Module1_Quiz

mindan

2021/10/3

```
knitr::opts_chunk$set(cache = TRUE)
```

```
rm(list=ls())
```

3.Create a `summarizedExperiment` object with the following code

```
rm(list=ls())
library(Biobase)
```

```
## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: ’BiocGenerics’

## The following objects are masked from ’package:parallel’:
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from ’package:stats’:
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from ’package:base’:
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     ’browseVignettes()’. To cite Bioconductor, see
##     ’citation("Biobase")’, and for packages ’citation("pkgname")’.
```

```
library(GenomicRanges)
```

```
## Loading required package: stats4
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: ’S4Vectors’
```

```
## The following object is masked from ’package:base’:
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: ’IRanges’
```

```
## The following object is masked from ’package:grDevices’:
##
##     windows
```

```
## Loading required package: GenomeInfoDb
```

```
library(SummarizedExperiment)
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: ’matrixStats’
```

```
## The following objects are masked from ’package:Biobase’:
##
##     anyMissing, rowMedians
```

```
##
## Attaching package: ’MatrixGenerics’
```

```
## The following objects are masked from ’package:matrixStats’:
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## The following object is masked from 'package:Biobase':
##
##     rowMedians
```

```
data(sample.ExpressionSet, package = "Biobase")
se = SummarizedExperiment(sample.ExpressionSet)
assays(se)
```

```
## List of length 1
```

```
colData(se)
```

```
## DataFrame with 26 rows and 0 columns
```

```
rowData(se)
```

```
## DataFrame with 500 rows and 0 columns
```

```
rowRanges(se)
```

```
## NULL
```

Load the Bottomly and the Bodymap data sets with the following code

```
con =url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bottomly_eset.RData")
load(file=con)
close(con)
bot = bottomly.eset
pdata_bot=pData(bot)
pdata_bot
```

```
##           sample.id num.tech.reps   strain experiment.number lane.number
## SRX033480 SRX033480             1 C57BL/6J                 6           1
## SRX033488 SRX033488             1 C57BL/6J                 7           1
## SRX033481 SRX033481             1 C57BL/6J                 6           2
## SRX033489 SRX033489             1 C57BL/6J                 7           2
## SRX033482 SRX033482             1 C57BL/6J                 6           3
## SRX033490 SRX033490             1 C57BL/6J                 7           3
## SRX033483 SRX033483             1 C57BL/6J                 6           5
## SRX033476 SRX033476             1 C57BL/6J                 4           6
## SRX033478 SRX033478             1 C57BL/6J                 4           7
## SRX033479 SRX033479             1 C57BL/6J                 4           8
## SRX033472 SRX033472             1   DBA/2J                 4           1
## SRX033473 SRX033473             1   DBA/2J                 4           2
## SRX033474 SRX033474             1   DBA/2J                 4           3
## SRX033475 SRX033475             1   DBA/2J                 4           5
## SRX033491 SRX033491             1   DBA/2J                 7           5
## SRX033484 SRX033484             1   DBA/2J                 6           6
## SRX033492 SRX033492             1   DBA/2J                 7           6
## SRX033485 SRX033485             1   DBA/2J                 6           7
## SRX033493 SRX033493             1   DBA/2J                 7           7
## SRX033486 SRX033486             1   DBA/2J                 6           8
## SRX033494 SRX033494             1   DBA/2J                 7           8
```

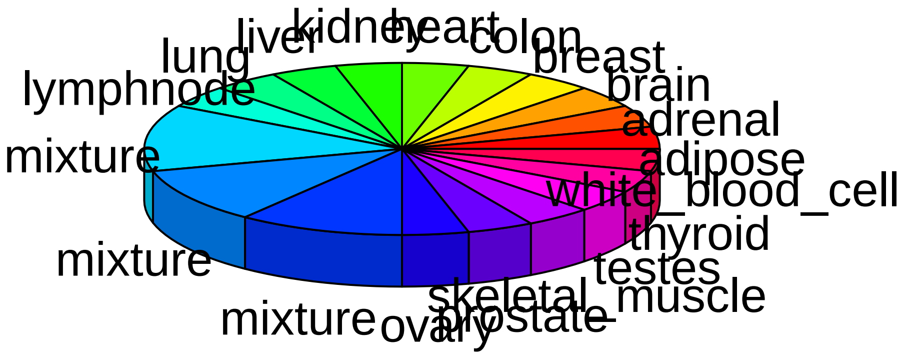
```
con =url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/bodymap_eset.RData")
load(file=con)
close(con)
bm = bodymap.eset
pdata_bm=pData(bm)
pdata_bm
```

```
##           sample.id num.tech.reps      tissue.type gender age             race
## ERS025098 ERS025098             2          adipose      F  73        caucasian
## ERS025092 ERS025092             2          adrenal      M  60        caucasian
## ERS025085 ERS025085             2            brain      F  77        caucasian
## ERS025088 ERS025088             2           breast      F  29        caucasian
## ERS025089 ERS025089             2            colon      F  68        caucasian
## ERS025082 ERS025082             2            heart      M  77        caucasian
## ERS025081 ERS025081             2           kidney      F  60        caucasian
## ERS025096 ERS025096             2            liver      M  37        caucasian
## ERS025099 ERS025099             2             lung      M  65        caucasian
## ERS025086 ERS025086             2        lymphnode      F  86        caucasian
## ERS025084 ERS025084             6          mixture   <NA>  NA        caucasian
## ERS025087 ERS025087             5          mixture   <NA>  NA        caucasian
## ERS025093 ERS025093             5          mixture   <NA>  NA        caucasian
## ERS025083 ERS025083             2            ovary      F  47 african_american
## ERS025095 ERS025095             2         prostate      M  73        caucasian
## ERS025097 ERS025097             2  skeletal_muscle      M  77        caucasian
## ERS025094 ERS025094             2           testes      M  19        caucasian
## ERS025090 ERS025090             2          thyroid      F  60        caucasian
## ERS025091 ERS025091             2 white_blood_cell      M  58        caucasian
```

```
edata_bm = exprs(bm)
```

5. Just considering the phenotype data what are some reasons that the Bottomly data set is likely a better experimental design than the Bodymap data? Imagine the question of interest in the Bottomly data is to compare strains and in the Bodymap data it is to compare tissues.

