007-Sensitivity Analysis of Many Paramters

Clustering Gene Expression Data

May 27, 2015

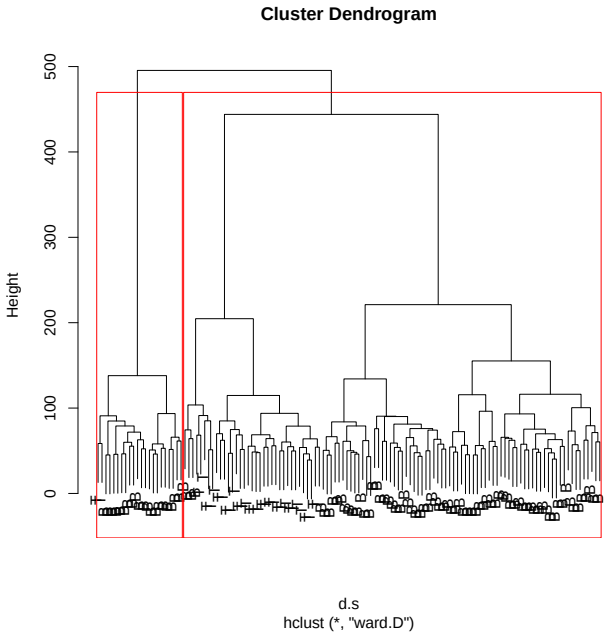
Abstract

DNA microarrays may be used to characterize the molecular variations among tumors by monitoring gene expression profiles on a genomic scale. This may lead to a finer and more reliable classification of tumors, and to the identification of marker genes that distinguish among these classes. Eventual clinical implications include an improved ability to understand and predict cancer survival (Dudoit and Gentleman, 2002). Therefore, a common task is to determine whether or not gene expression data can reliably identify or classify different types of a disease. We consider gene expression data from patients with acute lymphoblastic leukemia (ALL) that were investigated using HGU95AV2 Affymetrix GeneChip arrays (Chiaretti et al., 2004). The data consist of 128 patients with 12,625 genes. A number of additional covariates are available such as the type and stage of the disease; "B" indicates B-cell ALL, while a "T" indicates T-cell ALL. Several clustering procedures require user imputs such as the type of clustering and the number of clusters. Pre-filtering the data based on the most variable genes can also lead to increased power. We are interested in the effect these parameters have on the clustering results. Here I provide an illustration of performing such a task in an efficient and reproducible way using the function `knitr::knit_expand` (Xie, 2015, 2013, 2014) with the ALL dataset (Li, 2009).

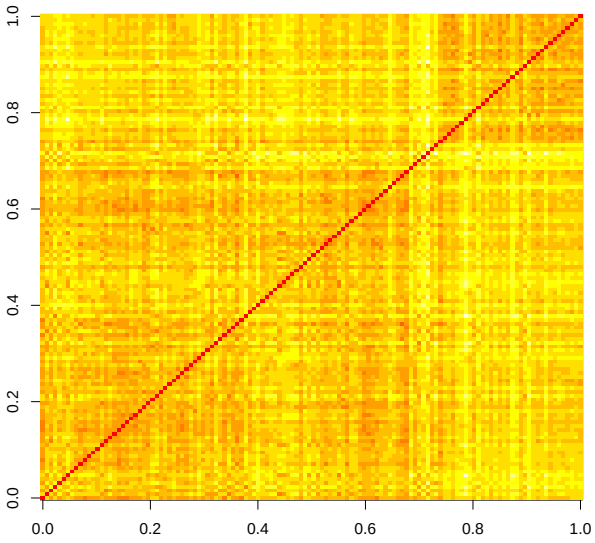
Contents

1 Method: ward.D, Filter: 10%, Groups: 2

```
dim(dat.filter)

## [1] 11362   128

table(groups, cl)

##       cl
## groups  B  T
##      1 75 31
##      2 20  2

fisher.test(groups, cl)$p.value

## [1] 0.061
```



(a) Dendogram

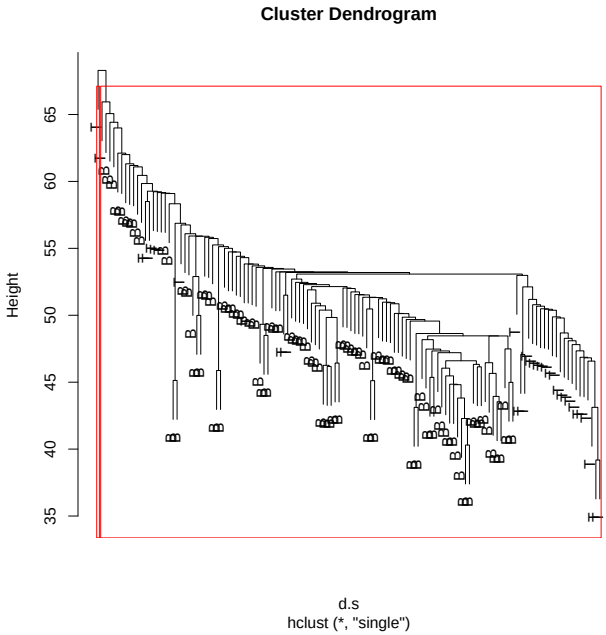


(b) Distance Matrix

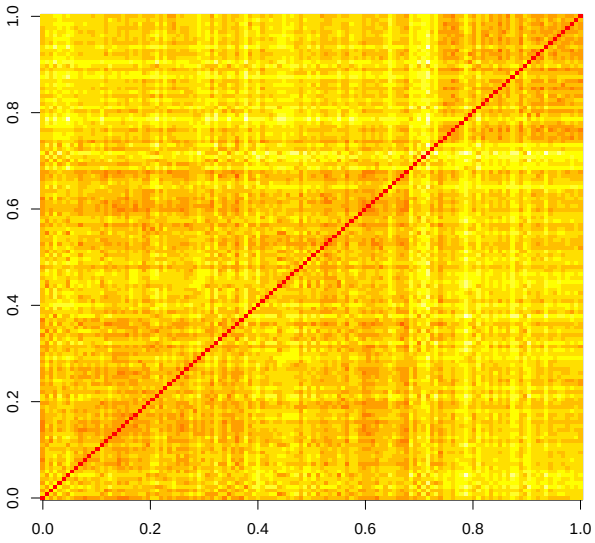
Figure 1: based on Method: ward.D, Filter: 10%, Groups: 2

2 Method: single, Filter: 10%, Groups: 2

```
dim(dat.filter)

## [1] 11362   128

table(groups, cl)

##       cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram

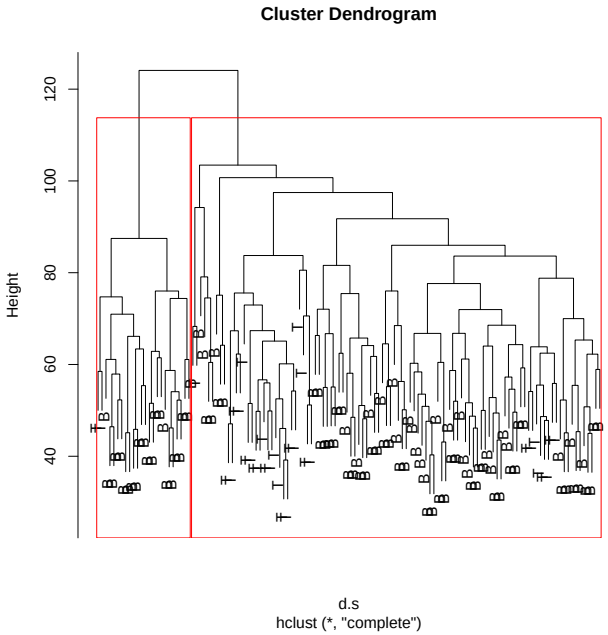


(b) Distance Matrix

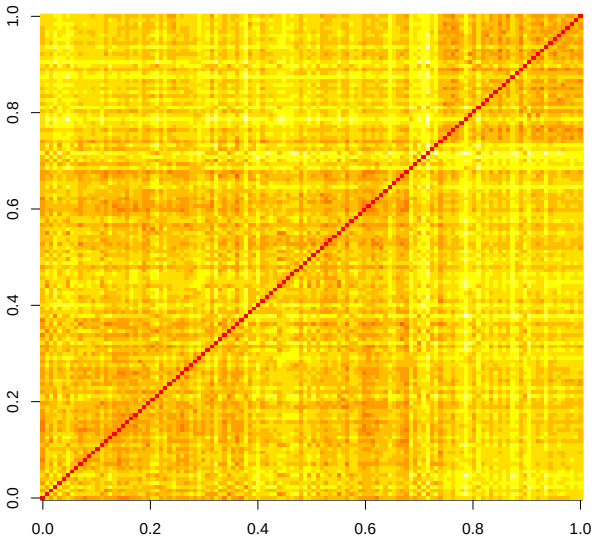
Figure 2: based on Method: single, Filter: 10%, Groups: 2

3 Method: complete, Filter: 10%, Groups: 2

```
dim(dat.filter)

## [1] 11362   128

table(groups, cl)

##       cl
## groups  B  T
##      1 73 31
##      2 22  2

fisher.test(groups, cl)$p.value

## [1] 0.037
```



(a) Dendogram

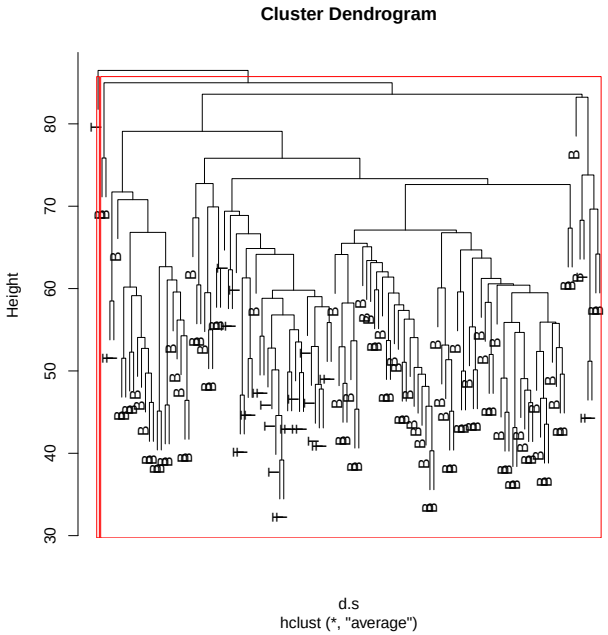


(b) Distance Matrix

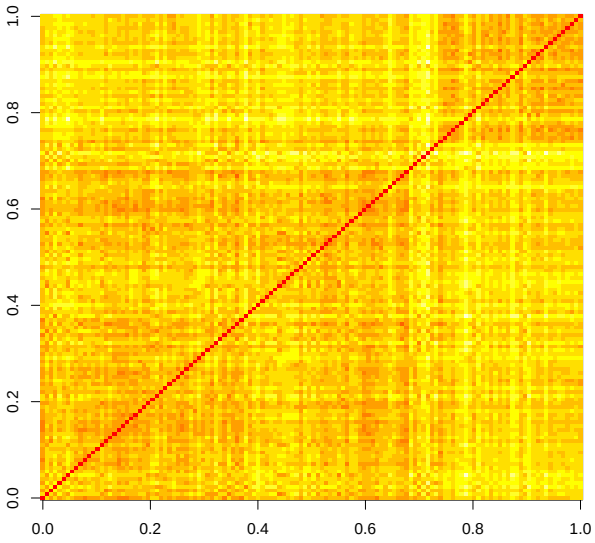
Figure 3: based on Method: complete, Filter: 10%, Groups: 2

4 Method: average, Filter: 10%, Groups: 2

```
dim(dat.filter)

## [1] 11362   128

table(groups, cl)

##       cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram



(b) Distance Matrix

Figure 4: based on Method: average, Filter: 10%, Groups: 2

5 Method: mcquitty, Filter: 10%, Groups: 2

```
dim(dat.filter)
```

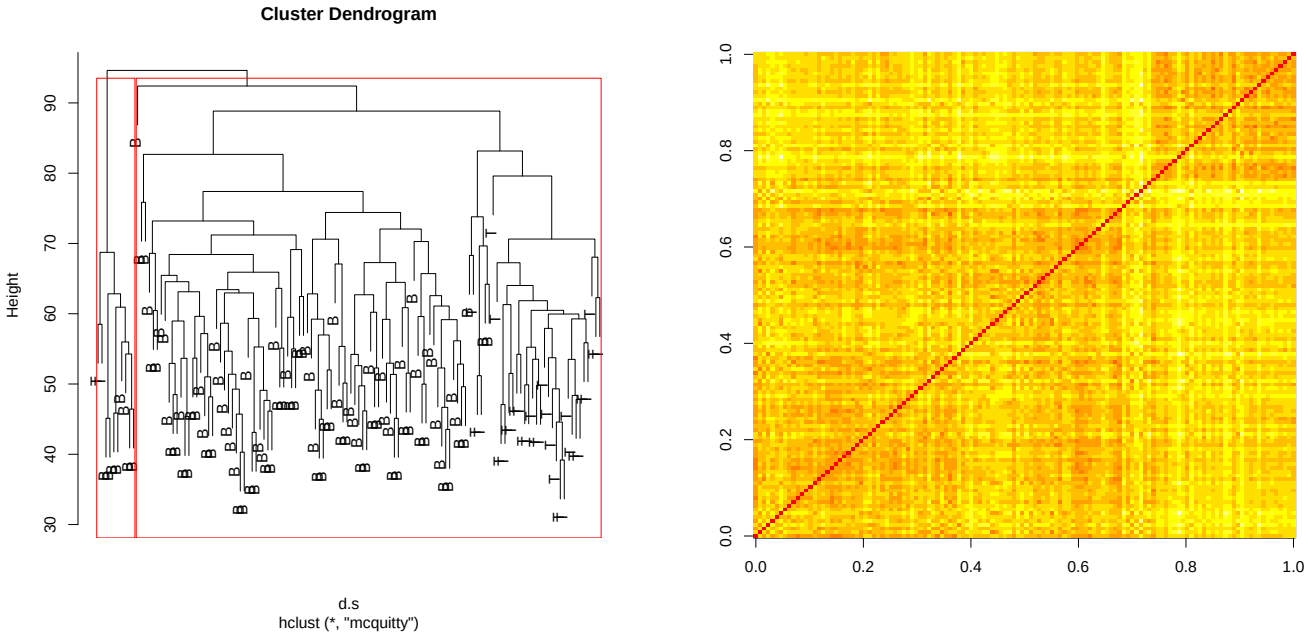
```
## [1] 11362   128
```

```
table(groups, cl)
```

```
##       cl
## groups  B  T
##      1 87 31
##      2  8  2
```

```
fisher.test(groups, cl)$p.value
```

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

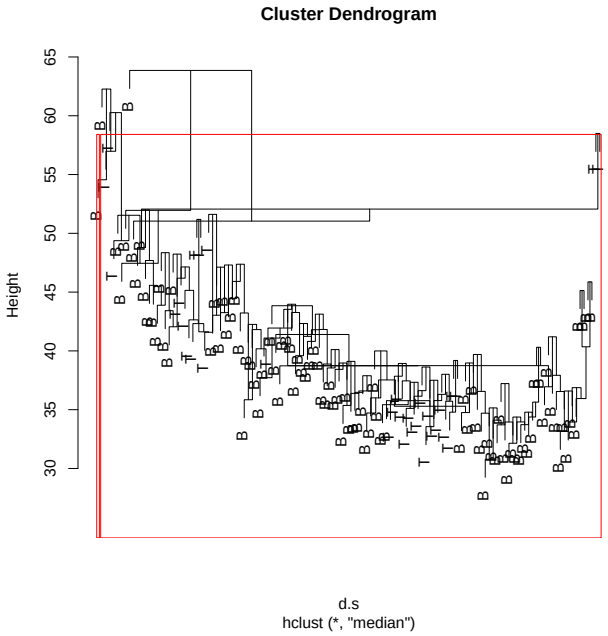


(a) Dendogram (b) Distance Matrix

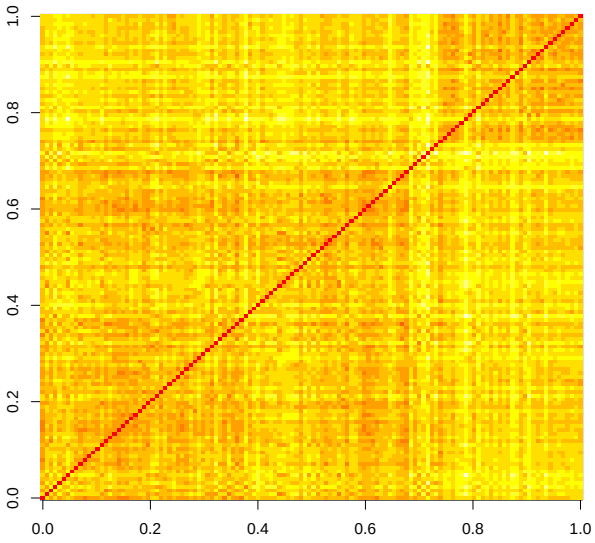
Figure 5: based on Method: mcquitty, Filter: 10%, Groups: 2

