# Tasks

irfan

2/10/2021

## Step 1: Read the datasets to R

```
getwd()
```

```
## [1] "/Users/Irfan/Downloads/Worker_R_test"
```

```
setwd("/Users/Irfan/Downloads/Worker_R_test/")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
file1 = "SampleTypeData.txt"
sampleTypeData = read.csv(file1, header = TRUE, sep = "\t", dec = ".")
dim(sampleTypeData)
```

```
## [1] 89 21
```

```
file2 = "PairData_Trimmo_ENSEMBL_HTSeq_Edited.table"
PairData = read.csv(file2, header = TRUE, sep = "\t", dec = ".")
dim(PairData)
```

```
## [1] 58889    91
```

## Step 2: Select relevant data columns and data rows

```
row.names(PairData) <- PairData$DB_ID
PairData_remove2col <- PairData[, -c(1:2)] # delete columns 1,2
dim(PairData_remove2col)
```

```
## [1] 58889    89
```

```
n<-dim(PairData_remove2col)[1]
PairData_remove2col4rows<-PairData_remove2col[1:(n-4),] # delete last 4 rows
dim(PairData_remove2col4rows)
```

```
## [1] 58885    89
```

```
last4rowsPairData <- tail(PairData, n=4)
row.names(last4rowsPairData) # control data row names
```

```
## [1] "suougibma__"            "lauQa_wol_oot__"        "dengila_ton__"
## [4] "euqinu_ton_tnemngila__"
```

### Step 3: Clean the column names from data matrix

```
PairData_remove2col4rows_cleanColumns  <- PairData_remove2col4rows %>%
                                          rename_all(~ str_remove(., "\\_S.*"))

names(PairData_remove2col4rows_cleanColumns) <- gsub(x = names(PairData_remove2col4rows_cleanColumns),
```

### Step 4: Reorder the sample types data

```
Ordered_sampleTypeData <- sampleTypeData[ order(match(sampleTypeData$Sample.ID.Corrected, colnames(Pair
dim(Ordered_sampleTypeData)
```

```
## [1] 89 21
```

```
row.names(Ordered_sampleTypeData) <- Ordered_sampleTypeData$Sample.ID.Corrected
```

### Step 5,6,7,8,9

```
## Step 5: Load data to DESeq2 object

library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## 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:dplyr':
##
##     combine, intersect, setdiff, union

## 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, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

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

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: Biobase

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

## Loading required package: DelayedArray

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

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

## The following object is masked from 'package:dplyr':
##
##     count

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
```

```
dds <- DESeqDataSetFromMatrix(countData=PairData_remove2col4rows_cleanColumns,
                              colData=Ordered_sampleTypeData[, c(4:8,15)],
                              design= ~ Sample.type.3, tidy = FALSE)

dds
```

```
## class: DESeqDataSet
## dim: 58885 89
## metadata(1): version
## assays(1): counts
## rownames(58885): 3000000000021 5000000000021 ... 2726820000021
##   erutaef_on__
## rowData names(0):
## colnames(89): J11L_1 J11LN_1 ... MMR7_LN_4 MMR868
## colData names(6): SampleType SampleDetailed ... Sample.type.4 CRCs
```

```
featureData <- data.frame(gene=rownames(PairData_remove2col4rows_cleanColumns))
mcols(dds) <- DataFrame(mcols(dds), featureData)
mcols(dds)
```

```
## DataFrame with 58885 rows and 1 column
##                         gene
##                     <factor>
## 3000000000021 3000000000021
## 5000000000021 5000000000021
## 9140000000021 9140000000021
## 7540000000021 7540000000021
## 0640000000021 0640000000021
## ...                      ...
## 9626820000021 9626820000021
## 0726820000021 0726820000021
## 1726820000021 1726820000021
## 2726820000021 2726820000021
## erutaef_on__   erutaef_on__
```

```
## Step 6: Relevel the design

dds$condition <- relevel(dds$Sample.type.3, ref = "CTRL_gut_CTRL")

### Step 7: Analyze Differential expression of genes

dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
## -- replacing outliers and refitting for 8814 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```r
dispersions
```

```
## standardGeneric for "dispersions" defined from package "DESeq2"
##
## function (object, ...)
## standardGeneric("dispersions")
## <bytecode: 0x7fd712a44220>
## <environment: 0x7fd713247698>
## Methods may be defined for arguments: object
## Use  showMethods("dispersions")  for currently available ones.
```

