Practicum 2

Peter Claussen Maggie Germundson Stephanie Liebl

10/26/2021

Introduction

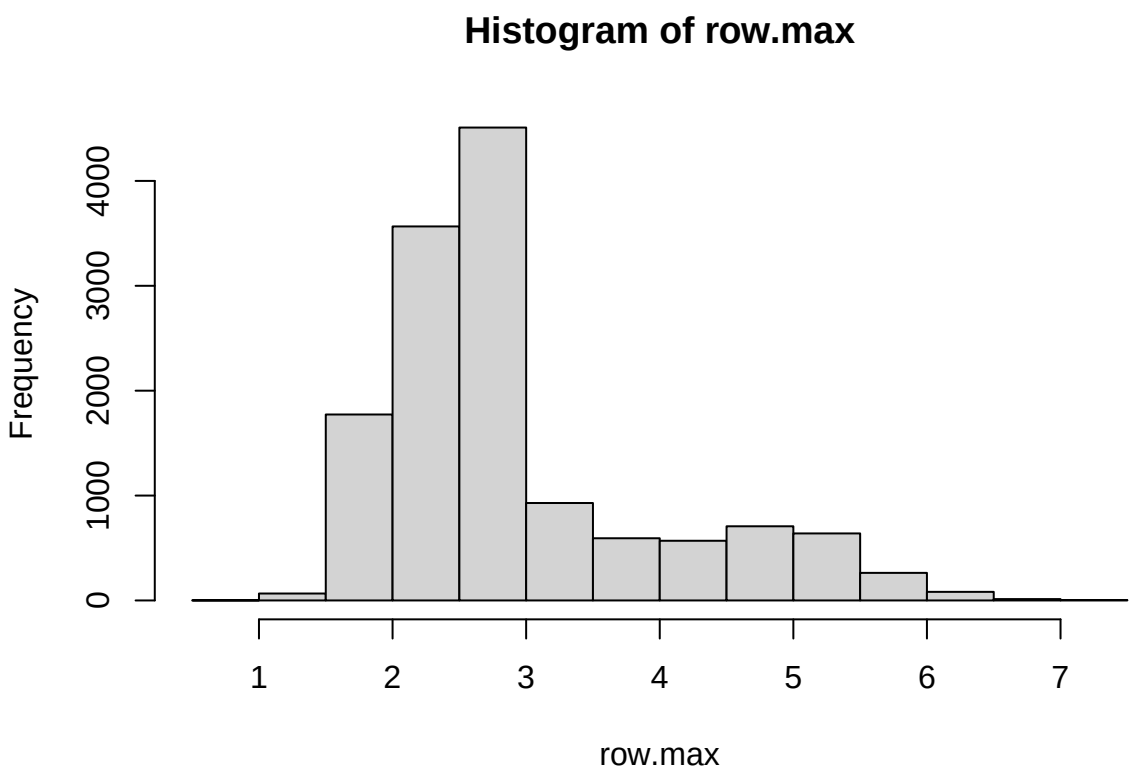
We have been give a large data file, consisting of expression levels of RNA for different genes. The mRNA were obtained from multiple cell lines. The client requests that we use RNA expression patterns to group cell lines. These data are expected to be very sparse.

Data Dimensions

```
## [1] 13714  2700
```

What is the variation in maximum expression levels?

```
row.max <- apply(cell.dat,1,max)
hist(row.max)
```



These data represent approximately 13000 genes and 2700 cell lines. For simplicity at this stage, we'll assume all 2700 cell lines are valid.

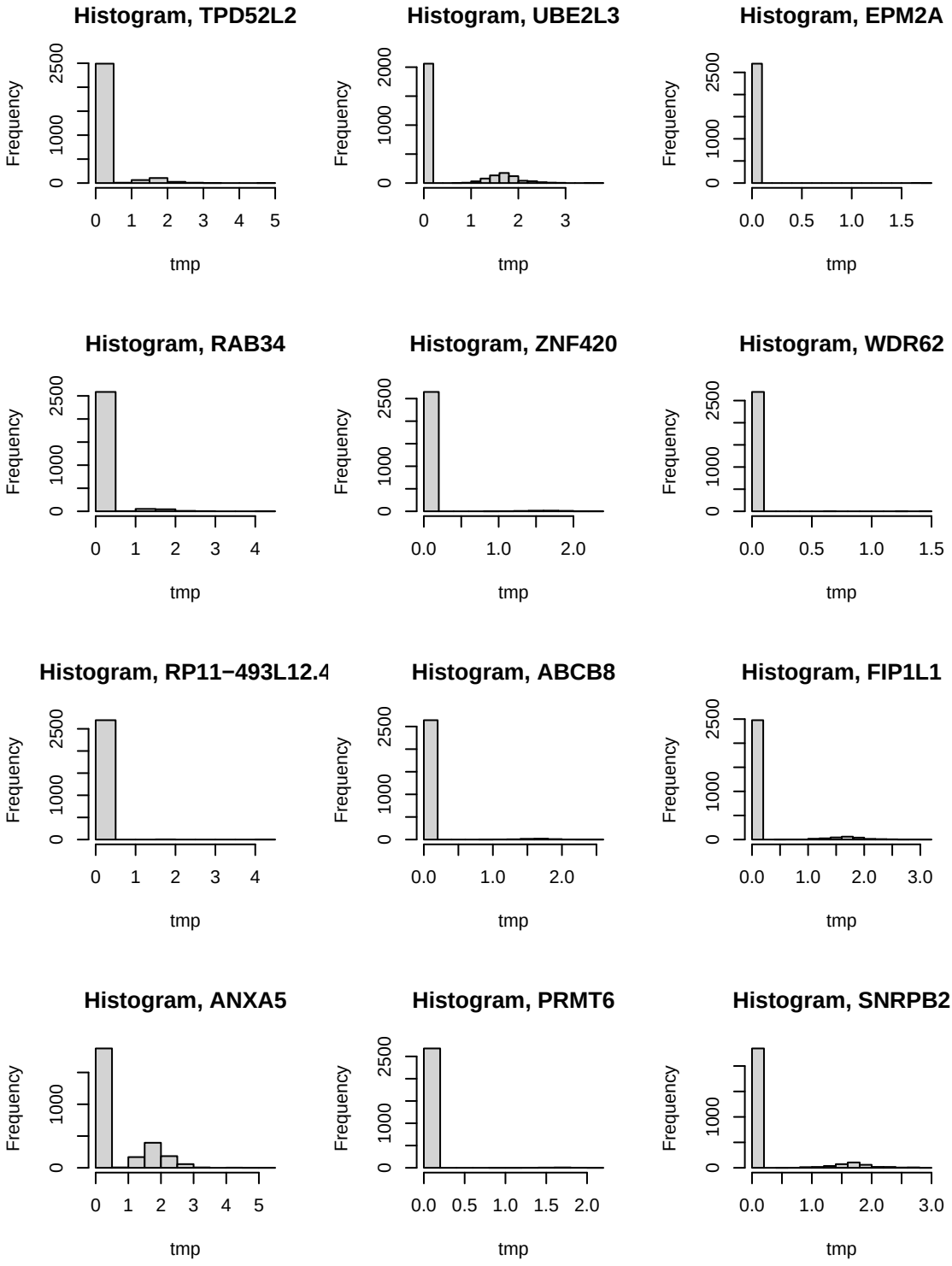
We could potentially use all 13000+ gene expression levels to generate a cluster analysis for cell lines. This, however, is computationally intractable, so we wish to reduce the dimensions of the problem. Briefly, we consider four dimension reduction procedures.

We consider the expression pattern of some randomly selected genes.

```
random.samples <- sample(1:dim(cell.dat)[1],12)
random.samples
```

```
##  [1] 12117 13263  4806 11014 12704 12680  8465  5469  3276  3453   690 11851
```

```
par(mfrow=c(4,3))
for(i in random.samples) {
  tmp <- unlist(cell.dat[i,])
  hist(tmp,main=paste("Histogram,",genes[i]))
}
```


Histogram, TPD52L2
Histogram, UBE2L3
Histogram, EPM2A
Histogram, RAB34
Histogram, ZNF420
Histogram, WDR62
Histogram, RP11−493L12.4
Histogram, ABCB8
Histogram, FIP1L1
Histogram, ANXA5
Histogram, PRMT6
Histogram, SNRPB2
Frequency
tmp

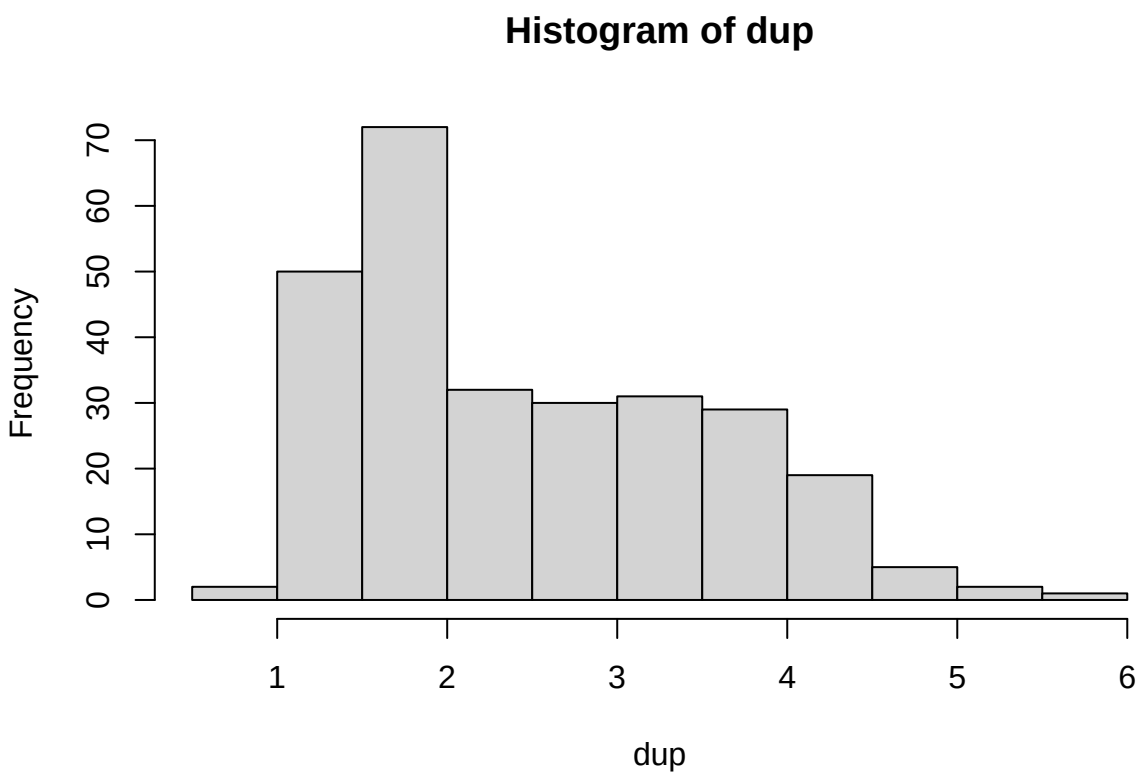
Commonly expressed genes

For many genes, it appears that only a small proportion of cell express these genes. We may use this as a dimension reduction procedure. We can, for example, exclude all genes where the median expression value is 0 (that is, genes that are not expressed in at least one-half of all cell lines.).

```
row.median <- apply(cell.dat,1,median)
dup <- row.median
summary(row.median)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.04935 0.00000 5.56168
```

```
dup <- row.median[row.median>0]
hist(dup)
```




```
work.dat <- cell.dat[row.median>0,]
genes.common <- genes[row.median>0]
dim(work.dat)
```

```
## [1]  273 2700
```