6 Method: median, Filter: 10%, Groups: 2

```
dim(dat.filter)

## [1] 11362   128

table(groups, cl)

##       cl
## groups  B  T
##      1 94 33
##      2  1  0

fisher.test(groups, cl)$p.value

## [1] 1
```



(a) Dendogram



(b) Distance Matrix

Figure 6: based on Method: median, Filter: 10%, Groups: 2

7 Method: centroid, Filter: 10%, Groups: 2

```
dim(dat.filter)
```

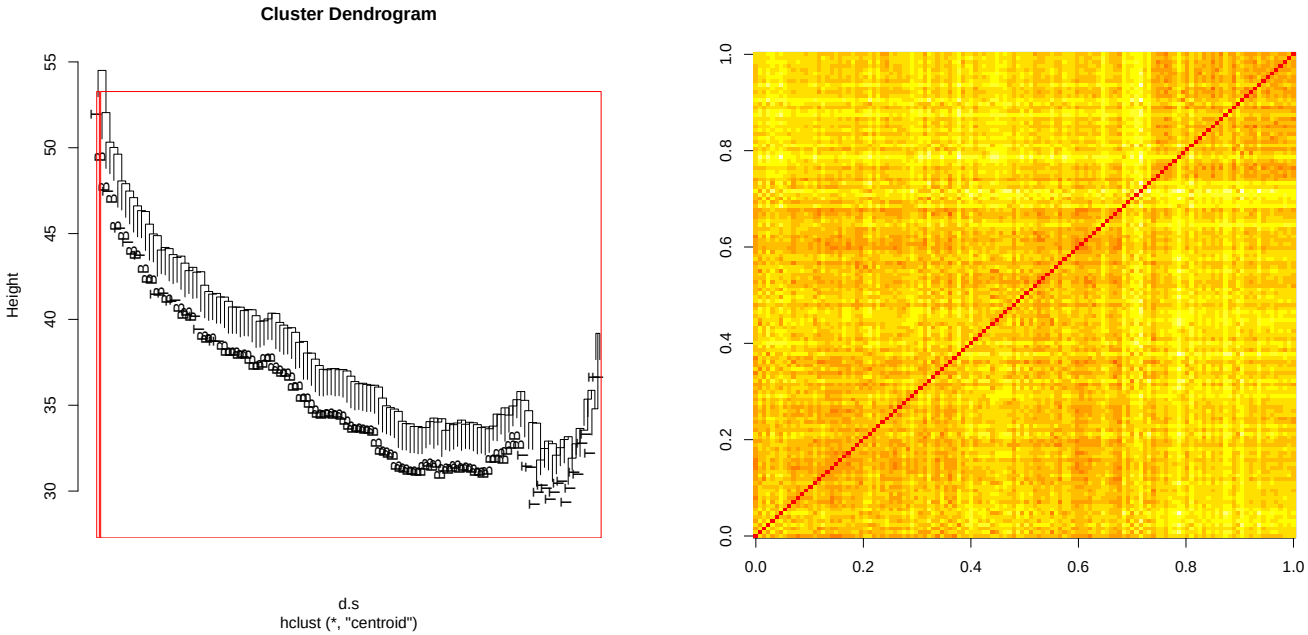
```
## [1] 11362   128
```

```
table(groups, cl)
```

```
##       cl
## groups  B  T
##      1 95 32
##      2  0  1
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 0.26
```

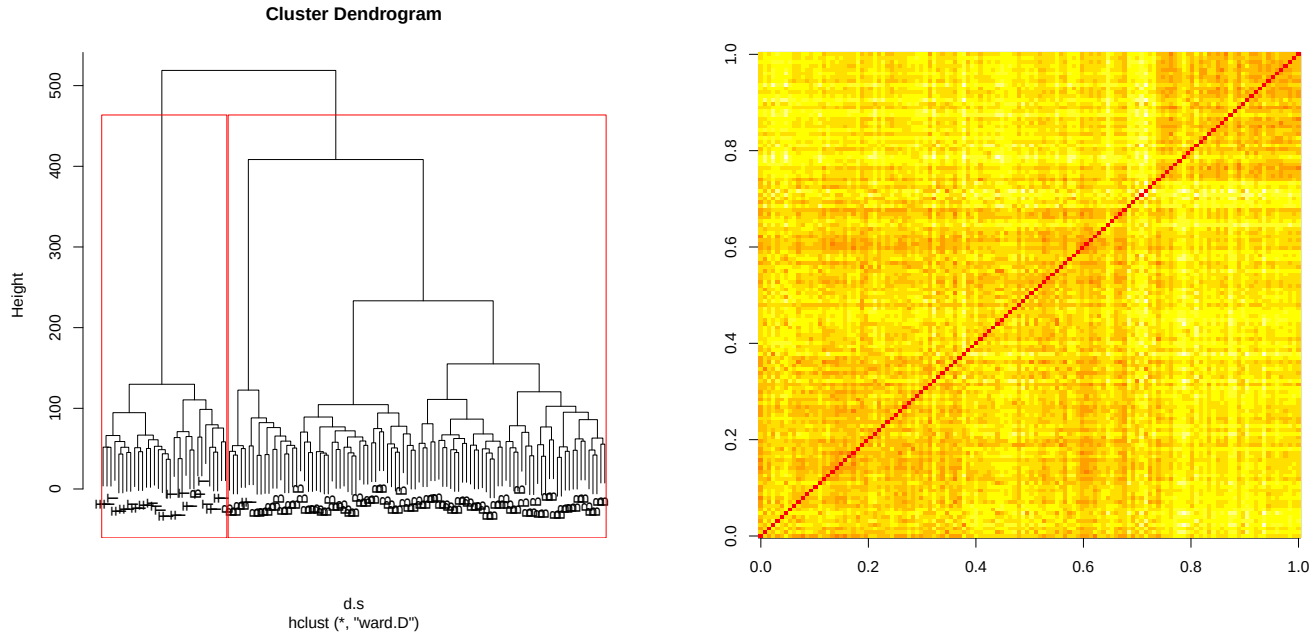


(a) Dendogram (b) Distance Matrix

Figure 7: based on Method: centroid, Filter: 10%, Groups: 2

8 Method: ward.D, Filter: 50%, Groups: 2

```
dim(dat.filter)

## [1] 6313  128

table(groups, cl)

##       cl
## groups  B  T
##      1 94  2
##      2  1 31

fisher.test(groups, cl)$p.value

## [1] 3.4e-26
```

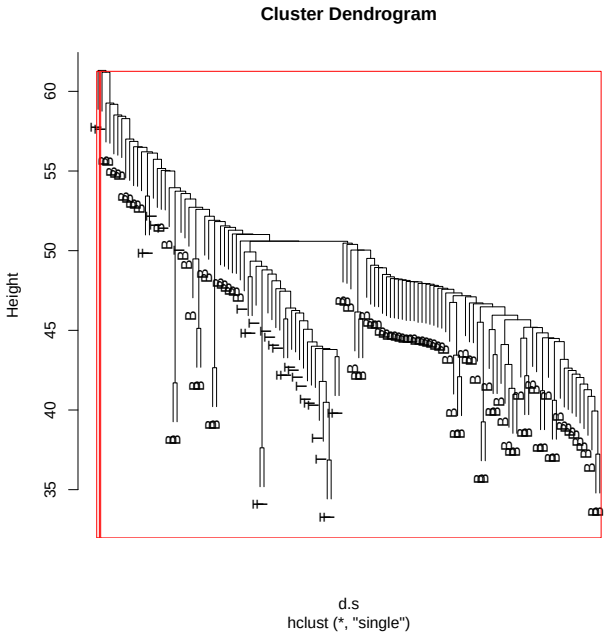


(a) Dendogram (b) Distance Matrix

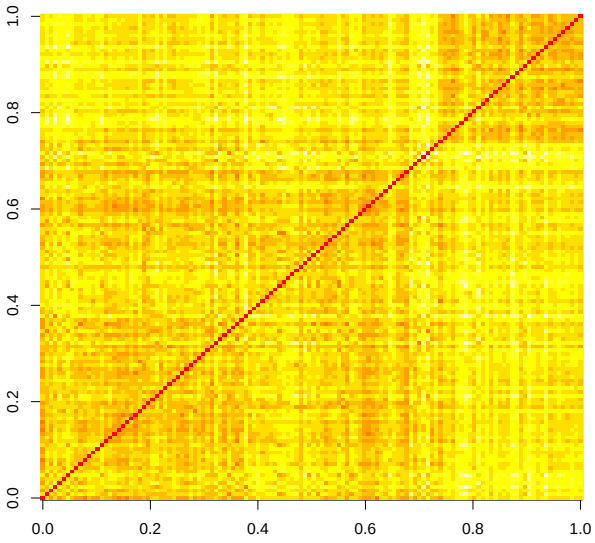
Figure 8: based on Method: ward.D, Filter: 50%, Groups: 2

9 Method: single, Filter: 50%, Groups: 2

```
dim(dat.filter)

## [1] 6313  128

table(groups, cl)

##        cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram



(b) Distance Matrix

Figure 9: based on Method: single, Filter: 50%, Groups: 2

10 Method: complete, Filter: 50%, Groups: 2

```
dim(dat.filter)
```

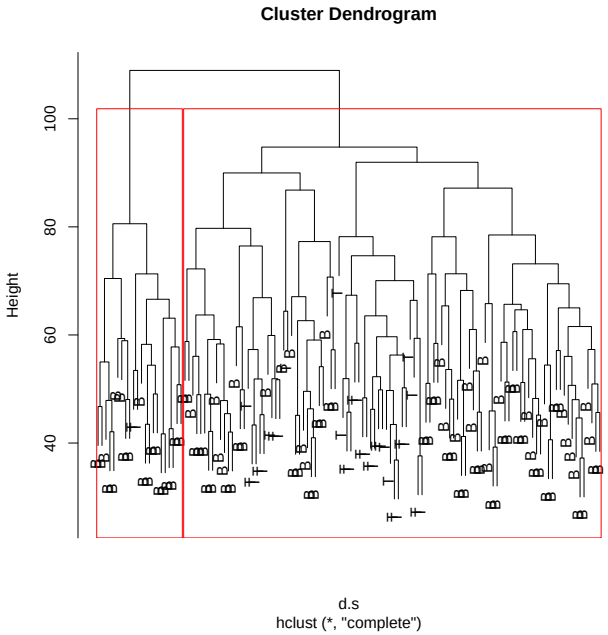
```
## [1] 6313  128
```

```
table(groups, cl)
```

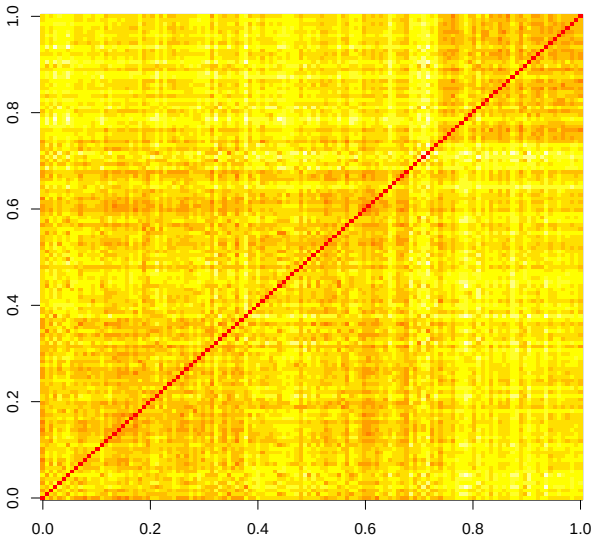
```
##       cl
## groups  B  T
##      1 75 31
##      2 20  2
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 0.061
```



(a) Dendogram



(b) Distance Matrix

Figure 10: based on Method: complete, Filter: 50%, Groups: 2

11 Method: average, Filter: 50%, Groups: 2

```
dim(dat.filter)
```

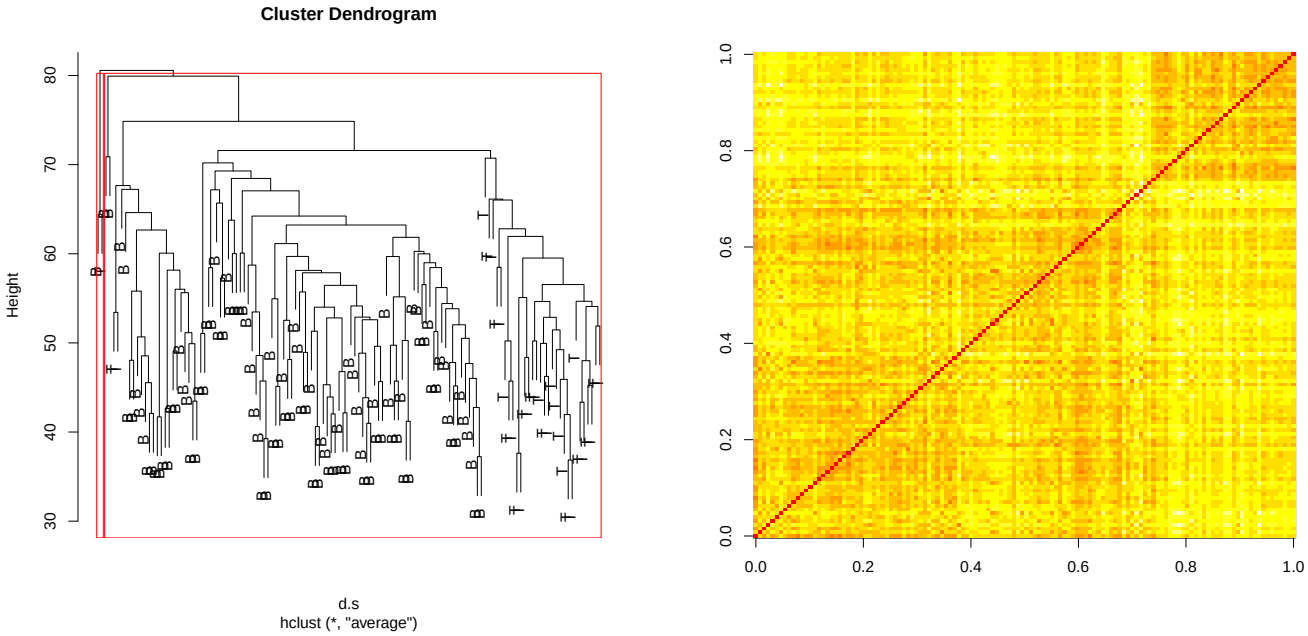
```
## [1] 6313  128
```

```
table(groups, cl)
```

```
##        cl
## groups  B  T
##      1 94 32
##      2  1  1
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 0.45
```

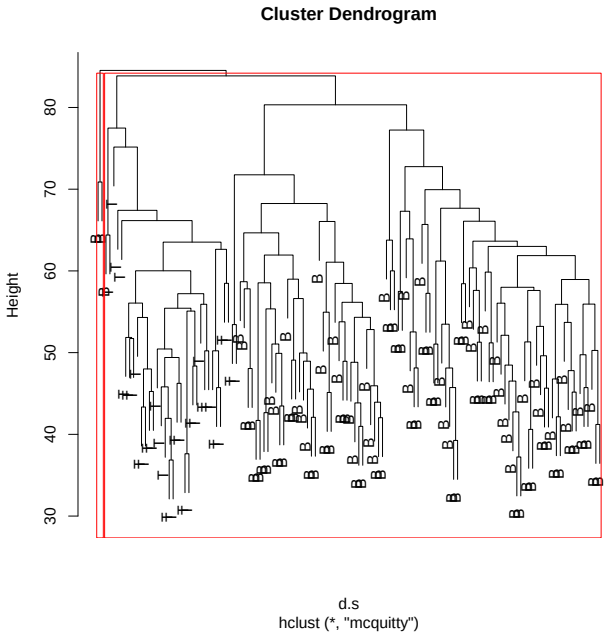


(a) Dendogram (b) Distance Matrix

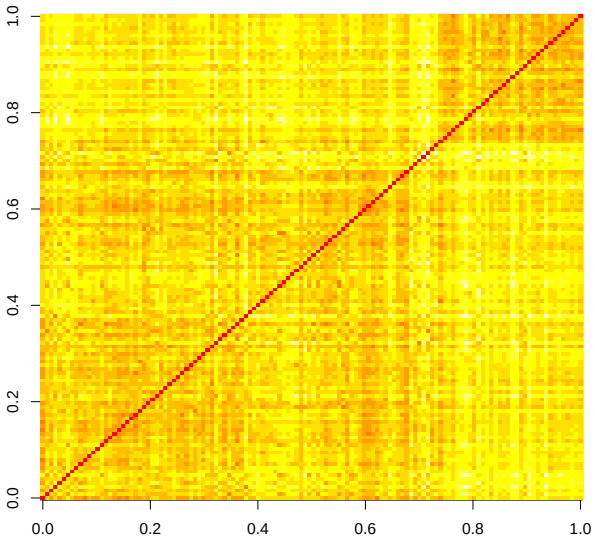
Figure 11: based on Method: average, Filter: 50%, Groups: 2

12 Method: mcquitty, Filter: 50%, Groups: 2