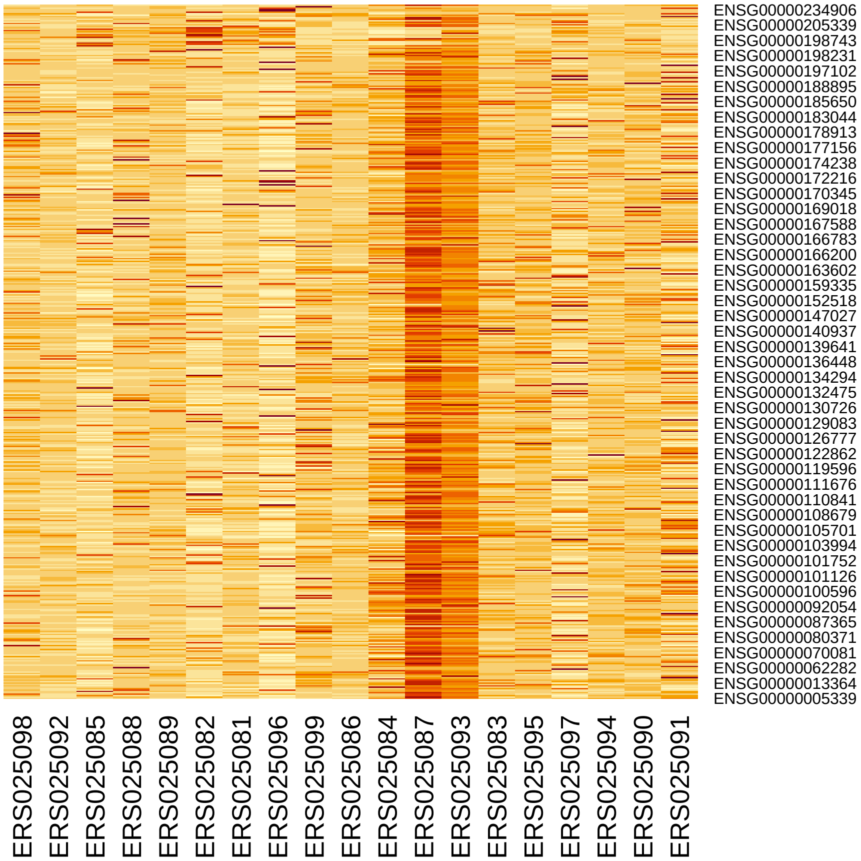
```
library(plotrix)
pie3D(pdata_bm$num.tech.reps,labels=pdata_bm$tissue.type)
```



6. What are some reasons why this plot is not useful for comparing the number of technical replicates by tissue (you may need to install the plotrix package).

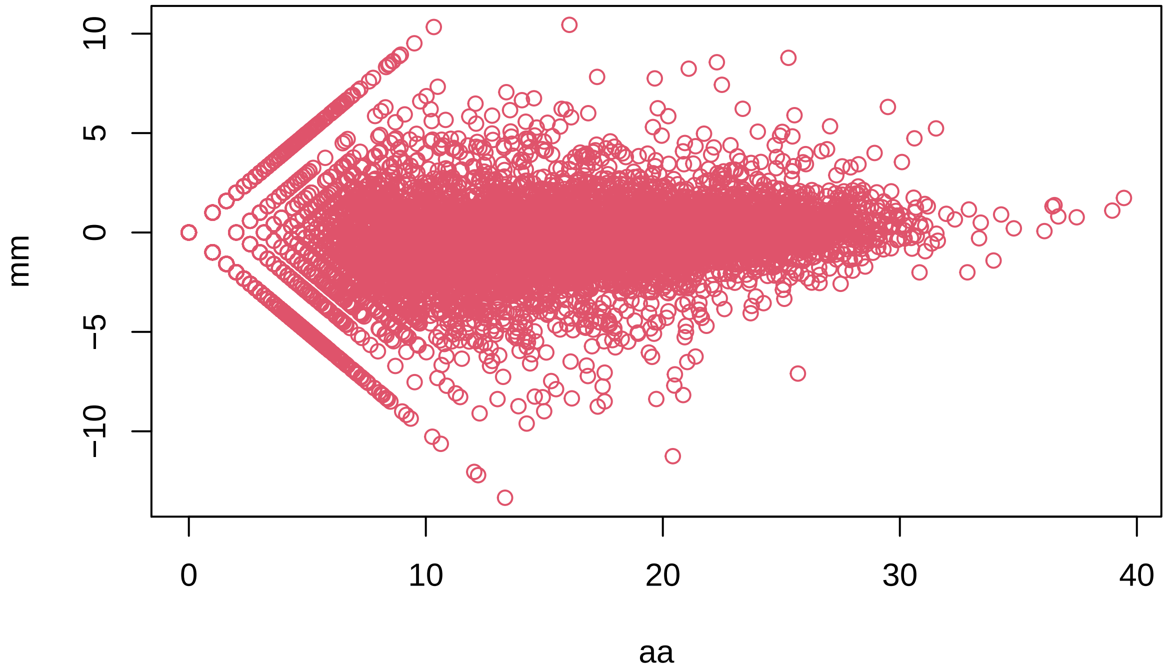
7.Which of the following code chunks will make a heatmap of the 500 most highly expressed genes (as defined by total count), without re-ordering due to clustering? Are the highly expressed samples next to each other in sample order?

```
row_sums = rowSums(edata_bm)
index = which(rank(-row_sums) < 500 )
heatmap(edata_bm[index,],Rowv=NA,Colv=NA)
```