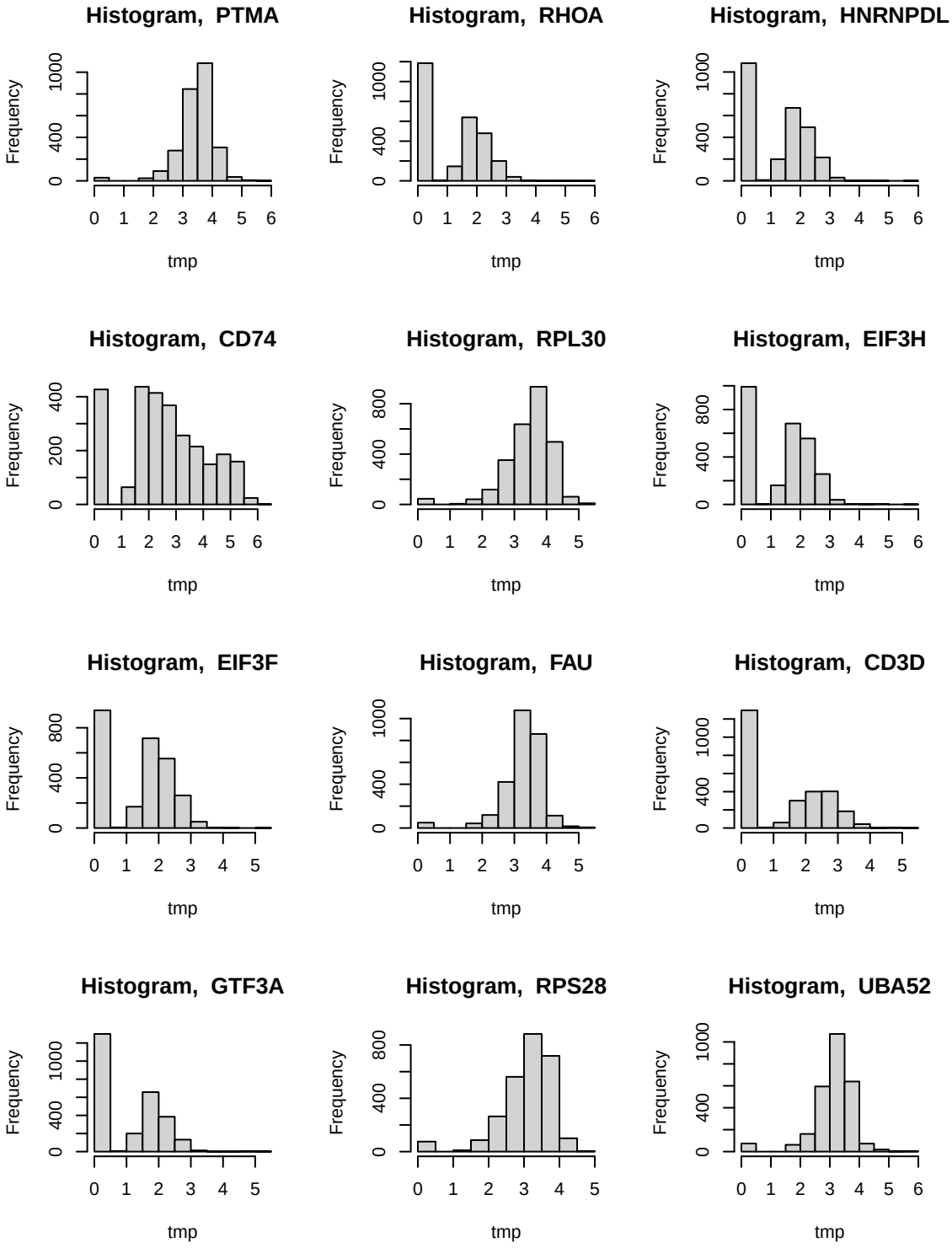
We plot the histograms of some of the common genes

```
print(common.sample <- sample(genes.common,12))
```

```
##  [1] "EIF3F"   "CD74"    "FAU"     "RPS28"   "HNRNPDL" "EIF3H"   "GTF3A"
##  [8] "PTMA"    "RPL30"   "CD3D"    "RHOA"    "UBA52"
```

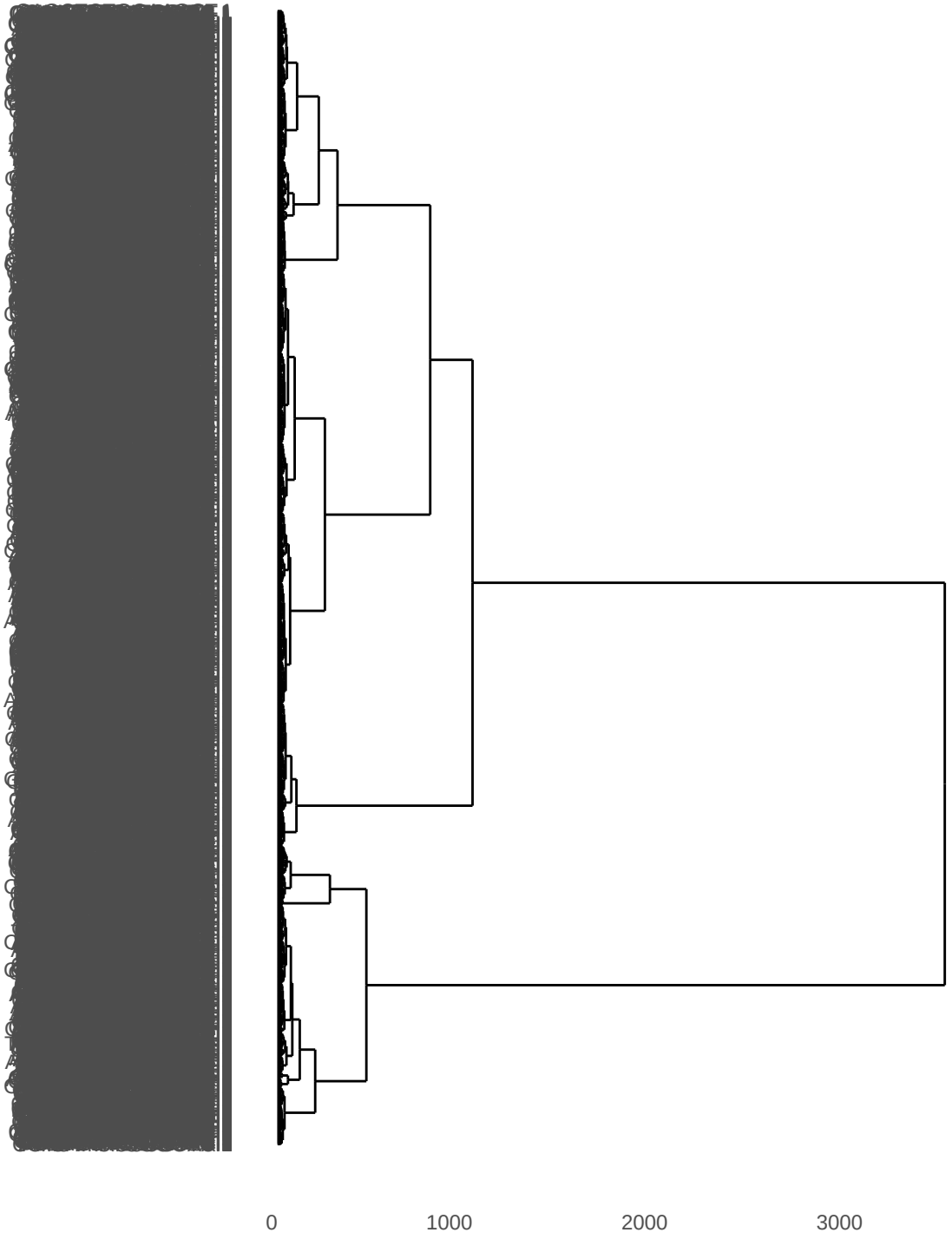
```
common.mask <- genes %in% common.sample
common.dat <- cell.dat[common.mask,]
common.names <- genes[common.mask]
par(mfrow=c(4,3))
```

```r
for(i in 1:dim(common.dat)[1]) {
  tmp <- unlist(common.dat[i,])
  hist(tmp,main=paste("Histogram, ",common.names[i]))
}
```

