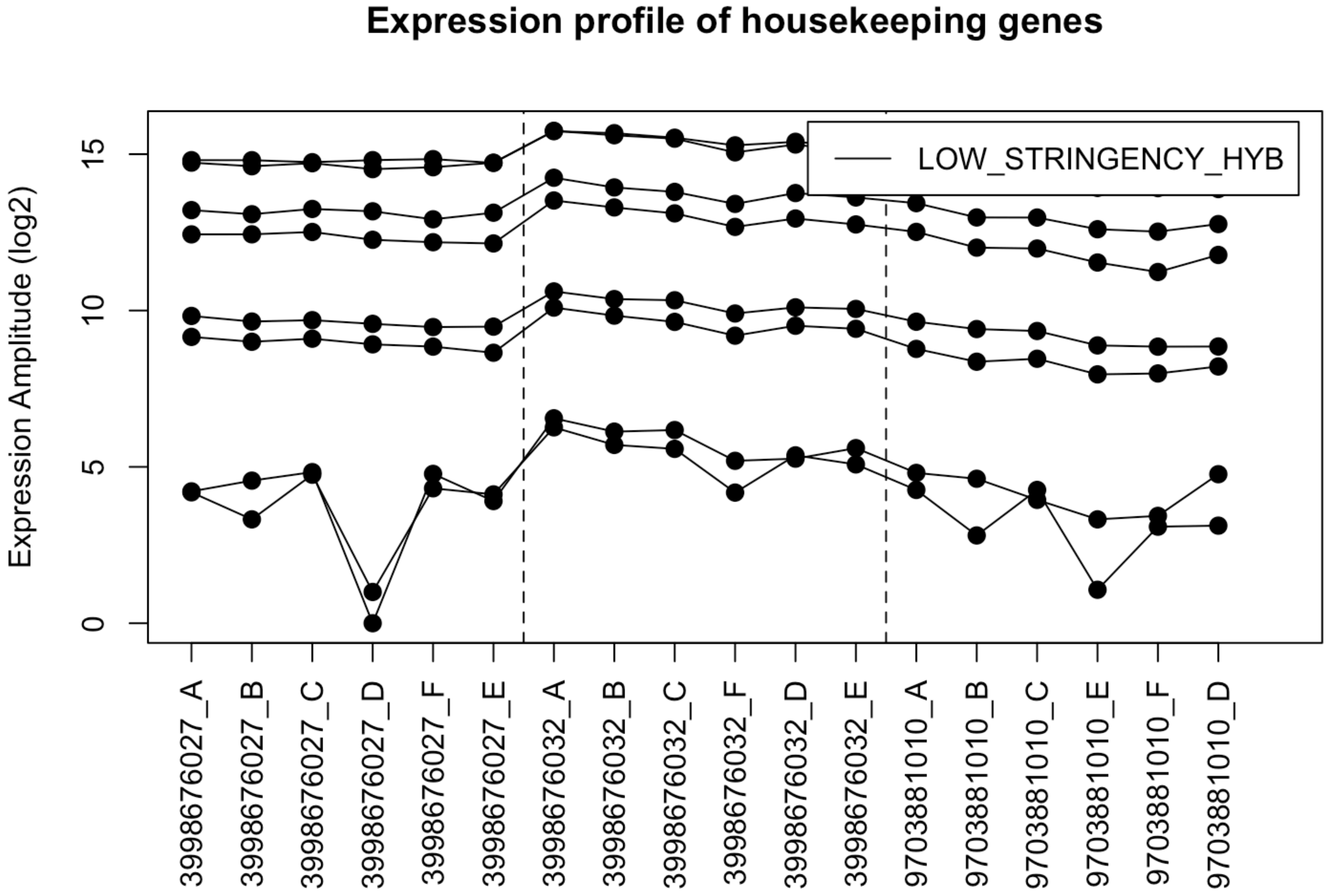
```
sampleLabels <- paste(targets$genotype, targets$treatment, targets$rep, sep=".")
myGroups <- factor(paste(targets$genotype, targets$treatment, sep="."))
```

Read in the raw array data:

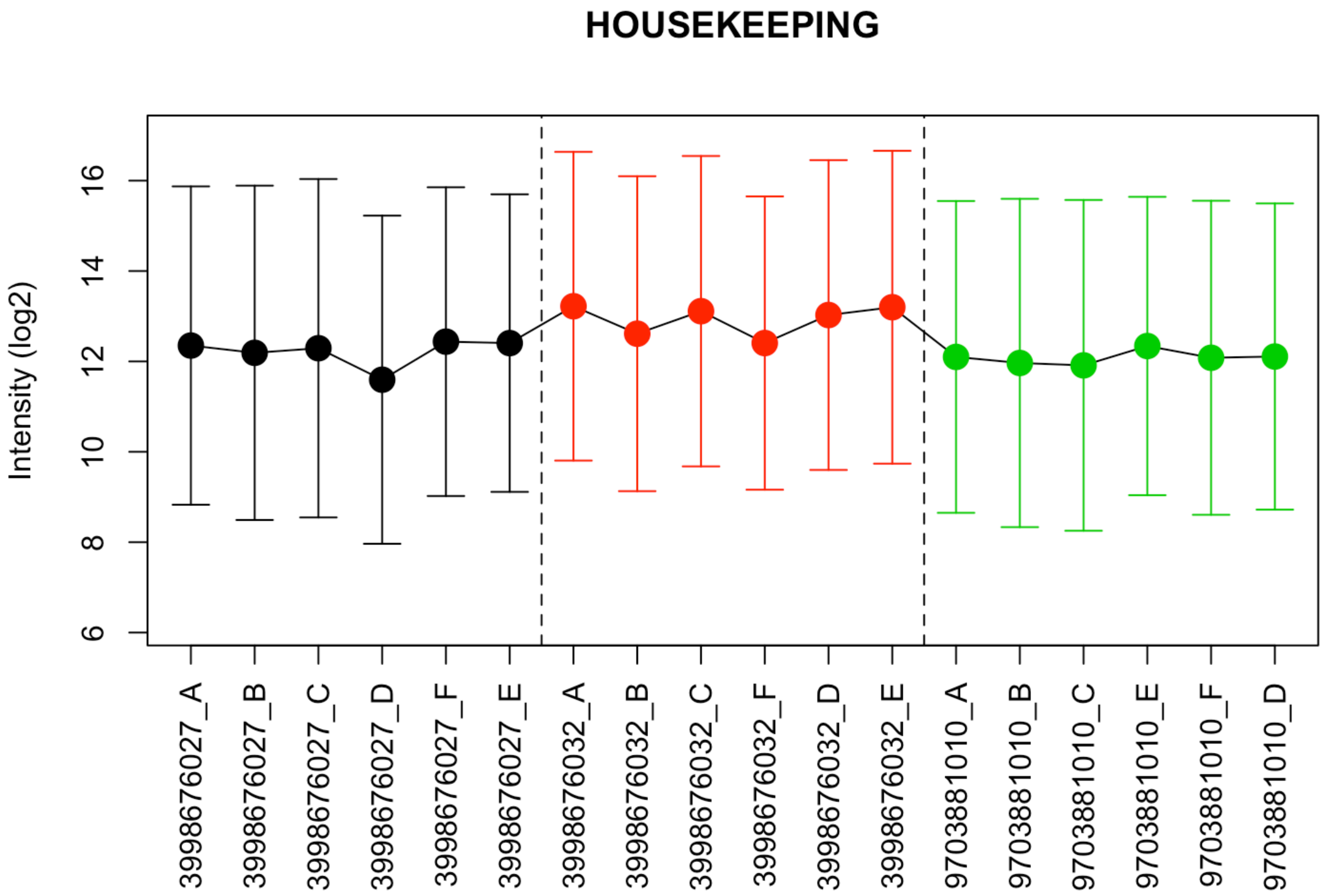
```
rawData <- lumiR("FinalReport_probes_NoNorm_NoBkrnd.txt", convertNuID = TRUE, sep = N
ULL, detectionTh = 0.01, na.rm = TRUE, lib = "lumiMouseIDMapping")
rawData <- addControlData2lumi("FinalReport_controls_NoNorm_NoBkrnd.txt", rawData)
rawData
controlData <- getControlData(rawData)
getControlType(controlData)
```

FIGURE 1: A measure of array quality and consistency across arrays

```
plotStringencyGene(rawData, lib = NULL, slideIndex = NULL, addLegend = TRUE, logMode
= TRUE)
```



```
plotControlData(rawData, type = 'HOUSEKEEPING', slideIndex = NULL, logMode = TRUE, ne
w = TRUE)
```



```
plotHousekeepingGene(rawData)
```