```
dim(dat.filter)

## [1] 6313  128

table(groups, cl)

##       cl
## groups  B  T
##      1 93 33
##      2  2  0

fisher.test(groups, cl)$p.value

## [1] 1
```



(a) Dendogram



(b) Distance Matrix

Figure 12: based on Method: mcquitty, Filter: 50%, Groups: 2

13 Method: median, Filter: 50%, Groups: 2

```
dim(dat.filter)
```

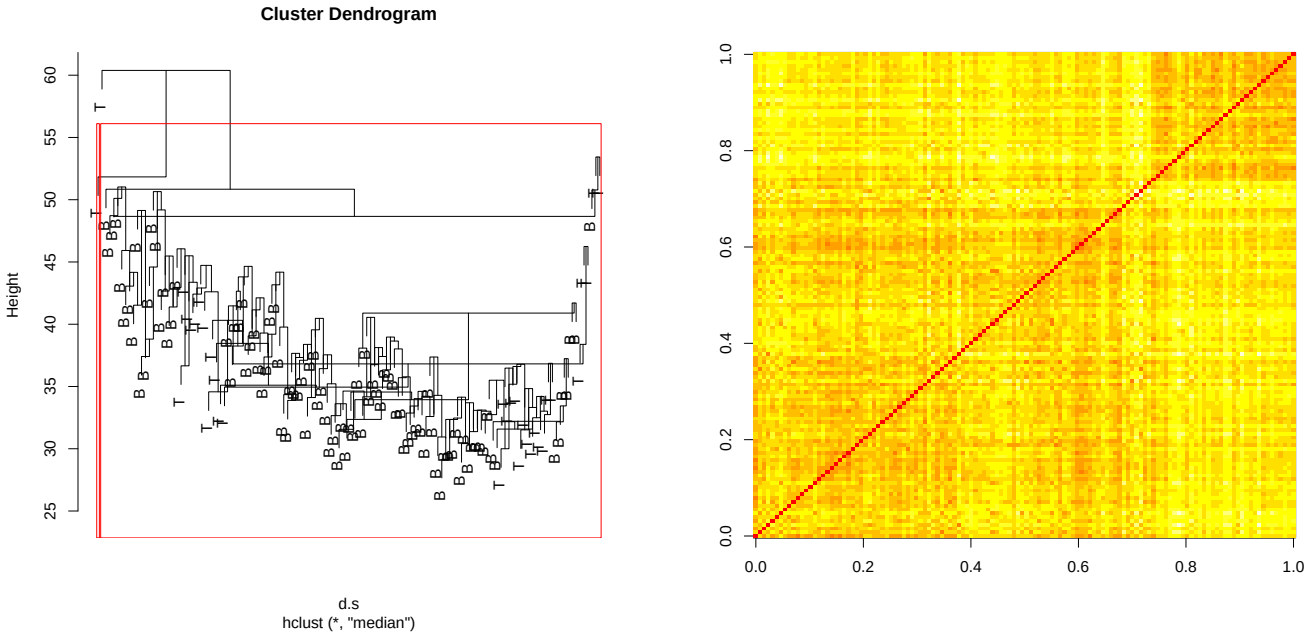
```
## [1] 6313  128
```

```
table(groups, cl)
```

```
##        cl
## groups  B  T
##      1 95 32
##      2  0  1
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 0.26
```

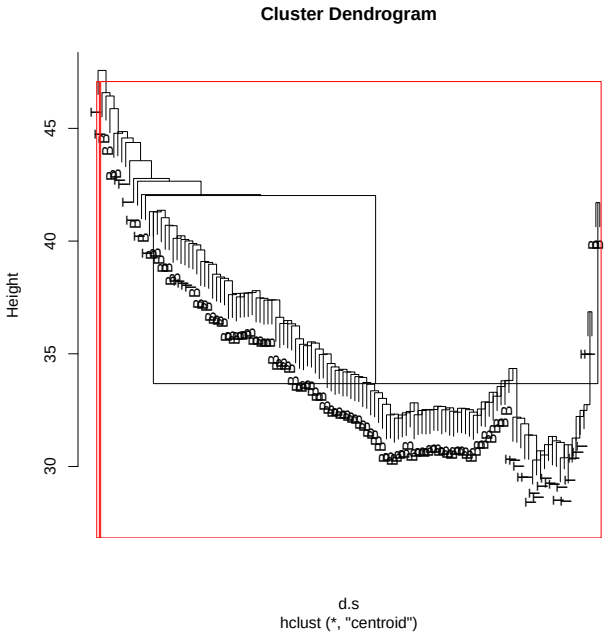


(a) Dendogram (b) Distance Matrix

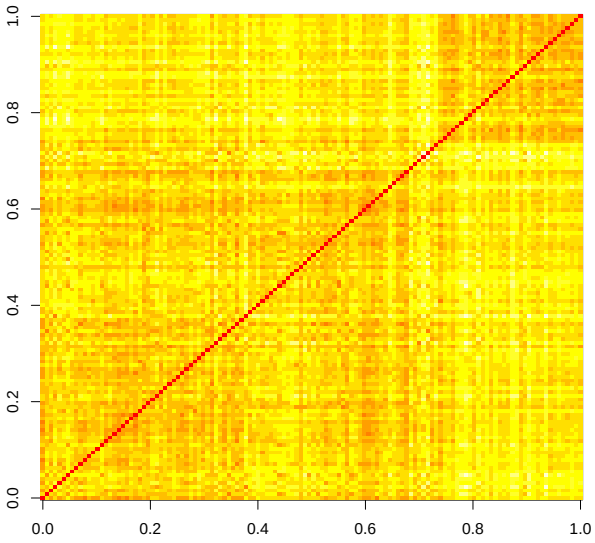
Figure 13: based on Method: median, Filter: 50%, Groups: 2

14 Method: centroid, Filter: 50%, Groups: 2

```
dim(dat.filter)

## [1] 6313  128

table(groups, cl)

##       cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram

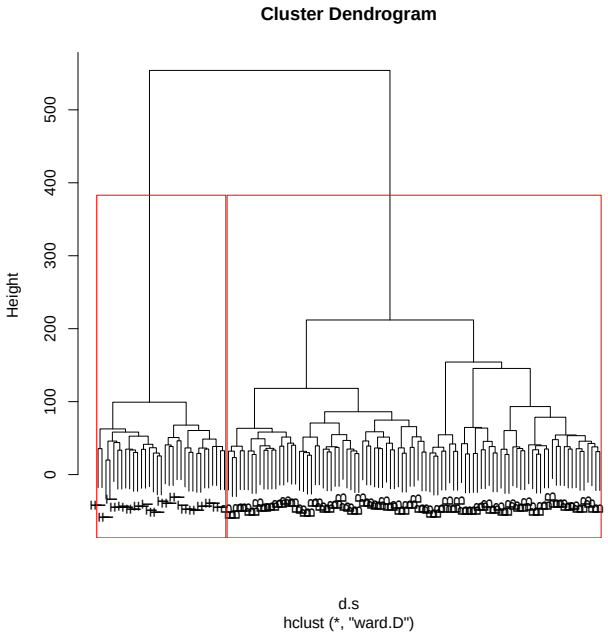


(b) Distance Matrix

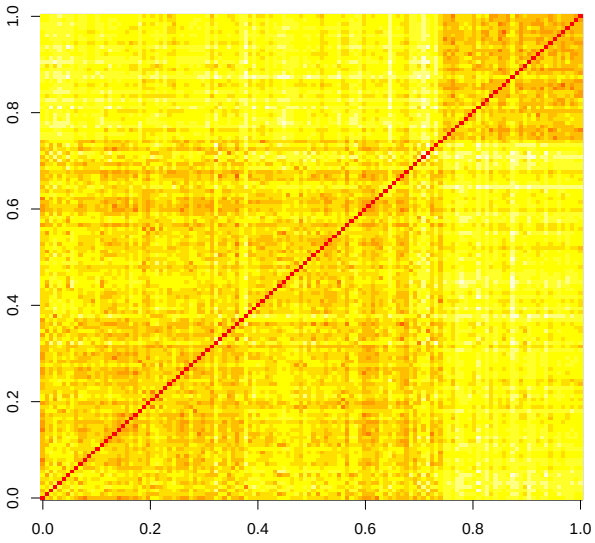
Figure 14: based on Method: centroid, Filter: 50%, Groups: 2

15 Method: ward.D, Filter: 90%, Groups: 2

```
dim(dat.filter)

## [1] 1263  128

table(groups, cl)

##       cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 15: based on Method: ward.D, Filter: 90%, Groups: 2

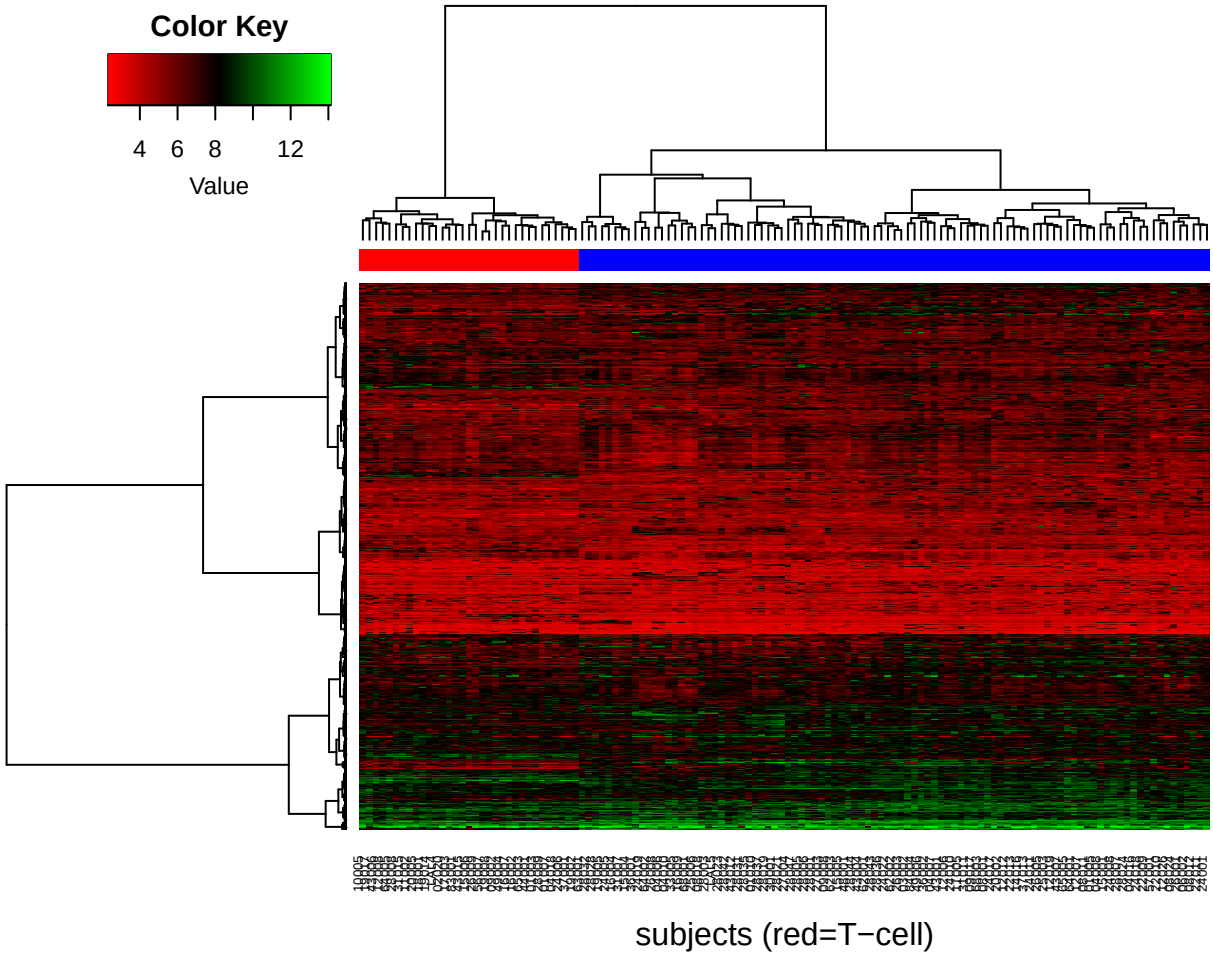


Figure 16: Heatmap of Gene expression values for genes that survived a filter of 90% and the ward.D clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

16 Method: single, Filter: 90%, Groups: 2

```
dim(dat.filter)
```

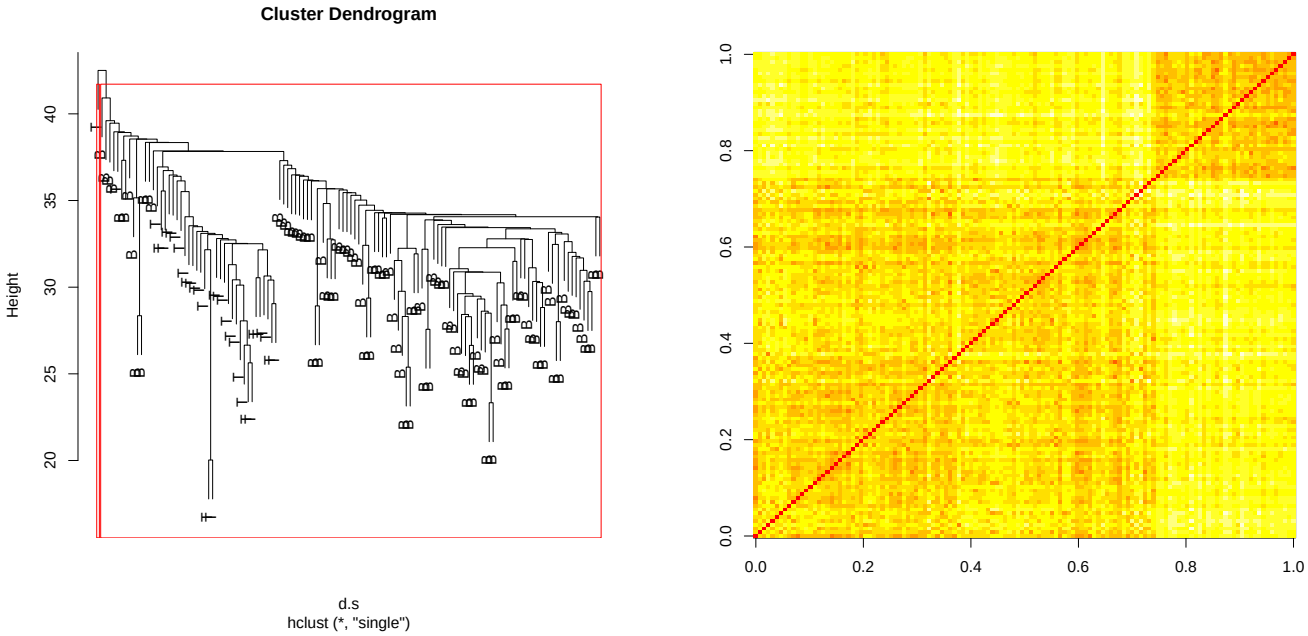
```
## [1] 1263  128
```

```
table(groups, cl)
```

```
##        cl
## groups  B  T
##      1 95 32
##      2  0  1
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 0.26
```



(a) Dendogram (b) Distance Matrix

Figure 17: based on Method: single, Filter: 90%, Groups: 2