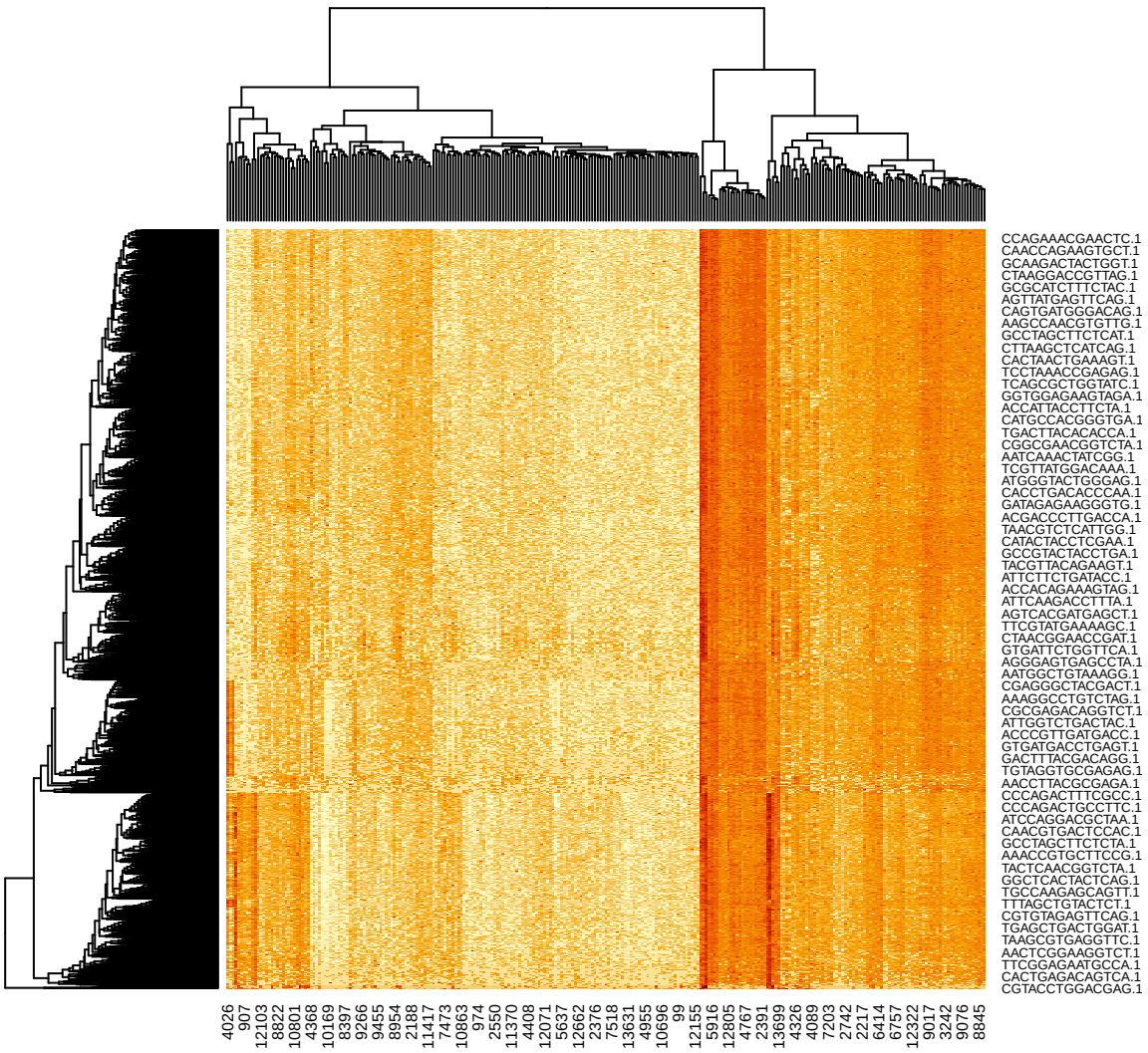


This leaves us with 273 genes, a more reasonable number to work with. A proposed dendrogram, showing potential cell clusters follows.

```
distances <- dist(t(work.dat), method="euclidean")
clusters <- hclust(distances,method="ward.D")
ggdendrogram(clusters,rotate=TRUE)
```

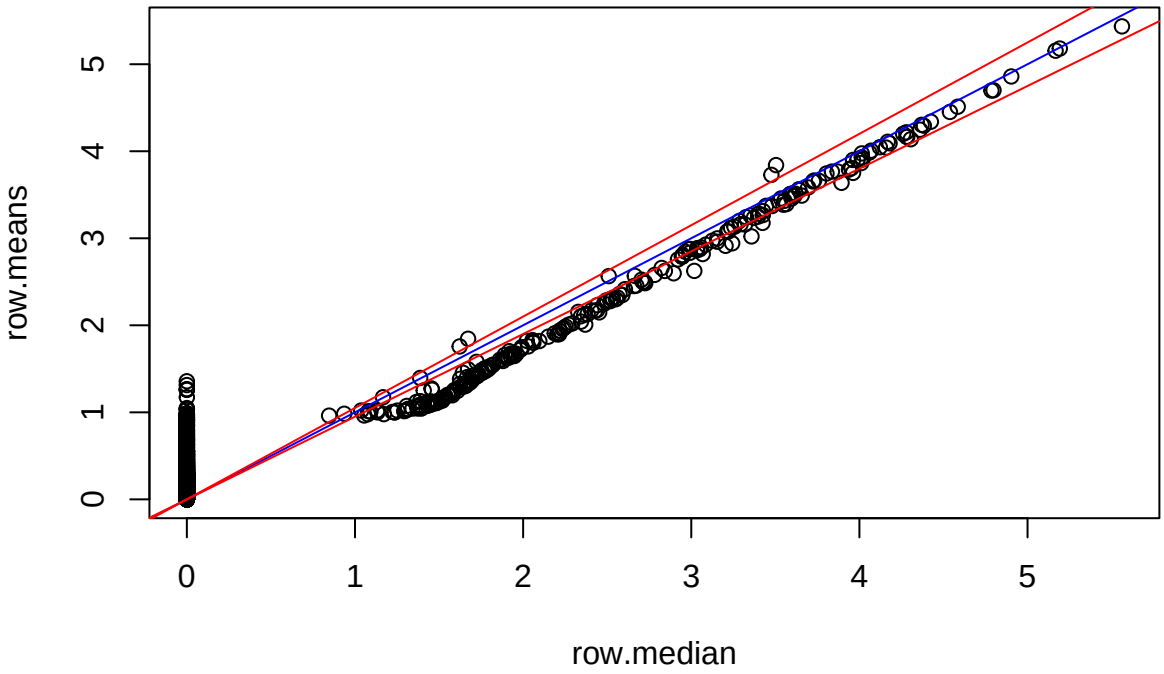



```
heatmap(t(work.dat))
```


CCAGAAACGAACTC.1
CAACCAGAAGTGCT.1
GCAAGACTACTGGT.1
CTAAGGACCGTTAG.1
GCGCATCTTTCTAC.1
AGTTATGAGTTCAG.1
CAGTGATGGGACAG.1
AAGCCAACGTGTTG.1
GCCTAGCTTCTCAT.1
CTTAAGCTCATCAG.1
CACTAACTGAAAGT.1
TCCTAAACCGAGAG.1
TCAGCGCTGGTATC.1
GGTGGAGAAGTAGA.1
ACCATTACCTTCTA.1
CATGCCACGGGTGA.1
TGACTTACACACCA.1
CGGCGAACGGTCTA.1
AATCAAACTATCGG.1
TCGTTATGGACAAA.1
ATGGGTACTGGGAG.1
CACCTGACACCCAA.1
GATAGAGAAGGGTG.1
ACGACCCTTGACCA.1
TAACGTCTCATTGG.1
CATACTACCTCGAA.1
GCCGTACTACCTGA.1
TACGTTACAGAAGT.1
ATTCTTCTGATACC.1
ACCACAGAAAGTAG.1
ATTCAAGACCTTTA.1
AGTCACGATGAGCT.1
TTCGTATGAAAAGC.1
CTAACGGAACCGAT.1
GTGATTCTGGTTCA.1
AGGGAGTGAGCCTA.1
AATGGCTGTAAAGG.1
CGAGGGCTACGACT.1
AAAGGCCTGTCTAG.1
CGCGAGACAGGTCT.1
ATTGGTCTGACTAC.1
ACCCGTTGATGACC.1
GTGATGACCTGAGT.1
GACTTTACGACAGG.1
TGTAGGTGCGAGAG.1
AACCTTACGCGAGA.1
CCCAGACTTTCGCC.1
CCCAGACTGCCTTC.1
ATCCAGGACGCTAA.1
CAACGTGACTCCAC.1
GCCTAGCTTCTCTA.1
AAACCGTGCTTCCG.1
TACTCAACGGTCTA.1
GGCTCACTACTCAG.1
TGCCAAGAGCAGTT.1
TTTAGCTGTACTCT.1
CGTGTAGAGTTCAG.1
TGAGCTGACTGGAT.1
TAAGCGTGAGGTTC.1
AACTCGGAAGGTCT.1
TTCGGAGAATGCCA.1
CACTGAGACAGTCA.1
CGTACCTGGACGAG.1
4026
907
12103
8822
10801
4368
10169
8397
9266
9455
8954
2188
11417
7473
10863
974
2550
11370
4408
12071
5637
12662
2376
7518
13631
4955
10696
99
12155
5916
12805
4767
2391
13699
4326
4089
7203
2742
2217
6414
6757
12322
9017
3242
9076
8845

Most commonly expressed genes

We might further reduce the number by selecting those with gene expression patterns approximately normal. Since there are still a large number, we use an approximation. When data are normally distributed (or, at least, symmetrically distributed), then the median will be close to the mean. For the current subset of genes, we can visualize this by plotting the median gene expression level against the corresponding mean. We bound this by a band representing 95-105% - that is, we wish to select genes where the mean is within 5%, plus or minus, of the mean.




```
ratio <- row.means/row.median
mask <- ratio<1.05 & ratio>0.95
sum(mask)
```

```
## [1] 81
```

```
work.dat <- cell.dat[mask,]
genes.work <- genes[mask]
dim(work.dat)
```