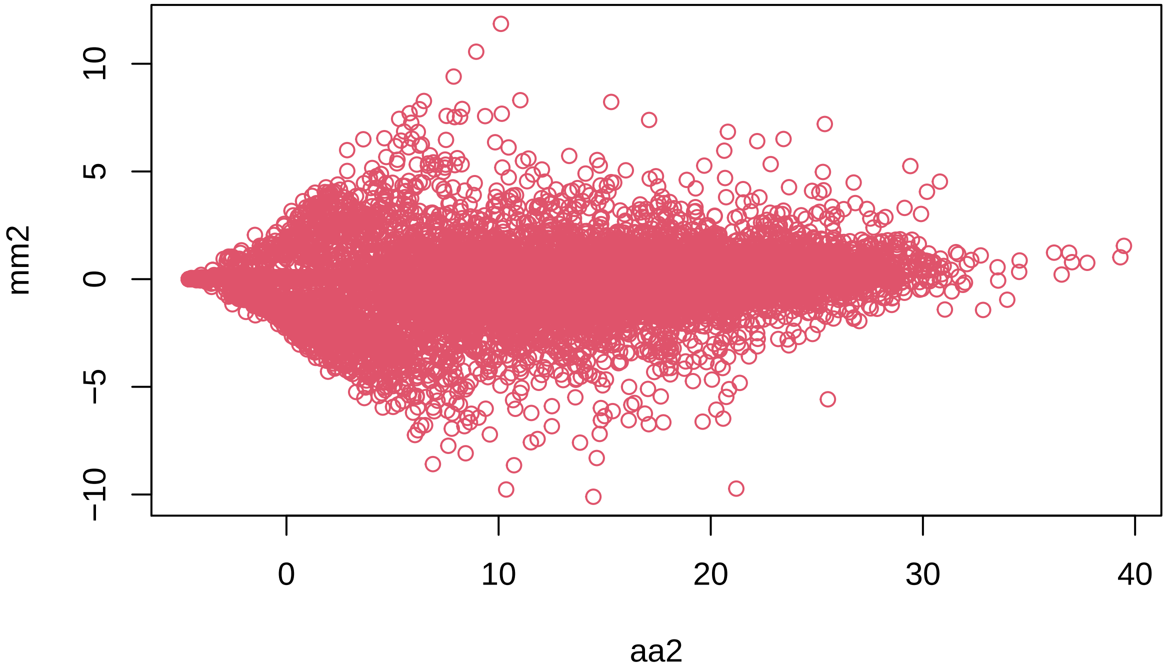


8.Make an MA-plot of the first sample versus the second sample using the log2 transform (hint: you may have to add 1 first) and the `rlog` transform from the DESeq2 package. How are the two MA-plots different? Which kind of genes appear most different in each plot?

```r
mm = log2(edata_bm[,1]+1) - log2(edata_bm[,2]+1)
aa = log2(edata_bm[,1]+1) + log2(edata_bm[,2]+1)
plot(aa,mm,col=2)
```



```
library(DESeq2)
edata1 <- rlog(edata_bm)
mm2 = edata1[,1] - edata1[,2]
aa2 = edata1[,1] + edata1[,2]
plot(aa2,mm2,col=2)
```



Load the Montgomery and Pickrell eSet

```
con =url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/montpick_eset.RData")
load(file=con)
close(con)
mp = montpick.eset
pdata=pData(mp)
edata=as.data.frame(exprs(mp))
fdata = fData(mp)
```

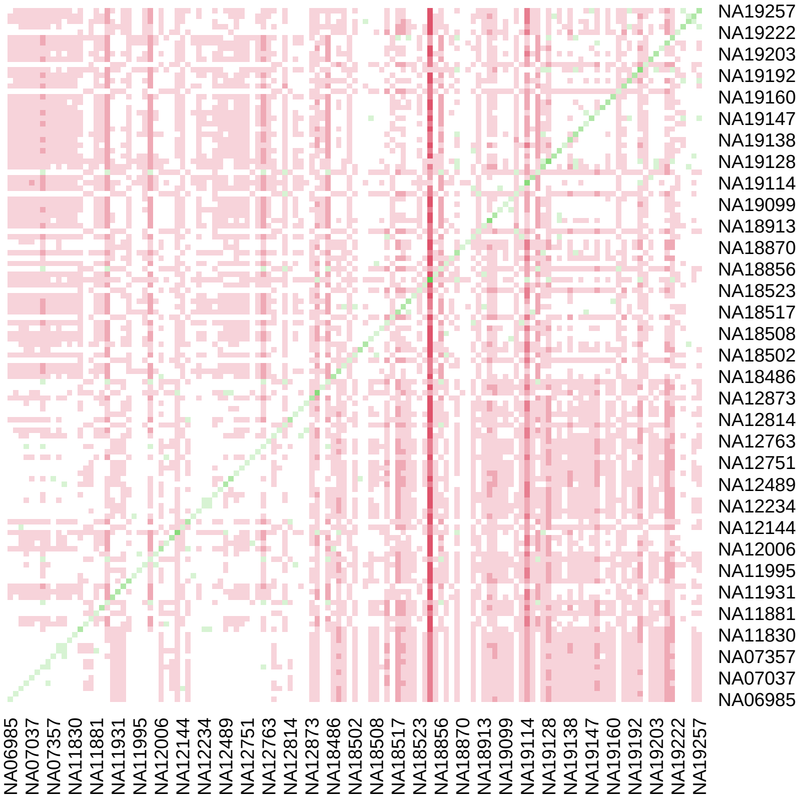
9. Cluster the data in three ways:

