Calcium control of NF-kB target gene transcription in T lymphocytes

Corbett Berry

August 10, 2015

Contents

Abstract

T cell fates initiated by engagement of the TCR and costimulatory molecules are encoded, in part, by intracellular calcium signals. However, how calcium encodes differences in antigen avidity and how these signals are decoded into unique transcriptional programs controlling distinct T cell fates and functions is unknown. NFAT and NF-kB are two calcium-regulated transcription factors that control T cell activation and differentiation. NFAT is activated by calcium/calcineurin-mediated tyrosine dephosphorylation and this mechanism has long-served as the paradigm for calcium regulated transcriptional control in lymphocyte. In contrast, we lack a comprehensive understanding of the mechanisms and the consequences of calcium's control of NF-B signaling. In order to investigate how calcium shapes the transcriptional response of T lymphocytes following TCR engagement, we have analyzed the transcriptome of T lymphocytes after initiating NF-kB signaling with a PKC activator (PMA) in the presence and absence of calcium release and calcium entry (via Ionomycin). Here, I provide an overview of the methods used to analyze our microarray data and present a number of interesting findings that provide evidence for a key role of calcium signaling in NF-kB target gene transcription in T lymphocytes.

R Packages

I will start by loading the packages we will use to analyze the microarray data.

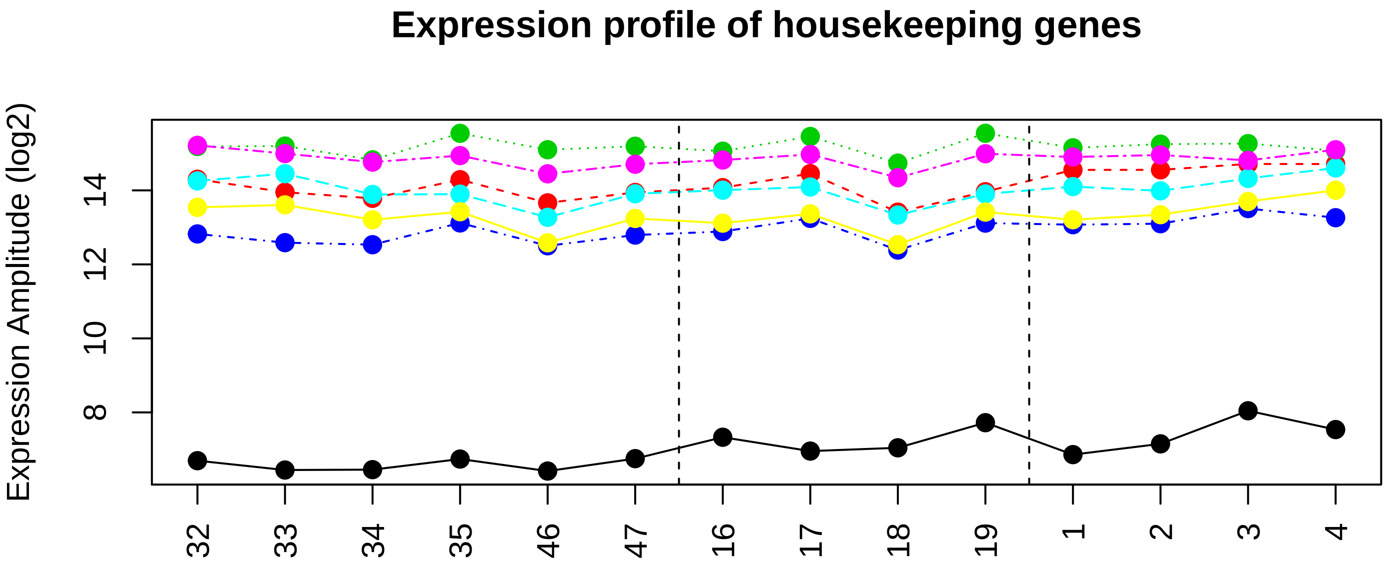
```r
library(rmarkdown)
library(knitr)
library(lumi)
library(lumiHumanIDMapping)
library(lumiHumanAll.db)
library(genefilter)
library(annotate)
library(ggplot2)
library(gplots)
library(reshape2)
library(dplyr)
library(ggvis)
library(limma)
library(Biobase)
library(RColorBrewer)
```

Loading the Data, Normalizing, and Performing Quality Control

Now that we have loaded the necessary R packages, I will load in the sample and control data. We can assess the quality of the data by taking a look at the house keeping genes and some boxplots of the raw data. Everything looks good so we will proceed with normalization using a robust spline normalization (RSN).

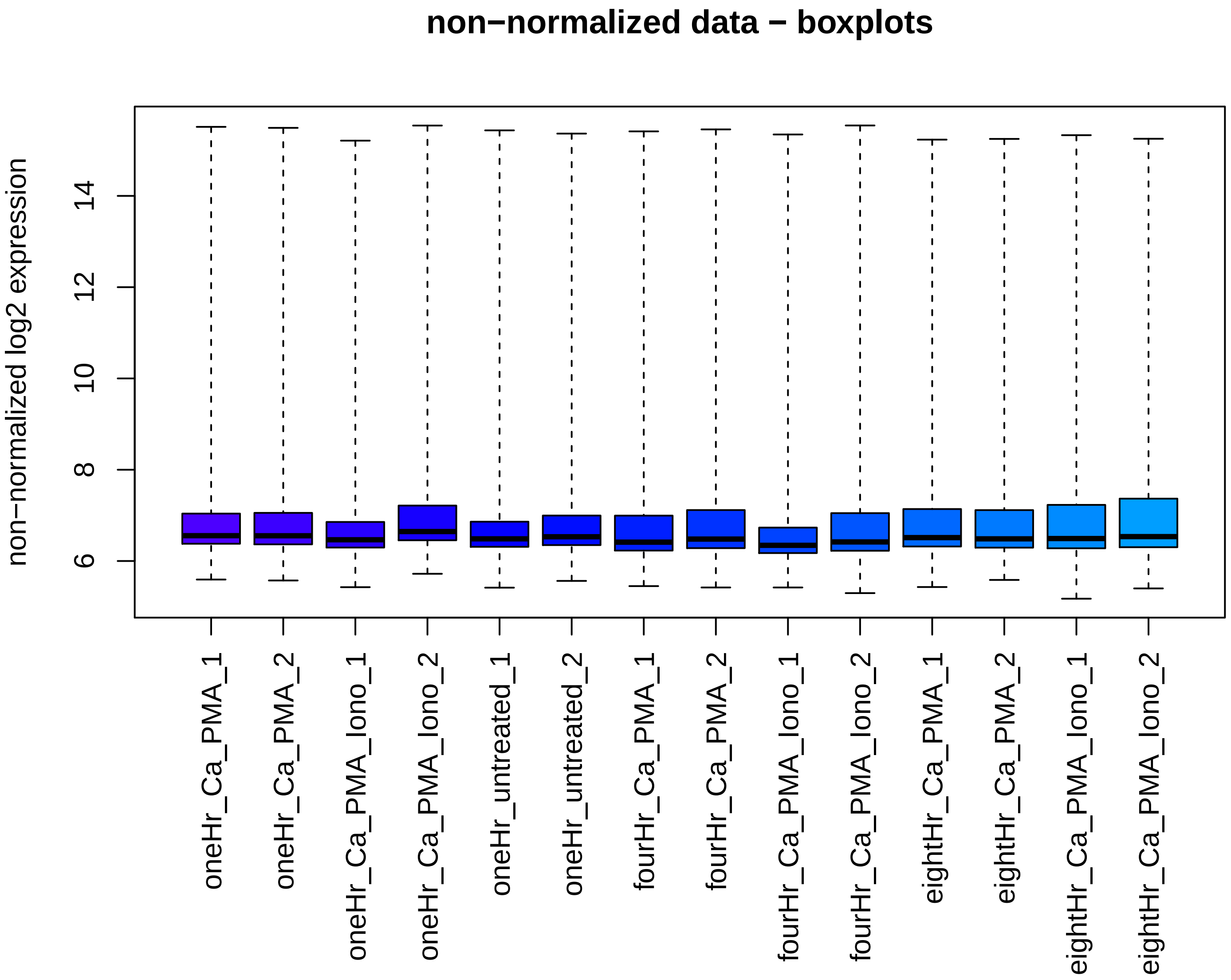
```r
rawData <- lumiR("FinalReport_probes_samples.txt", convertNuID = TRUE,
                 sep = NULL, detectionTh = 0.01, na.rm = TRUE, lib = "lumiHumanIDMapping")
rawData <- rawData[,-46]
rawData <- addControlData2lumi("FinalReport_probes_controls.txt", rawData)
controlData <- getControlData(rawData)

targets <- read.delim("CalciumJurkat_studyDesign_2.txt", sep="\t")
myGroups <- factor(targets$groupName)
design <- model.matrix(~0+myGroups)
colnames(design) <- levels(myGroups)
sampleLabels <- targets$sampleName
sampleLabels <- sampleLabels[c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]

rawData2 <- rawData[,c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]
colnames(rawData2) <- sampleLabels
```

```r
plotHousekeepingGene(rawData2, addLegend = FALSE)
```