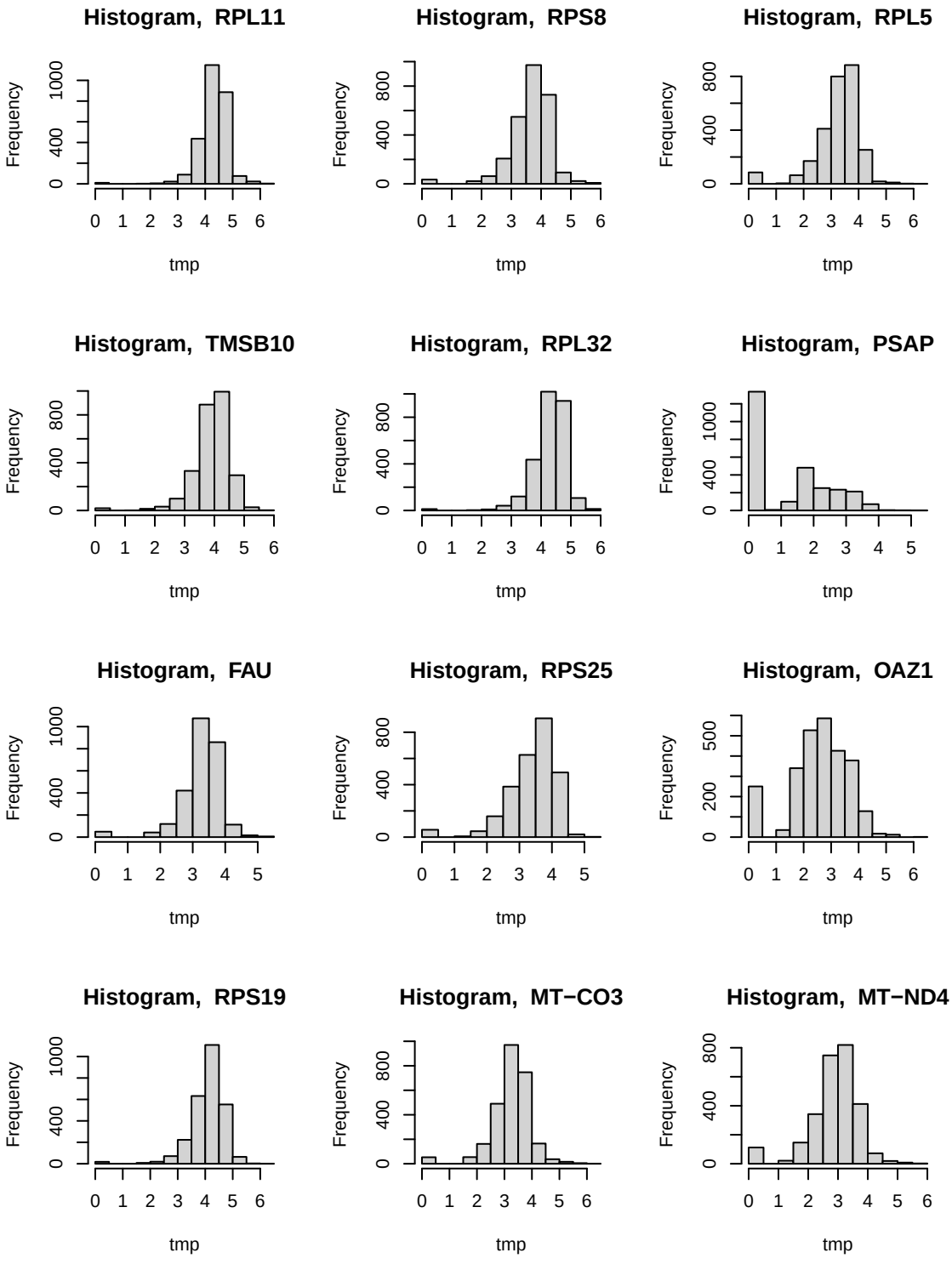
```
## [1]   81 2700
```

We print some sample histograms for common genes with approximately symmetric distributions.

```
print(work.sample <- sample(genes.work,12))
```

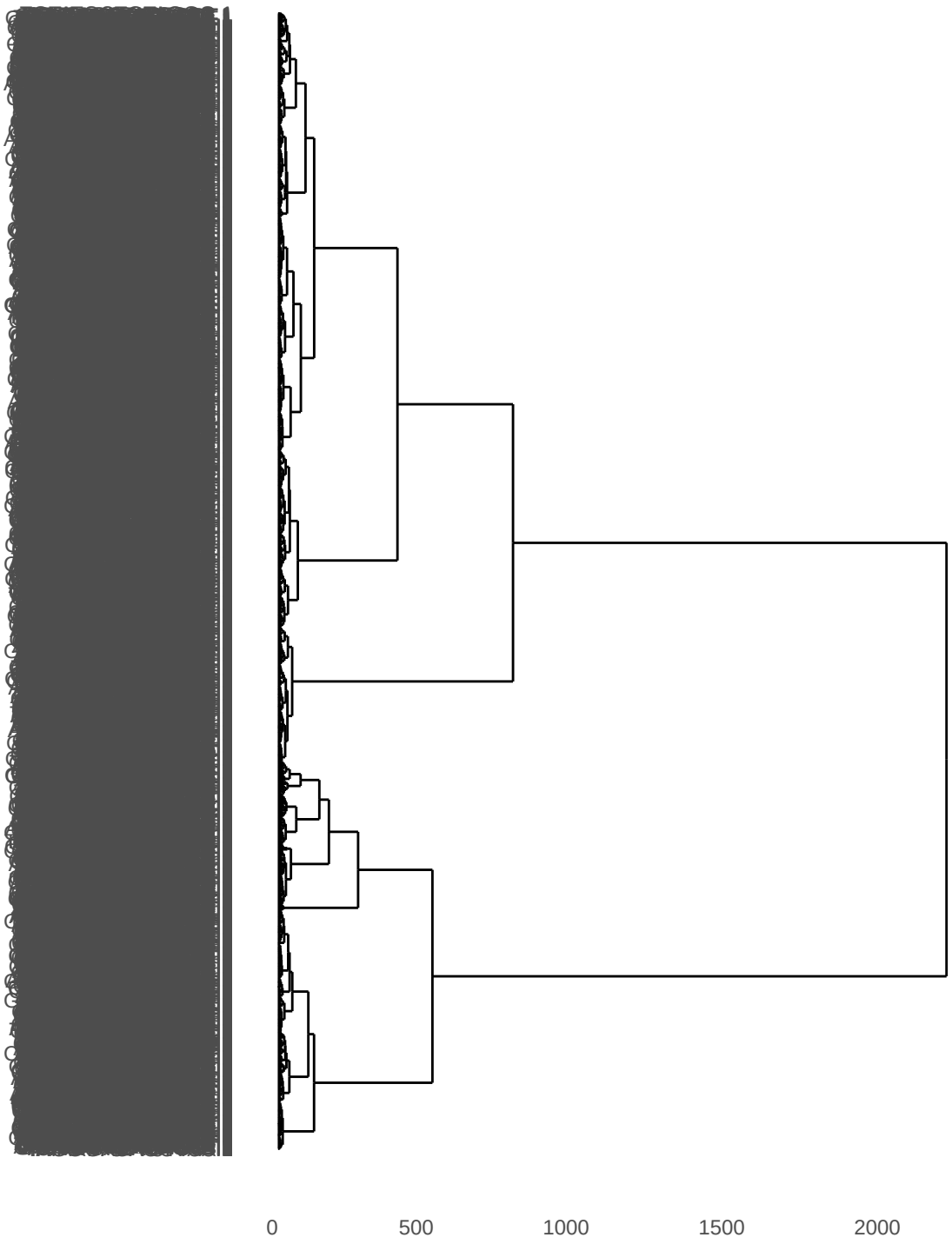
```
##  [1] "MT-CO3" "TMSB10" "RPL32"  "RPS19"  "FAU"    "MT-ND4" "RPS8"   "OAZ1"
##  [9] "RPL5"   "RPL11"  "RPS25"  "PSAP"
```

```
work.mask <- genes %in% work.sample
common.dat <- cell.dat[work.mask,]
common.names <- genes[work.mask]
par(mfrow=c(4,3))
for(i in 1:dim(common.dat)[1]) {
  tmp <- unlist(common.dat[i,])
  hist(tmp,main=paste("Histogram, ",common.names[i]))
}
```

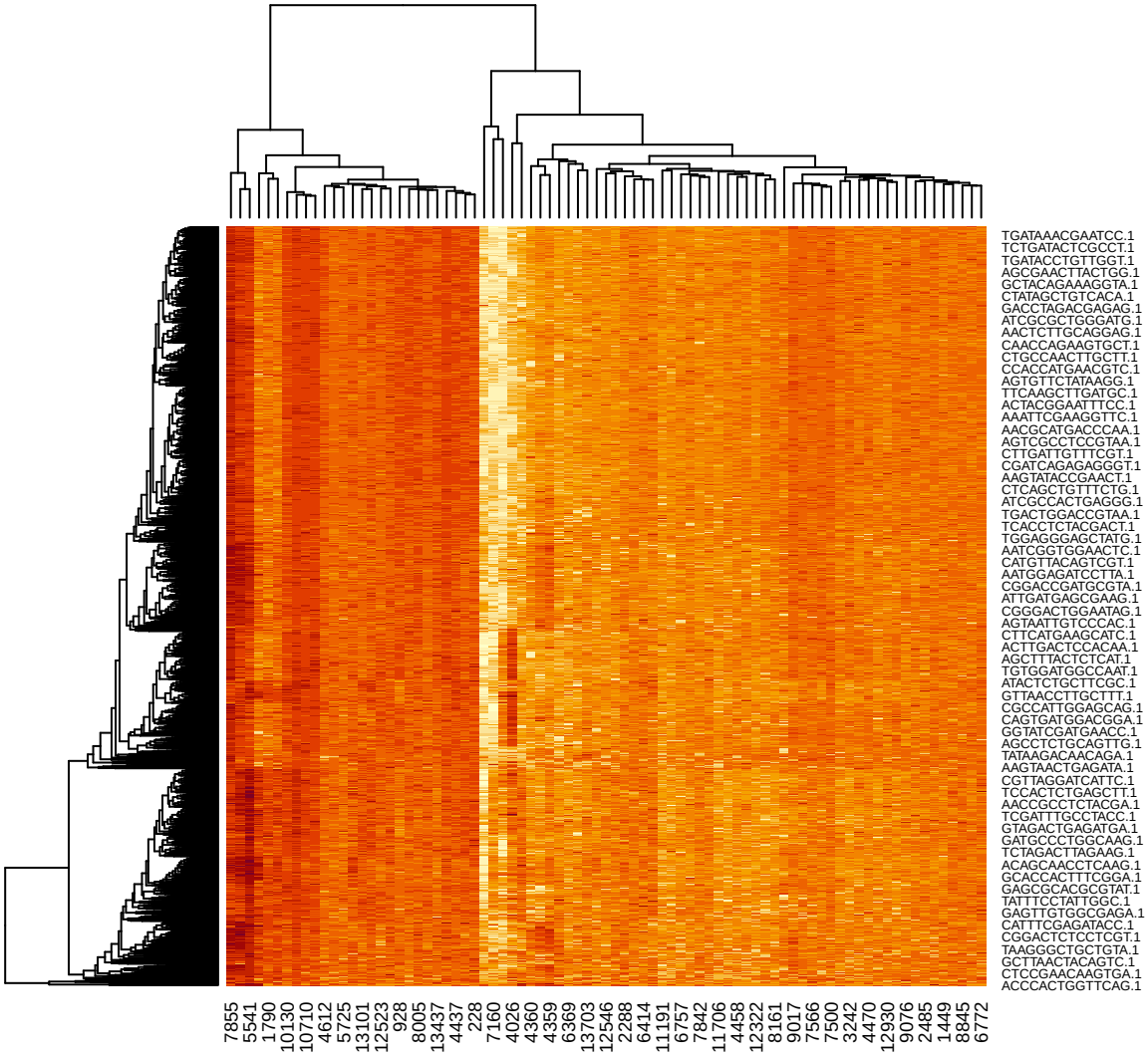


A potential dendrogram based on this subset of genes follows.

```
distances <- dist(t(work.dat), method="euclidean")
clusters <- hclust(distances,method="ward.D")
ggdendrogram(clusters,rotate=TRUE)
```




```
heatmap(t(work.dat))
```