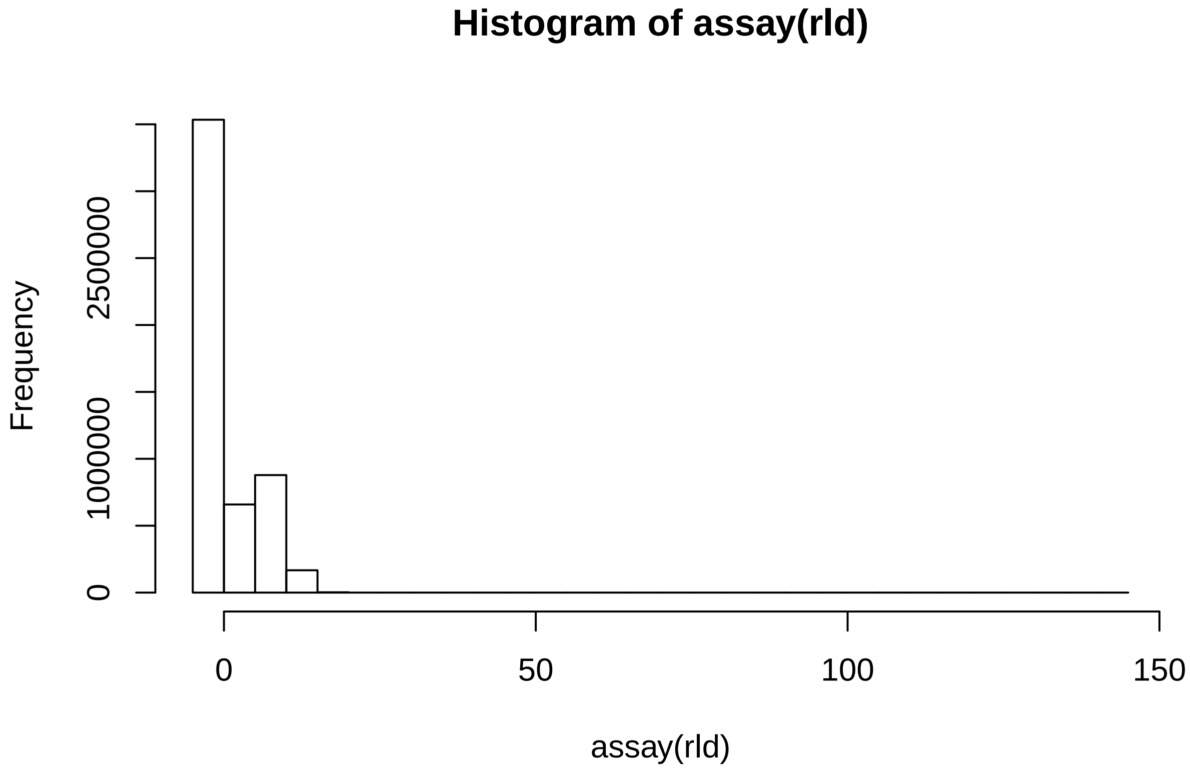
```r
# Regularized log transformation for clustering/heatmaps, etc
rld <- rlogTransformation(dds)
```

```
## rlog() may take a long time with 50 or more samples,
## vst() is a much faster transformation
```

```r
head(assay(rld))
```

```
##                  J11L_1   J11LN_1   J14N_1  J16LN_2    J1N_1    J22L_1   J22N_1
## 3000000000021 10.718680 10.153991 9.685769 9.829832 9.893464 10.129952 9.993270
## 5000000000021  3.643246  3.380268 3.300466 3.155225 3.513895  3.220763 3.639837
## 9140000000021  8.482604  7.786142 7.971661 7.633986 7.827865  7.700126 8.043492
## 7540000000021  7.235069  7.314915 7.433608 7.454977 7.632544  7.226185 7.635030
## 0640000000021  5.603465  5.755221 5.885160 5.445896 5.918108  5.694169 5.198483
## 8390000000021  5.006288  4.768813 5.104326 4.476990 5.382750  5.258261 4.892020
##                  J23N_2    J25N_1    J26N_2      J27L    J27N_2      J28L
## 3000000000021 10.215281 10.243697 10.353773 10.424892 10.333135 10.594496
## 5000000000021  3.501941  3.444321  3.386431  3.496852  3.273443  3.353447
## 9140000000021  8.069383  8.188890  8.141625  7.913618  7.784061  7.753066
## 7540000000021  7.864515  7.547848  7.628110  7.657610  7.660973  7.838427
## 0640000000021  5.723116  5.547292  5.735392  5.422472  5.940574  5.613552
## 8390000000021  5.168610  4.646826  5.233367  4.801593  4.859852  4.714291
##                  J28N_2    J2L_2     J2N_2   J30N_2   J35N_1    J3N_1    J41N_2
## 3000000000021 10.688079 9.975350 10.666419 9.734163 9.907792 9.965283 10.376829
## 5000000000021  3.239218 3.481944  3.594139 3.257812 3.351823 3.496528  3.493591
## 9140000000021  7.943805 7.839864  7.680881 7.824270 7.863487 8.084570  8.048361
## 7540000000021  7.877428 7.510636  7.597749 7.691235 7.654897 7.739519  7.721139
## 0640000000021  5.845200 5.762689  5.507104 5.559782 5.443627 5.693199  5.951574
## 8390000000021  5.005521 5.107666  4.543220 5.098439 5.045441 4.742601  4.803818
##                  J42N_2   J44N_1   J46N_1    J5N_1     J7L_1    J7LN_1
## 3000000000021 10.225658 9.801618 9.948672 9.900040 10.228274 10.116276
## 5000000000021  3.327477 3.321959 3.282337 3.377747  3.255387  3.261717
## 9140000000021  7.790875 7.831470 8.078099 7.807316  7.988289  8.039915
## 7540000000021  8.042225 7.807553 7.557032 7.702051  7.511894  7.329711
## 0640000000021  6.090478 5.854497 5.457877 5.747947  5.871665  6.045166
## 8390000000021  5.302475 4.984244 4.693409 4.978602  4.852921  4.899537
##                   J8N_2     J9L_1    J9LN_1  ME100N1 ME104NP1-II  ME104P1
## 3000000000021 10.594820 10.010198 10.549886 9.832703   10.046503 9.591814
## 5000000000021  3.415615  3.203411  3.507422 3.215425    3.253847 3.150659
## 9140000000021  7.948516  8.011071  7.900929 7.910254    7.857163 8.215951
## 7540000000021  7.723544  7.386263  7.760905 7.618210    8.121853 7.302607
## 0640000000021  5.444514  5.637580  6.098158 4.755572    4.989612 6.597359
## 8390000000021  4.676825  5.087077  4.662452 4.175873    4.266102 5.351013
##                ME105N1 ME124N1-I ME19N1-III    ME20N1  ME21N1-I ME21Np1-III
## 3000000000021 9.973188 10.545028   9.436230 10.717438 10.555802   10.991068
## 5000000000021 3.213771  3.660425   3.466475  3.500757  3.507494    3.803309
## 9140000000021 7.982217  7.768633   8.018577  7.825697  7.767053    7.897960
## 7540000000021 7.569156  7.684673   7.941254  7.600155  7.207442    7.710787
## 0640000000021 5.144241  5.351385   5.202466  5.302848  5.292931    5.523595
## 8390000000021 4.634181  4.153048   4.111274  4.346223  4.405327    4.595405
##                  ME21P1  ME23N1-I   ME23P1 ME25Np1-II    ME25P1    ME25T
## 3000000000021 10.333493 10.121593 9.608063  10.022029 10.047351 9.766409
## 5000000000021  3.123353  3.504231 3.707052   3.480900  3.141700 3.215151
## 9140000000021  8.351211  7.933387 7.627809   7.803449  8.786427 8.190142
## 7540000000021  7.071663  7.365512 7.358595   7.935343  7.287035 7.969176
## 0640000000021  6.601249  5.575773 5.712111   5.241929  6.390963 6.507363
## 8390000000021  5.758715  4.931405 5.106616   5.046008  5.249754 6.457724
##                 ME37N1    ME41N1  ME44N1-I ME46N1-III   ME50N1 ME55N1-I
## 3000000000021 9.868906 10.353443 10.502397  10.165640 9.953995 9.628544
## 5000000000021 3.972283  3.495409  3.805552   3.412922 3.217009 3.337078
## 9140000000021 8.215400  7.833215  8.074849   7.807384 7.561892 7.791067
## 7540000000021 7.690523  8.197915  7.987854   8.080007 7.487526 7.852152
## 0640000000021 5.192748  5.202521  4.983912   4.814164 5.799546 5.407874
## 