- With no changes to the data
- After filtering all genes withrowMeans less than 100
- After taking the log2 transform of the data without filtering

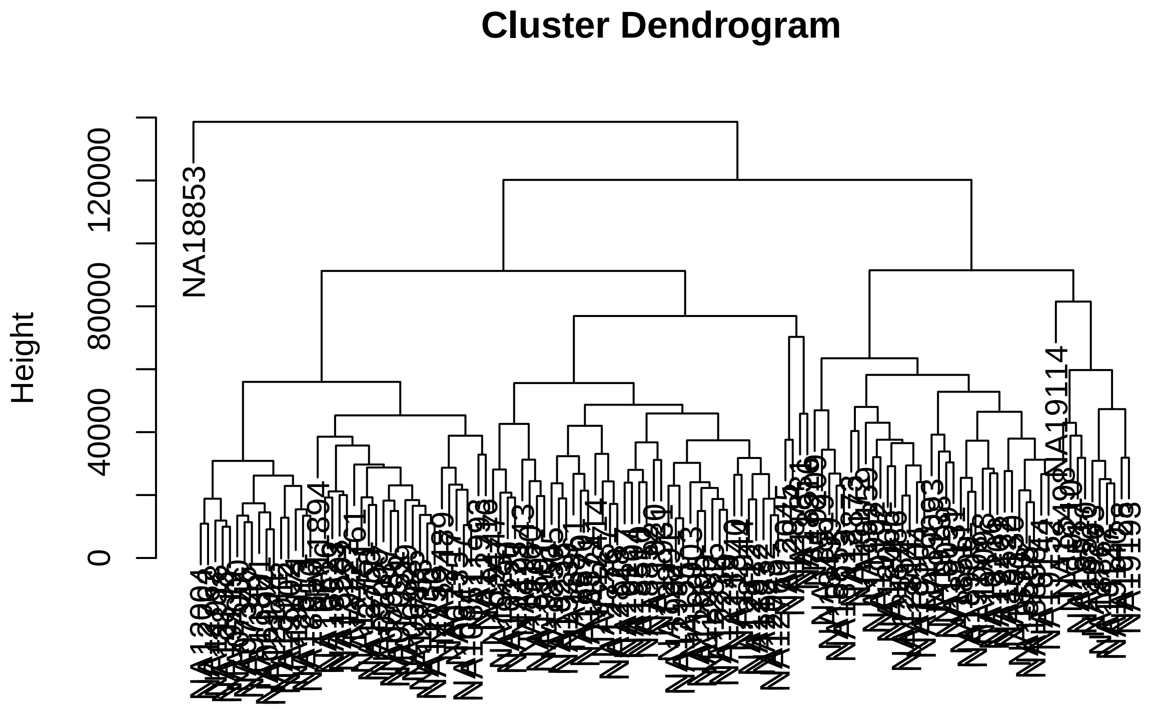
Color the samples by which study they came from (Hint: consider using the function myplclust.R in the package rafalib available from CRAN and looking at the argument lab.col.)

How do the methods compare in terms of how well they cluster the data by study? Why do you think that is?

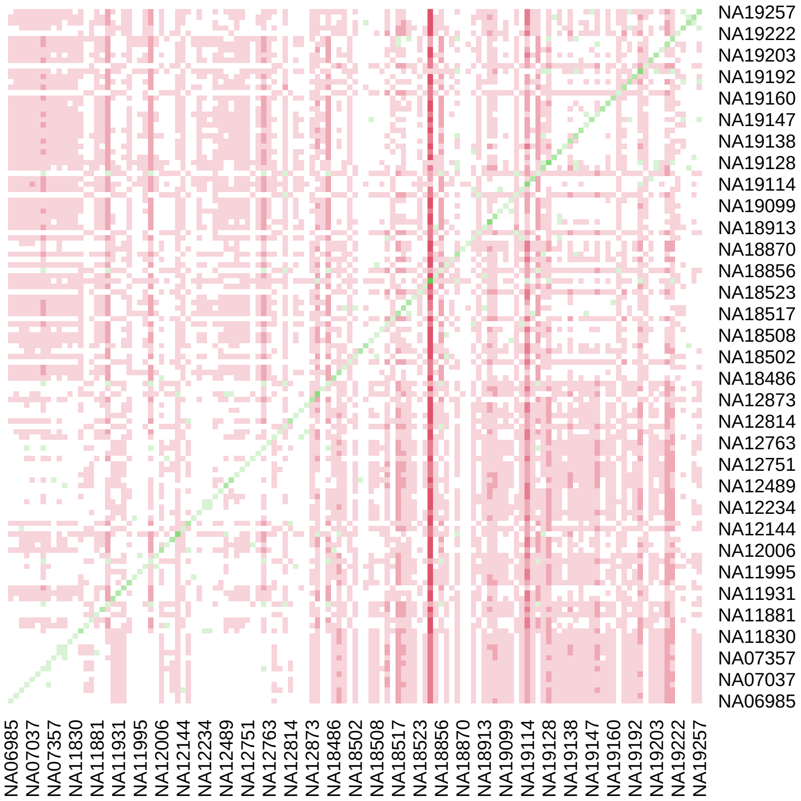
```
library(rafalib)
colramp = colorRampPalette(c(3,"white",2))(9)
# cluster With no changes to the data
dist1 = dist(t(edata))
heatmap(as.matrix(dist1),col=colramp,Colv=NA,Rowv=NA)
```