```
## Error in gplots::heatmap.2(dat.filter, density.info = "none", hclustfun = function(x) hclust(x,
:  Row dendrogram too deeply nested, recursion limit exceeded.  Try increasing option("expressions
```

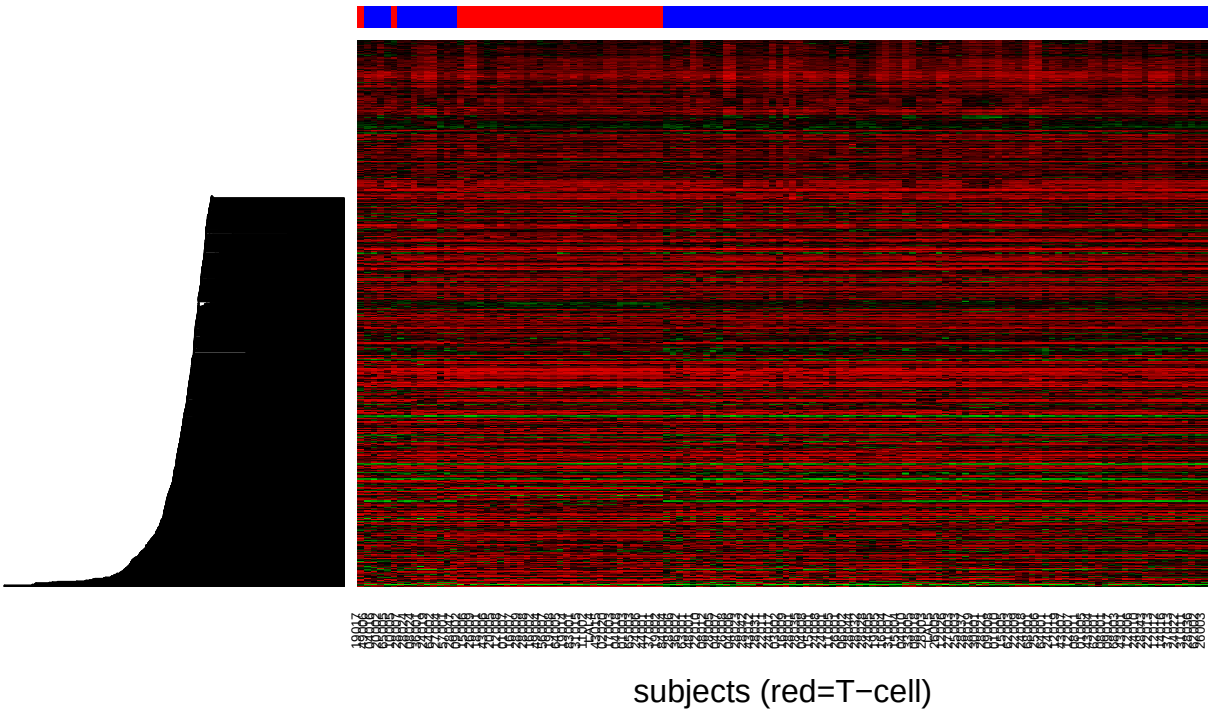
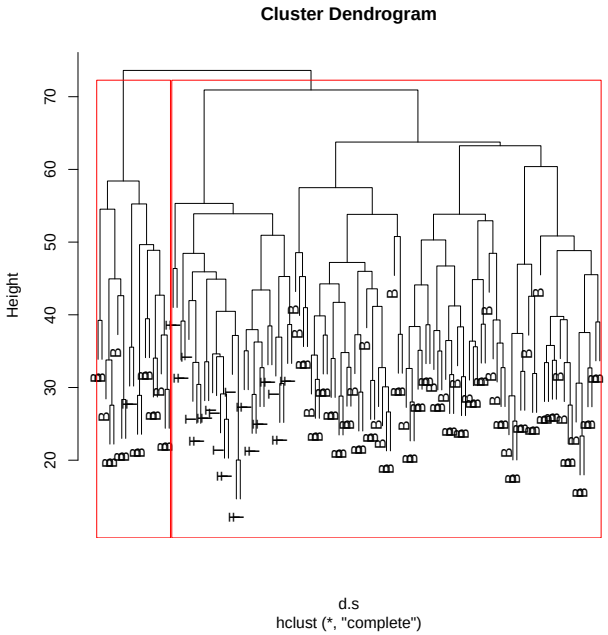


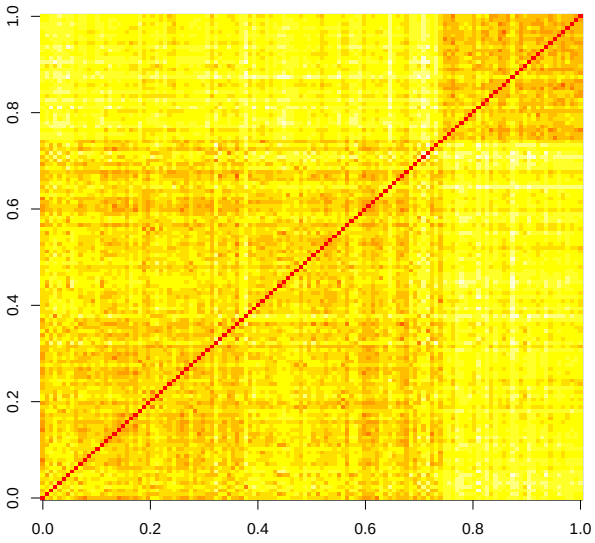
Figure 18: Heatmap of Gene expression values for genes that survived a filter of 90% and the single clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

17 Method: complete, Filter: 90%, Groups: 2

```
dim(dat.filter)

## [1] 1263  128

table(groups, cl)

##       cl
## groups  B  T
##      1 78 31
##      2 17  2

fisher.test(groups, cl)$p.value

## [1] 0.15
```



(a) Dendogram



(b) Distance Matrix

Figure 19: based on Method: complete, Filter: 90%, Groups: 2

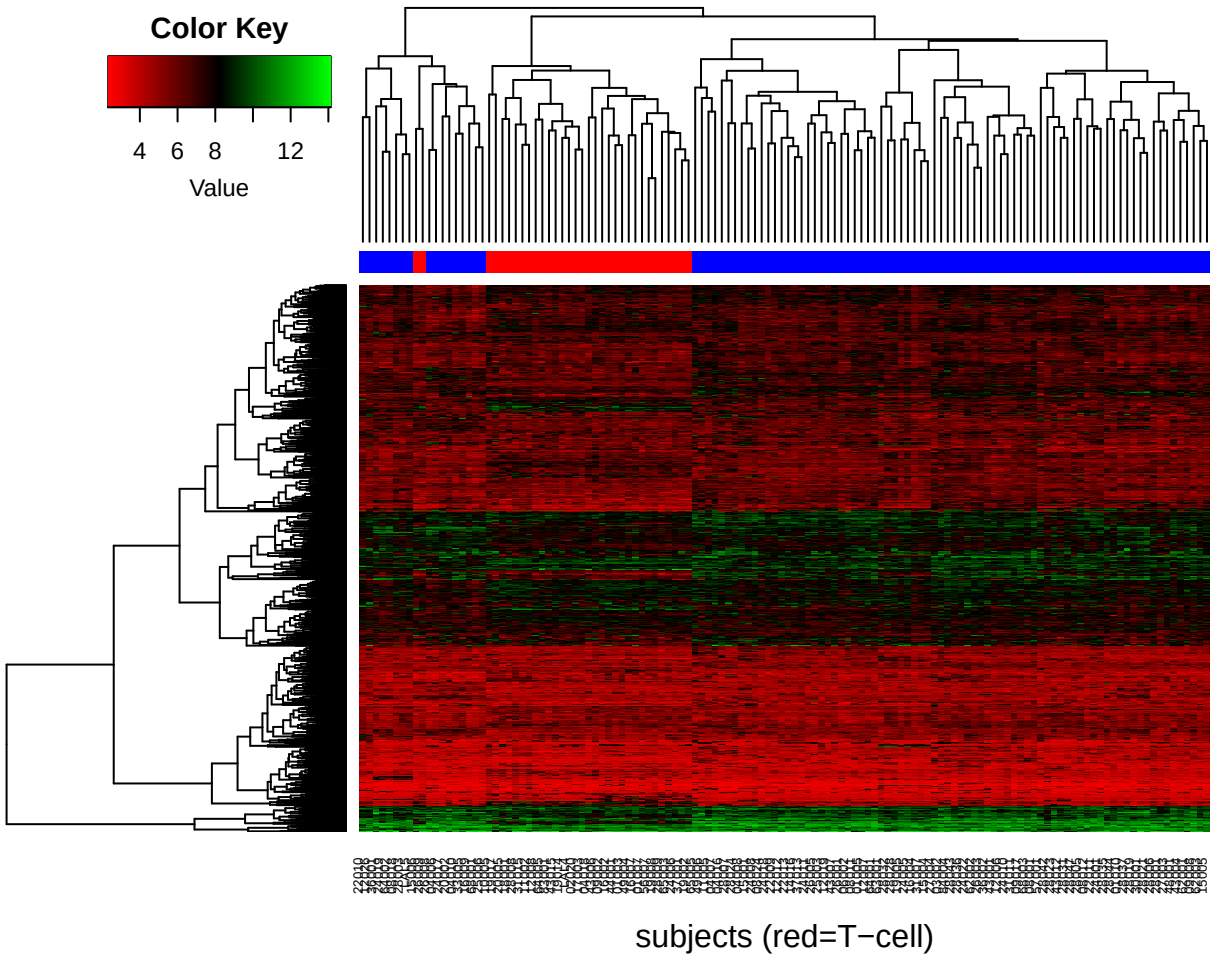
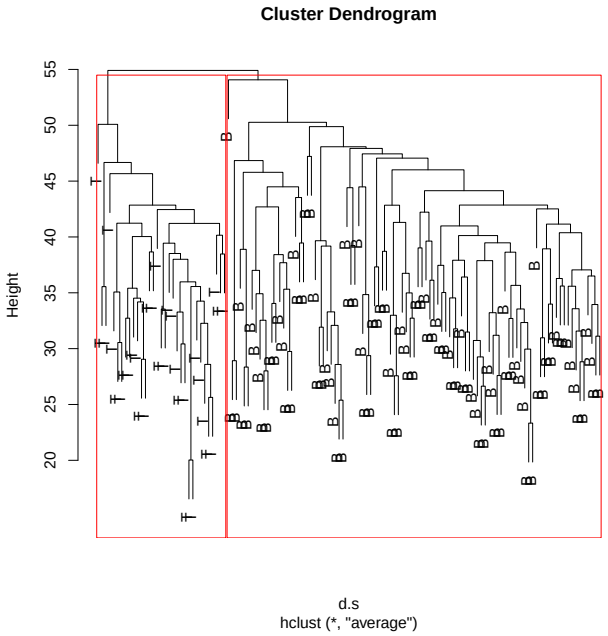


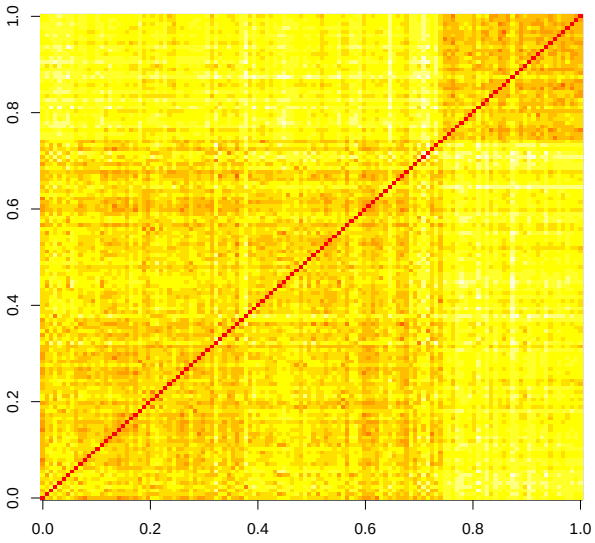
Figure 20: Heatmap of Gene expression values for genes that survived a filter of 90% and the complete clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

18 Method: average, Filter: 90%, Groups: 2

```
dim(dat.filter)

## [1] 1263  128

table(groups, cl)

##        cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 21: based on Method: average, Filter: 90%, Groups: 2

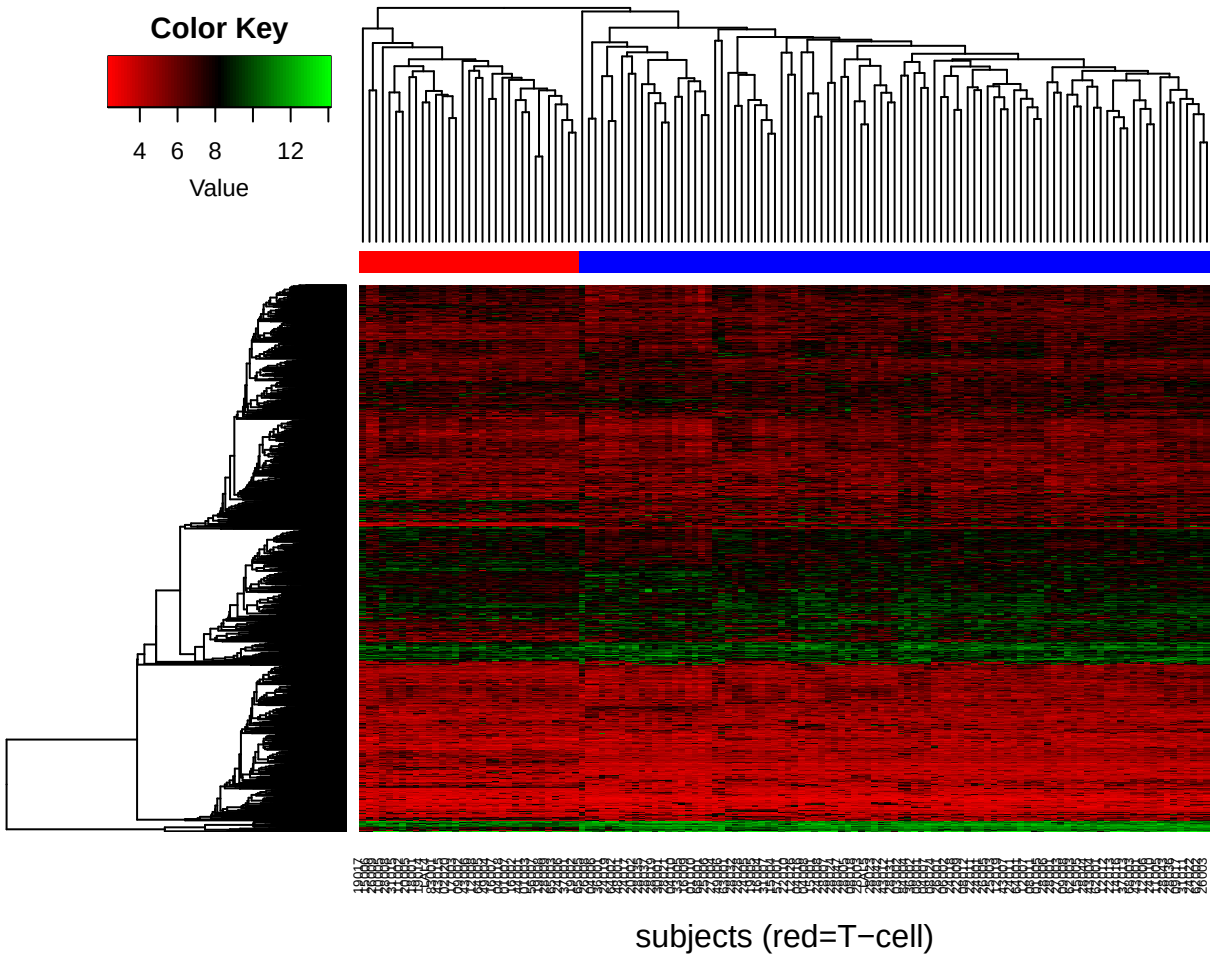


Figure 22: Heatmap of Gene expression values for genes that survived a filter of 90% and the average clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