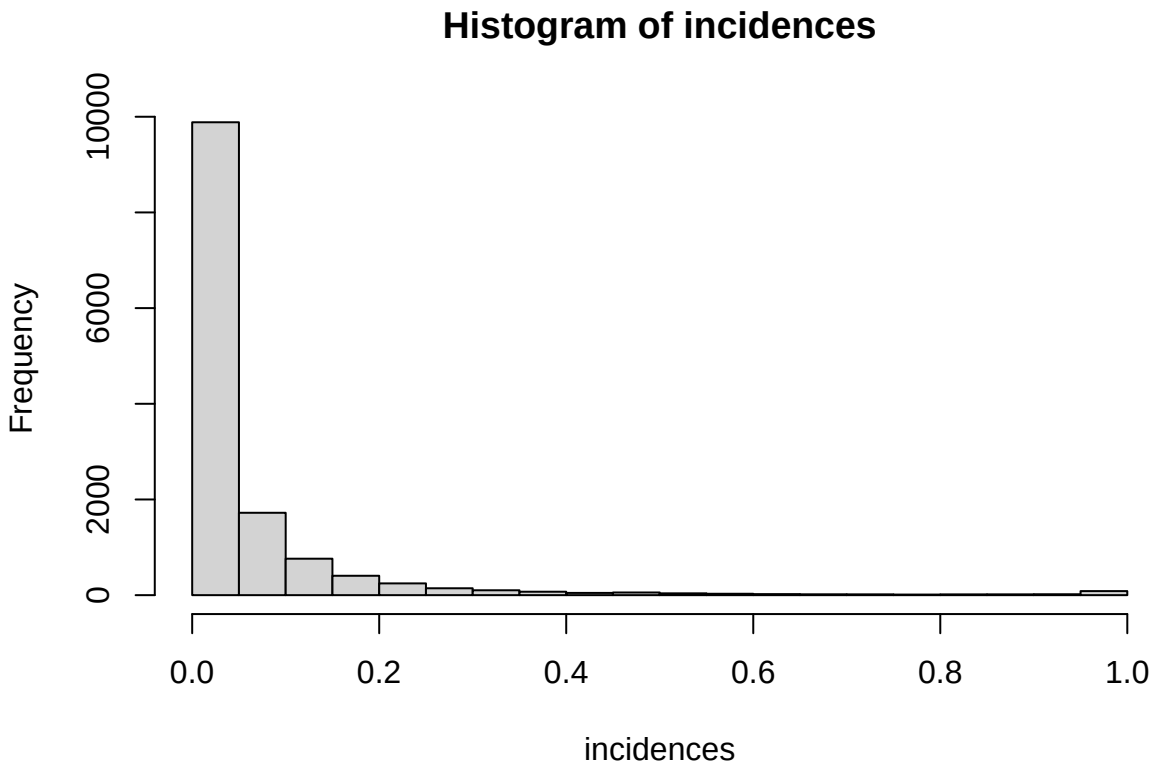

TGATAAACGAATCC.1
TCTGATACTCGCCT.1
TGATACCTGTTGGT.1
AGCGAACTTACTGG.1
GCTACAGAAAGGTA.1
CTATAGCTGTCACA.1
GACCTAGACGAGAG.1
ATCGCGCTGGGATG.1
AACTCTTGCAGGAG.1
CAACCAGAAGTGCT.1
CTGCCAACTTGCTT.1
CCACCATGAACGTC.1
AGTGTTCTATAAGG.1
TTCAAGCTTGATGC.1
ACTACGGAATTTCC.1
AAATTCGAAGGTTC.1
AACGCATGACCCAA.1
AGTCGCCTCCGTAA.1
CTTGATTGTTTCGT.1
CGATCAGAGAGGGT.1
AAGTATACCGAACT.1
CTCAGCTGTTTCTG.1
ATCGCCACTGAGGG.1
TGACTGGACCGTAA.1
TCACCTCTACGACT.1
TGGAGGGAGCTATG.1
AATCGGTGGAACTC.1
CATGTTACAGTCGT.1
AATGGAGATCCTTA.1
CGGACCGATGCGTA.1
ATTGATGAGCGAAG.1
CGGGACTGGAATAG.1
AGTAATTGTCCCAC.1
CTTCATGAAGCATC.1
ACTTGACTCCACAA.1
AGCTTTACTCTCAT.1
TGTGGATGGCCAAT.1
ATACTCTGCTTCGC.1
GTTAACCTTGCTTT.1
CGCCATTGGAGCAG.1
CAGTGATGGACGGA.1
GGTATCGATGAACC.1
AGCCTCTGCAGTTG.1
TATAAGACAACAGA.1
AAGTAACTGAGATA.1
CGTTAGGATCATTC.1
TCCACTCTGAGCTT.1
AACCGCCTCTACGA.1
TCGATTTGCCTACC.1
CTAGACTGAGATGA.1
GATGCCCTGGCAAG.1
TCTAGACTTAGAAG.1
ACAGCAACCTCAAG.1
GCACCACTTTCGGA.1
GAGCGCACGCGTAT.1
TATTTCCTATTGGC.1
GAGTTGTGGCGAGA.1
CATTTCGAGATACC.1
CGGACTCTCCTCGT.1
TAAGGGCTGCTGTA.1
GCTTAACTACAGTC.1
CTCCGAACAAGTGA.1
ACCCACTGGTTCAG.1
7855
5541
1790
10130
10710
4612
5725
13101
12523
928
8005
13437
4437
228
7160
4026
4360
4359
6369
13703
12546
2288
6414
11191
6757
7842
11706
4458
12322
8161
9017
7566
7500
3242
4470
12930
9076
2485
1449
8845
6772

Uncommon, but not rare, genes

Distinct cell types may be distiquished not simply by gene expression levels, but by gene expression patterns. That is, a certain gene may only be expressed in certain cell types and could thus be usefully diagnostic, while other genes (i.e. glycolytic pathway) are expressed in nearly all cells and may be of little use in determining cell groups.

Since we have seen that the majority genes are expressed in less than half the cells, we don't wish to use very rare genes. Instead, we propose to consider genes that are expressed in at least 10% of cell lines, but no more than 20%.

```
incidence.dat <- cell.dat
incidence.dat[incidence.dat>0.01] <- 1

incidences <- apply(incidence.dat,1,mean)
hist(incidences)
```




```
incidence.dat <- incidence.dat[incidences>0.1 & incidences<0.2,]
genes.incidence <- genes[incidences>0.1 & incidences<0.2]
dim(incidence.dat)
```

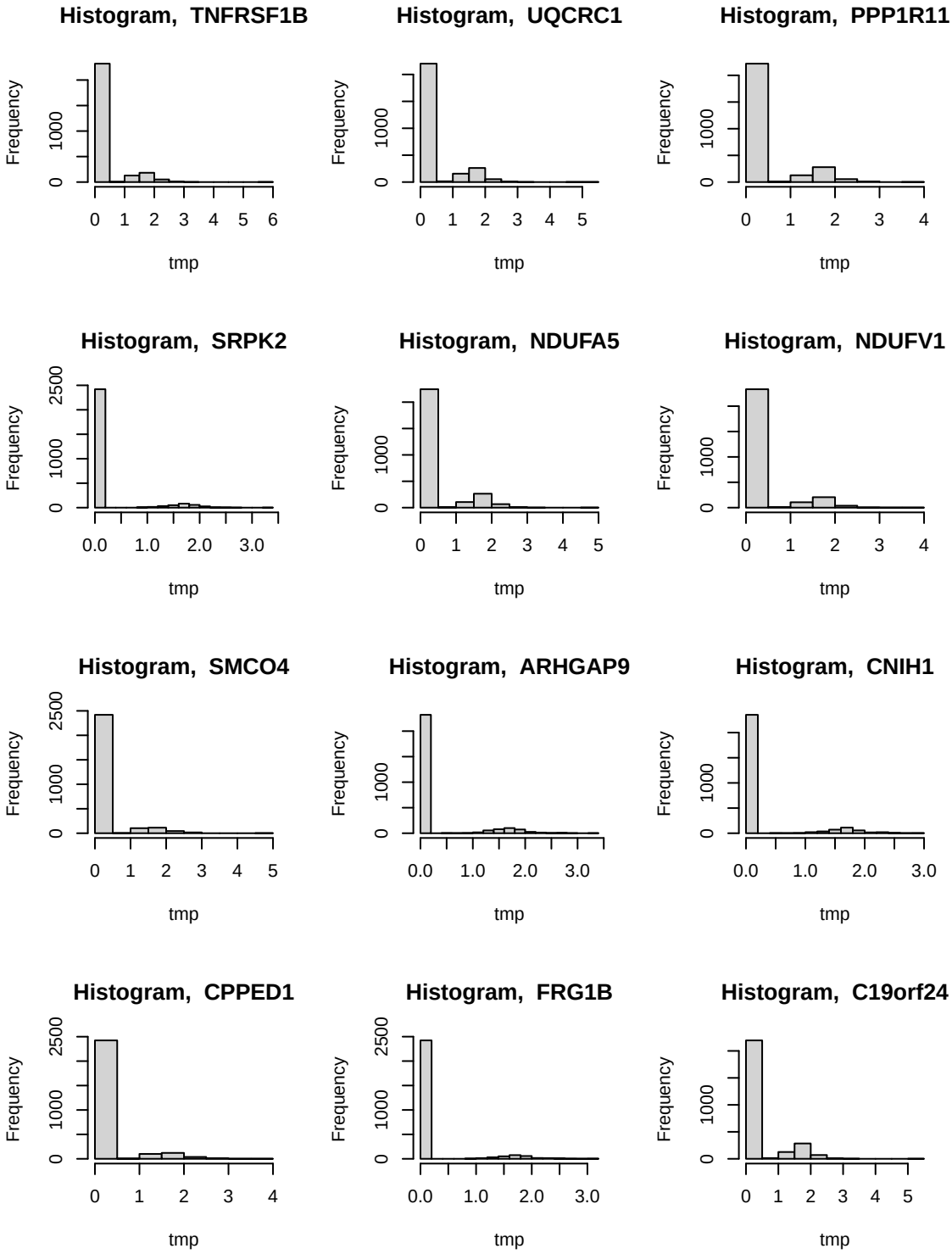
```
## [1] 1164 2700
```

```
print(incidence.sample <- sample(genes.incidence,12))
```

```
##  [1] "CNIH1"    "TNFRSF1B" "SMCO4"    "FRG1B"    "CPPED1"   "UQCRC1"
##  [7] "ARHGAP9"  "PPP1R11"  "SRPK2"    "NDUFV1"   "NDUFA5"   "C19orf24"
```