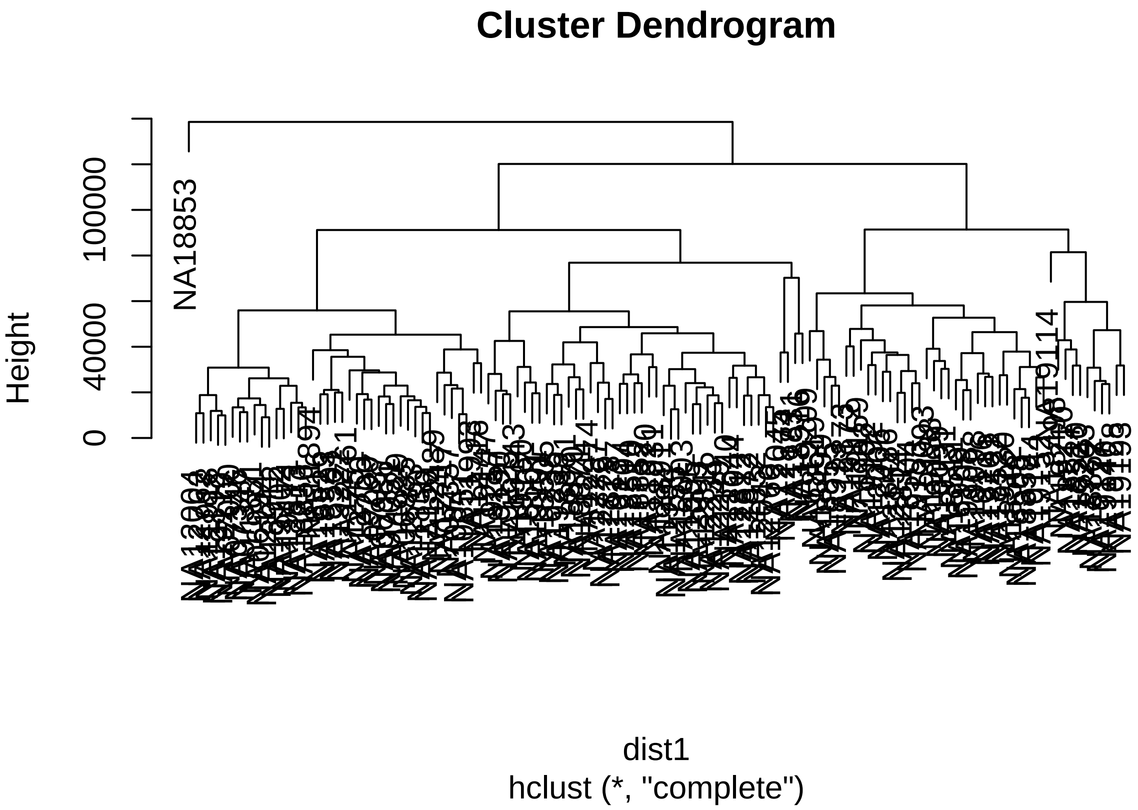
```
hclust1 = hclust(dist1)
myplclust(hclust1, labels = hclust1$labels, lab.col = rep(1, length(hclust1$labels)), hang = 0.1)
```



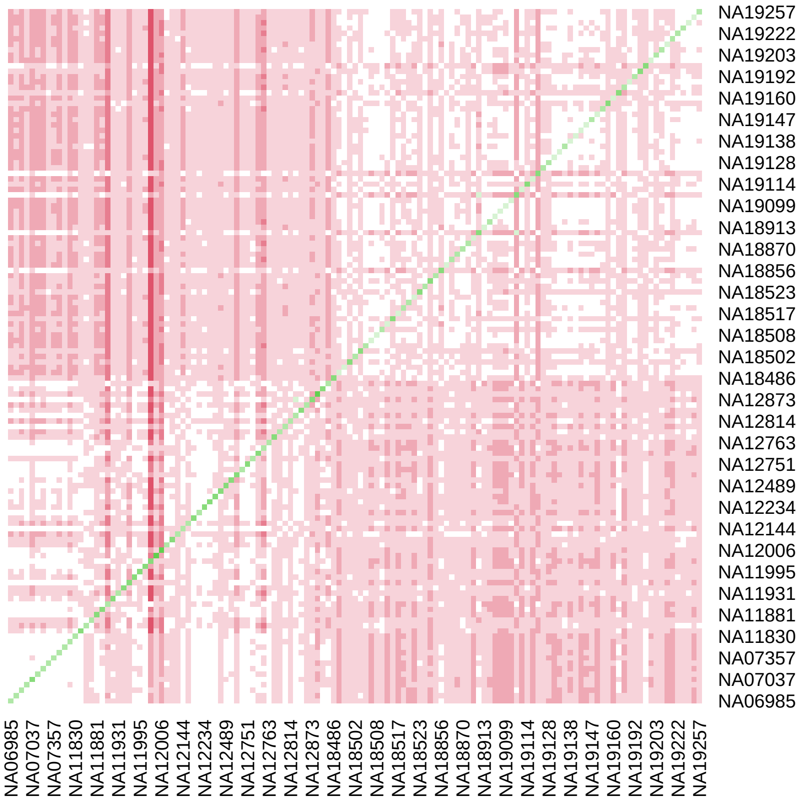
```
# After filtering all genes with rowMeans less than 100
edata2 = edata[rowMeans(edata) > 100,]
dist1 = dist(t(edata2))
heatmap(as.matrix(dist1),col=colramp,Colv=NA,Rowv=NA)
```