8390000000021 4.023239  4.320731  4.120820   3.982990 4.168215 4.931103
##               ME56kohouma  ME56N-I ME59N1-III    ME61N1 ME62N1-I ME63N1-III
## 3000000000021   10.342535 9.619853  10.416771 10.218111 9.951586  10.359846
## 5000000000021    3.551607 3.344860   3.311216  3.269790 3.543605   3.544672
## 9140000000021    8.361791 7.776459   7.651878  7.872642 8.060477   7.358218
## 7540000000021    7.618978 7.467658   7.614954  7.878008 7.208661   7.945275
## 0640000000021    5.595828 5.198418   5.329789  5.289254 5.440795   4.649781
## 8390000000021    5.191546 4.403523   4.629244  4.998709 5.002330   4.077586
##                 ME72N1 ME73colonT ME73N1-I-1 ME73N1-I-2 ME73P1-II ME76N1-III
## 3000000000021 9.657910  10.060063  10.541475  10.155026  9.275954  10.126179
## 5000000000021 3.531985   3.116514   3.587446   3.428517  3.511625   3.481927
## 9140000000021 7.709331   8.684071   8.238018   7.832890  8.244315   7.956222
## 7540000000021 8.102311   7.608017   7.502189   7.802806  7.369992   7.773683
## 0640000000021 4.994514   6.184630   4.666992   4.929792  5.313152   4.926236
## 8390000000021 4.320354   5.519978   4.468952   4.467951  4.754815   4.318944
##                 ME76P1   ME83C1 ME83N1-III ME87N1-III     ME87P ME89colonCA
## 3000000000021 9.488365 9.907815   9.556023  10.344052 10.088328    9.511000
## 5000000000021 3.277185 3.386344   3.694226   3.314791  3.279351    3.399164
## 9140000000021 8.172598 8.307040   7.602596   8.069931  7.707190    8.209071
## 7540000000021 7.084886 7.033662   7.333669   7.700924  7.360632    7.275231
## 0640000000021 5.929241 6.225665   5.690270   5.192717  5.734180    6.207436
## 8390000000021 4.246710 5.638021   5.086448   4.312915  4.389072    5.250209
##               ME89colonN-II  ME89N1-I    ME89P1    ME99N1  MMR1002   MMR1101
## 3000000000021     10.610917 10.036035 10.279449 10.387405 9.865254 10.005305
## 5000000000021      3.722810  3.429749  3.320204  3.212489 3.304537  3.344697
## 9140000000021      7.658783  7.795066  7.667232  7.914625 8.074670  7.895542
## 7540000000021      7.389216  7.585769  7.630528  7.883307 7.790403  7.804368
## 0640000000021      5.738942  5.201199  4.671377  4.868572 5.615342  5.984027
## 8390000000021      5.131393  4.702651  4.651828  4.065199 4.476737  5.070061
##                 MMR1102 MMR13_N_1 MMR16_LN_4 MMR23_LN  MMR23_L MMR36_LN_4
## 3000000000021 10.175933  9.860202  11.167200 9.933088 9.146933  10.054264
## 5000000000021  3.391470  3.419061   3.165623 3.130046 3.238470   3.340899
## 9140000000021  7.957559  7.922797   8.439787 8.247898 7.867819   7.751146
## 7540000000021  7.962557  7.367459   8.009108 7.573149 7.008971   7.345524
## 0640000000021  5.549938  6.087211   5.494239 4.602237 5.515220   5.980983
## 8390000000021  5.365476  4.863942   4.406208 6.370907 6.063289   5.041273
##               MMR42_L_2 MMR42_LN_2 MMR43_N_1 MMR48_N_1   MMR612   MMR658
## 3000000000021  9.493886  10.789017 10.252997  9.464586 9.875003 9.860173
## 5000000000021  3.145070   3.372249  3.401205  3.358793 3.196057 3.214705
## 9140000000021  8.163841   8.067844  8.001036  8.169968 8.112043 7.799193
## 7540000000021  7.144412   7.784463  7.490042  7.520872 7.880167 7.601771
## 0640000000021  6.334178   5.749551  5.409999  5.781193 5.691980 5.834325
## 8390000000021  5.273352   4.640936  5.333312  5.254946 5.218956 4.726580
##                MMR7_L_4 MMR7_LN_4    MMR868
## 3000000000021 10.336825 10.174492 10.299542
## 5000000000021  3.631721  3.164188  3.370519
## 9140000000021  7.913070  7.830614  7.888624
## 7540000000021  7.639466  7.667006  8.024102
## 0640000000021  5.571007  5.455813  5.673397
## 8390000000021  5.146883  5.341954  4.780363
```

```
hist(assay(rld))
```



```
resultsNames(dds)
```

```
##  [1] "Intercept"
##  [2] "Sample.type.3_CTRL_gut_Cars_vs_CTRL_gut_Ade"
##  [3] "Sample.type.3_CTRL_gut_CTRL_vs_CTRL_gut_Ade"
##  [4] "Sample.type.3_CTRL_Tumor_Ade_vs_CTRL_gut_Ade"
##  [5] "Sample.type.3_CTRL_Tumor_Cars_vs_CTRL_gut_Ade"
##  [6] "Sample.type.3_LS_gut_Ade_vs_CTRL_gut_Ade"
##  [7] "Sample.type.3_LS_gut_Cars_vs_CTRL_gut_Ade"
##  [8] "Sample.type.3_LS_gut_CTRL_vs_CTRL_gut_Ade"
##  [9] "Sample.type.3_LS_Tumor_Ade_vs_CTRL_gut_Ade"
## [10] "Sample.type.3_LS_Tumor_Cars_vs_CTRL_gut_Ade"
```

```
library(RColorBrewer)
(mycols <- brewer.pal(8, "Dark2")[1:length(unique(dds$condition))])
```

```
##  [1] "#1B9E77" "#D95F02" "#7570B3" "#E7298A" "#66A61E" "#E6AB02" "#A6761D"
##  [8] "#666666" NA        NA
```

```
# Sample distance heatmap
sampleDists <- as.matrix(dist(t(assay(rld))))
library(gplots)
```