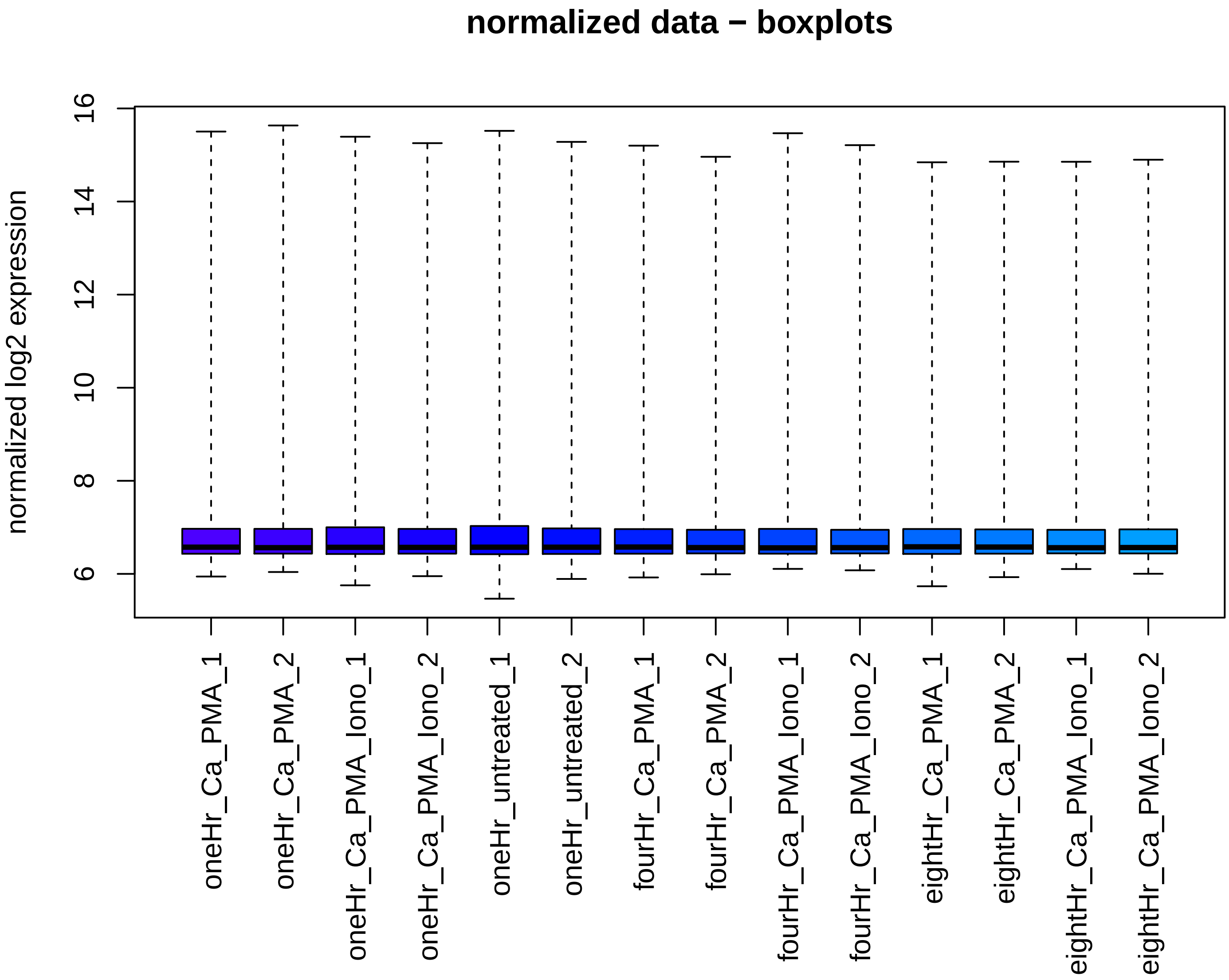


```r
cols.ALL <- topo.colors(n=50, alpha=1)
```

```r
boxplot(rawData2, ylab = "non-normalized log2 expression",
        main = "non-normalized data - boxplots", col=cols.ALL)
```



```
normData <- lumiExpresso(rawData, QC.evaluation=TRUE,
                         normalize.param=list(method='rsn'))
normData2 <-normData[,c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]
colnames(normData2) <- sampleLabels
boxplot(normData2, ylab = "normalized log2 expression",
        main = "normalized data - boxplots", col=cols.ALL)
```



Filtering and Annotating

Now that we have normalized the data, let's go ahead and filter as well as annotate the data. I will remove the probes with low variance across the different time and stimulatory conditions. I will also remove duplicate genes and remove coding sequences with no entrezIDs. For annotation, I will add labels to the sample groups and genes (including gene symbol and entrezID). This includes, in duplicate, T cells stimulated with PMA and PMA/Ionomycin for 1 hour, 4 hours, and 8 hours along with a 1 hour unstimulated control group.

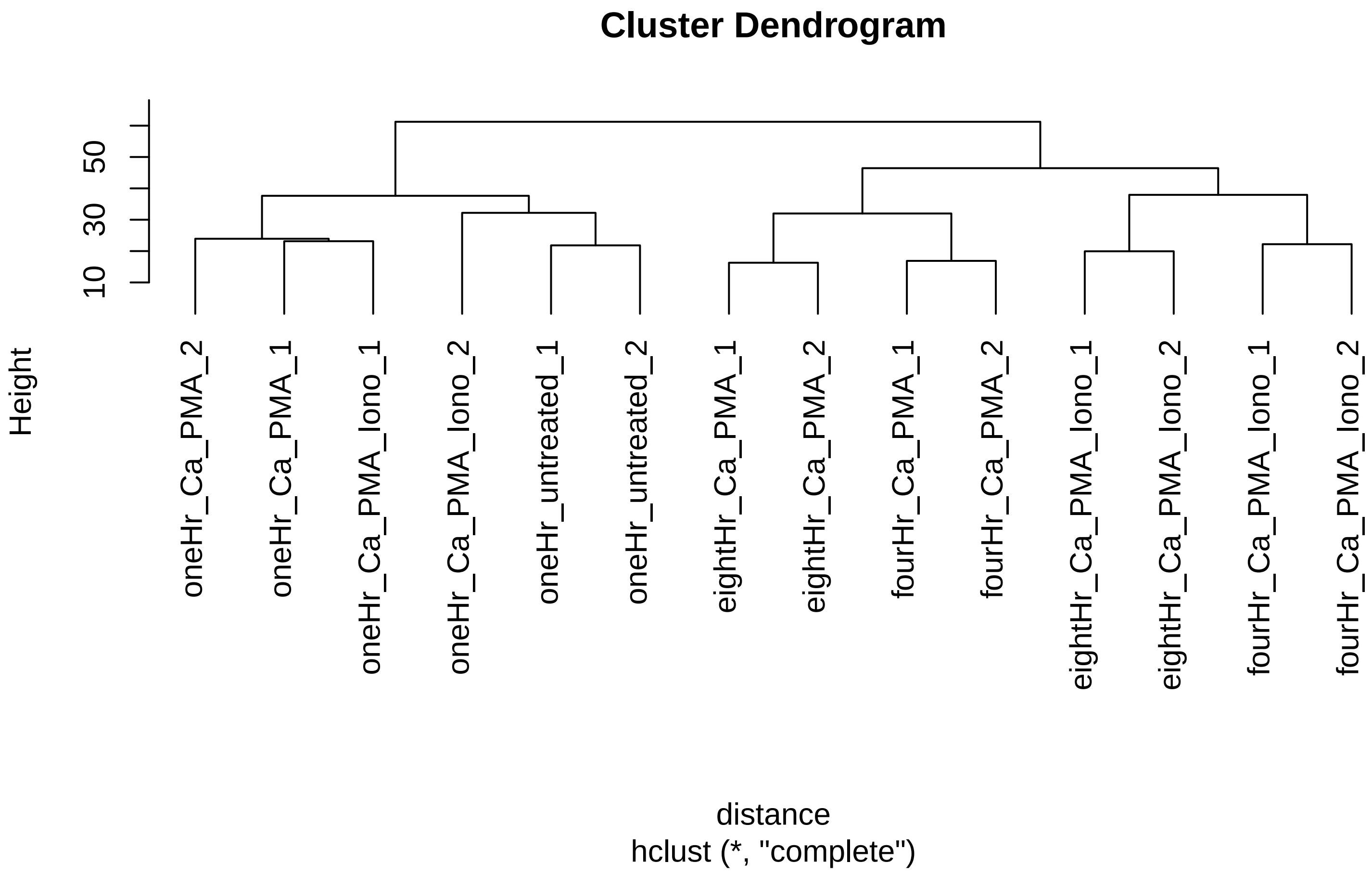
```
filtered_geneList <- nsFilter(normData, require.entrez=TRUE,
                              remove.dupEntrez=TRUE, var.func=IQR, var.filter=TRUE,
                              var.cutoff=0.5, filterByQuantile=TRUE)
filtered.eset <- filtered_geneList$eset
filtered.matrix <- as.matrix(filtered.eset)
probeList <- rownames(filtered.matrix)
myGenesAll <- getSYMBOL(probeList, "lumiHumanAll.db")
myGenesAll <- as.matrix(myGenesAll)
myEntrezAll <- getEG(probeList, "lumiHumanAll.db")
myEntrezAll <- as.matrix(myEntrezAll)
write.table(cbind(myGenesAll, myEntrezAll,
                  filtered.matrix[,c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]),
                  "normalizedFilteredData.xls", sep="\t", quote=FALSE)
write.exprs(normData, file='robustSplineNorm.txt')
```

```
normData.delim <- read.delim("robustSplineNorm.txt", header=TRUE, row.names=1)
normData.matrix <- as.matrix(normData.delim)
probeList2 <- rownames(normData.matrix)
symbols <- getSYMBOL(probeList2, "lumiHumanAll.db")
entrezIDs <- getEG(probeList2, "lumiHumanAll.db")
write.table(cbind(symbols,normData.matrix), "unfiltered_expressionData.xls",
                  sep="\t", quote=FALSE)
```

Hierarchical Clustering

Let's look at how our samples and duplicates look with some hierarchical clustering. These dendograms provide good evidence for consistent replicates. They also suggest that there is a significant difference in gene expression between cells stimulated with PMA vs PMA and Ionomycin.