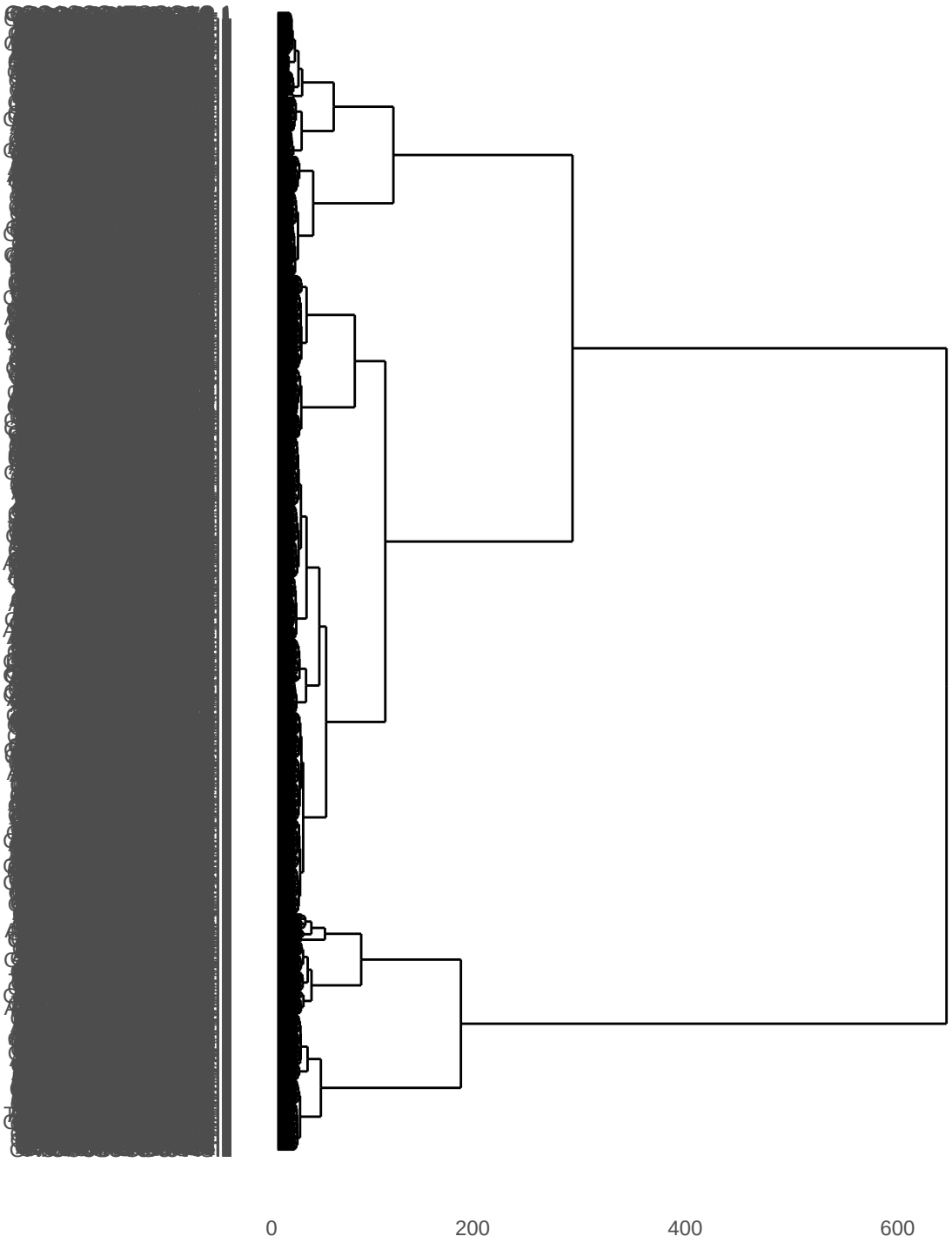
```
incidence.mask <- genes %in% incidence.sample
tmp.dat <- cell.dat[incidence.mask,]
tmp.names <- genes[incidence.mask]
par(mfrow=c(4,3))
for(i in 1:dim(tmp.dat)[1]) {
```

```
  tmp <- unlist(tmp.dat[i,])
  hist(tmp,main=paste("Histogram, ",tmp.names[i]))
}
```

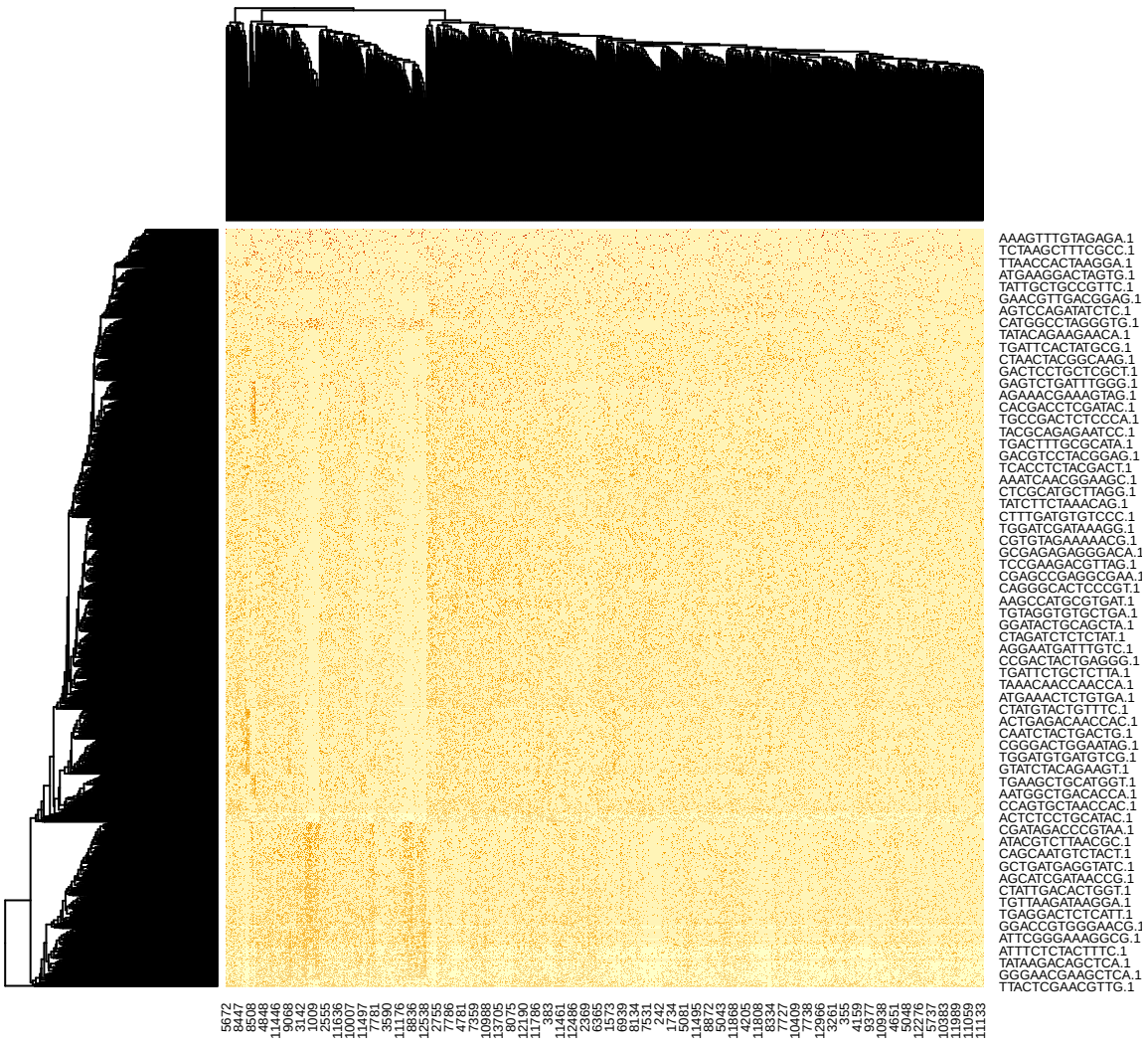


This results in approximately 1100 candidate genes. A potential dendrogram is presented below:

```
distances <- dist(t(incidence.dat), method="euclidean")
clusters <- hclust(distances,method="ward.D")
ggdendrogram(clusters,rotate=TRUE)
```




```
heatmap(t(incidence.dat))
```


AAAGTTTGTAGAGA.1
TCTAAGCTTTCGCC.1
TTAACCACTAAGGA.1
ATGAAGGACTAGTG.1
TATTGCTGCCGTTC.1
GAACGTTGACGGAG.1
AGTCCAGATATCTC.1
CATGGCCTAGGGTG.1
TATACAGAAGAACA.1
TGATTCACTATGCG.1
CTAACTACGGCAAG.1
GACTCCTGCTCGCT.1
GAGTCTGATTTGGG.1
AGAAACGAAAGTAG.1
CACGACCTCGATAC.1
TGCCGACTCTCCCA.1
TACGCAGAGAATCC.1
TGACTTTGCGCATA.1
GACGTCCTACGGAG.1
TCACCTCTACGACT.1
AAATCAACGGAAGC.1
CTCGCATGCTTAGG.1
TATCTTCTAAACAG.1
CTTTGATGTGTCCC.1
TGGATCGATAAAGG.1
CGTGTAGAAAAACG.1
GCGAGAGAGGGACA.1
TCCGAAGACGTTAG.1
CGAGCCGAGGCGAA.1
CAGGGCACTCCCGT.1
AAGCCATGCGTGAT.1
TGTAGGTGTGCTGA.1
GGATACTGCAGCTA.1
CTAGATCTCTCTAT.1
AGGAATGATTTGTC.1
CCGACTACTGAGGG.1
TGATTCTGCTCTTA.1
TAAACAACCAACCA.1
ATGAAACTCTGTGA.1
CTATGTACTGTTTC.1
ACTGAGACAACCAC.1
CAATCTACTGACTG.1
CGGGACTGGAATAG.1
TGGATGTGATGTCG.1
GTATCTACAGAAGT.1
TGAAGCTGCATGGT.1
AATGGCTGACACCA.1
CCAGTGCTAACCAC.1
ACTCTCCTGCATAC.1
CGATAGACCCGTAA.1
ATACGTCTTAACGC.1
CAGCAATGTCTACT.1
GCTGATGAGGTATC.1
AGCATCGATAACCG.1
CTATTGACACTGGT.1
TGTTAAGATAAGGA.1
TGAGGACTCTCATT.1
GGACCGTGGGAACG.1
ATTCGGGAAAGGCG.1
ATTTCTCTACTTTC.1
TATAAGACAGCTCA.1
GGGAACGAAGCTCA.1
TTACTCGAACGTTG.1
5672
8447
8508
4848
11446
9068
3142
2009
2555
11636
10007
11497
7781
3590
11176
8836
12538
2755
7786
4781
7359
10988
13705
8075
12190
11796
3861
11461
12486
2369
6365
1573
6939
8134
7531
742
1734
5081
11495
8872
5043
11888
4205
11698
8834
7727
10409
7738
12966
3261
355
4159
9377
10938
4651
5048
12276
5737
10383
11989
11059
11133