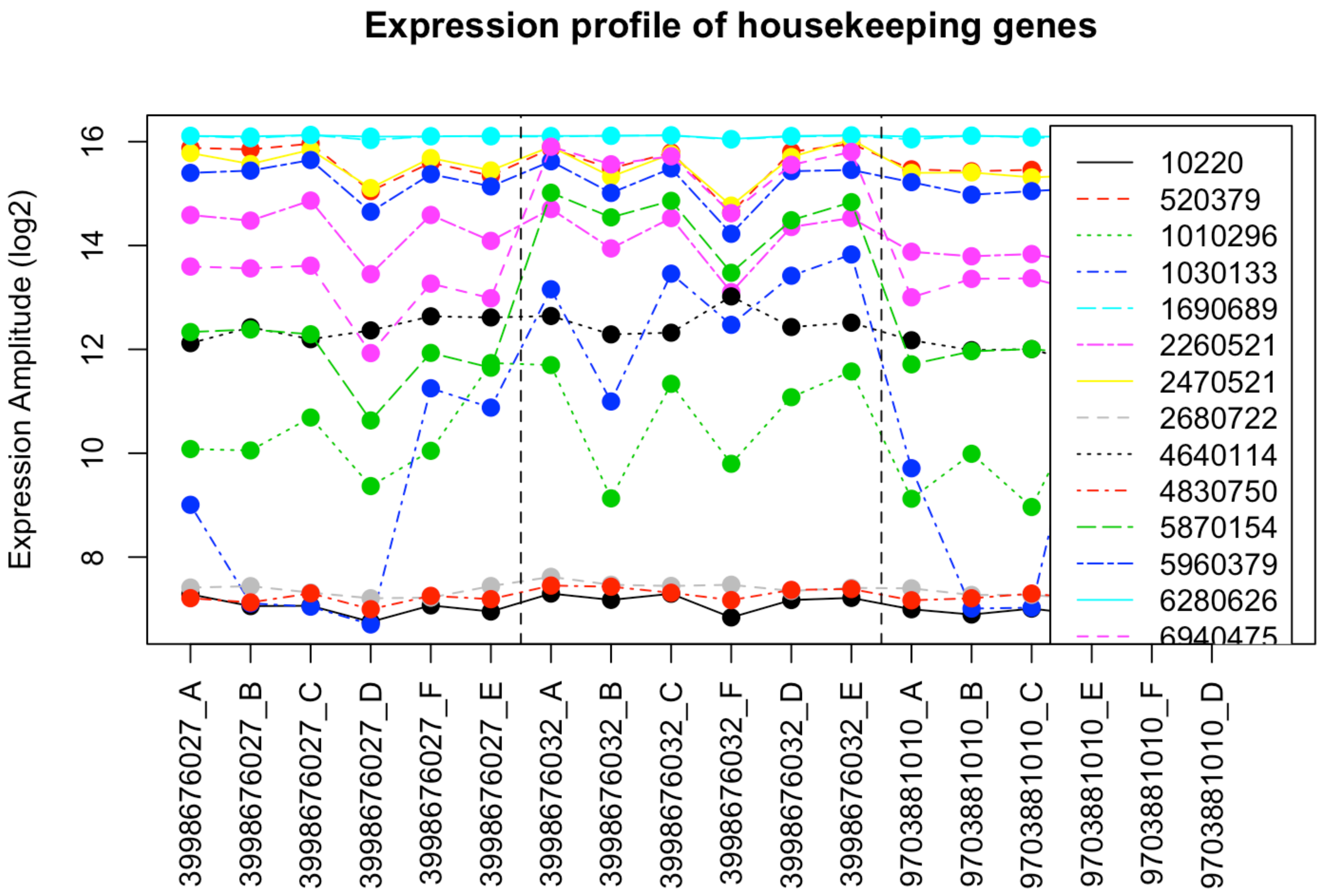


FIGURE 2: Distribution of signal intensity from each array (note: this is before any normalization or filtering is applied to the data)

```
#choose a color scheme for the next graph
cols.ALL <- topo.colors(n=18, alpha=1)
boxplot(rawData, ylab = "non-normalized log2 expression", main = "non-normalized data
- boxplots", col=cols.ALL)
```

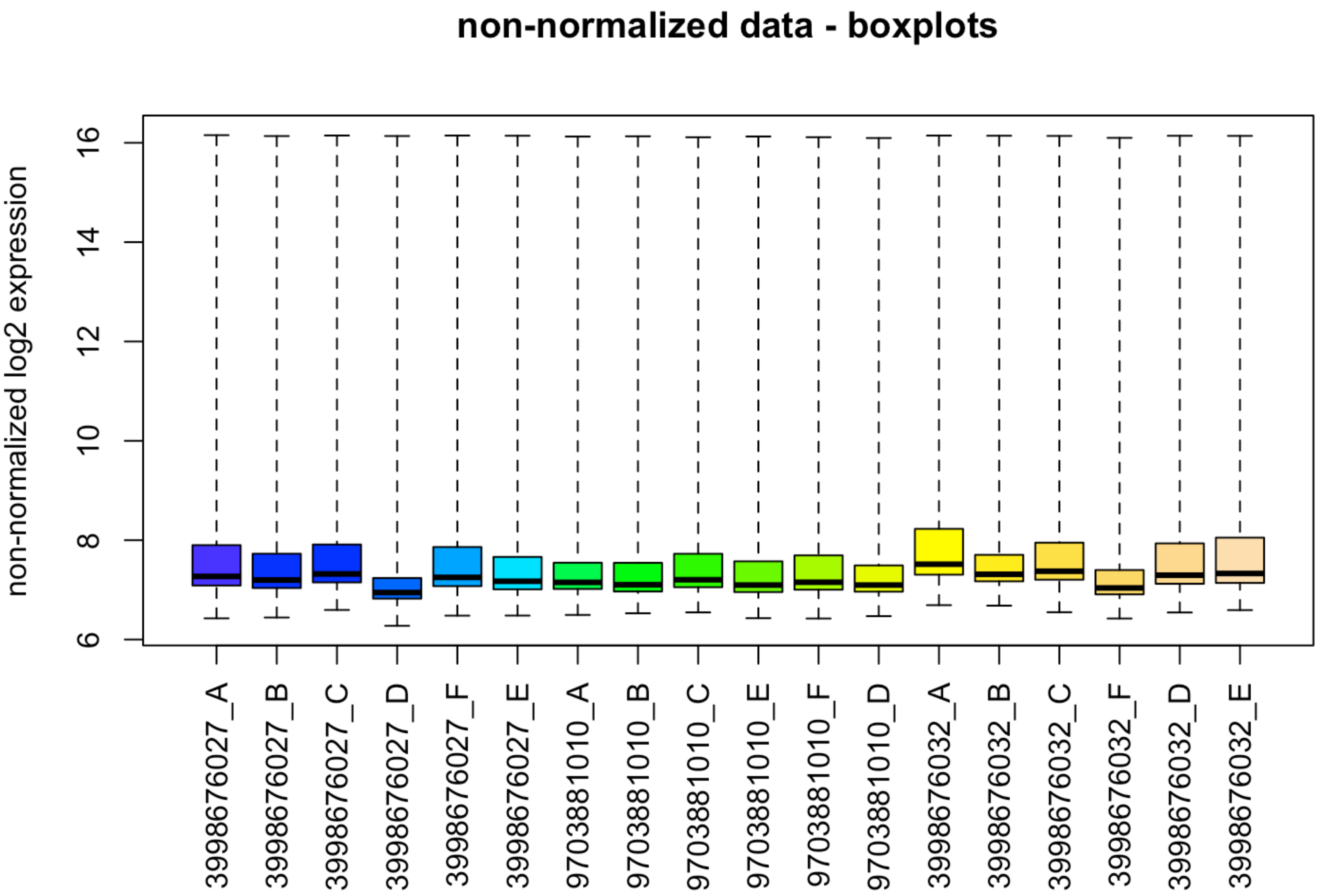
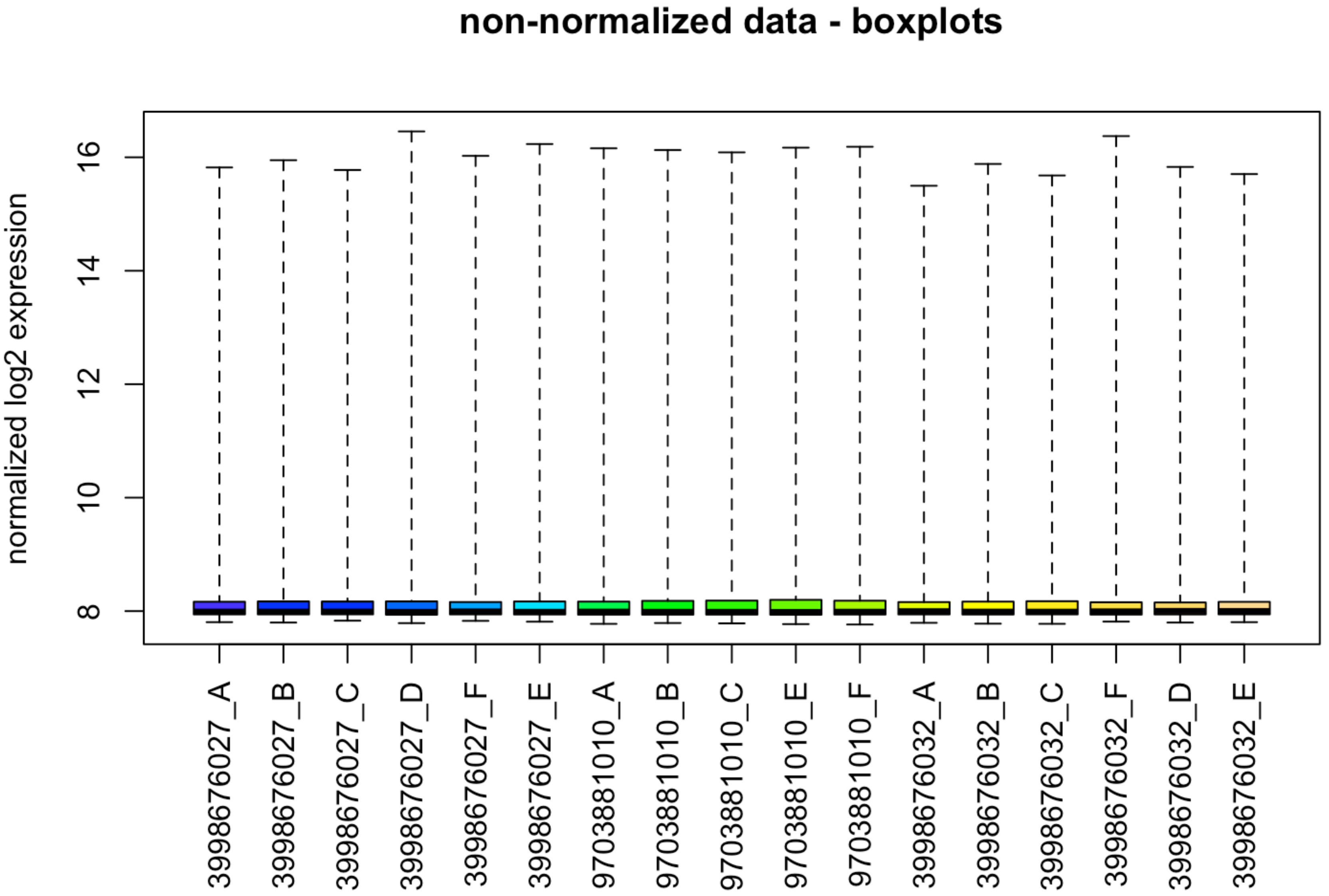