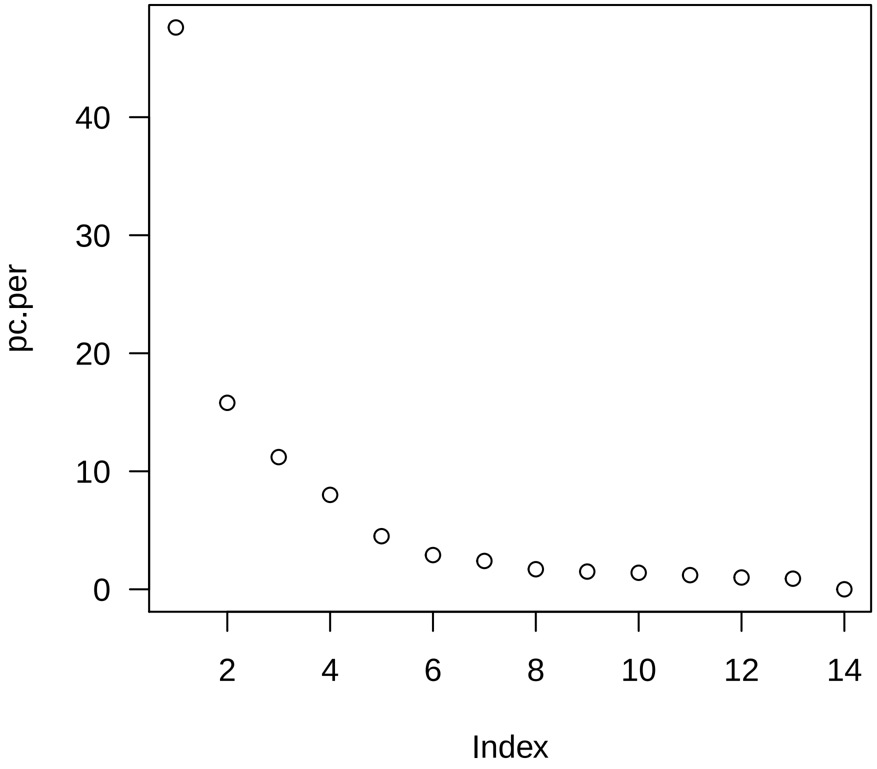
```
distance <- dist(t(filtered.matrix[,c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]),
                 method="euclidian")
clusters <- hclust(distance, method = "complete")
plot(clusters, label = sampleLabels, hang = -1)
```



Principal Component Analysis

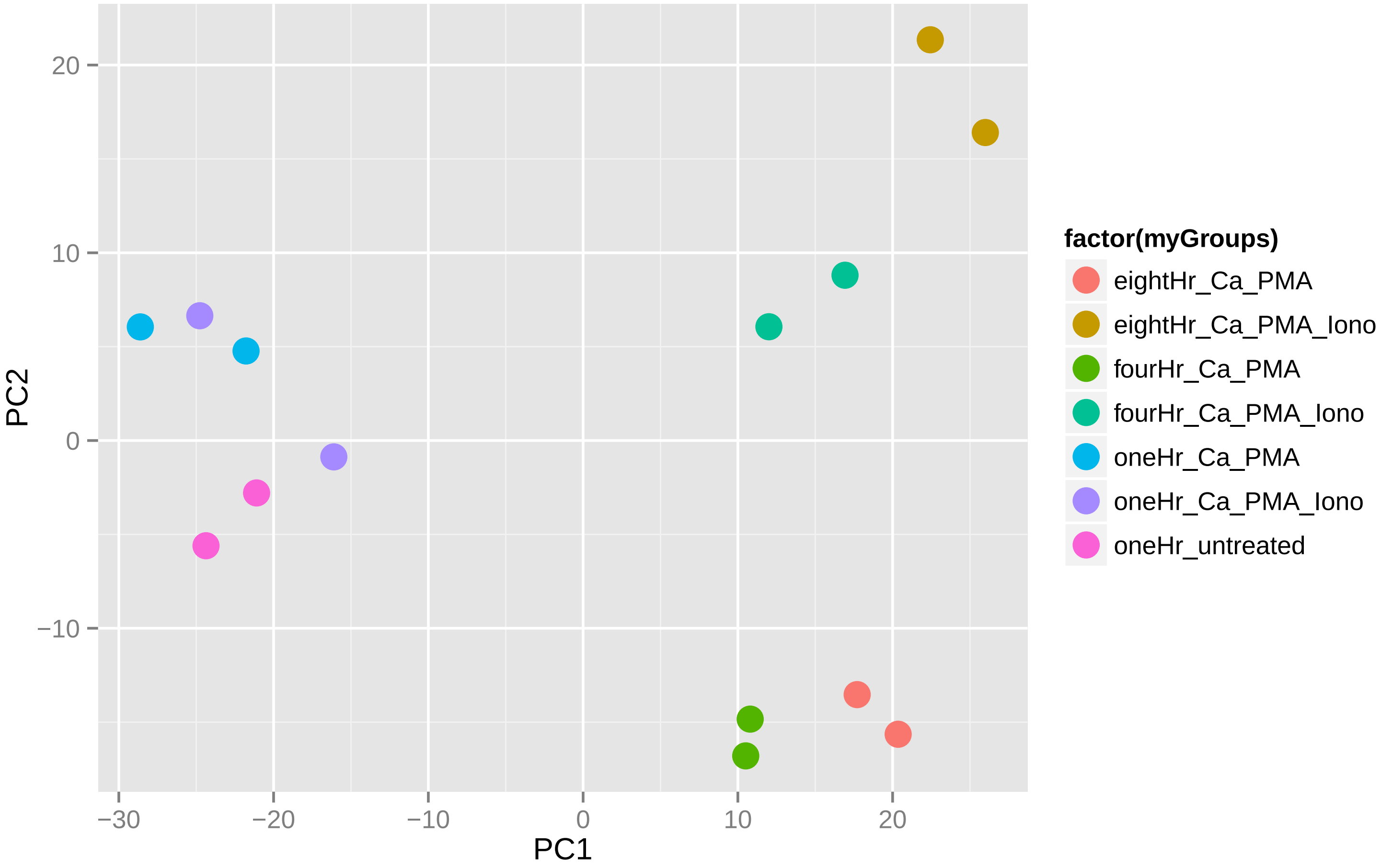
We can use principal component analysis (PCA) to explore the data and see if we can see any patterns between the groups. The first graph below shows a plot of how much each principal component accounts for the variablity in the dataset. For example, the first principal component in this analysis explains nearly 50% of the data's variability. Furthermore, the combination of the first three principal components explain nearly 75% of the dataset's variability.

```
pca.res <- prcomp(t(filtered.matrix[,c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]),
                  scale.=F, retx=T)
ls(pca.res)
summary(pca.res)
pc.var<-pca.res$sdev^2
pc.per<-round(pc.var/sum(pc.var)*100, 1)
plot(pc.per, las=1)
```