Gene Ontology

We may be able to reduce the number of genes considered with further guidance from the researcher. A brief survey of the descriptors associate with rows suggest families of genes may be used to reduce the dimensions of the problem. For example, there appear to be 12 different aldehyde dehydrogenase genes represented in these data.

```
sum(grepl('^ALDH',genes))
```

```
## [1] 12
```

It also appears these data include non-protein coding genes, for example, we find examples of Long Intergenic Non-Protein Coding RNA (LINC). We also find non-cytoplasmic proteins such as Mitochondrial Ribosomal Protein (MRPL)

```
sum(grepl('^LINC',genes))
```

```
## [1] 85
```

```
sum(grepl('^MRPL',genes))
```

```
## [1] 47
```

Other examples of proteins with known functionality include membrane protiens, i.e.

```
#Solute Carrier
sum(grepl('^SLC',genes))
```

```
## [1] 218
```

```
#Transmembrane Protein
sum(grepl('^TMEM',genes))
```

```
## [1] 171
```

Other groups include proteins with identifiable secondary or tertiary structural characteristics,

```
#coiled coil domain containing
sum(grepl('^CCDC',genes))
```

```
## [1] 92
```

```
#Leucine-rich repeat-containing protein
sum(grepl('^LRRC',genes))
```

```
## [1] 36
```

```
#WD Repeat Domain
sum(grepl('^WDR',genes))
```

```
## [1] 54
```

```
#Zinc Finger Protein
sum(grepl('^ZNF',genes))
```

```
## [1] 438
```

```
#Family with sequence similarity
sum(grepl('^FAM',genes))
```

```
## [1] 182
```

while some genes have not clear ontology (at least, based on naming conventions)

```
#open reading frame
sum(grepl('orf',genes))
```

```
## [1] 313
```

```
sum(grepl('A[C|L|P][0-9]',genes))
```

```
## [1] 600
```

```
sum(grepl('^CT[B|C]',genes))
```

```
## [1] 63
```

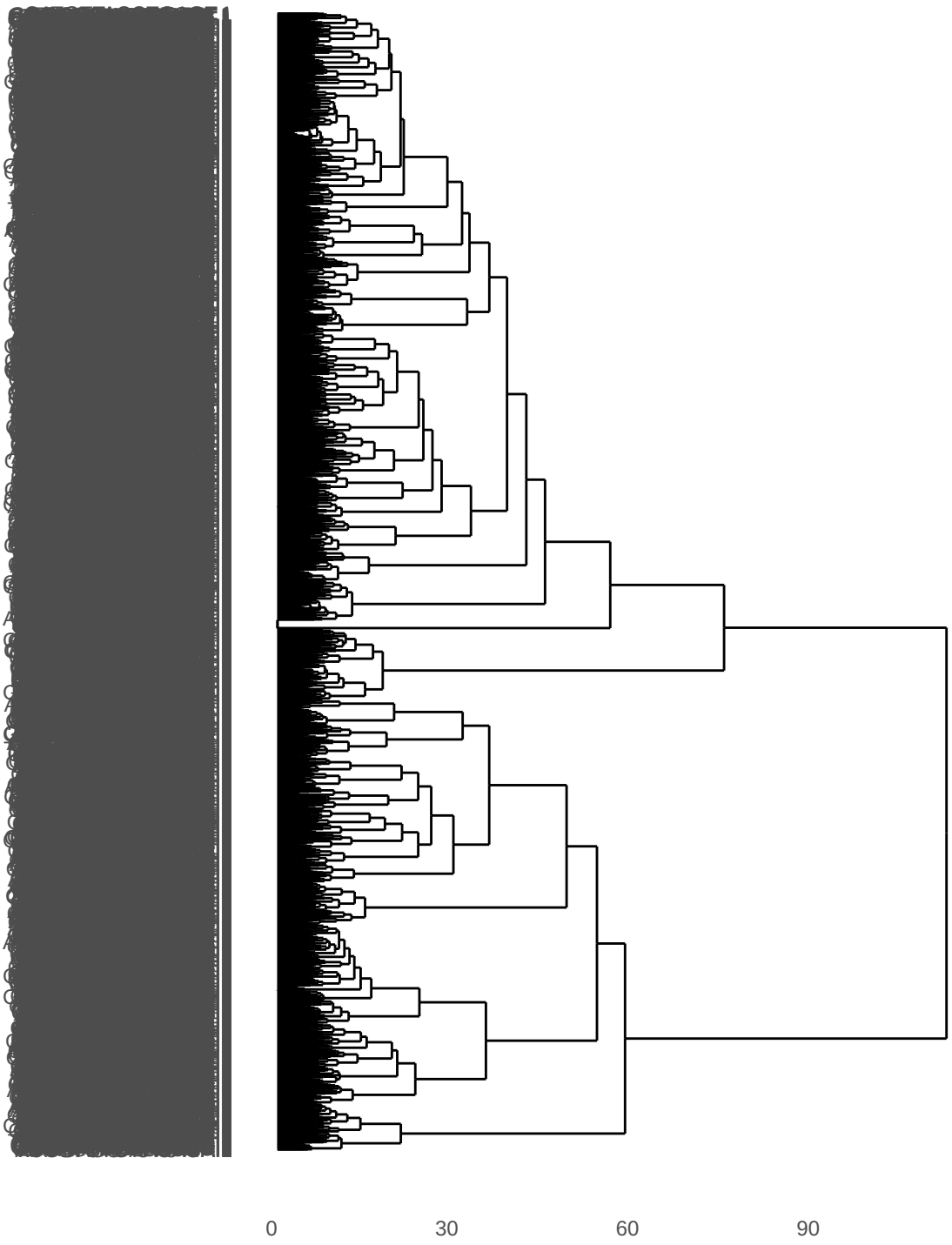
```
sum(grepl('^RP[1-9]',genes))
```

```
## [1] 854
```

Question 1.

Is there are subset of genes that, *a priori* would be preferred to differentiate among cell lineages? For example, perhaps zinc-finger protein expression is discriminatory?

```
znf.genes <- genes[grepl('^ZNF',genes)]
znf.dat <- cell.dat[grepl('^ZNF',genes),]
distances <- dist(t(znf.dat), method="euclidean")
clusters <- hclust(distances,method="ward.D")
ggdendrogram(clusters,rotate=TRUE)
```

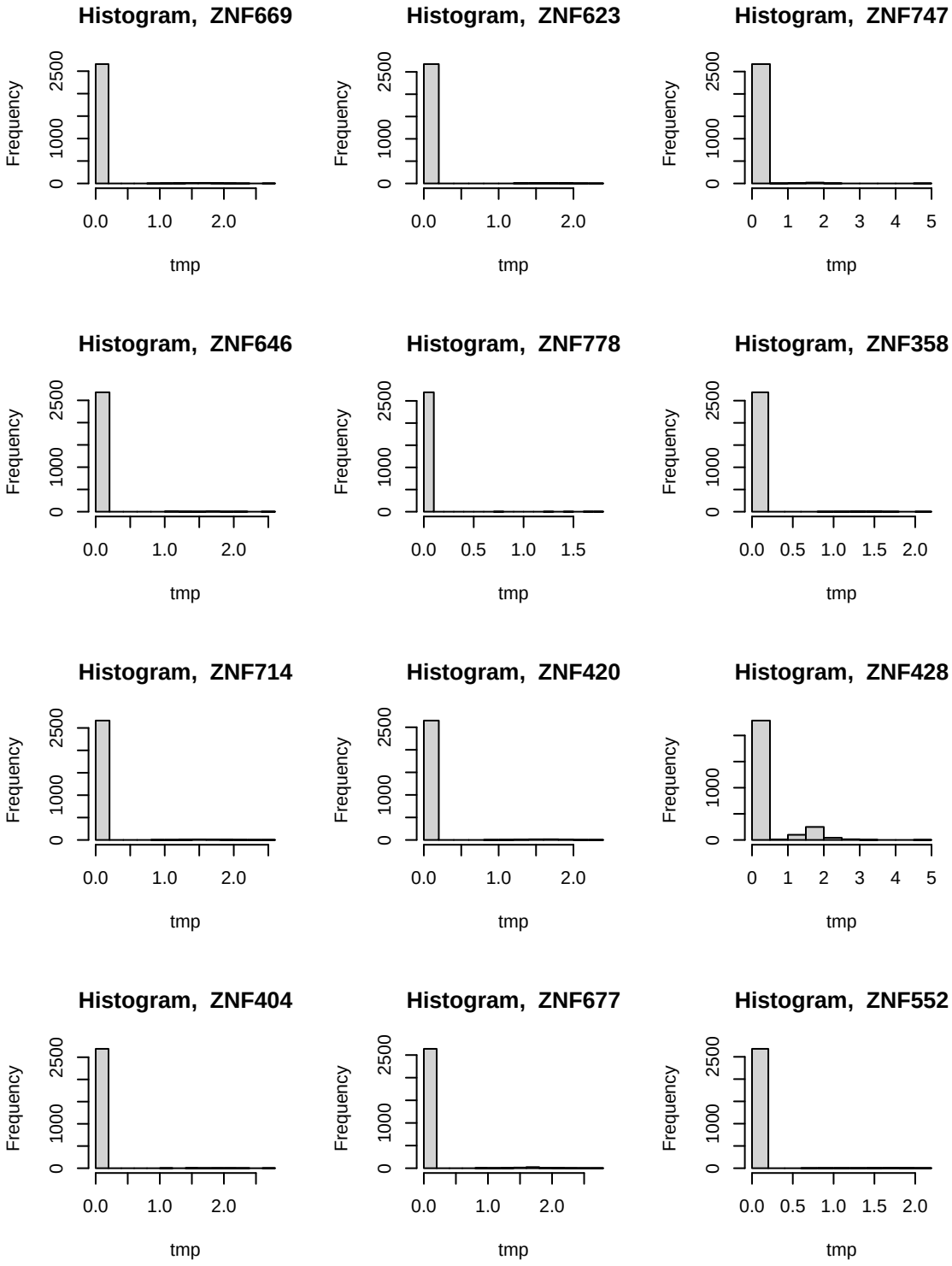



```
print(znf.sample <- sample(znf.genes,12))
```