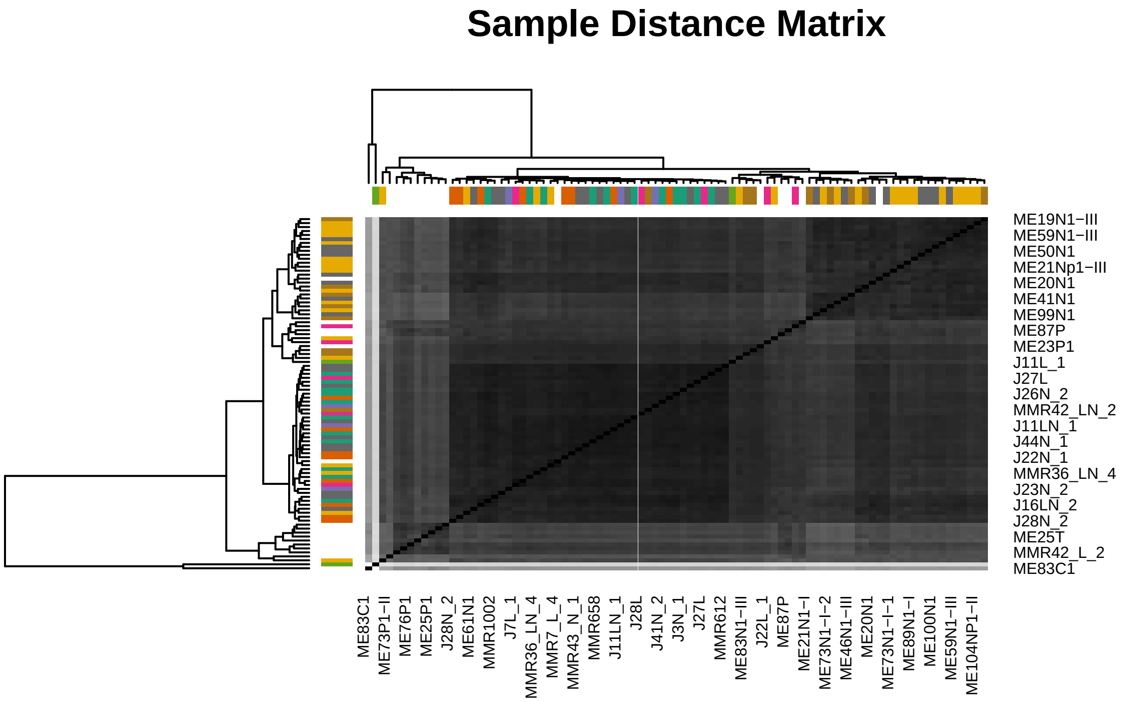
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:IRanges':
##
##     space
```

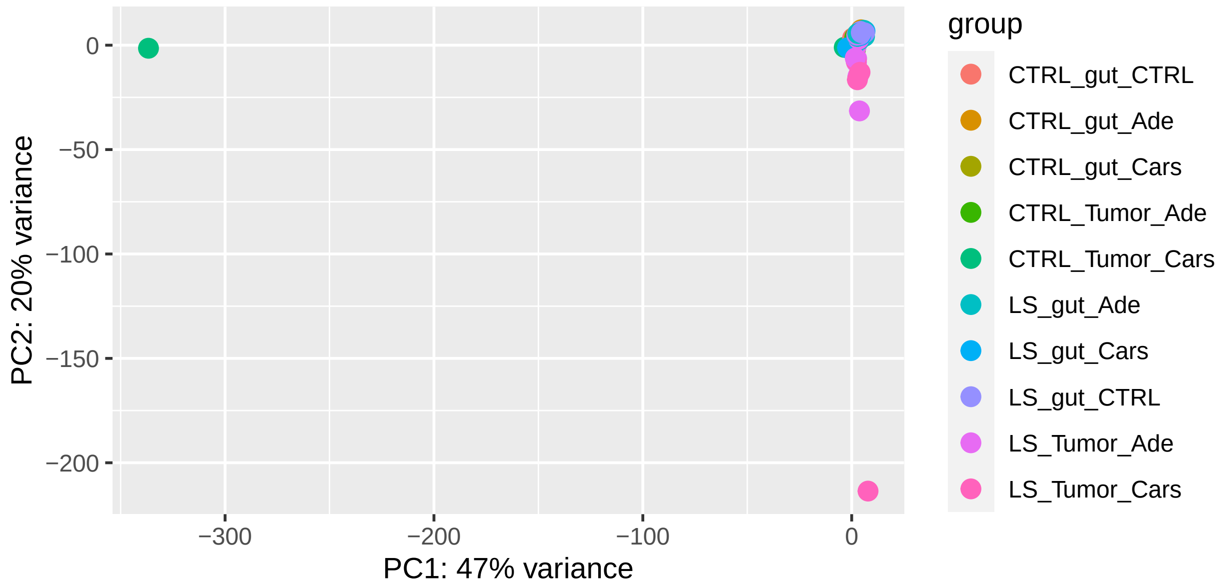
```
## The following object is masked from 'package:S4Vectors':
##
##     space
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
heatmap.2(as.matrix(sampleDists), key=F, trace="none",
          col=colorpanel(100, "black", "white"),
          ColSideColors=mycols[dds$condition], RowSideColors=mycols[dds$condition],
          margin=c(10, 10), main="Sample Distance Matrix")
```