FIGURE 3: Normalized data to control for the variation between arrays then repeat the above graph

```
normData <- lumiExpresso(rawData, QC.evaluation=TRUE, normalize.param=list(method='rs
n'))
#subset normData to remove outlier
normData <- normData[,-12]
summary(normData, 'QC')
boxplot(normData, ylab = "normalized log2 expression", main = "non-normalized data -
boxplots", col=cols.ALL)
```



Filter data to remove: 1) genes that were not detected above background, 2) genes that had low variation across all samples, 3) genes without a EntrezID

```
filtered_geneList <- nsFilter(normData, require.entrez=TRUE, remove.dupEntrez=TRUE, v
ar.func=IQR, var.filter=TRUE, var.cutoff=0.5, filterByQuantile=TRUE)
# extract the ExpressionSet from this filtered list
filtered.eset <- filtered_geneList$eset
#now convert to a datamatrix that will contain only the probes after filtering
filtered.matrix <- as.matrix(filtered.eset)
probeList <- rownames(filtered.matrix)
```

Exploratory Analysis

FIGURE 4: Dendrogram of hierarchical sample clustering based on similarity/disimilarity

```
distance <- dist(t(filtered.matrix),method="maximum")
clusters <- hclust(distance, method = "average")
plot(clusters, label = sampleLabels, hang = -1)
```