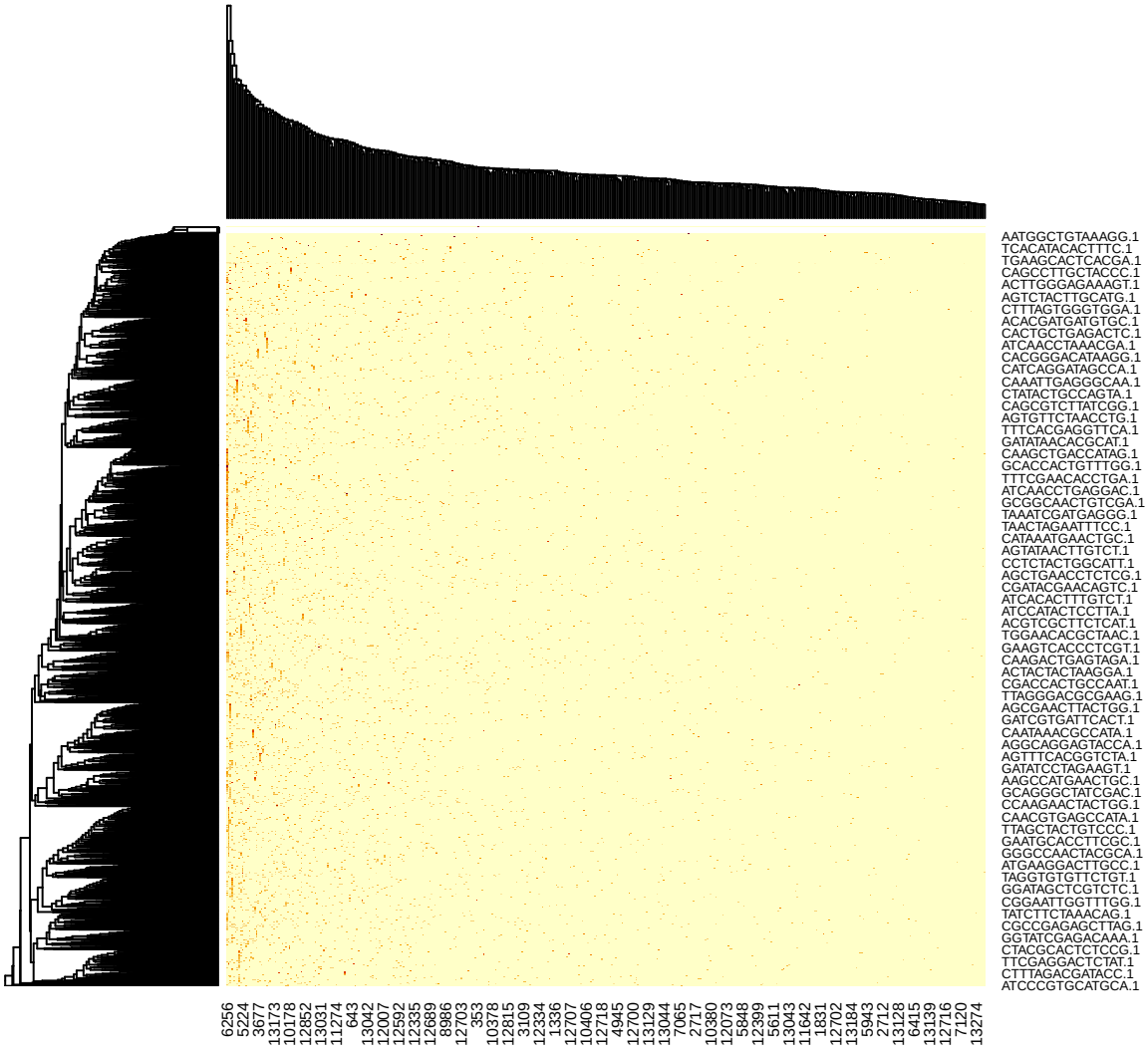
```
##  [1] "ZNF623" "ZNF747" "ZNF420" "ZNF428" "ZNF646" "ZNF358" "ZNF714" "ZNF669"
##  [9] "ZNF552" "ZNF404" "ZNF677" "ZNF778"
```

```
znf.mask <- genes %in% znf.sample
tmp.dat <- cell.dat[znf.mask,]
tmp.names <- genes[znf.mask]
par(mfrow=c(4,3))
```

```
for(i in 1:dim(tmp.dat)[1]) {
  tmp <- unlist(tmp.dat[i,])
  hist(tmp,main=paste("Histogram, ",tmp.names[i]))
}
```



```
heatmap(t(znf.dat))
```



PCA

A final approach to dimension reduction is to use principal component analysis to identify pseudovariables that capture variation in gene expression patterns. We present a PCA of the most commonly expressed genes identified in the previous step. PCA may be computationally intractable for the full data set.

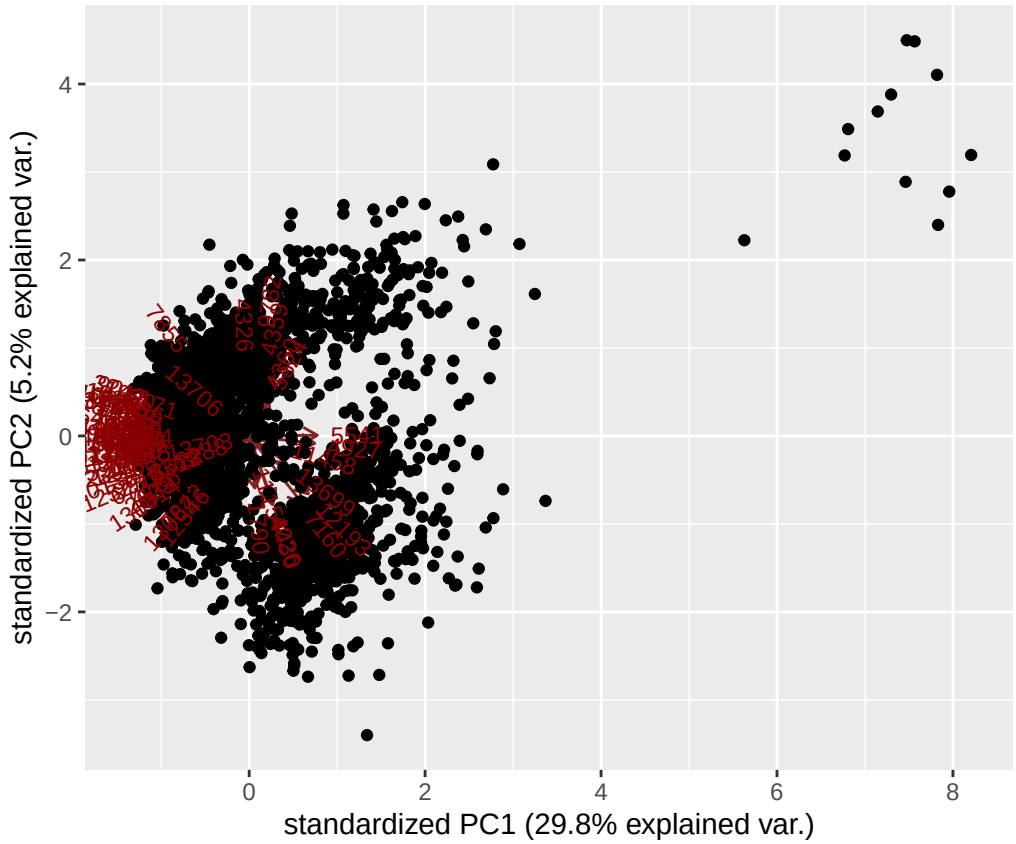
```
cell.pca <- prcomp(t(work.dat), center = TRUE,scale. = TRUE)
#summary(cell.pca)
```

A biplot suggests the two component may be discriminatory, but further components may not be discriminatory.

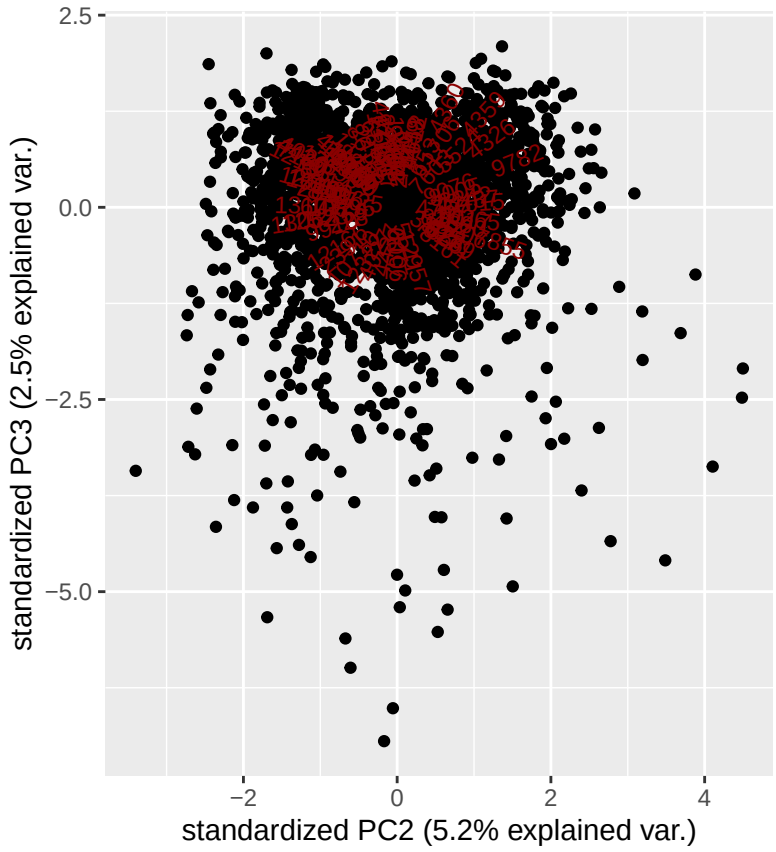
```
#library(devtools)
#install_github("vqv/ggbiplot", force = TRUE)

library(ggbiplot)
```

```
## Loading required package: ggplot2
## Loading required package: plyr
## Loading required package: scales
## Loading required package: grid
```

```
ggbiplot(cell.pca)
```




```
ggbiplot(cell.pca,choices=c(2,3))
```




```
col.median <- apply(cell.dat,2,median)
col.mean <- apply(cell.dat,2,mean)
dup <- col.median
hist(col.mean)
summary(col.median)
summary(col.mean)
keep.cols <- col.mean>0.2
work.dat <- cell.dat[,keep.cols]
dim(work.dat)
work.dat <- work.dat[,col.median>1]
bar.code <- bar.code[col.median>1]
```