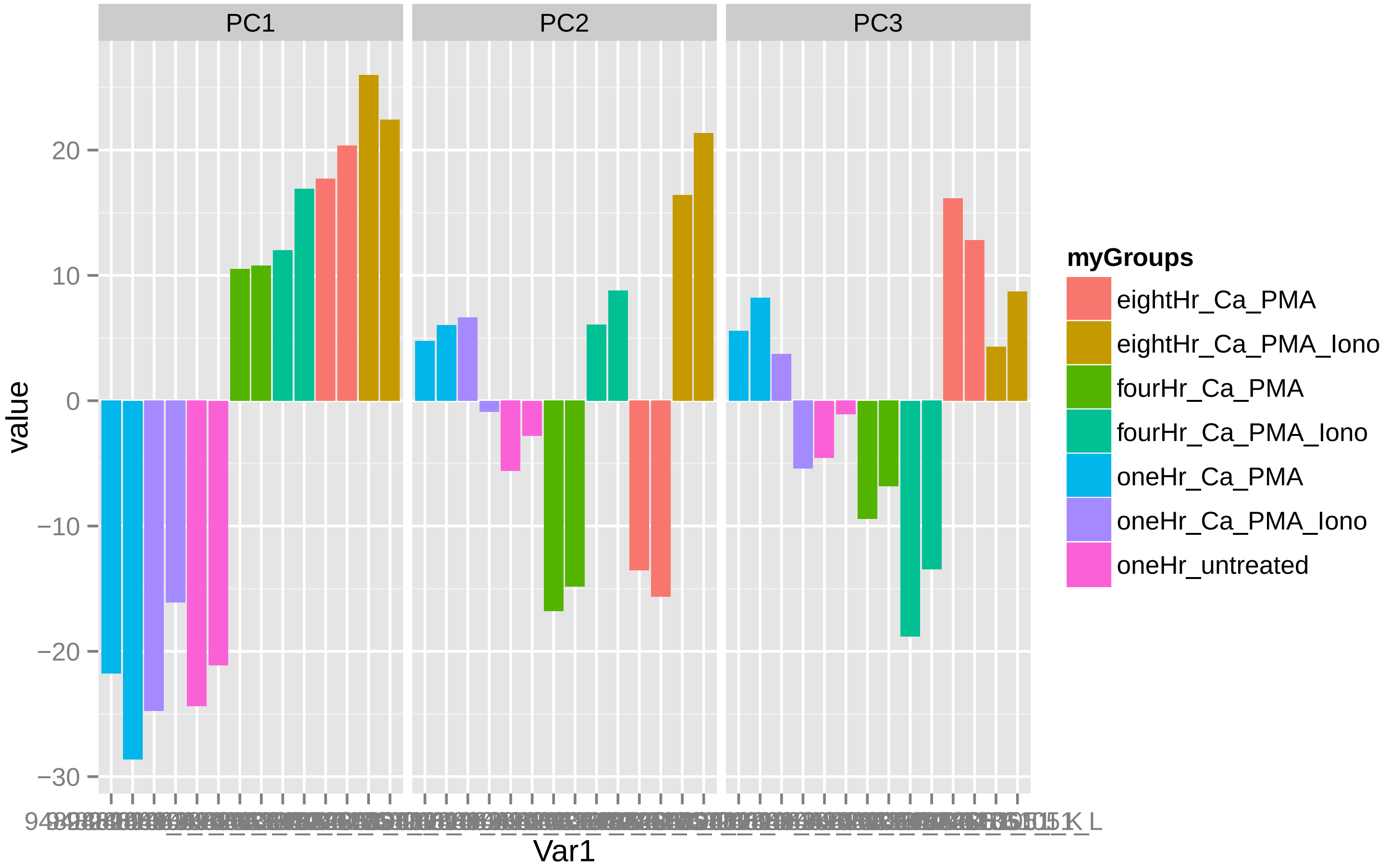


We can plot the top two principal components (PC1 and PC2) against each other and get a good visualization of how the groups separate. PC1 appears to separate the early and late responses. Thus, PC1 can be thought of as time. PC2 appears to separate the treatment groups PMA vs PMA and Ionomycin. Thus, PC2 can be thought of as drug treatment or even calcium signaling since both groups are stimulated with PMA. Not surprisingly, it appears that calcium signaling greatly influences the transcriptome of T lymphocytes. We will have to dig deeper to assess how calcium signaling influences NF-kB target gene transcription.

```
myGroups <- myGroups[c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]
data.frame <- as.data.frame(pca.res$x)
ggplot(data.frame, aes(x=PC1, y=PC2, colour=factor(myGroups))) +
  geom_point(size=5) +
  theme(legend.position="right")
```



```
melted <- cbind(myGroups, melt(pca.res$x[,1:3]))
ggplot(melted) +
  geom_bar(aes(x=Var1, y=value, fill=myGroups), stat="identity") +
  facet_wrap(~Var2)
```



Let's reload the normalized data set and prep everything for the next stage in analysis…

```
myData <- read.delim("normalizedFilteredData.xls", header=TRUE)
myData <- myData[,-2]
row.names(myData) <- myData[,1]
myData <- myData[,-1]
sampleLabels <- as.character(targets$sampleName)
sampleLabels <- sampleLabels[c(1,2,3,4,7,8, 17, 18, 19, 20, 33, 34, 35, 36)]
colnames(myData) <- sampleLabels
geneSymbols <- row.names(myData)
row.names(myData) <- geneSymbols
```

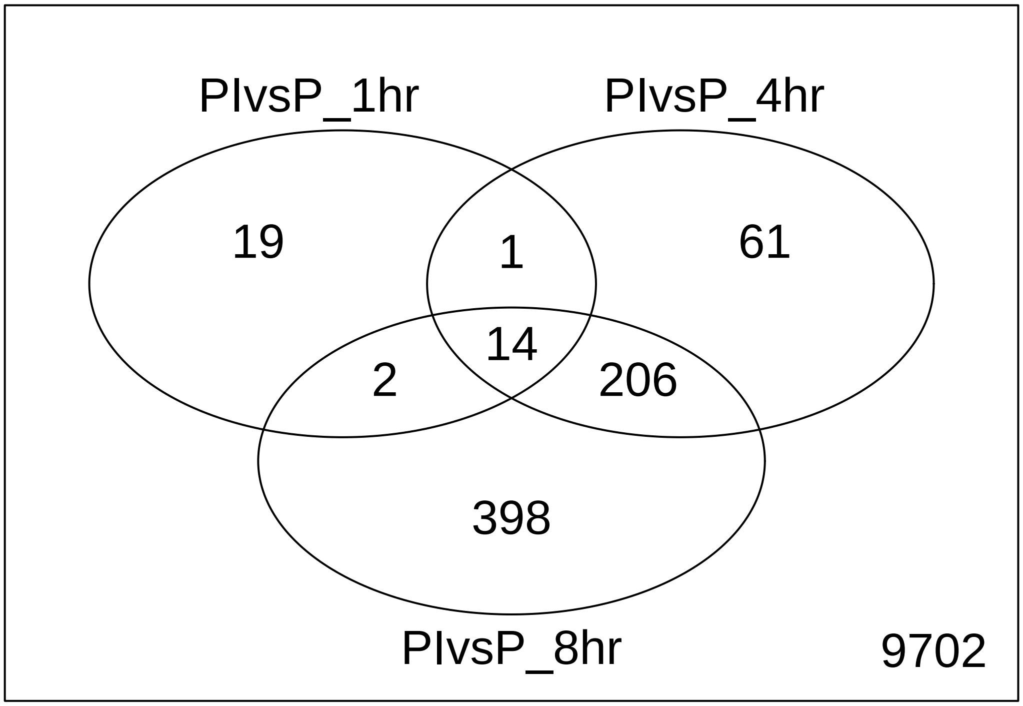
Differential Gene Expression

We have generated some pretty intriguing results so far. Now, let's take a closer look at those genes that significantly change when you compare cells stimulated with PMA with and without Ionomycin. We will use an R package called limma (linear models for microarray data) to determine which genes are differentially expressed. I will compare drug treatment with PMA vs PMA and Ionomycin separately at 1 hour, 4 hours, and 8 hours and use bayesian statistics to generate p-values for assessing statistical significance.

```
fit <- lmFit(filtered.matrix, design)
fit$genes$Symbol <- getSYMBOL(probeList, "lumiHumanAll.db")
fit$genes$Entrez <- getEG(probeList, "lumiHumanAll.db")

contrast.matrix.PIvsP_8hr.2 <- makeContrasts(eightPMAIonovsPMA = eightHr_Ca_PMA_Iono-eightHr_Ca_PMA,
                                             levels=design)
contrast.matrix.PIvsP_4hr.2 <- makeContrasts( fourPMAIonovsPMA = fourHr_Ca_PMA_Iono-fourHr_Ca_PMA,
                                              levels=design)
contrast.matrix.PIvsP_1hr.2 <- makeContrasts(onePMAIonovsPMA =oneHr_Ca_PMA_Iono-oneHr_Ca_PMA,
                                             levels=design)

fits_PIvsP_8 <- contrasts.fit(fit, contrast.matrix.PIvsP_8hr.2)
fits_PIvsP_4 <- contrasts.fit(fit, contrast.matrix.PIvsP_4hr.2)
fits_PIvsP_1 <- contrasts.fit(fit, contrast.matrix.PIvsP_1hr.2)

ebFit_PIvsP_8 <- eBayes(fits_PIvsP_8)
ebFit_PIvsP_4 <- eBayes(fits_PIvsP_4)
ebFit_PIvsP_1 <- eBayes(fits_PIvsP_1)

probeList_PIvsP_8 <- topTable(ebFit_PIvsP_8, adjust ="BH", coef=1, number=100, sort.by="logFC")
probeList_PIvsP_4 <- topTable(ebFit_PIvsP_4, adjust ="BH", coef=1, number=100, sort.by="logFC")
probeList_PIvsP_1 <- topTable(ebFit_PIvsP_1, adjust ="BH", coef=1, number=100, sort.by="logFC")

results_PIvsP_1 <- decideTests(ebFit_PIvsP_1, method="global", adjust.method="BH",
                               p.value=0.01, lfc=0.59)
results_PIvsP_4 <- decideTests(ebFit_PIvsP_4, method="global", adjust.method="BH",
                               p.value=0.01, lfc=0.59)
results_PIvsP_8 <- decideTests(ebFit_PIvsP_8, method="global", adjust.method="BH",
                               p.value=0.01, lfc=0.59)
```