```r
DESeq2::plotPCA(rld, intgroup="condition")
```

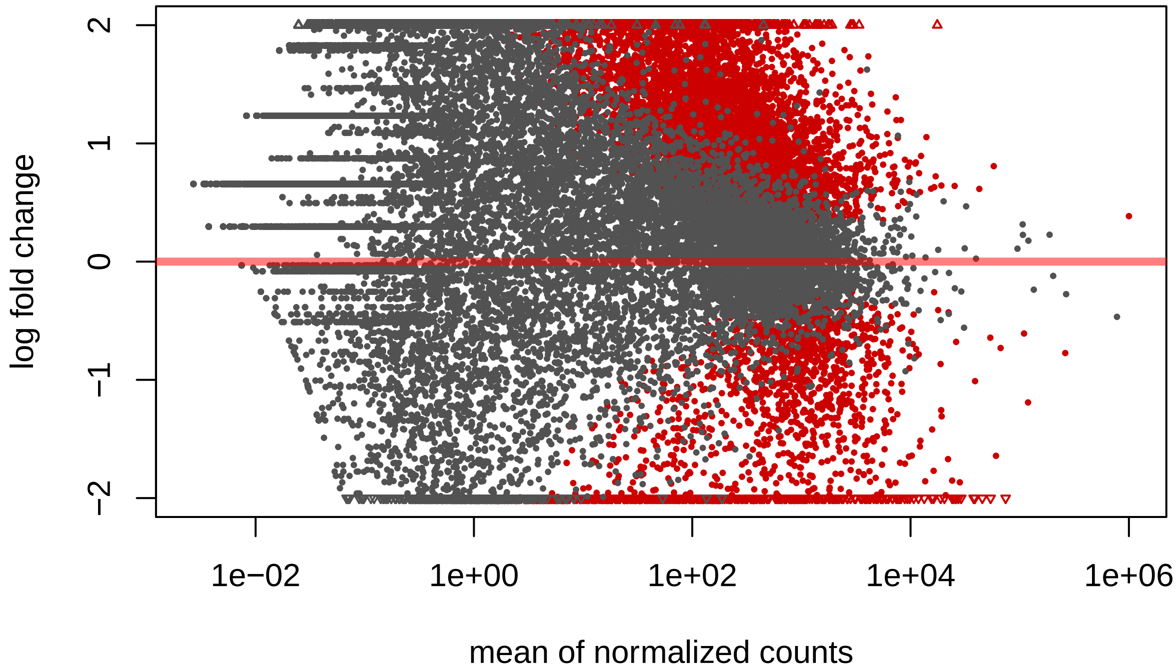