19 Method: mcquitty, Filter: 90%, Groups: 2

```
dim(dat.filter)
```

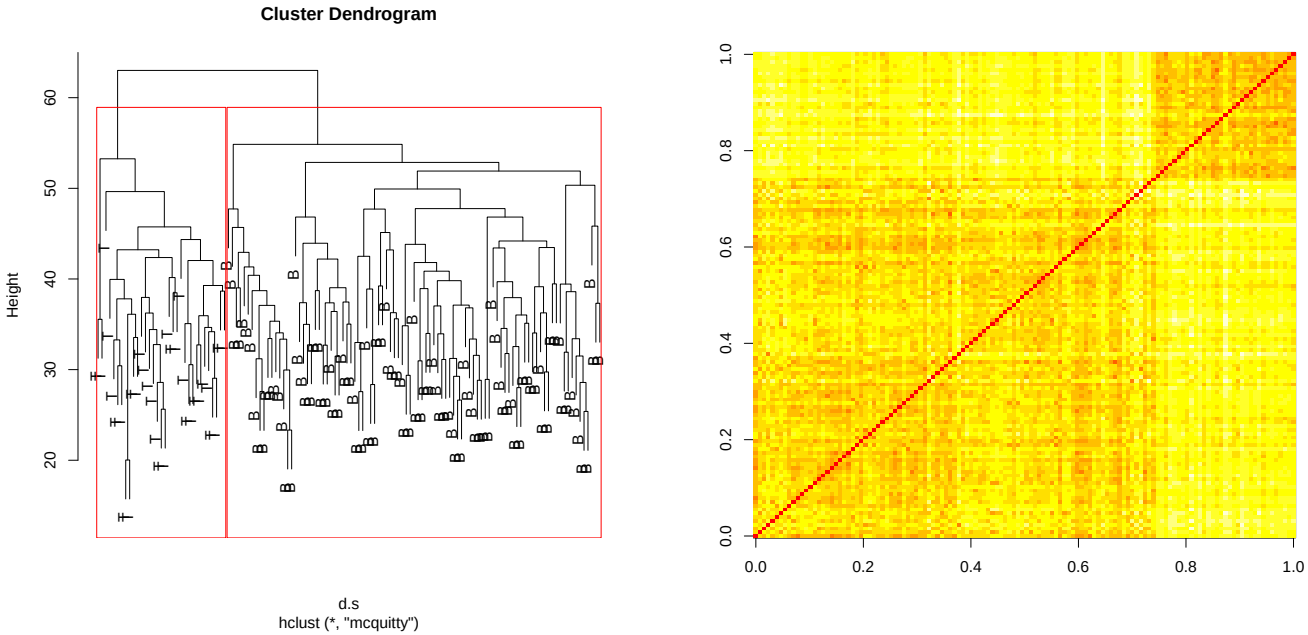
```
## [1] 1263  128
```

```
table(groups, cl)
```

```
##        cl
## groups  B  T
##      1 95  0
##      2  0 33
```

```
fisher.test(groups, cl)$p.value
```

```
## [1] 2.3e-31
```



(a) Dendogram (b) Distance Matrix

Figure 23: based on Method: mcquitty, Filter: 90%, Groups: 2

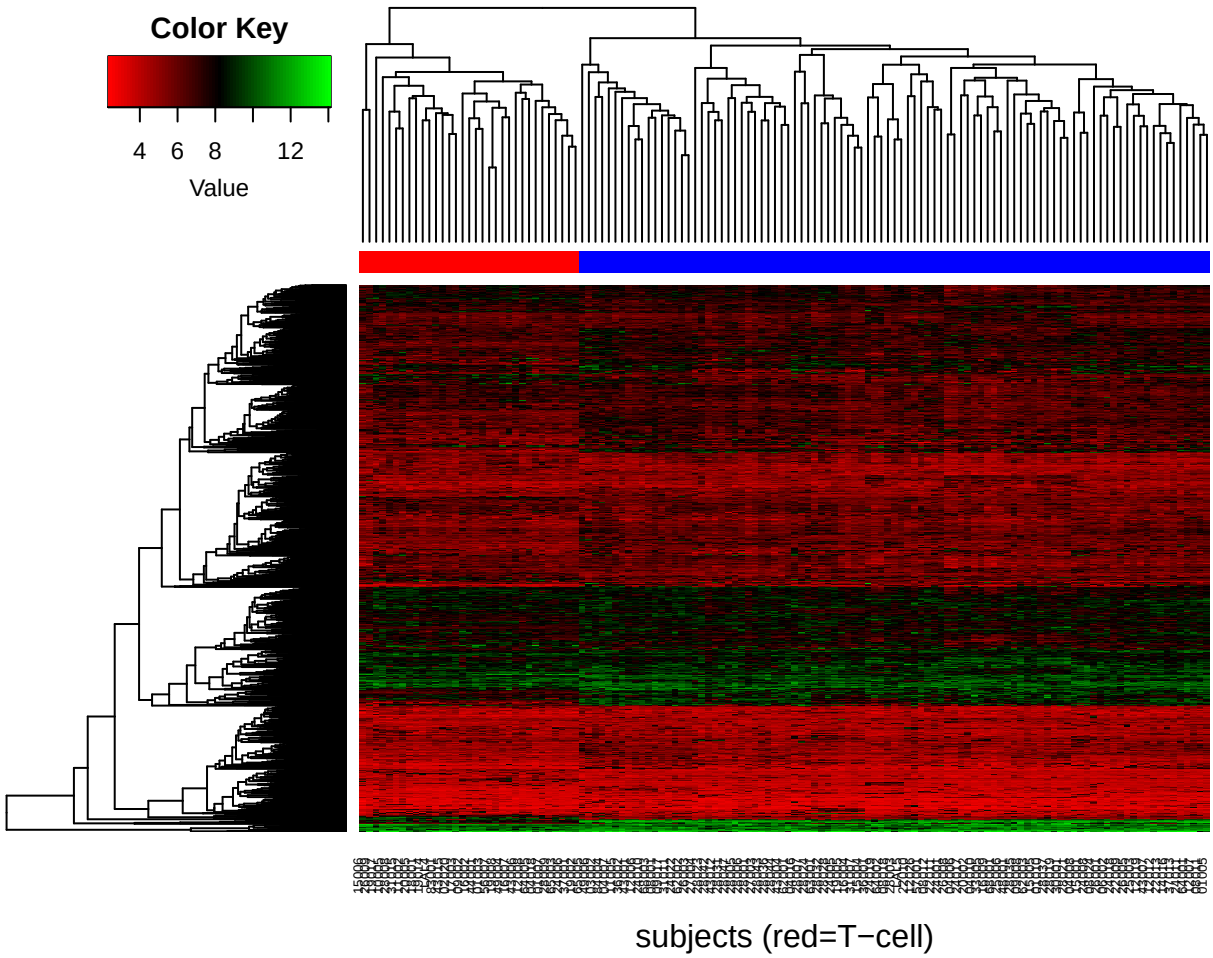


Figure 24: Heatmap of Gene expression values for genes that survived a filter of 90% and the mcquitty clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

20 Method: median, Filter: 90%, Groups: 2

```
dim(dat.filter)
```

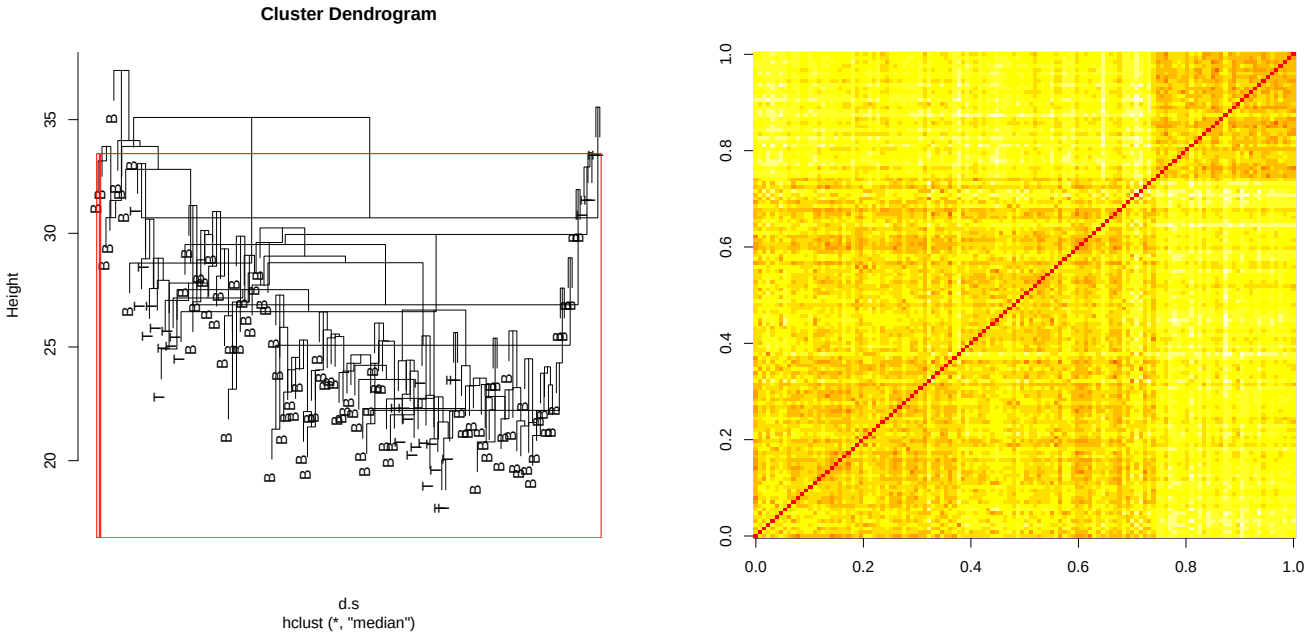
```
## [1] 1263  128
```

```
table(groups, cl)
```

```
##       cl
## groups  B  T
##      1 94 33
##      2  1  0
```

```
fisher.test(groups, cl)$p.value
```

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



(a) Dendogram (b) Distance Matrix

Figure 25: based on Method: median, Filter: 90%, Groups: 2

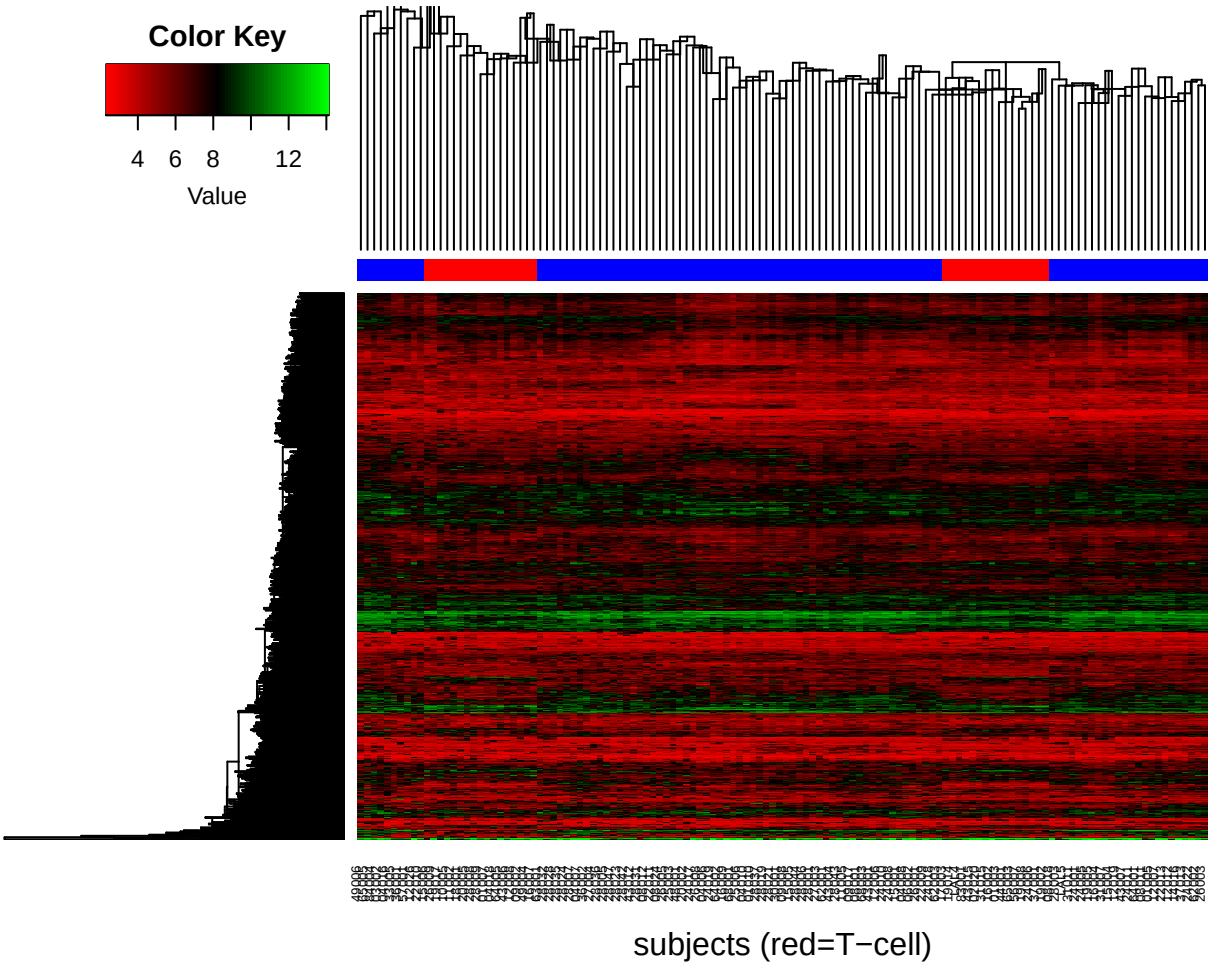
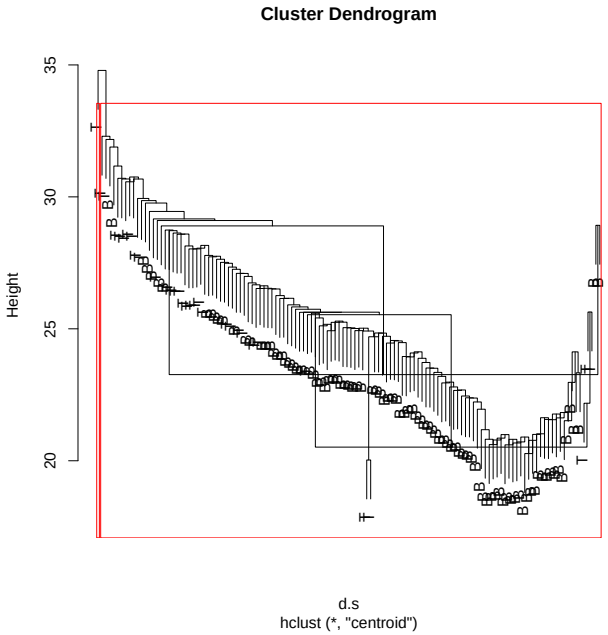


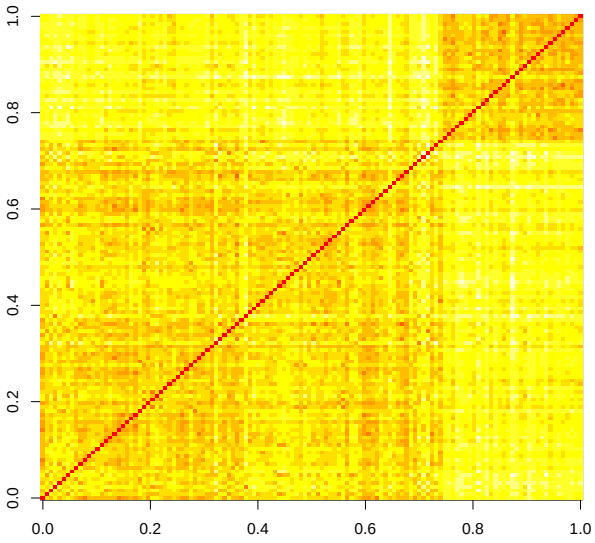
Figure 26: Heatmap of Gene expression values for genes that survived a filter of 90% and the median clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