Now we have a list of differentially expressed genes. Let's take a look at the overall number and relationship between each time point using a venn Diagram. Then we will move forward with annotation and some hierarchical clustering.

```
colnames(results_PIvsP_1)<- "PIvsP_1hr"
colnames(results_PIvsP_4)<- "PIvsP_4hr"
colnames(results_PIvsP_8)<- "PIvsP_8hr"
vennDiagram(cbind(results_PIvsP_1,results_PIvsP_4,results_PIvsP_8))
```



```
diffProbes_PIvsP_1 <- which(results_PIvsP_1[,1] !=0)
diffProbes_PIvsP_4 <- which(results_PIvsP_4[,1] !=0)
diffProbes_PIvsP_8 <- which(results_PIvsP_8[,1] !=0)

diffSymbols_PIvsP_1 <- fit$genes$Symbol[results_PIvsP_1[,1] !=0]
diffSymbols_PIvsP_4 <- fit$genes$Symbol[results_PIvsP_4[,1] !=0]
diffSymbols_PIvsP_8 <- fit$genes$Symbol[results_PIvsP_8[,1] !=0]

diffEntrez_PIvsP_1 <- fit$genes$Entrez[results_PIvsP_1[,1] !=0]
diffEntrez_PIvsP_4 <- fit$genes$Entrez[results_PIvsP_4[,1] !=0]
diffEntrez_PIvsP_8 <- fit$genes$Entrez[results_PIvsP_8[,1] !=0]

myEset.ALL <- new("ExpressionSet", exprs = filtered.matrix[,c(1,2,3,4,7,8, 17, 18,
                                                              19, 20, 33, 34, 35, 36)])
annotation(myEset.ALL) <- "lumiHumanAll.db"

diffData_PIvsP_1 <- filtered.eset[results_PIvsP_1[,1] !=0]
diffData_PIvsP_4 <- filtered.eset[results_PIvsP_4[,1] !=0]
diffData_PIvsP_8 <- filtered.eset[results_PIvsP_8[,1] !=0]

diffData_PIvsP_1 <- exprs(diffData_PIvsP_1)
diffData_PIvsP_4 <- exprs(diffData_PIvsP_4)
diffData_PIvsP_8 <- exprs(diffData_PIvsP_8)

diffData_PIvsP_1 <- diffData_PIvsP_1[,c(1,2,3,4,7,8, 17, 18,
                                        19, 20, 33, 34, 35, 36)]
diffData_PIvsP_4 <- diffData_PIvsP_4[,c(1,2,3,4,7,8, 17, 18,
                                        19, 20, 33, 34, 35, 36)]
diffData_PIvsP_8 <- diffData_PIvsP_8[,c(1,2,3,4,7,8, 17, 18,
                                        19, 20, 33, 34, 35, 36)]

colnames(diffData_PIvsP_1) <- sampleLabels
colnames(diffData_PIvsP_4) <- sampleLabels
colnames(diffData_PIvsP_8) <- sampleLabels

rownames(diffData_PIvsP_1) = diffSymbols_PIvsP_1
rownames(diffData_PIvsP_4) = diffSymbols_PIvsP_4
rownames(diffData_PIvsP_8) = diffSymbols_PIvsP_8

hr_PIvsP_1 <- hclust(as.dist(1-cor(t(diffData_PIvsP_1), method="pearson")), method="complete")
hr_PIvsP_4 <- hclust(as.dist(1-cor(t(diffData_PIvsP_4), method="pearson")), method="complete")
hr_PIvsP_8 <- hclust(as.dist(1-cor(t(diffData_PIvsP_8), method="pearson")), method="complete")

mycl_PIvsP_1 <- cutree(hr_PIvsP_1, k=2)
mycolhc_PIvsP_1 <- rainbow(length(unique(mycl_PIvsP_1)), start=0.1, end=0.9)
mycolhc_PIvsP_1 <- mycolhc_PIvsP_1[as.vector(mycl_PIvsP_1)]

mycl_PIvsP_4 <- cutree(hr_PIvsP_4, k=4)
mycolhc_PIvsP_4 <- rainbow(length(unique(mycl_PIvsP_4)), start=0.1, end=0.9)
mycolhc_PIvsP_4 <- mycolhc_PIvsP_4[as.vector(mycl_PIvsP_4)]

mycl_PIvsP_8 <- cutree(hr_PIvsP_8, k=4)
mycolhc_PIvsP_8 <- rainbow(length(unique(mycl_PIvsP_8)), start=0.1, end=0.9)
```

```
mycolhc_PIvsP_8 <- mycolhc_PIvsP_8[as.vector(mycl_PIvsP_8)]

myheatcol <- greenred(75)
colnames(diffData_PIvsP_1) <- as.character(paste(targets$groupName[c(1,2,3,4,7,8, 17, 18,
                                                                    19, 20, 33, 34, 35, 36)]))
```

Using our replicates, we were able to determine which genes were differentially expressed. Now, we can go ahead and average the replicates and prepare to visualize our genes using heat maps.

```
diffData_PIvsP_1.AVG <- avearrays(diffData_PIvsP_1)
rownames(diffData_PIvsP_1.AVG)<- diffSymbols_PIvsP_1
colnames(diffData_PIvsP_4) <- as.character(paste(targets$groupName[c(1,2,3,4,7,8, 17, 18,
                                                                    19, 20, 33, 34, 35, 36)]))
diffData_PIvsP_4.AVG <- avearrays(diffData_PIvsP_4)
rownames(diffData_PIvsP_4.AVG)<- diffSymbols_PIvsP_4
colnames(diffData_PIvsP_8) <- as.character(paste(targets$groupName[c(1,2,3,4,7,8, 17, 18,
                                                                    19, 20, 33, 34, 35, 36)]))
diffData_PIvsP_8.AVG <- avearrays(diffData_PIvsP_8)
rownames(diffData_PIvsP_8.AVG)<- diffSymbols_PIvsP_8

diffData_avg_PIvsPonly_oneHr <- diffData_PIvsP_1.AVG[,c(3, 1, 2)]
diffData_avg_PIvsPonly_fourHr <- diffData_PIvsP_4.AVG[,c(3, 4, 5)]
diffData_avg_PIvsPonly_eightHr <- diffData_PIvsP_8.AVG[,c(3, 6, 7)]

nam.PIvsP.oneHr.unstimPMAPI <- colnames(diffData_avg_PIvsPonly_oneHr)
nam.PIvsP.fourHr.unstimPMAPI <- colnames(diffData_avg_PIvsPonly_fourHr)
nam.PIvsP.eightHr.unstimPMAPI <- colnames(diffData_avg_PIvsPonly_eightHr)
```

Let's take a peak at the genes that are most different from each other in cells stimulated with PMA vs PMA and Ionomycin at 1 hour. This LogFC value is the log2 fold change in gene expression or the difference in the log2(expression) for gene A (e.g. TNF has a logFC around 2.99 or the difference between log2(TNF from PMA+Iono) and log2(TNF from PMA only) is 2.99. This is equal to a fold change of 2ˆ2.99 = 7.94.

```
probeList_PIvsP_1_list <- topTable(ebFit_PIvsP_1 , adjust ="BH", coef=1, number=20, sort.by="logFC")
row.names(probeList_PIvsP_1_list) <- probeList_PIvsP_1_list[,1]
probeList_PIvsP_1_list <- probeList_PIvsP_1_list[,c(2,3,7)]
knitr::kable(probeList_PIvsP_1_list)
```

	ID.Entrez	logFC	adj.P.Val
TNF	7124	2.9868310	0.0000000
CCL4L1	388372	1.7521939	0.0000000
FOSB	2354	1.7494462	0.0000000
CXCL8	3576	1.5844914	0.0000000
JUN	3725	1.5325477	0.0000000
CD69	969	1.5115143	0.0000010
RNA28S5	100008589	1.4395505	0.3271991
ICOS	29851	1.3644876	0.0000000
NBPF10	100132406	1.2405327	0.0942461
HES1	3280	-1.2198178	0.0000000
FOS	2353	1.1653411	0.0000005
RPS4X	6191	-1.1580746	0.0088396
PTGER4	5734	1.1509975	0.0000576

	ID.Entrez	logFC	adj.P.Val
PLEKHF1	79156	-1.0195967	0.0000059
KLF10	7071	-1.0173275	0.0002212
NFKBIA	4792	1.0005583	0.0000005
NBPF9	400818	0.9889608	0.2012816
FAM177A1	283635	0.9863615	0.2028800
MTHFD1	4522	-0.9858125	0.0248560
SH3BGRL3	83442	-0.9840437	0.0225719

Here are the top differentially expressed genes between PMA + Ionomycin and PMA alone at 4 hours.

```
probeList_PIvsP_4_list <- topTable(ebFit_PIvsP_4 , adjust ="BH", coef=1, number=20, sort.by="logFC")
row.names(probeList_PIvsP_4_list) <- probeList_PIvsP_4_list[,1]
probeList_PIvsP_4_list <- probeList_PIvsP_4_list[,c(2,3,7)]
knitr::kable(probeList_PIvsP_4_list)
```

	ID.Entrez	logFC	adj.P.Val
CCL4L1	388372	5.992669	0
CCL3L3	414062	5.365917	0
CCL3L1	6349	4.151259	0
IL3	3562	4.047089	0
CXCL8	3576	3.813500	0
TNFSF14	8740	3.419750	0
TNF	7124	3.064485	0
GZMB	3002	2.982663	0
XIRP1	165904	2.978672	0
TNS3	64759	2.600022	0
CCL20	6364	2.528188	0
PLAU	5328	2.479378	0
IFNG	3458	2.340236	0
CD70	970	2.332177	0
NFKBIA	4792	2.248448	0
SPRY1	10252	2.247784	0
GEM	2669	2.214692	0
MMP10	4319	2.142373	0
POU2AF1	5450	2.123783	0
PTGER4	5734	2.077269	0

Here are the top differentially expressed genes between PMA + Ionomycin and PMA alone at 8 hours.

```
probeList_PIvsP_8_list <- topTable(ebFit_PIvsP_8 , adjust ="BH", coef=1, number=20, sort.by="logFC")
row.names(probeList_PIvsP_8_list) <- probeList_PIvsP_8_list[,1]
probeList_PIvsP_8_list <- probeList_PIvsP_8_list[,c(2,3,7)]
knitr::kable(probeList_PIvsP_4_list)
```