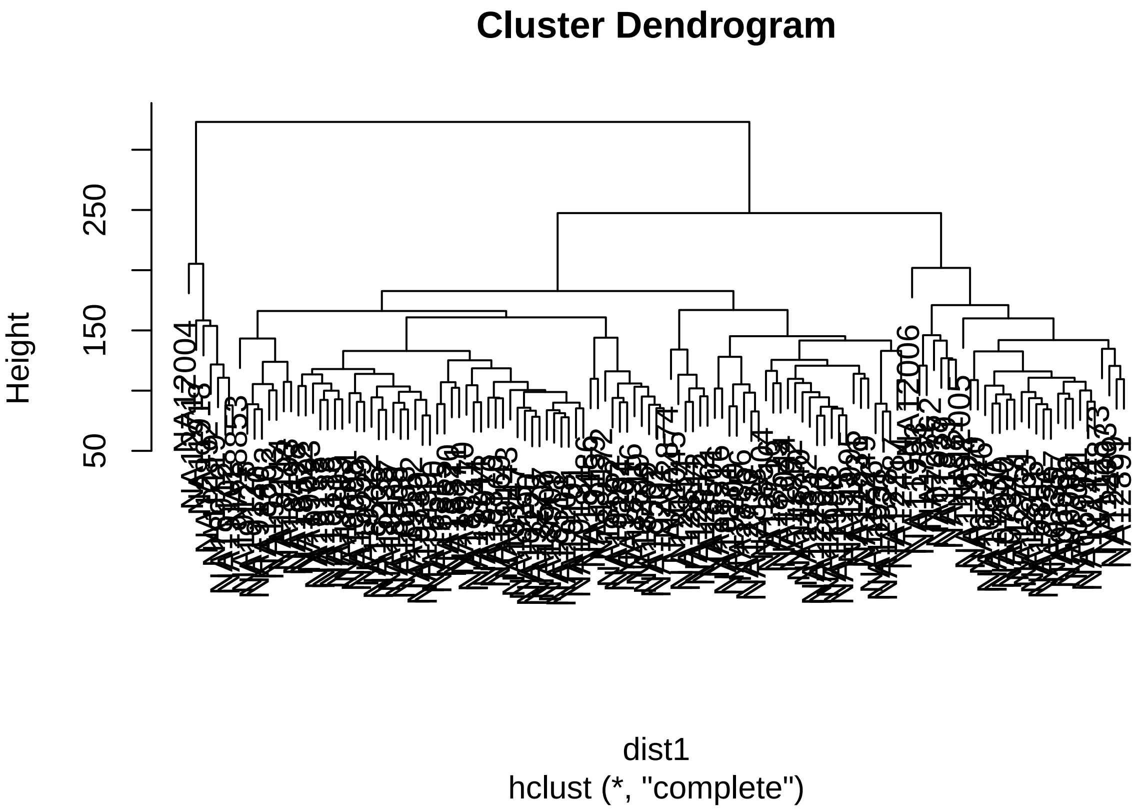
```
hclust1 = hclust(dist1)
plot(hclust1)
```



```
# After taking the log2 transform of the data without filtering
edata3 = log2(edata + 1)
dist1 = dist(t(edata3))
heatmap(as.matrix(dist1),col=colramp,Colv=NA,Rowv=NA)
```

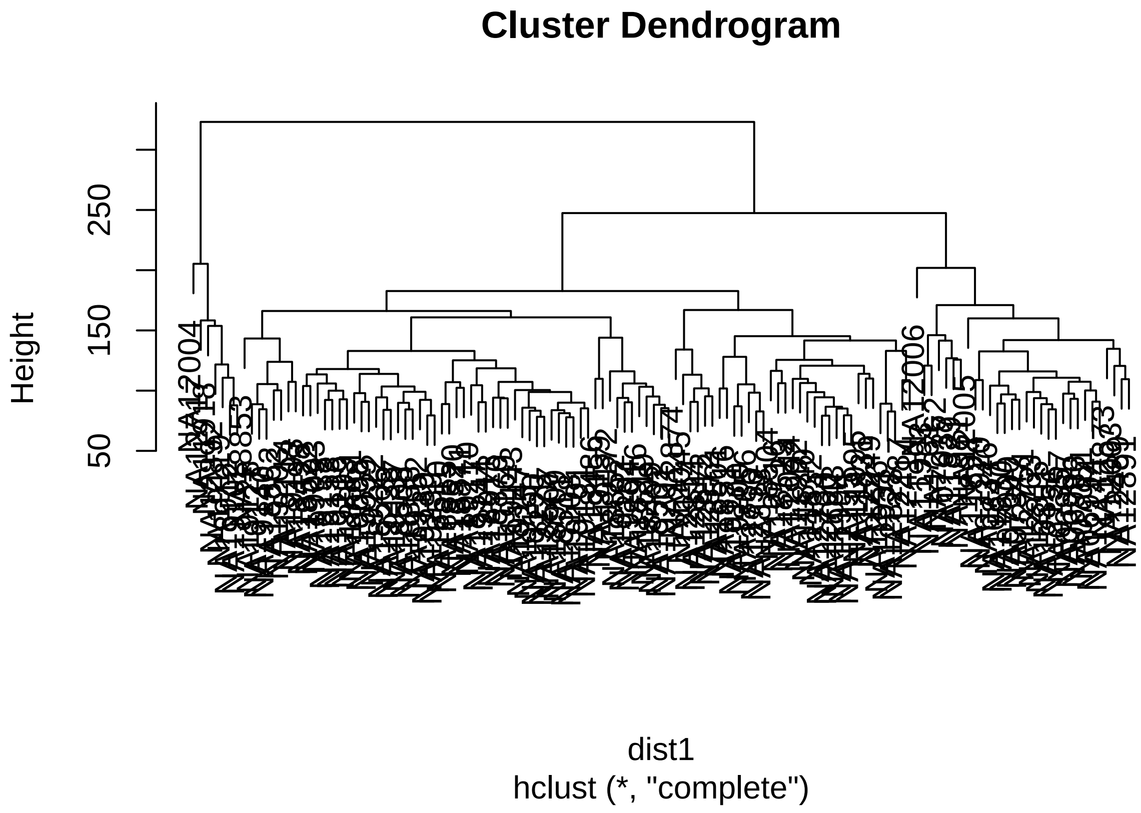


```
hclust1 = hclust(dist1)
plot(hclust1)
```