21 Method: centroid, Filter: 90%, Groups: 2

```
dim(dat.filter)

## [1] 1263  128

table(groups, cl)

##       cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram



(b) Distance Matrix

Figure 27: based on Method: centroid, Filter: 90%, Groups: 2

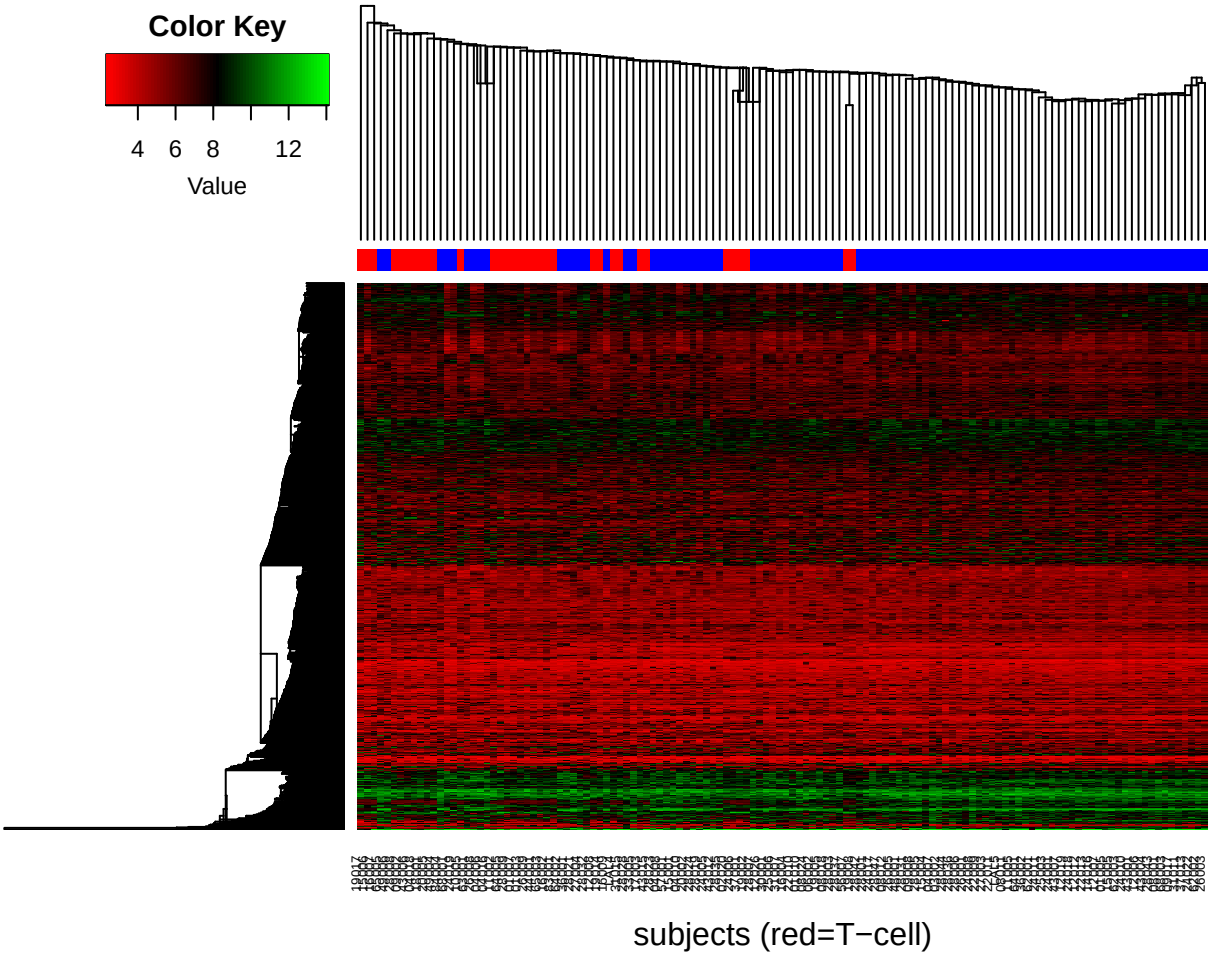
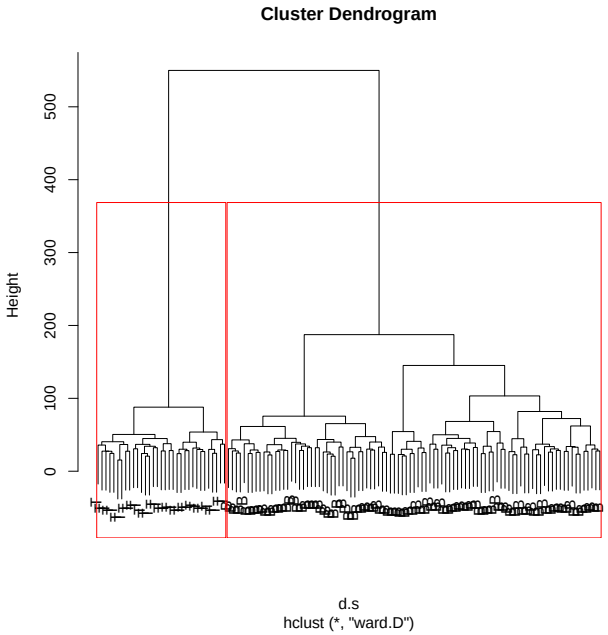


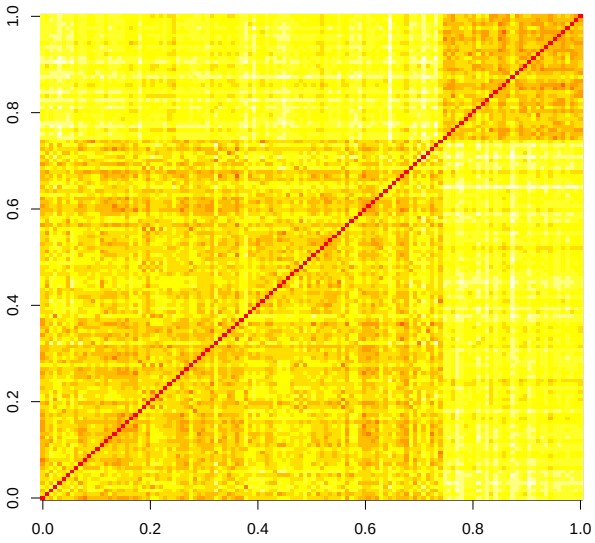
Figure 28: Heatmap of Gene expression values for genes that survived a filter of 90% and the centroid clustering algoritm. Rows are genes and columns are subjects. There are a total of 1263 genes and 128 subjects in this plot.

22 Method: ward.D, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 29: based on Method: ward.D, Filter: 95%, Groups: 2

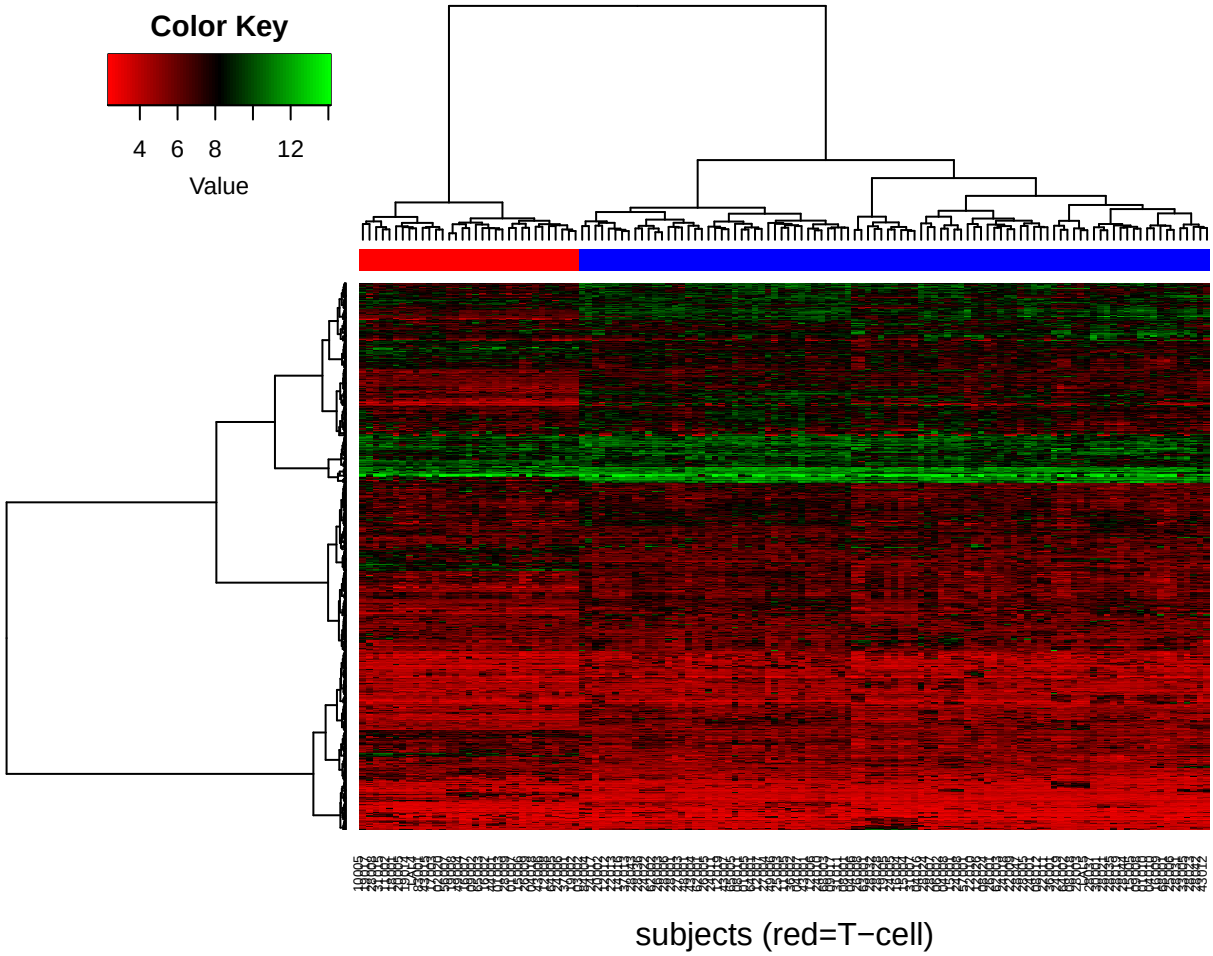
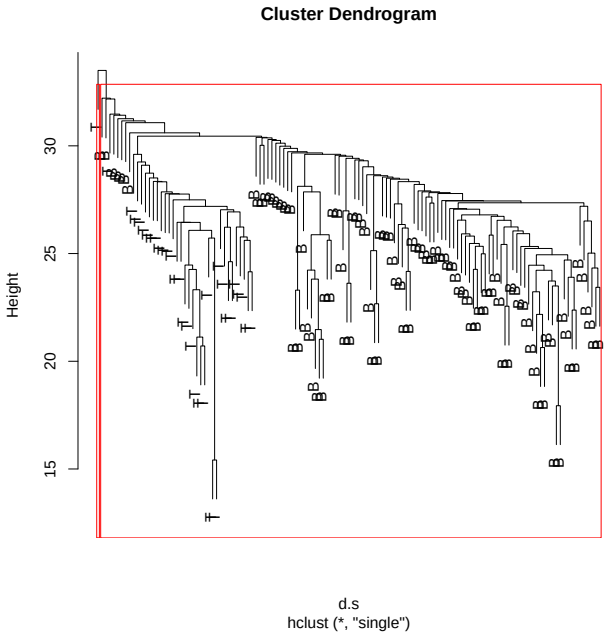


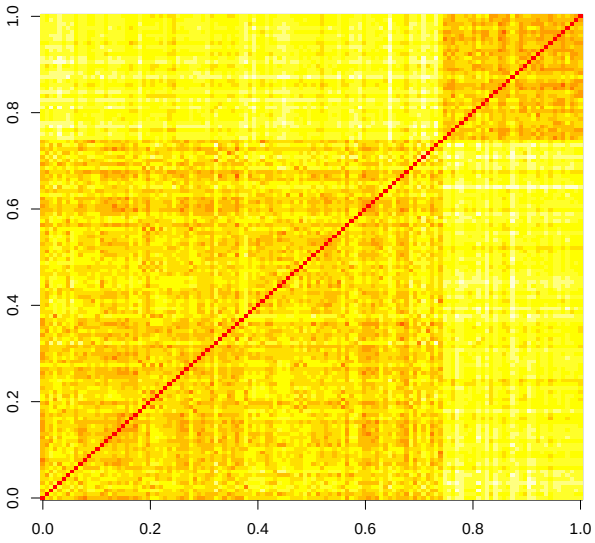
Figure 30: Heatmap of Gene expression values for genes that survived a filter of 95% and the ward.D clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

23 Method: single, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram



(b) Distance Matrix

Figure 31: based on Method: single, Filter: 95%, Groups: 2

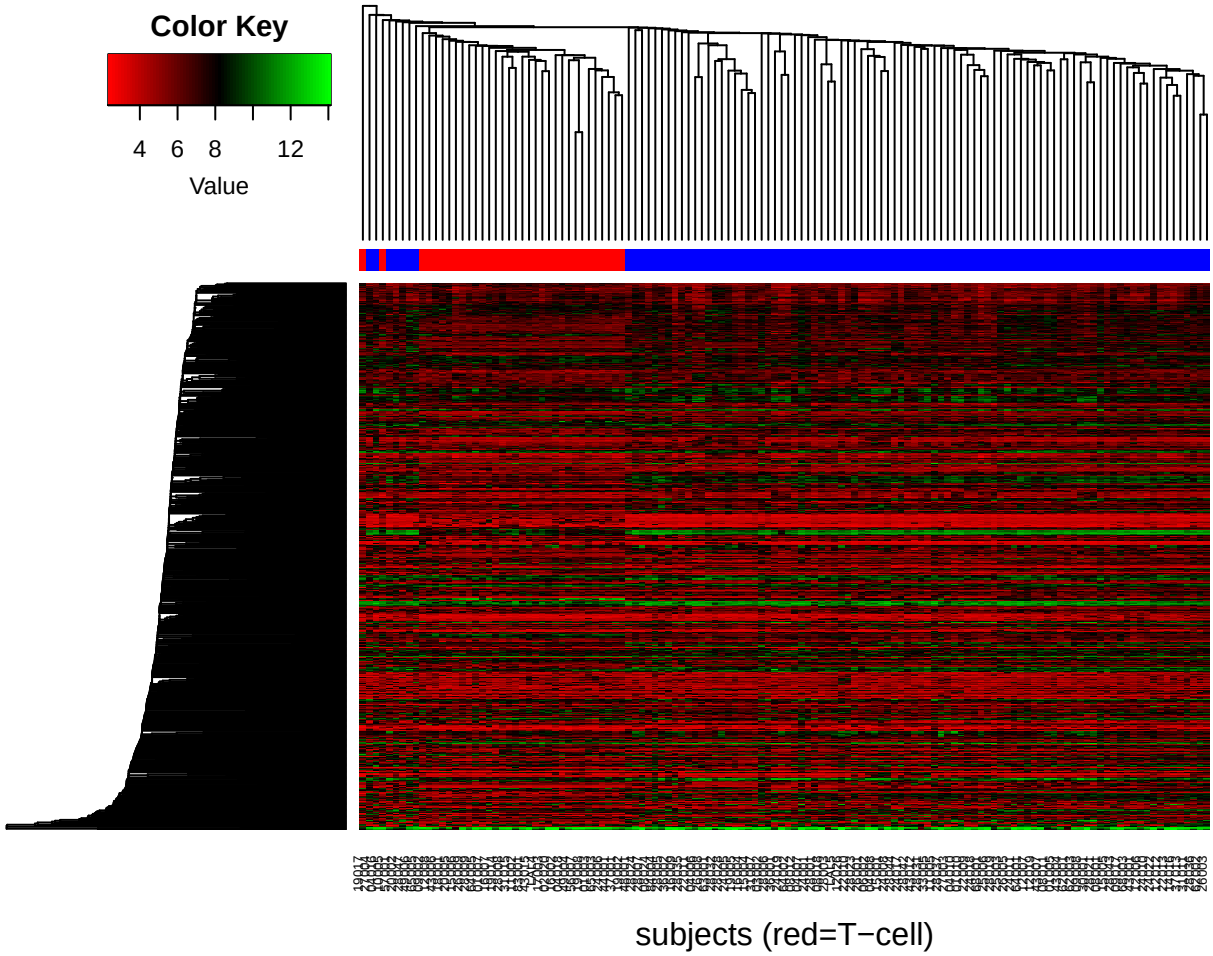