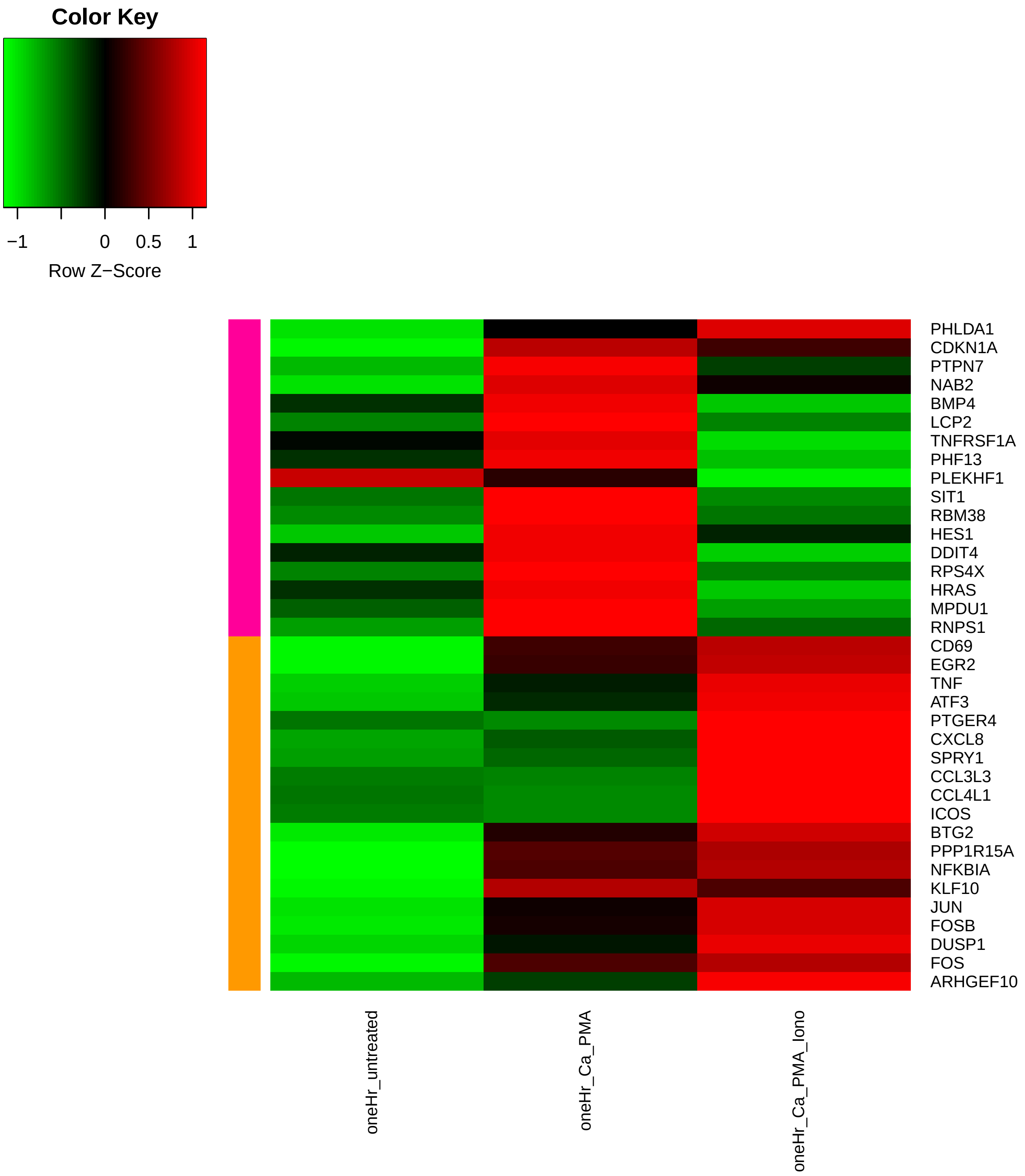
	ID.Entrez	logFC	adj.P.Val
CCL4L1	388372	5.992669	0
CCL3L3	414062	5.365917	0
CCL3L1	6349	4.151259	0
IL3	3562	4.047089	0

	ID.Entrez	logFC	adj.P.Val
CXCL8	3576	3.813500	0
TNFSF14	8740	3.419750	0
TNF	7124	3.064485	0
GZMB	3002	2.982663	0
XIRP1	165904	2.978672	0
TNS3	64759	2.600022	0
CCL20	6364	2.528188	0
PLAU	5328	2.479378	0
IFNG	3458	2.340236	0
CD70	970	2.332177	0
NFKBIA	4792	2.248448	0
SPRY1	10252	2.247784	0
GEM	2669	2.214692	0
MMP10	4319	2.142373	0
POU2AF1	5450	2.123783	0
PTGER4	5734	2.077269	0

We can take a look at all the differentially expressed genes at once using a heat map. The grouping of the genes shown below are determined by the same method clustering method we used here before. The color bars on the left indicate individual clusters within the entire population of differentially expressed genes. We can use this clustering information to further explore groups of genes and their role in particular cell functions using Gene Ontology.

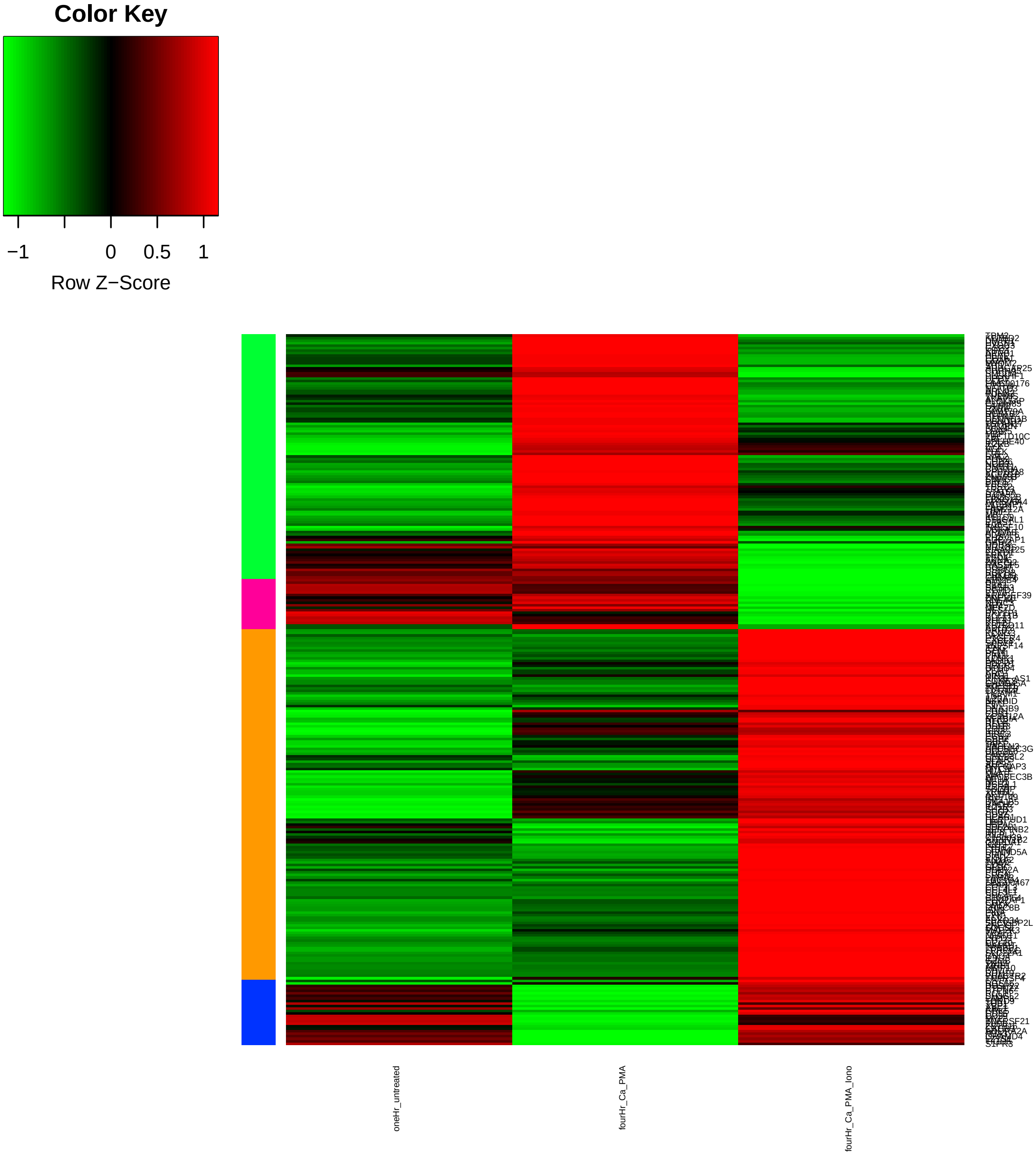
Here's the heat map for gene expression after 1 hour. From left to right, unstimulated, PMA only, and PMA and Ionomycin.

```
heatmap.2(diffData_avg_PIvsPonly_oneHr, Rowv=as.dendrogram(hr_PIvsP_1), Colv=NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none", RowSideColors=mycolhc_PIvsP_1,
          cexRow=1.0, cexCol=1.0, margins=c(10,10))
```