```r
### Step 8: Extract results from DESeq2 Object

# Get differential expression results
res <- results(dds)
table(res$padj<0.05)
```

```
##
## FALSE  TRUE
## 11306  8007
```

```r
## Order by adjusted p-value
res <- res[order(res$padj), ]
plotMA(res, ylim=c(-2,2))
```

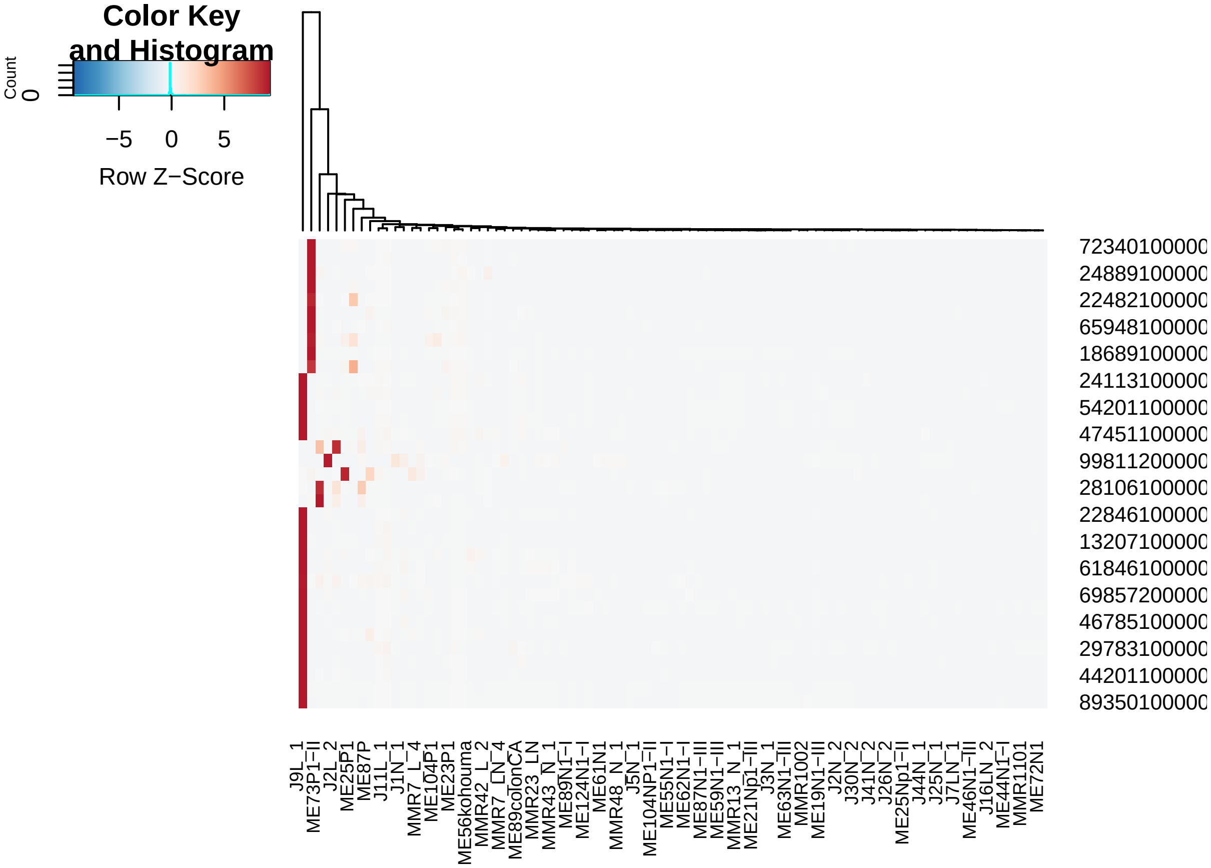