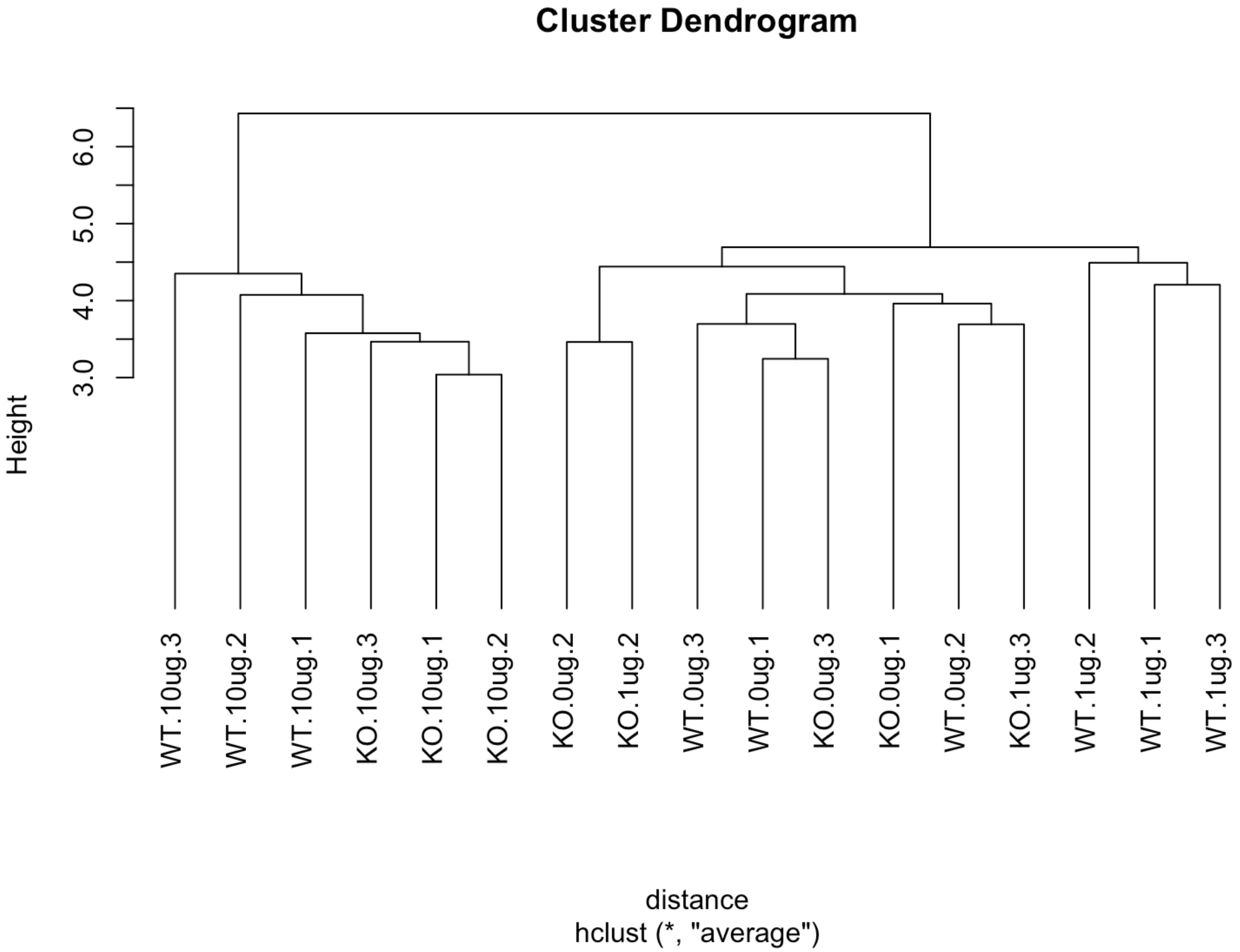
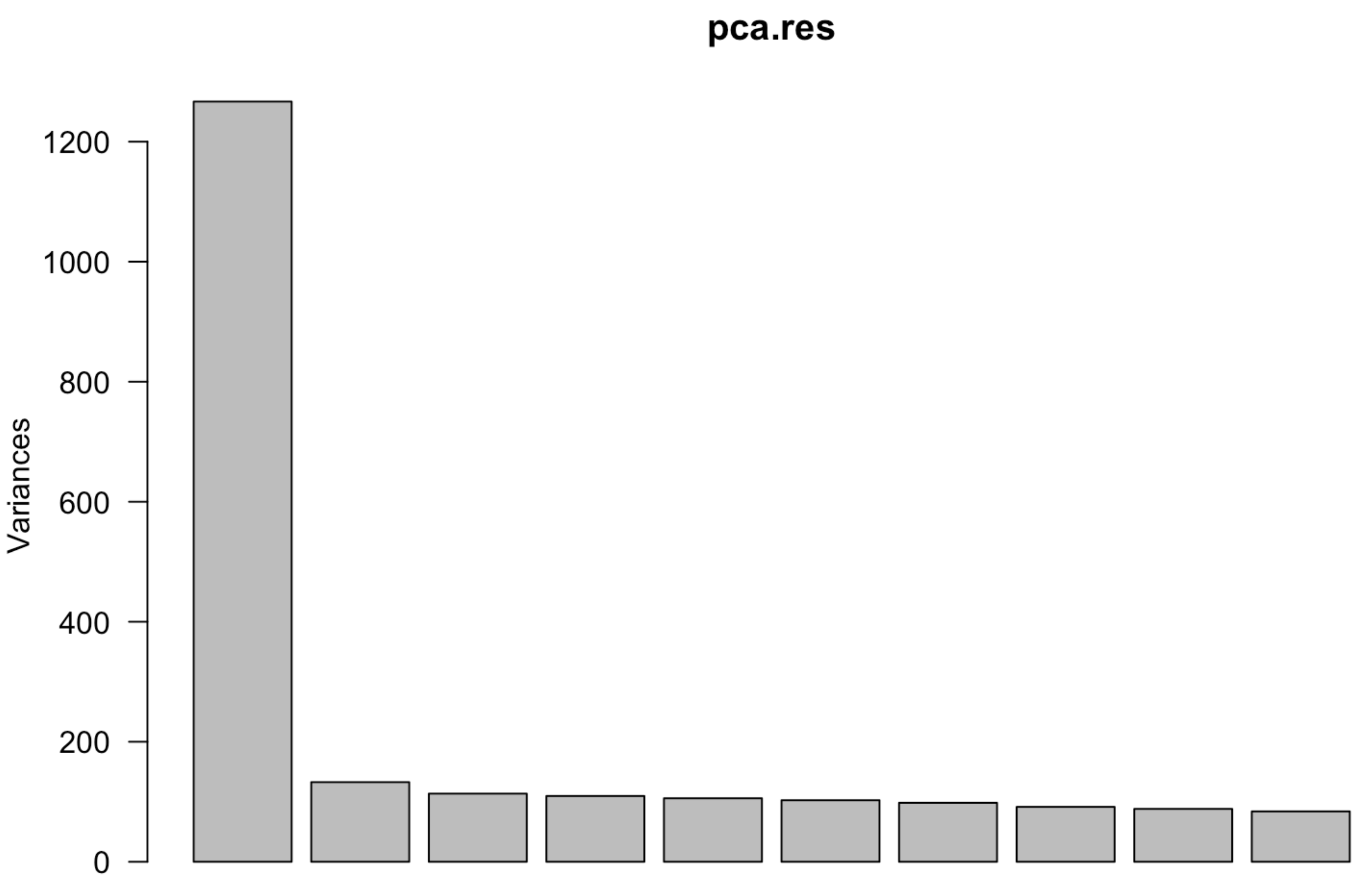