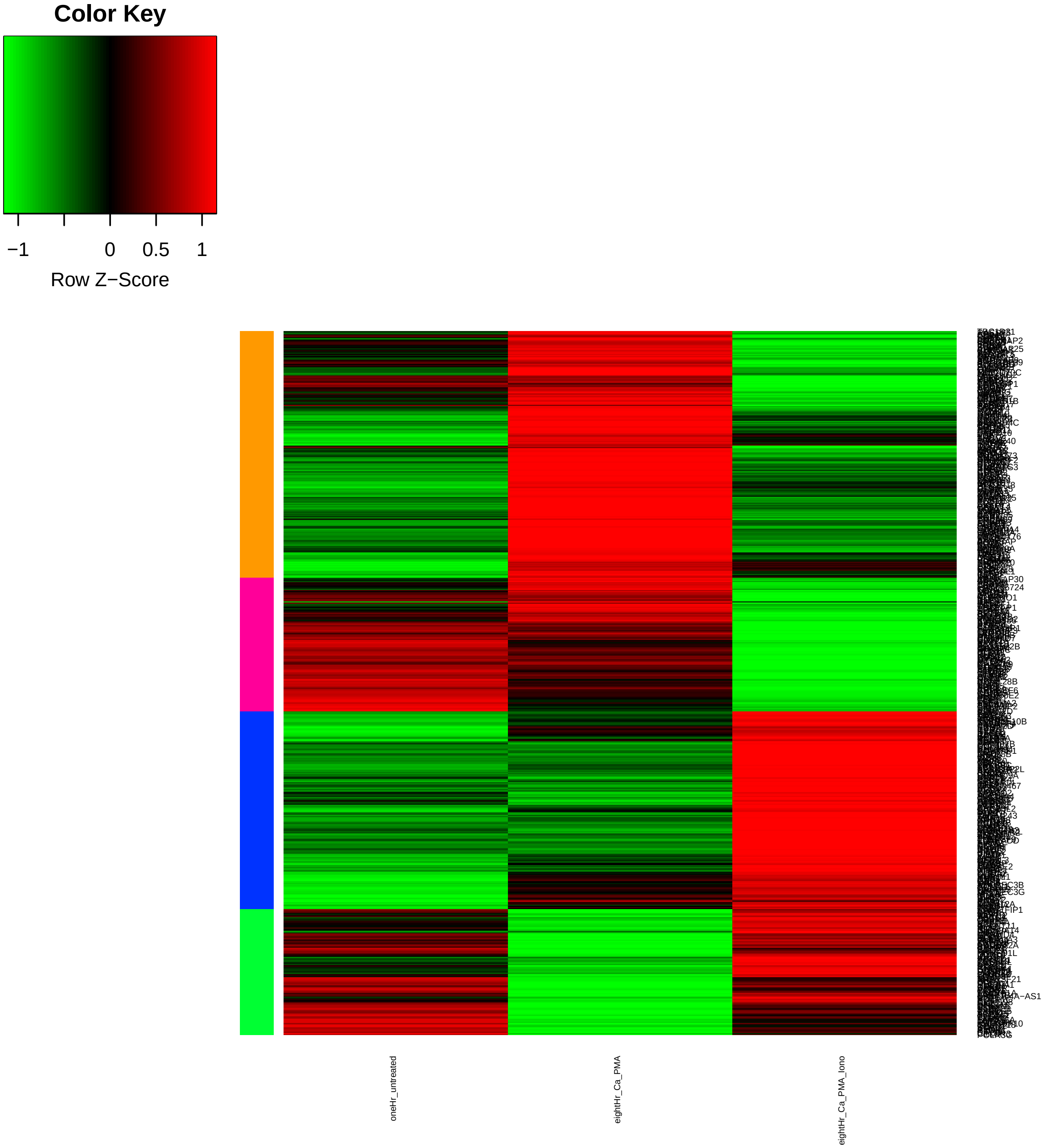
Next, we have the heat map for gene expression at 4 hours. Clearly the number of differentially expressed genes has increased greatly since 1 hour.

```
heatmap.2(diffData_avg_PIvsPonly_fourHr, Rowv=as.dendrogram(hr_PIvsP_4), Colv=NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none", RowSideColors=mycolhc_PIvsP_4,
          cexRow=0.5, cexCol=0.5, margins=c(10,10))
```



Lastly, we have the heat map for gene expression at 8 hours.

```
heatmap.2(diffData_avg_PIvsPonly_eightHr, Rowv=as.dendrogram(hr_PIvsP_8), Colv=NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none", RowSideColors=mycolhc_PIvsP_8,
          cexRow=0.5, cexCol=0.5, margins=c(10,10))
```



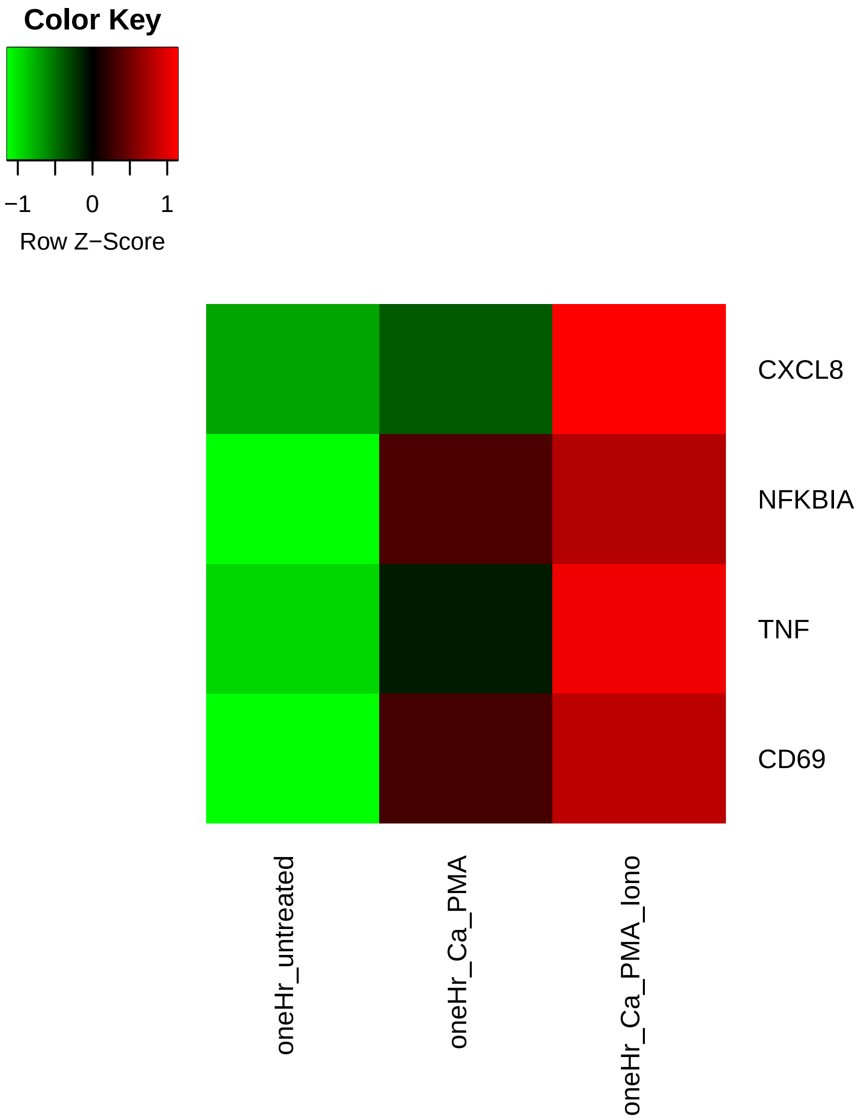
Calcium control of NF-kB target gene transcription

Calcium release from the ER and calcium entry from the extracellular space directly influence the activation of numerous transcription factors including NF-kB, NFAT, Oct, MEF2, and others. Therefore, differences in gene expression in T cells stimulated with PMA versus PMA + Ionomycin have complex etiologies. We are specifically interested in understanding how calcium signaling influences NF-kB specific target gene transcription and how differences in calcium signal dynamics can lead to distinct transcriptional outputs via calcium-modulated NF-kB activation. Over 150 NF-kB target genes have been identified including genes with checked binding sites and putative target genes. In an attempt to assess how calcium signaling influences the expression of NF-kB target genes, I have curated a list of NF-kB target genes and filtered my list of differentially expressed genes (displayed above) to include only genes that are associated with NF-kB signaling. Now let's take a look at those genes that appear to be differentially expressed when you compare NF-kB activation in the presence and absence of calcium.

```
NFkBtarget <-  read.table("NFkBtarget.txt", sep="\t", stringsAsFactors=FALSE)
diffNFkB_all_one <- diffData_avg_PIvsPonly_oneHr[diffSymbols_PIvsP_1%in%NFkBtarget[,1],]
diffNFkB_all_four <- diffData_avg_PIvsPonly_fourHr[diffSymbols_PIvsP_4%in%NFkBtarget[,1],]
diffNFkB_all_eight <- diffData_avg_PIvsPonly_eightHr[diffSymbols_PIvsP_8%in%NFkBtarget[,1],]

hr_PIvsP_NFkB_1 <- hclust(as.dist(1-cor(t(diffNFkB_all_one), method="pearson")), method="complete")
hr_PIvsP_NFkB_4 <- hclust(as.dist(1-cor(t(diffNFkB_all_four), method="pearson")), method="complete")
hr_PIvsP_NFkB_8 <- hclust(as.dist(1-cor(t(diffNFkB_all_eight), method="pearson")), method="complete")

mycl_PIvsP_NFkB_1 <- cutree(hr_PIvsP_NFkB_1, k=1)
mycolhc_PIvsP_NFkB_1 <- rainbow(length(unique(mycl_PIvsP_NFkB_1)), start=0.1, end=0.9)
mycolhc_PIvsP_NFkB_1 <- mycolhc_PIvsP_NFkB_1[as.vector(mycl_PIvsP_NFkB_1)]

mycl_PIvsP_NFkB_4 <- cutree(hr_PIvsP_NFkB_4, k=2)
mycolhc_PIvsP_NFkB_4 <- rainbow(length(unique(mycl_PIvsP_NFkB_4)), start=0.1, end=0.9)
mycolhc_PIvsP_NFkB_4 <- mycolhc_PIvsP_NFkB_4[as.vector(mycl_PIvsP_NFkB_4)]

mycl_PIvsP_NFkB_8 <- cutree(hr_PIvsP_NFkB_8, k=2)
mycolhc_PIvsP_NFkB_8 <- rainbow(length(unique(mycl_PIvsP_NFkB_8)), start=0.1, end=0.9)
mycolhc_PIvsP_NFkB_8 <- mycolhc_PIvsP_NFkB_8[as.vector(mycl_PIvsP_NFkB_8)]
```