```
# Gene clustering
library( "genefilter" )
```

```
##
## Attaching package: 'genefilter'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

```
topVarGenes <- head( order( rowVars( assay(rld) ), decreasing=TRUE ), 35 )
heatmap.2( assay(rld)[ topVarGenes, ], scale="row",
           trace="none", dendrogram="column",
           col = colorRampPalette( rev(brewer.pal(9, "RdBu")) )(255))
```



```
## Merge with normalized count data
resdata <- merge(as.data.frame(res), as.data.frame(counts(dds, normalized=TRUE)), by="row.names", sort=
names(resdata)[1] <- "Gene"
head(resdata)
```

```
##             Gene  baseMean log2FoldChange     lfcSE       stat       pvalue
## 1 0144700000021 8147.7911      -2.663427 0.2286378 -11.649110 2.318690e-31
## 2 7161710000021  557.9673       3.094349 0.3031889  10.206010 1.863723e-24
## 3 2942810000021  201.6368       4.940083 0.4976506   9.926810 3.182737e-23
## 4 2868910000021 5000.4270      -2.354809 0.2368870  -9.940643 2.770342e-23
## 5 4303810000021  696.4234      -8.191907 0.8352500  -9.807731 1.042874e-22
## 6 3738810000021 2629.6828      -2.321348 0.2421037  -9.588237 8.960280e-22
##           padj  J11L_1    J11LN_1    J14N_1     J16LN_2     J1N_1    J22L_1
## 1 4.478086e-27 5350.637 7507.90035 8930.9484 11775.25457 7702.3516 8912.9600
## 2 1.799704e-20 1189.030  335.53142  532.8939   197.18688  373.7138  391.3292
## 3 1.536705e-19    0.000   69.26107  123.8260    65.95102  130.9830  136.6732
## 4 1.536705e-19 2972.576 5465.46826 5665.0383  5687.77571 4728.2125 4645.7198
## 5 4.028203e-19    0.000  304.74872  261.6561  1862.61664  180.4451  348.1077
## 6 2.884165e-18 2080.803 3186.77894 4246.1990  3955.06259 2093.8965 2719.4457
##       J22N_1    J23N_2      J25N_1     J26N_2      J27L     J27N_2       J28L
## 1 9395.54541 8775.2098 10859.72525 6531.41516 7422.4419 8830.00633 7863.16043
## 2  325.54149  489.4275   324.70780  378.89215  348.3607  385.21564  416.11918
## 3   28.98195  122.0253    23.02212   55.59623  107.1650   33.43644   44.20921
## 4 6013.41866 4256.2951  5673.51338 4260.72786 4691.6758 4743.09822 4928.22157
## 5  520.32719 1054.4577   839.34808 1240.63378  153.4339  712.61411  405.61949
## 6 2671.73185 3282.7455  2859.53900 2630.53964 2203.8966 3041.32279 4056.19492
##       J28N_2     J2L_2      J2N_2     J30N_2     J35N_1      J3N_1     J41N_2
## 1 10085.71357 5207.4796 7465.11669 8287.0361 9598.0018 8930.54869 7176.35166
## 2   244.78582  530.5287  279.23144  370.3678  373.9318  448.25816  310.83693
## 3    26.89954  165.2345   59.30579  103.4114   77.4842   74.42124   35.25697
## 4  6794.28584 4398.3493 5134.89324 4423.0022 5402.2483 7056.17190 4248.34454
## 5   960.85157  228.2162  929.12409  374.4678  423.1829  303.74250  814.98756
## 6  3501.78213 1184.7979 3270.46737 2890.9643 2796.9599 1976.48965 2689.84271
##      J42N_2     J44N_1     J46N_1      J5N_1     J7L_1      J7LN_1      J8N_2
## 1 6968.8476 7017.45281 7986.36721 8143.41970 8596.7185 10092.69641 7122.82608
## 2  375.3412  426.24003  332.44766  307.72328  355.9255   338.32856  386.80979
## 3  103.2479   85.51385   56.01561   87.27824  122.0939    79.81876   39.78615
## 4 3986.3019 5343.06500 3544.89302 5965.96258 6354.3333  5153.71732 5037.36870
## 5  845.3910  163.93847  837.25103  399.50040  258.9045   271.38380  452.56746
## 6 2979.0523 2436.03710 2429.88413 2642.19125 2846.3140  2351.30628 3971.98401
##       J9L_1     J9LN_1   ME100N1 ME104NP1-II     ME104P1    ME105N1  ME124N1-I
## 1 1668.7160 8438.12233 6973.6936   9125.3203 3222.112849 10441.1286 9004.99752
## 2  313.1161  326.85449  178.4899    266.7094 1489.920350   303.6667  140.81201
## 3   91.2637   49.10146  108.6958     99.2407  282.828720   120.8136   50.19042
## 4 6500.4982 6880.79994 6144.7452   5771.1225 1362.914246  4199.0904 5933.62277
## 5  234.4661  835.45766  688.2160    647.1321    4.269113  1483.5045  421.04184
## 6 1141.9092 2390.58147 2751.1479   4383.8202 2008.617549  3573.2540 3022.57852
##    ME19N1-III      ME20N1    ME21N1-I ME21Np1-III      ME21P1  ME23N1-I
## 1 11480.01138 10341.78524 12630.67711  9463.93024 3170.958120 5724.2337
## 2   486.30224   309.64286   255.60401   231.60361 1639.874911  320.4782
## 3    99.13384    73.95168    88.90574    59.19581    7.999390  268.7086
## 4  6117.41682  6244.38481  8252.06932  6643.98980 1471.887725 5189.2817
## 5  1425.34173  3191.85012   281.87161  1342.26501    1.599878  559.6042
## 6  3023.19196  3978.12351  2178.19068  3681.97943 1855.858435 2811.5799
##      ME23P1 ME25Np1-II      ME25P1       ME25T     ME37N1      ME41N1
## 1 6771.6009 9640.39539 2186.960946 2215.833256 8737.96057 17964.82684
## 2  451.4401  257.97961 2613.644222 3207.272351  269.37714   282.72768
## 3  451.4401  147.70129  974.678463 1352.464244   