FIGURE 5: Principle Component Analysis

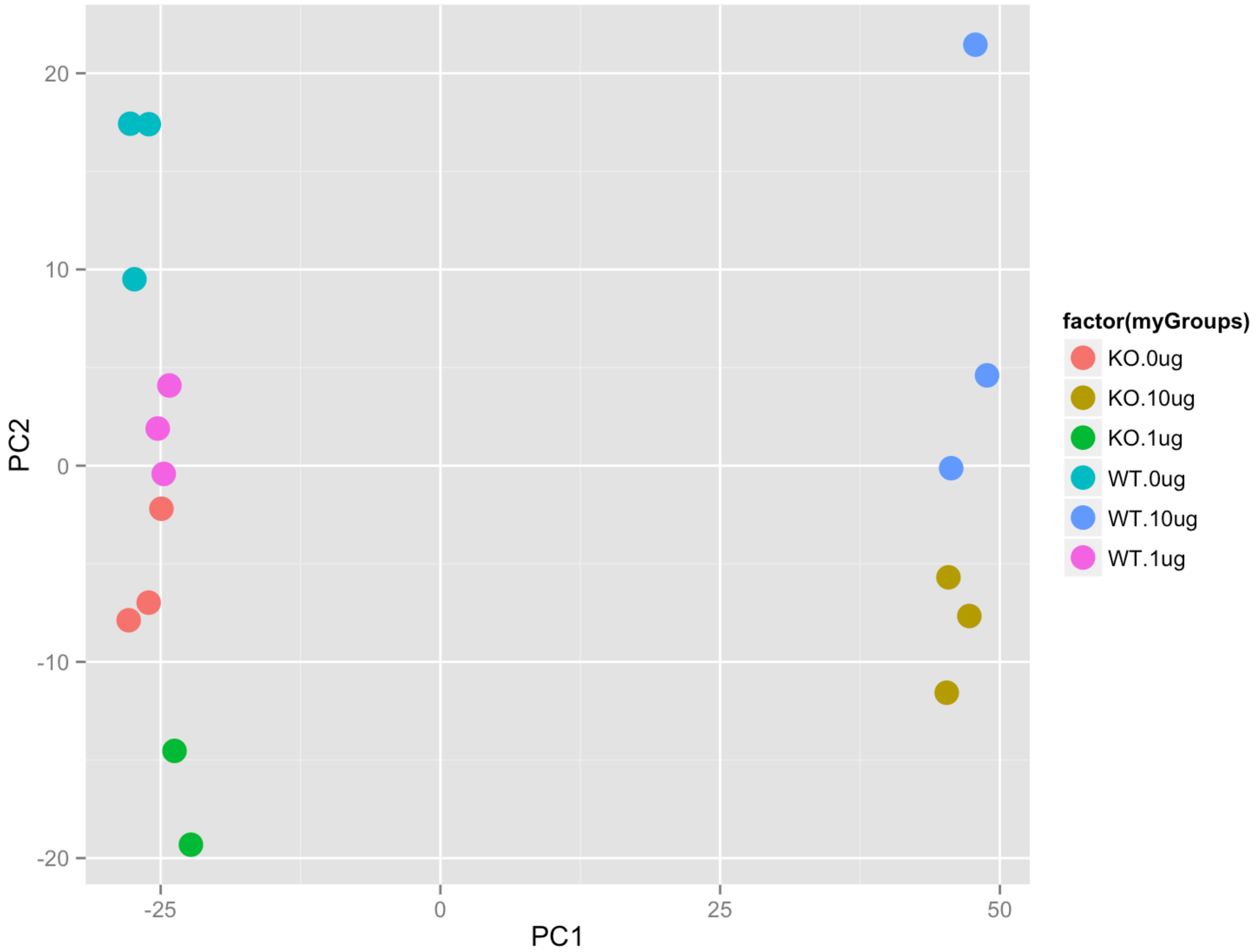
```
pca.res <- prcomp(t(filtered.matrix), scale.=F, retx=T)
ls(pca.res)
summary(pca.res) # Prints variance summary for all principal components.
head(pca.res$rotation) #$rotation shows you how much each GENE influenced each PC (ca
llled 'eigenvalues', or loadings)
head(pca.res$x) #$x shows you how much each SAMPLE influenced each PC (called 'scores
')
plot(pca.res, las=1)
```



```
pc.var<-pca.res$sdev^2 #sdev^2 gives you the eigenvalues
pc.per<-round(pc.var/sum(pc.var)*100, 1)
pc.per
```

Figure 6: Sample contribution to the first two principle components

```
data.frame <- as.data.frame(pca.res$x)
dim(data.frame)
ggplot(data.frame, aes(x=PC1, y=PC2, colour=factor(myGroups))) +
  geom_point(size=5) +
  theme(legend.position="right")
```



Targeted Analysis

GSEA highlighted histones as potential T-bet targets a priori heatmaps with list of selected genes (all the histones)

Figure 7: Most histones are upregulated in WT cells after activation

```
library(gplots)
myheatcol <- greenred(75)
histonesWT <- as.matrix(read.delim("histones_WT.txt", header=TRUE, row.names=1))
heatmap.2(histonesWT, Rowv=F, Colv=F, col=myheatcol, scale="row", density.info="none"
, trace="none", cexRow=0.75, cexCol=1, margins=c(7,20))
```