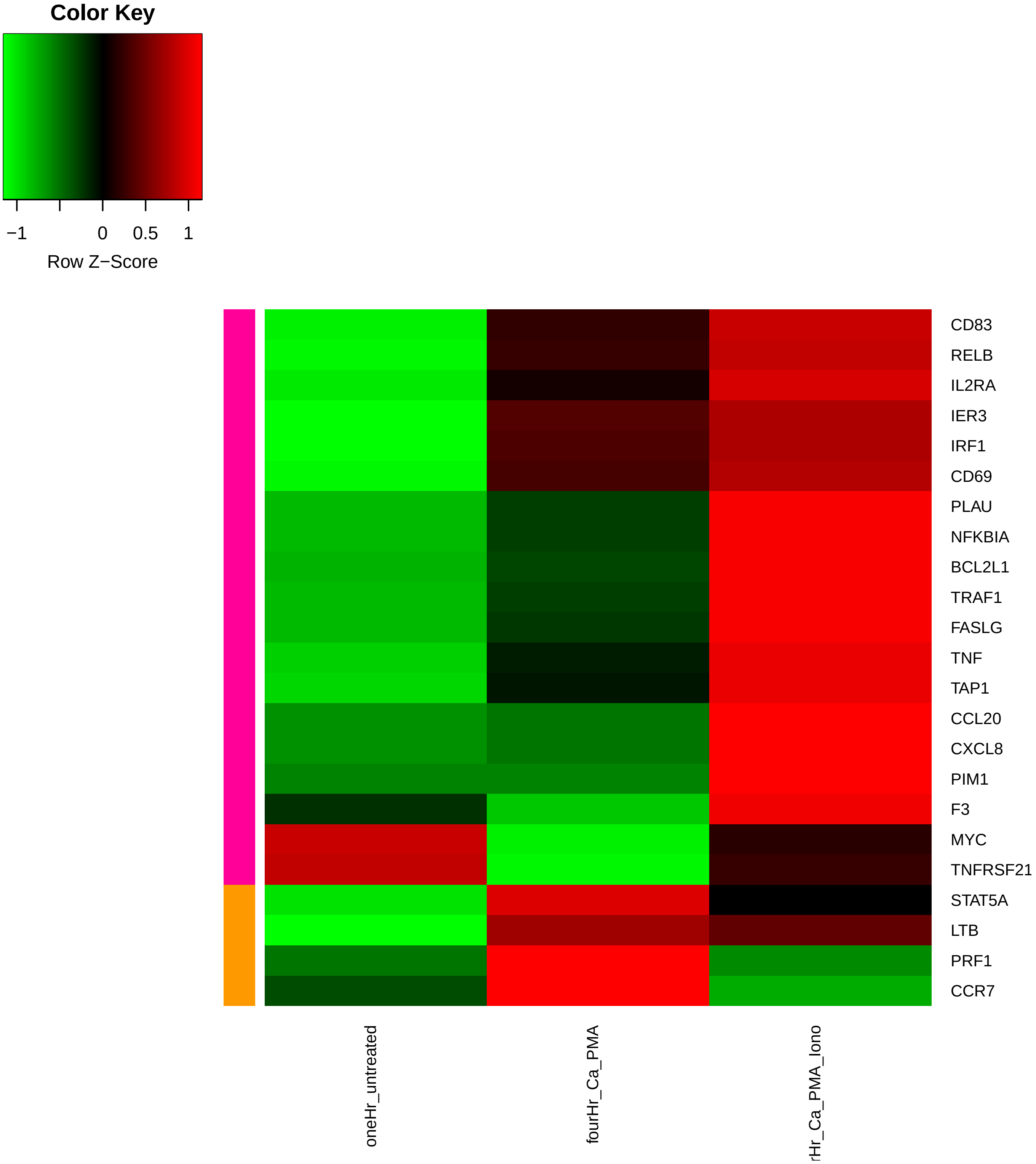
Let's now take a look at the genes that are differentially expressed between PMA and PMA + Ionomycin at 1 hour.

```
heatmap.2(diffNFkB_all_one, Rowv = NA,Colv = NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none",
          cexRow=1.0, cexCol=1.0, margins=c(10,10))
```



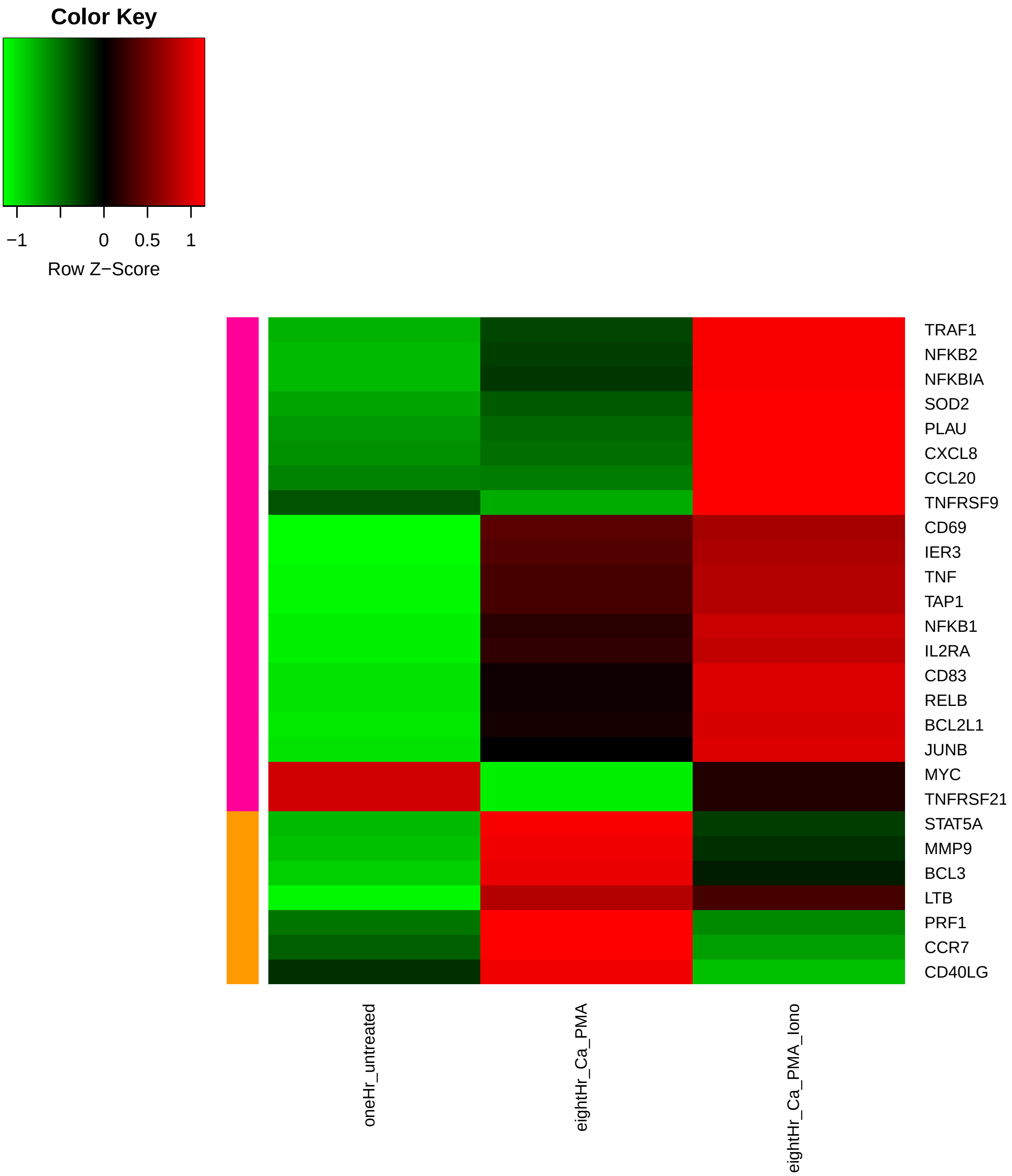
Now let's look at 4 hours...

```
heatmap.2(diffNFkB_all_four, Rowv=as.dendrogram(hr_PIvsP_NFkB_4), Colv=NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none", RowSideColors=mycolhc_PIvsP_NFkB_4,
          cexRow=1.0, cexCol=1.0, margins=c(8,8))
```



And finally 8 hours...

```
heatmap.2(diffNFkB_all_eight, Rowv=as.dendrogram(hr_PIvsP_NFkB_8), Colv=NA,
          col=myheatcol, scale="row",
          density.info="none", trace="none", RowSideColors=mycolhc_PIvsP_NFkB_8,
          cexRow=1.0, cexCol=1.0, margins=c(10,10))
```



These results provide evidence for the importance of calcium signaling in NF-kB dependent gene transcription in lymphocytes. We are now working to understand the exact mechanisms by which calcium influences the activation of NF-kB signaling by investigating novel, calcium-dependent post-translational modifications of NF-kB family members p65 and c-Rel using mass spectroscopy. Furthermore, we are working to understand how these post-translational modifications influence the activation of specific NF-kB target genes and how dynamic signals are able to tune the transcriptional response of T cells to ultimately determine cell fate, differentiation, and function.