10.Cluster the samples using k-means clustering after applying the `log2` transform (be sure to add 1). Set a seed for reproducible results (use `set.seed(1235)`). If you choose two clusters, do you get the same two clusters as you get if you use the `cutree` function to cluster the samples into two groups? Which cluster matches most closely to the study labels?

```r
edata3 = log2(edata + 1)
dist1 = dist(t(edata3))
hclust1 = hclust(dist1)
plot(hclust1)
```



```
cut <- cutree(hclust1, k = 2)
table(cut)
```

```
## cut
##   1   2
## 122   7
```

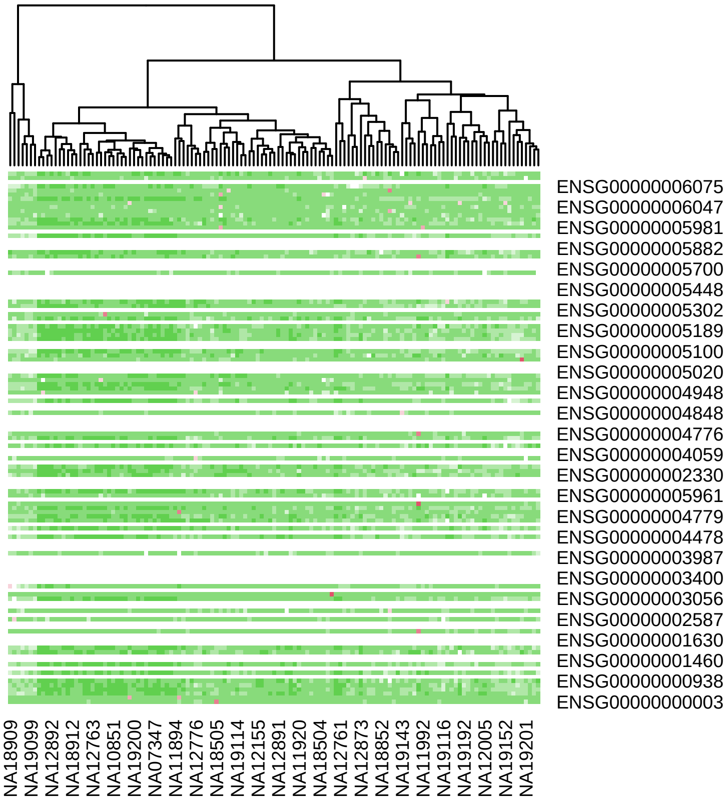
```
set.seed(1235)
kmeans1 = kmeans(t(edata3),centers=2)
names(kmeans1)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
table(kmeans1$cluster)
```

```
##
##  1  2
## 52 77
```

```
heatmap(as.matrix(edata)[order(kmeans1$cluster),],col=colramp,Rowv=NA)
```



```
all.equal(kmeans1$cluster,cut)
```

```
## [1] "Mean relative difference: 0.5217391"
```

```
#labels_colors(dend) = c(rep(1,60),rep(2,69))
#plot(dend)
```

```
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.0.5 (2021-03-31)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  Chinese (Simplified)_China.936
##  ctype    Chinese (Simplified)_China.936
##  tz       Asia/Taipei
##  date     2021-10-11
##
## - Packages -------------------------------------------------------------------
##  package              * version  date       lib source
##  annotate               1.68.0   2020-10-27 [1] Bioconductor
##  AnnotationDbi          1.52.0   2020-10-27 [1] Bioconductor
##  Biobase              * 2.50.0   2020-10-27 [1] Bioconductor
##  BiocGenerics         * 0.36.1   2021-04-16 [1] Bioconductor
##  BiocParallel           1.24.1   2020-11-06 [1] Bioconductor
##  bit                    4.0.4    2020-08-04 [1] CRAN (R 4.0.5)
##  bit64                  4.0.5    2020-08-30 [1] CRAN (R 4.0.5)
##  bitops                 1.0-7    2021-04-24 [1] CRAN (R 4.0.5)
##  blob                   1.2.2    2021-07-23 [1] CRAN (R 4.0.5)
##  cachem                 1.0.6    2021-08-19 [1] CRAN (R 4.0.5)
##  callr                  3.7.0    2021-04-20 [1] CRAN (R 4.0.5)
##  cli                    3.0.1    2021-07-17 [1] CRAN (R 4.0.5)
##  codetools              0.2-18   2020-11-04 [1] CRAN (R 4.0.5)
##  colorspace             2.0-2    2021-06-24 [1] CRAN (R 4.0.5)
##  crayon                 1.4.1    2021-02-08 [1] CRAN (R 4.0.5)
##  DBI                    1.1.1    2021-01-15 [1] CRAN (R 4.0.5)
##  DelayedArray           0.16.3   2021-03-24 [1] Bioconductor
##  desc                   1.4.0    2021-09-28 [1] CRAN (R 4.0.5)
##  DESeq2               * 1.30.1   2021-02-19 [1] Bioconductor
##  devtools               2.4.2    2021-06-07 [1] CRAN (R 4.0.5)
##  digest                 0.6.27   2020-10-24 [1] CRAN (R 4.0.5)
##  dplyr                  1.0.7    2021-06-18 [1] CRAN (R 4.0.5)
##  ellipsis               0.3.2    2021-04-29 [1] CRAN (R 4.0.5)
##  evaluate               0.14     2019-05-28 [1] CRAN (R 4.0.5)
##  fansi                  0.5.0    2021-05-25 [1] CRAN (R 4.0.5)
##  fastmap                1.1.0    2021-01-25 [1] CRAN (R 4.0.5)
##  fs                     1.5.0    2020-07-31 [1] CRAN (R 4.0.5)
##  genefilter             1.72.1   2021-01-21 [1] Bioconductor
##  geneplotter            1.68.0   2020-10-27 [1] Bioconductor
##  generics               0.1.0    2020-10-31 [1] CRAN (R 4.0.5)
##  GenomeInfoDb         * 1.26.7   2021-04-09 [1] Bioconductor
##  GenomeInfoDbData       1.2.4    2021-09-26 [1] Bioconductor
##  GenomicRanges        * 1.42.0   2020-10-27 [1] Bioconductor
##  ggplot2                3.3.5    2021-06-25 [1] CRAN (R 4.0.5)
##  glue                   1.4.2    