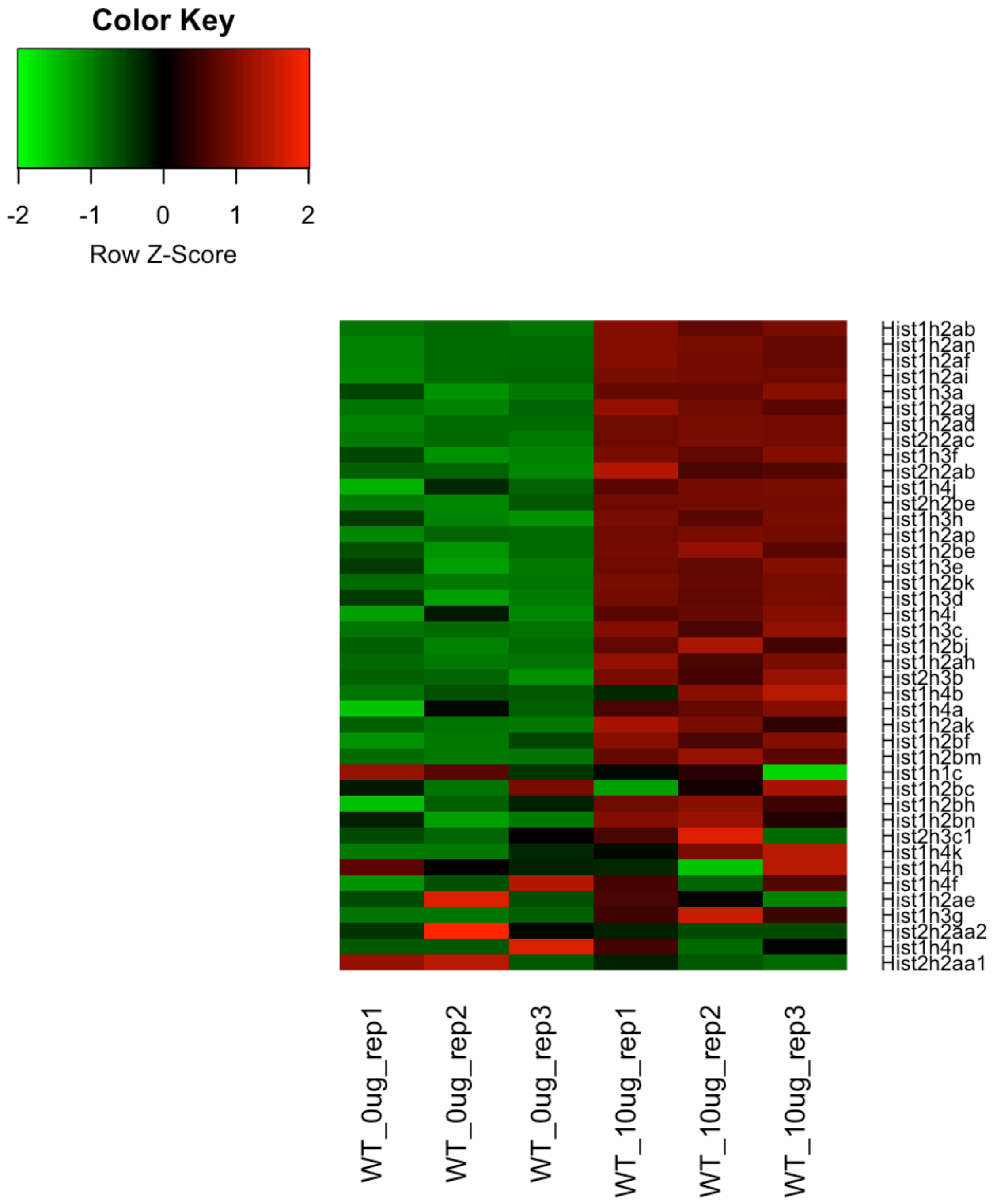
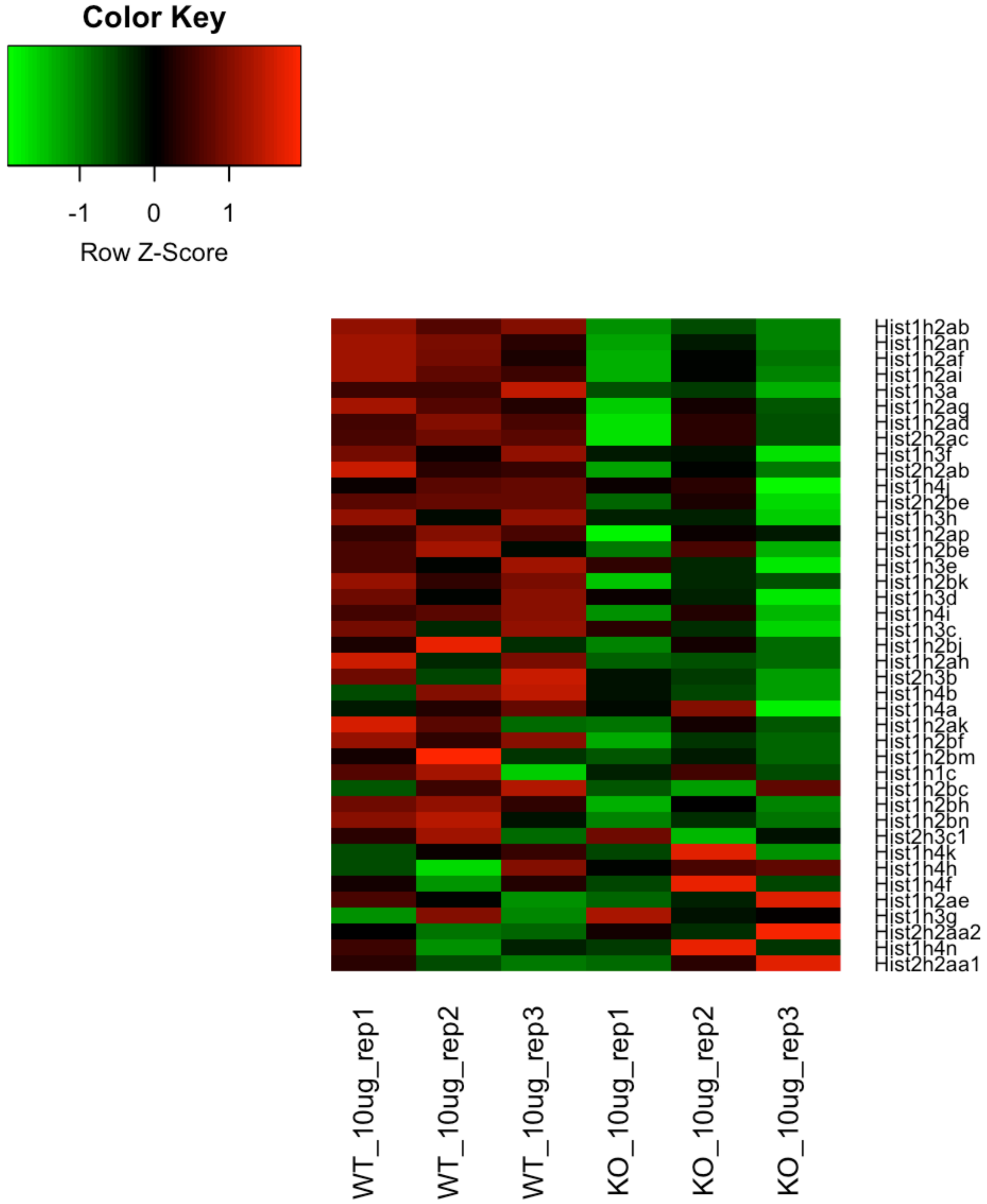