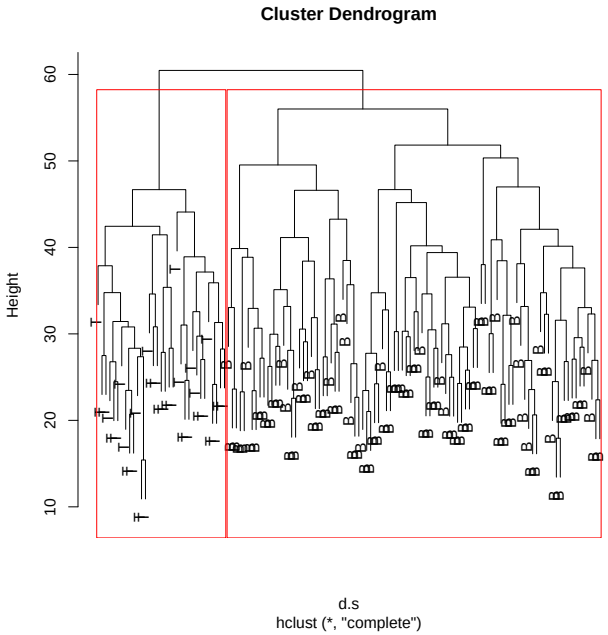


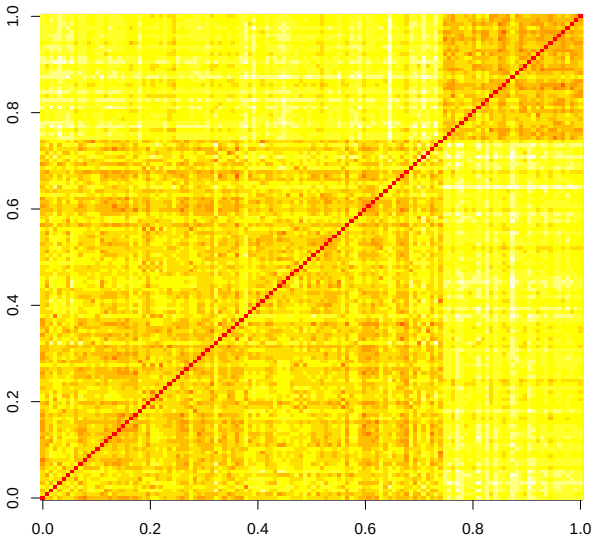
Figure 32: Heatmap of Gene expression values for genes that survived a filter of 95% and the single clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

24 Method: complete, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 33: based on Method: complete, Filter: 95%, Groups: 2

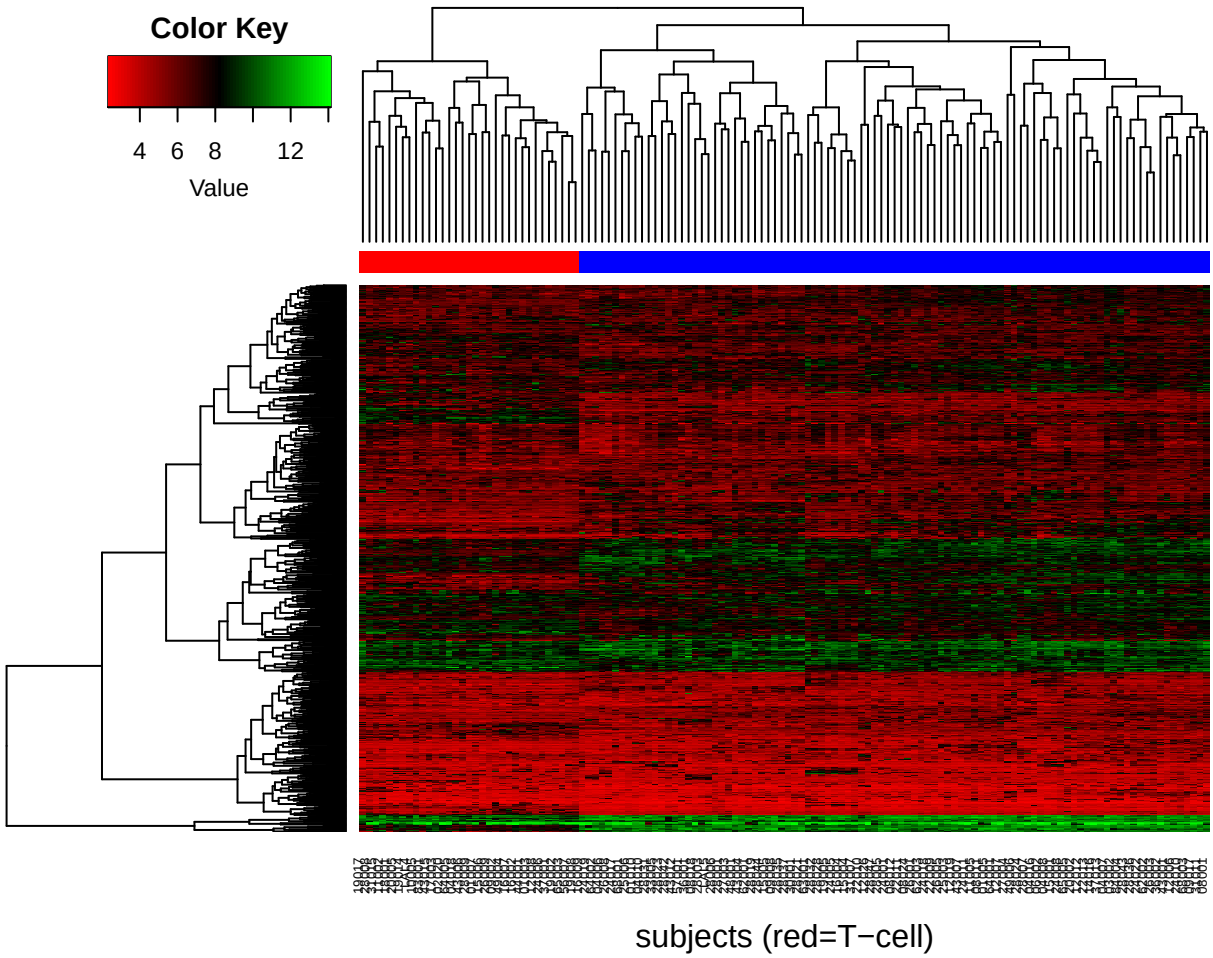
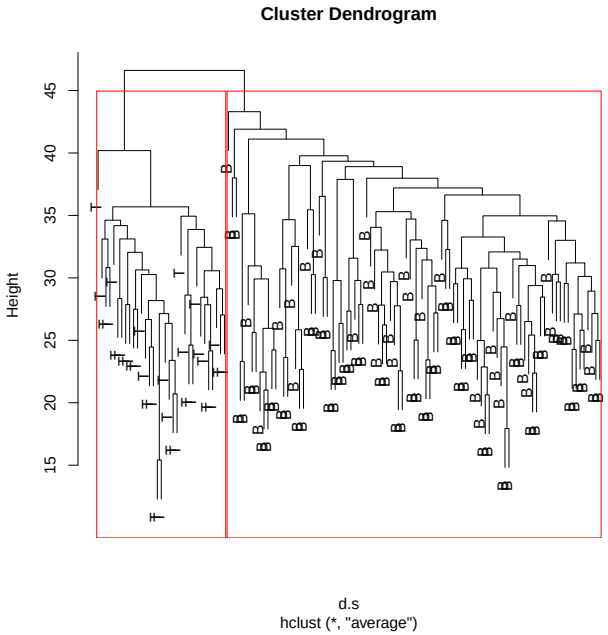


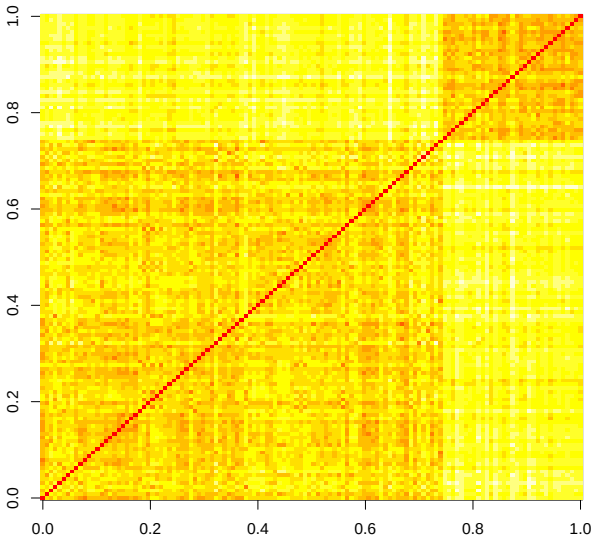
Figure 34: Heatmap of Gene expression values for genes that survived a filter of 95% and the complete clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

25 Method: average, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 35: based on Method: average, Filter: 95%, Groups: 2

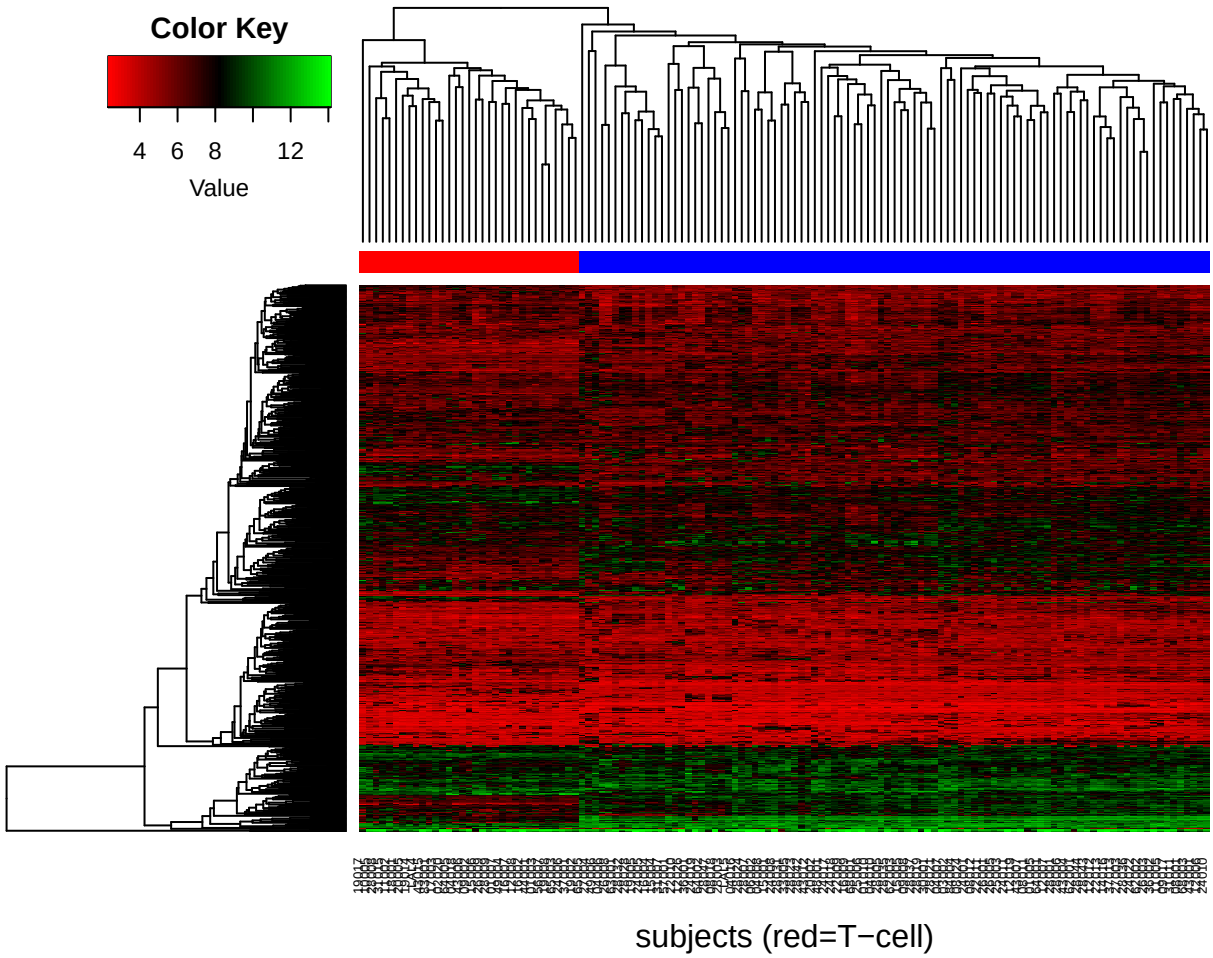
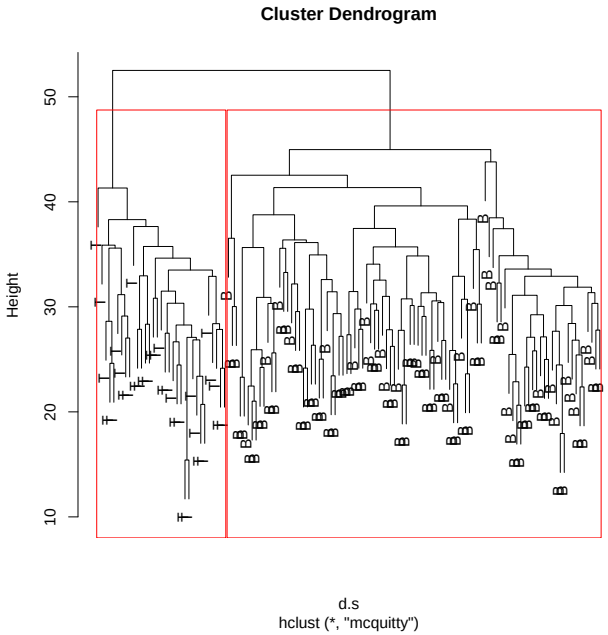


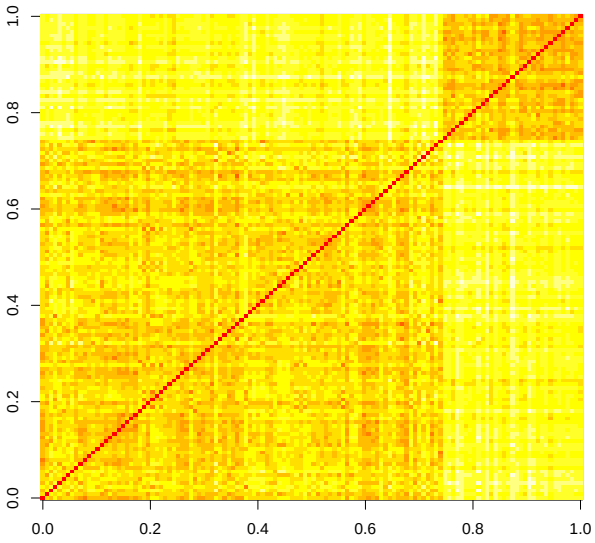
Figure 36: Heatmap of Gene expression values for genes that survived a filter of 95% and the average clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

26 Method: mcquitty, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95  0
##      2  0 33

fisher.test(groups, cl)$p.value

## [1] 2.3e-31
```