2020-08-27 [1] CRAN (R 4.0.5)
##  gtable                 0.3.0    2019-03-25 [1] CRAN (R 4.0.5)
##  highr                  0.9      2021-04-16 [1] CRAN (R 4.0.5)
##  htmltools              0.5.2    2021-08-25 [1] CRAN (R 4.0.5)
##  httr                   1.4.2    2020-07-20 [1] CRAN (R 4.0.5)
##  IRanges              * 2.24.1   2020-12-12 [1] Bioconductor
##  knitr                  1.36     2021-09-29 [1] CRAN (R 4.0.5)
##  lattice                0.20-45  2021-09-22 [1] CRAN (R 4.0.5)
##  lifecycle              1.0.1    2021-09-24 [1] CRAN (R 4.0.5)
##  locfit                 1.5-9.4  2020-03-25 [1] CRAN (R 4.0.5)
##  magrittr               2.0.1    2020-11-17 [1] CRAN (R 4.0.5)
##  Matrix                 1.3-4    2021-06-01 [1] CRAN (R 4.0.5)
##  MatrixGenerics       * 1.2.1    2021-01-30 [1] Bioconductor
##  matrixStats          * 0.61.0   2021-09-17 [1] CRAN (R 4.0.5)
##  memoise                2.0.0    2021-01-26 [1] CRAN (R 4.0.5)
##  munsell                0.5.0    2018-06-12 [1] CRAN (R 4.0.5)
##  pillar                 1.6.3    2021-09-26 [1] CRAN (R 4.0.5)
##  pkgbuild               1.2.0    2020-12-15 [1] CRAN (R 4.0.5)
##  pkgconfig              2.0.3    2019-09-22 [1] CRAN (R 4.0.5)
##  pkgload                1.2.2    2021-09-11 [1] CRAN (R 4.0.5)
##  plotrix              * 3.8-2    2021-09-08 [1] CRAN (R 4.0.5)
##  prettyunits            1.1.1    2020-01-24 [1] CRAN (R 4.0.5)
##  processx               3.5.2    2021-04-30 [1] CRAN (R 4.0.5)
##  ps                     1.6.0    2021-02-28 [1] CRAN (R 4.0.5)
##  purrr                  0.3.4    2020-04-17 [1] CRAN (R 4.0.5)
##  R6                     2.5.1    2021-08-19 [1] CRAN (R 4.0.5)
##  rafalib              * 1.0.0    2015-08-09 [1] CRAN (R 4.0.3)
##  RColorBrewer           1.1-2    2014-12-07 [1] CRAN (R 4.0.3)
##  Rcpp                   1.0.7    2021-07-07 [1] CRAN (R 4.0.5)
##  RCurl                  1.98-1.5 2021-09-17 [1] CRAN (R 4.0.5)
##  remotes                2.4.1    2021-09-29 [1] CRAN (R 4.0.5)
##  rlang                  0.4.11   2021-04-30 [1] CRAN (R 4.0.5)
##  rmarkdown              2.11     2021-09-14 [1] CRAN (R 4.0.5)
##  rprojroot              2.0.2    2020-11-15 [1] CRAN (R 4.0.5)
##  RSQLite                2.2.8    2021-08-21 [1] CRAN (R 4.0.5)
##  rstudioapi             0.13     2020-11-12 [1] CRAN (R 4.0.5)
##  S4Vectors            * 0.28.1   2020-12-09 [1] Bioconductor
##  scales                 1.1.1    2020-05-11 [1] CRAN (R 4.0.5)
##  sessioninfo            1.1.1    2018-11-05 [1] CRAN (R 4.0.5)
##  stringi                1.7.5    2021-10-04 [1] CRAN (R 4.0.5)
##  stringr                1.4.0    2019-02-10 [1] CRAN (R 4.0.5)
##  SummarizedExperiment * 1.20.0   2020-10-28 [1] Bioconductor
##  survival               3.2-13   2021-08-24 [1] CRAN (R 4.0.5)
##  testthat               3.0.4    2021-07-01 [1] CRAN (R 4.0.5)
##  tibble                 3.1.4    2021-08-25 [1] CRAN (R 4.0.5)
##  tidyselect             1.1.1    2021-04-30 [1] CRAN (R 4.0.5)
##  usethis                2.0.1    2021-02-10 [1] CRAN (R 4.0.5)
##  utf8                   1.2.2    2021-07-24 [1] CRAN (R 4.0.5)
##  vctrs                  0.3.8    2021-04-29 [1] CRAN (R 4.0.5)
##  withr                  2.4.2    2021-04-18 [1] CRAN (R 4.0.5)
##  xfun                   0.26     2021-09-14 [1] CRAN (R 4.0.5)
##  XML                    3.99-0.8 2021-09-17 [1] CRAN (R 4.0.5)
##  xtable                 1.8-4    2019-04-21 [1] CRAN (R 4.0.5)
##  XVector                0.30.0   2020-10-28 [1] Bioconductor
##  yaml                   2.2.1    2020-02-01 [1] CRAN (R 4.0.5)
##  zlibbioc               1.36.0   2020-10-28 [1] Bioconductor
##
## [1] D:/R/R-4.0.5/library
```