Figure 8: Histone upregulation is T-bet dependent

```
library(gplots)
myheatcol <- greenred(75)
histones10ug <- as.matrix(read.delim("10ughistones.txt", header=TRUE, row.names=1))
heatmap.2(histones10ug, Rowv=F, Colv=F, col=myheatcol, scale="row", density.info="non
e", trace="none", cexRow=0.75, cexCol=1, margins=c(7,20))
```



All analyses were carried out on the following system:

```
sessionInfo()
```

R version 3.1.3 (2015-03-09) Platform: x86_64-apple-darwin13.4.0 (64-bit) Running under: OS X 10.10.4 (Yosemite)

locale: [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages: [1] stats4 parallel stats graphics grDevices utils datasets [8] methods base

other attached packages: [1] devtools_1.8.0 knitr_1.10.5
[3] rmarkdown_0.7 knitcitations_1.0.6
[5] reshape2_1.4.1 annotate_1.44.0
[7] XML_3.98-1.1 limma_3.22.7
[9] genefilter_1.48.1 ggplot2_1.0.1
[11] gplots_2.17.0 RColorBrewer_1.1-2
[13] lumiMouseAll.db_1.22.0 org.Mm.eg.db_3.0.0
[15] lumiMouseIDMapping_1.10.0 RSQLite_1.0.0
[17] DBI_0.3.1 AnnotationDbi_1.28.2

```
[19] GenomeInfoDb_1.2.5 IRanges_2.0.1
[21] S4Vectors_0.4.0 lumi_2.18.0
[23] Biobase_2.26.0 BiocGenerics_0.12.1

loaded via a namespace (and not attached): [1] affy_1.44.0 affyio_1.34.0
[3] base64_1.1 base64enc_0.1-2
[5] BatchJobs_1.6 BBmisc_1.9
[7] beanplot_1.2 bibtex_0.4.0
[9] BiocInstaller_1.16.4 BiocParallel_1.0.3
[11] biomaRt_2.22.0 Biostrings_2.34.1
[13] bitops_1.0-6 brew_1.0-6
[15] bumphunter_1.6.0 caTools_1.17.1
[17] checkmate_1.5.2 codetools_0.2-11
[19] colorspace_1.2-6 curl_0.9.1
[21] digest_0.6.8 doRNG_1.6
[23] evaluate_0.7 fail_1.2
[25] foreach_1.4.2 formatR_1.2
[27] gdata_2.16.1 GenomicAlignments_1.2.2 [29] GenomicFeatures_1.18.7 GenomicRanges_1.18.4
[31] git2r_0.10.1 grid_3.1.3
[33] gtable_0.1.2 gtools_3.4.2
[35] htmltools_0.2.6 httr_1.0.0
[37] illuminaio_0.8.0 iterators_1.0.7
[39] KernSmooth_2.23-14 labeling_0.3
[41] lattice_0.20-31 locfit_1.5-9.1
[43] lubridate_1.3.3 MASS_7.3-40
[45] Matrix_1.2-0 matrixStats_0.14.0
[47] mclust_5.0.0 memoise_0.2.1
[49] methylumi_2.12.0 mgcv_1.8-6
[51] minfi_1.12.0 multtest_2.22.0
[53] munsell_0.4.2 nleqslv_2.7
[55] nlme_3.1-120 nor1mix_1.2-0
[57] pkgmaker_0.22 plyr_1.8.2
[59] preprocessCore_1.28.0 proto_0.3-10
[61] quadprog_1.5-5 R6_2.0.1
[63] Rcpp_0.11.5 RCurl_1.95-4.5
[65] RefManageR_0.8.63 registry_0.2
[67] reshape_0.8.5 RJSONIO_1.3-0
[69] rngtools_1.2.4 Rsamtools_1.18.3
[71] rtracklayer_1.26.3 rversions_1.0.2
[73] scales_0.2.4 sendmailR_1.2-1
[75] siggenes_1.40.0 splines_3.1.3
[77] stringr_0.6.2 survival_2.38-1
[79] tools_3.1.3 xml2_0.1.1
[81] xtable_1.7-4 XVector_0.6.0
[83] yaml_2.1.13 zlibbioc_1.12.0
```