(a) Dendogram



(b) Distance Matrix

Figure 37: based on Method: mcquitty, Filter: 95%, Groups: 2

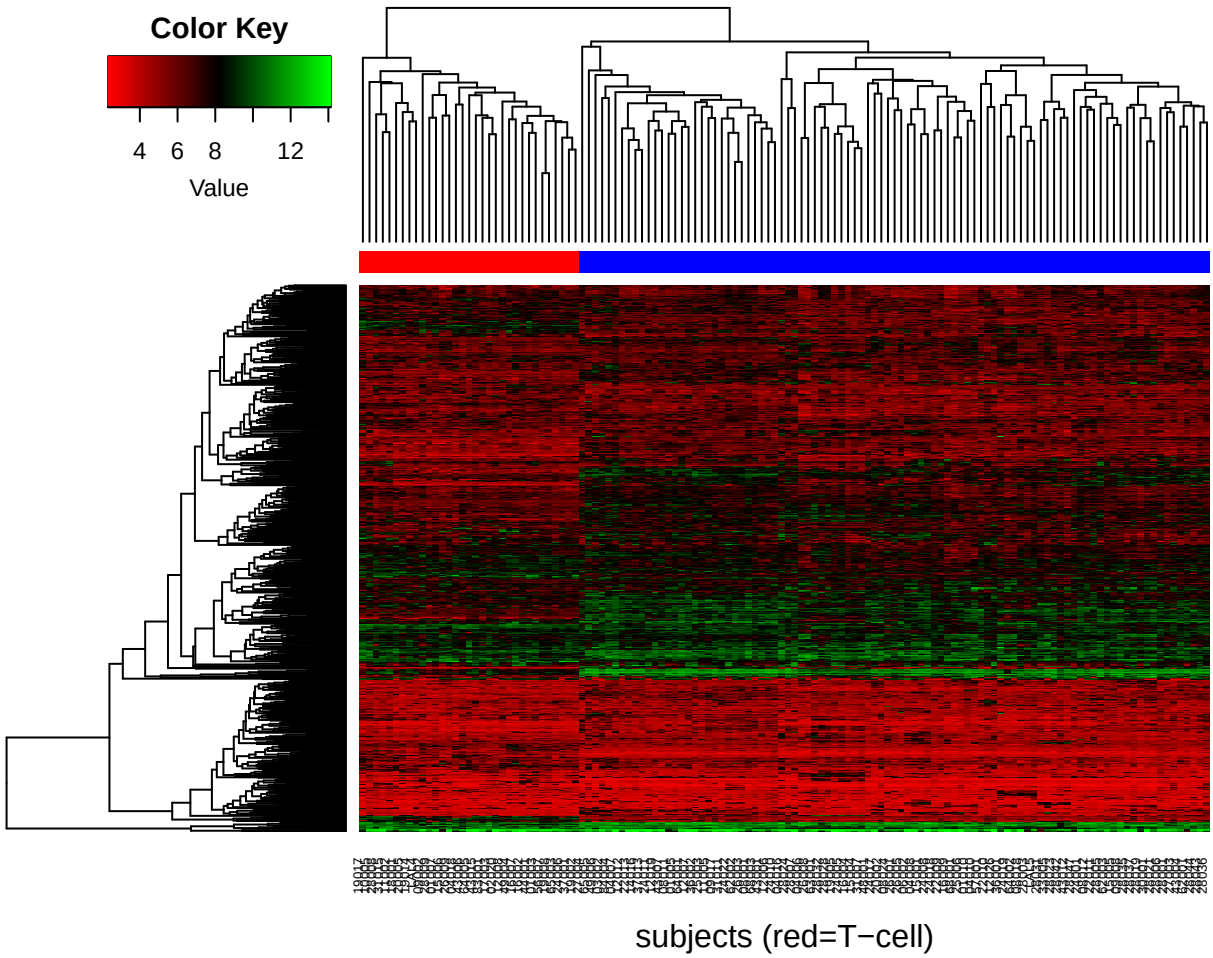
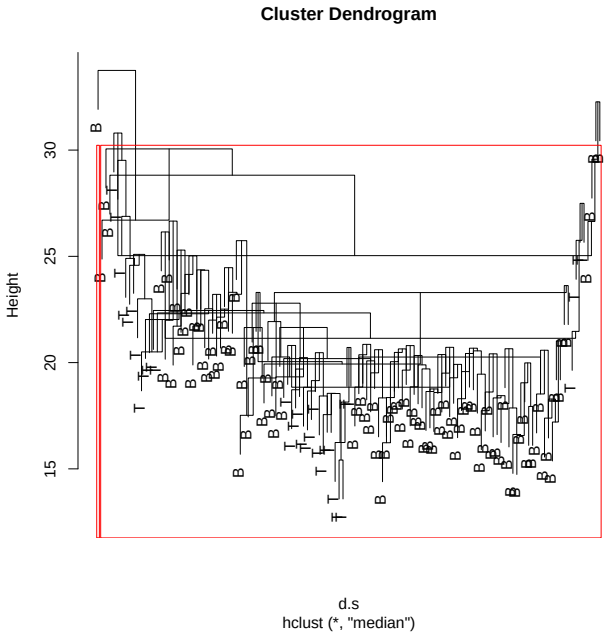


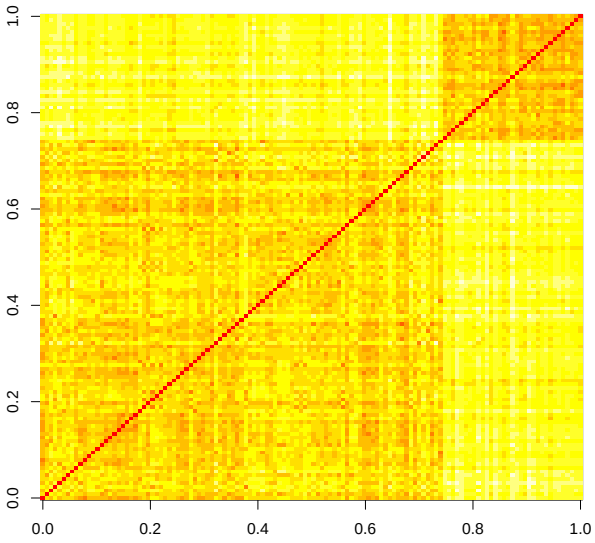
Figure 38: Heatmap of Gene expression values for genes that survived a filter of 95% and the mcquitty clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

27 Method: median, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##       cl
## groups  B  T
##      1 94 33
##      2  1  0

fisher.test(groups, cl)$p.value

## [1] 1
```



(a) Dendogram



(b) Distance Matrix

Figure 39: based on Method: median, Filter: 95%, Groups: 2

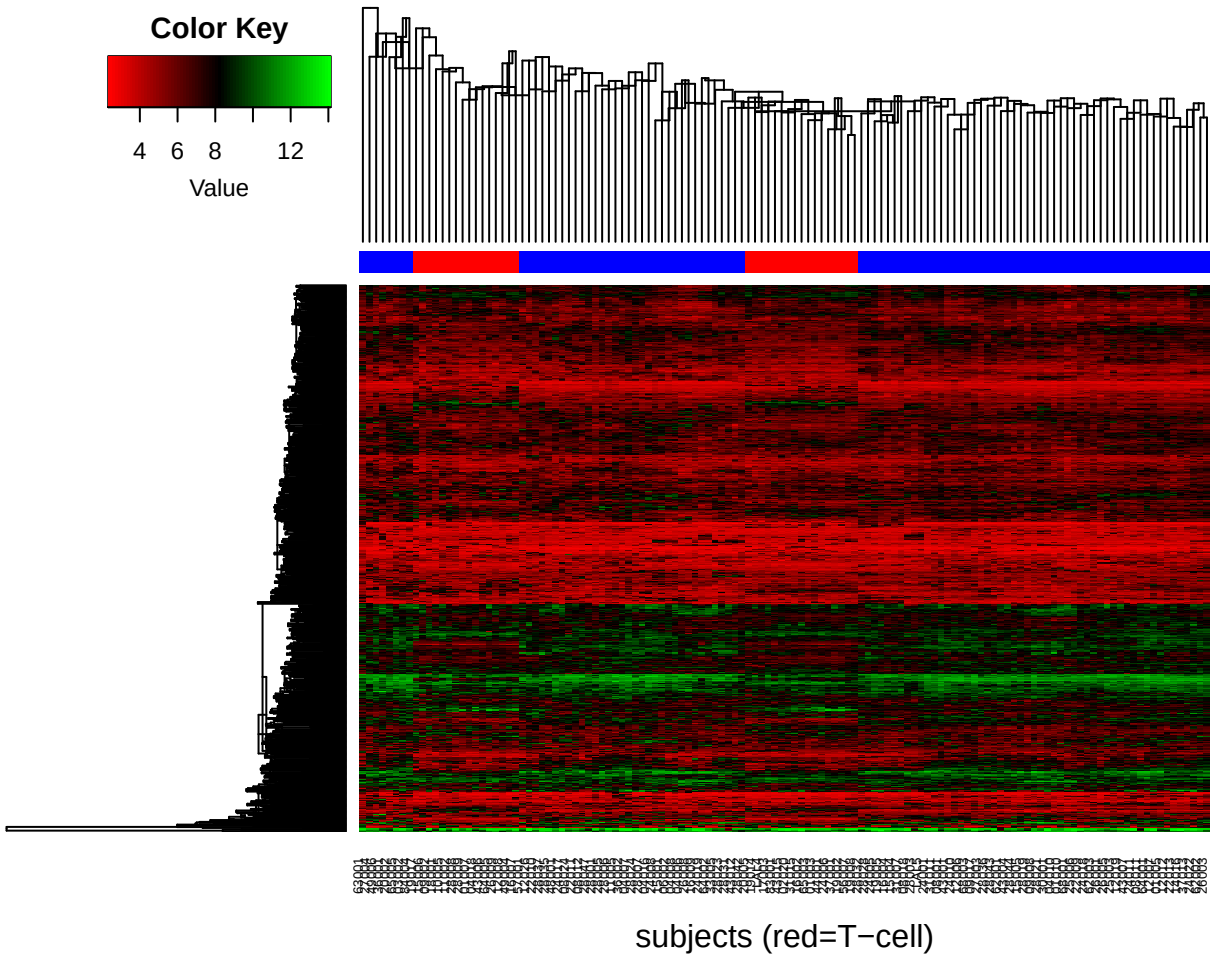
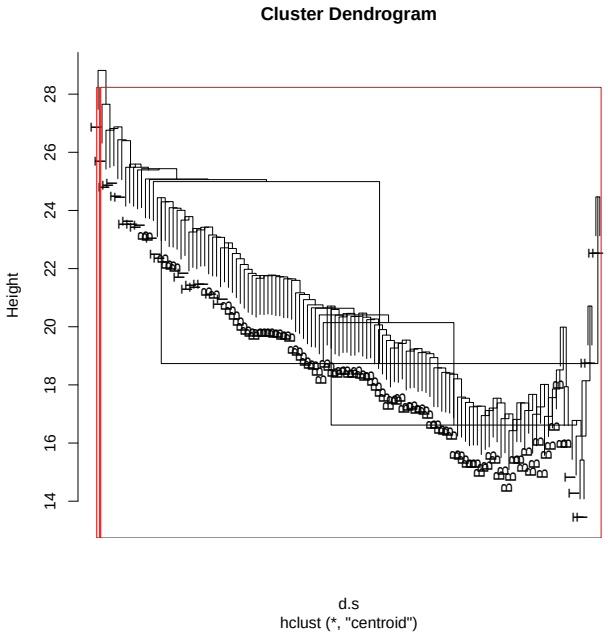


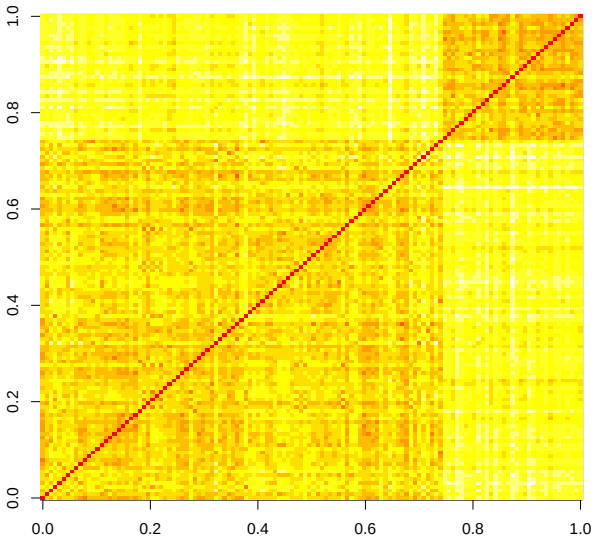
Figure 40: Heatmap of Gene expression values for genes that survived a filter of 95% and the median clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

28 Method: centroid, Filter: 95%, Groups: 2

```
dim(dat.filter)

## [1] 632 128

table(groups, cl)

##        cl
## groups  B  T
##      1 95 32
##      2  0  1

fisher.test(groups, cl)$p.value

## [1] 0.26
```



(a) Dendogram



(b) Distance Matrix

Figure 41: based on Method: centroid, Filter: 95%, Groups: 2

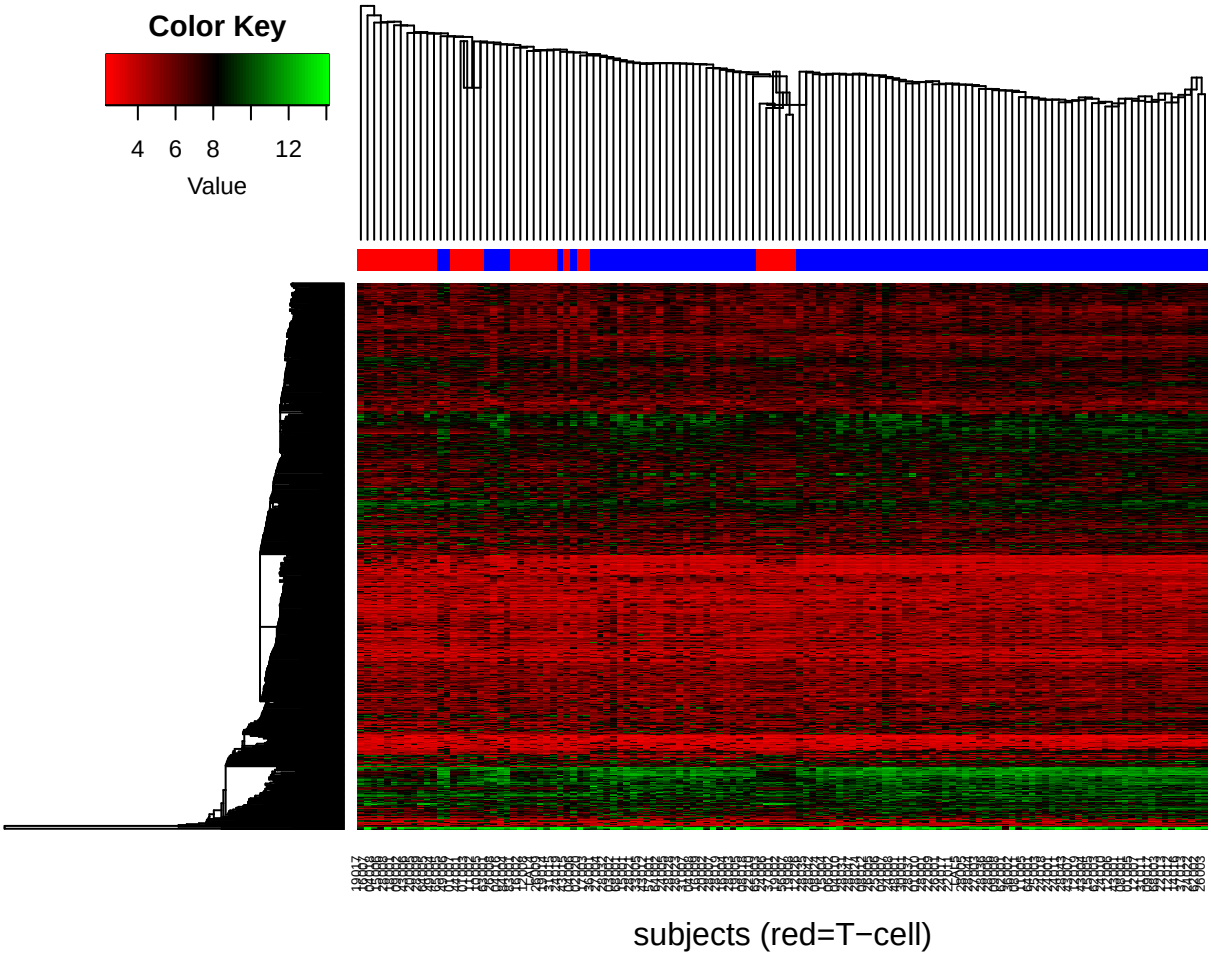


Figure 42: Heatmap of Gene expression values for genes that survived a filter of 95% and the centroid clustering algoritm. Rows are genes and columns are subjects. There are a total of 632 genes and 128 subjects in this plot.

References

Sabina Chiaretti, Xiaochun Li, Robert Gentleman, Antonella Vitale, Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa. Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103(7):2771–2778, 2004. 1

Sandrine Dudoit and Robert Gentleman. Cluster Analysis in DNA Microarray Experiments. Technical report, 2002. URL http://www.bioconductor.org/help/course-materials/2002/Seattle02/Cluster/cluster.pdf. 1

Xiaochun Li. *ALL: A data package*, 2009. R package version 1.10.0. 1

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2013. URL http://yihui.name/knitr/. ISBN 978-1482203530. 1

Yihui Xie. knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch, and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. URL http://www.crcpress.com/product/isbn/9781466561595. ISBN 978-1466561595. 1

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2015. URL http://yihui.name/knitr/. R package version 1.10.5. 1

A Session Information

```
sessionInfo()
```

```
## R version 3.2.0 (2015-04-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04 LTS
##
## locale:
##  [1] LC_CTYPE=en_CA.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_CA.UTF-8
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_CA.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils
## [6] datasets  methods   base
##
## other attached packages:
## [1] gplots_2.17.0       ALL_1.10.0
## [3] Biobase_2.28.0      BiocGenerics_0.14.0
## [5] knitr_1.10
##
## loaded via a namespace (and not attached):
##  [1] gtools_3.4.2       bitops_1.0-6
##  [3] formatR_1.2        magrittr_1.5
##  [5] evaluate_0.7       highr_0.5
##  [7] KernSmooth_2.23-14 stringi_0.4-1
##  [9] gdata_2.16.1       tools_3.2.0
## [11] stringr_1.0.0      caTools_1.17.1
```