59.57988    23.23789
## 4 4062.9605 5043.74021 1162.921085 1503.719429 6955.63465  6853.88708
## 5 1354.3202   72.45724    1.673268    3.312157  671.22445  1773.82574
## 6 3160.0804 3003.39223  834.960606  792.709655 1789.93186  2734.32527
##     ME44N1-I ME46N1-III     ME50N1  ME55N1-I ME56kohouma   ME56N-I ME59N1-III
## 1 8933.23878 9981.49765 9647.37768 8397.0088   4917.6674 9082.3810 11330.4847
## 2  288.91275  353.84409  281.29746  320.1665    997.1593  262.7483   260.5997
## 3   61.64099   70.36956   93.39076  203.0019    251.4576  136.2919   121.8934
## 4 8392.11554 5115.01847 7076.31890 6655.8293   3261.5160 6893.2799  4661.3729
## 5  541.59378 1730.29262  466.95378  675.4197    315.8703  824.7769  1742.9359
## 6 2175.31540 2106.09600 3148.28117 2295.0487   2958.0327 2964.7005  3006.7046
##       ME61N1  ME62N1-I ME63N1-III      ME72N1  ME73colonT ME73N1-I-1 ME73N1-I-2
## 1 14309.7228 8073.6439 9464.92871 15782.78595 1410.514331  7154.3232 9837.05092
## 2   485.9587  460.2029  207.39400   373.24809 2625.350373   116.5440  134.44502
## 3   220.8903  215.1935   16.97963    89.60533 2425.142494   100.1293   47.94892
## 4  4885.3993 6046.1585 6481.36584  5771.48563  982.921033  6261.3665 8314.90761
## 5  2586.1543  208.7117  811.38356  3426.92032    1.811836   535.1174 1114.10736
## 6  4043.0378 3450.8735 3645.76828  3303.14893  595.188128  3313.2962 2724.62712
##   ME73P1-II ME76N1-III      ME76P1       ME83C1 ME83N1-III  ME87N1-III
## 1 4744.2266  9988.9694 5957.757664 1122.9821831  8301.3869 12541.23463
## 2  421.6276   267.0103  502.630132 2645.9808955     0.0000   306.02956
## 3  150.3194    40.8778  134.246782 2629.0994640   415.0693    78.89824
## 4 2812.0725  6307.7929 1703.852523  734.7092584  5395.9015  5235.97442
## 5  342.8015  1843.4150    1.908722    0.7339753   830.1387  2218.71428
## 6 1063.2347  3916.4414 2350.909288  634.8886199  2075.3467  3387.84281
##        ME87P ME89colonCA ME89colonN-II   ME89N1-I     ME89P1      ME99N1
## 1 7939.97098 1764.289206     7007.1856 11173.7921 10958.3225 15319.22765
## 2 1293.84999 3130.120480     1001.0265   648.0083   626.8059   303.91458
## 3  196.09614  776.287251        0.0000   181.1966   229.0109    20.40416
## 4 2481.96187 1639.431816     3003.0796  8328.9026  5839.7791  5994.52708
## 5   99.97058    6.514299      500.5133   314.2789   932.4017  1570.04633
## 6 2671.32922  309.429184     2002.0530  2570.5352  2994.9873  2931.75545
##      MMR1002    MMR1101    MMR1102  MMR13_N_1 MMR16_LN_4   MMR23_LN    MMR23_L
## 1 9005.17627 8896.59593 8682.59942 8404.85997 8391.47767 8017.64975 5239.81000
## 2  361.33566  428.86643  486.06308  263.89580  376.98818  180.95390  780.54511
## 3   41.86761   78.89025   61.63336   82.55958   20.94379   90.47695  316.49500
## 4 5181.57151 3978.39801 4268.81048 6293.69360 7288.43818 4528.48736 4075.94240
## 5 1299.26108  875.20521  305.36528  740.08765  376.98818  837.49177   12.83088
## 6 2684.53279 2579.02272 2509.45823 2594.72962 1103.03949 4848.63657  855.92652
##   MMR36_LN_4   MMR42_L_2 MMR42_LN_2  MMR43_N_1  MMR48_N_1     MMR612     MMR658
## 1  9271.2455  766.705292 7904.36003 8281.79381 7691.73637 6850.09598 7991.30432
## 2   307.0299 1261.234819  468.12799  394.97268  311.10603  383.08292  308.32460
## 3   150.8746 1219.716482   40.01684   54.74035   40.02222   72.47515   68.70276
## 4  4820.4442  400.421296 4855.49110 5521.19587 4943.54418 5051.75663 5444.27505
## 5   277.6093    1.845259  249.07031  239.17322  451.45060  518.07783  568.89239
## 6  3238.5237  651.376578 2708.38086 2518.05608 2128.11466 2766.40023 2399.56968
##     MMR7_L_4  MMR7_LN_4     MMR868
## 1 6950.71065 7687.47990 8238.43920
## 2  299.73365  260.24075  357.49149
## 3   80.96562   91.80272   62.01897
## 4 4880.99352 5908.90199 4449.23113
## 5  194.10301  500.52439  749.26988
## 6 2747.46905 2527.36878 2429.83282
```

```
### Step 9: Extract the overall view of the results

output <- as.data.frame(res)
summary(output)
```

```
##     baseMean         log2FoldChange        lfcSE            stat
##  Min.   :      0.0   Min.   :-15.489   Min.   :0.000   Min.   :-11.649
##  1st Qu.:      0.0   1st Qu.:  0.125   1st Qu.:0.345   1st Qu.:  0.046
##  Median :      0.0   Median :  0.657   Median :0.994   Median :  0.339
##  Mean   :    219.5   Mean   :  0.835   Mean   :2.922   Mean   :  0.851
##  3rd Qu.:      8.2   3rd Qu.:  1.392   3rd Qu.:7.809   3rd Qu.:  1.998
##  Max.   :1000588.9   Max.   : 26.161   Max.   :7.845   Max.   : 10.206
##                      NA's   :28232     NA's   :28232   NA's   :28232
##      pvalue           padj
##  Min.   :0.000   Min.   :0.00
##  1st Qu.:0.016   1st Qu.:0.01
##  Median :0.376   Median :0.11
##  Mean   :0.439   Mean   :0.26
##  3rd Qu.:0.875   3rd Qu.:0.46
##  Max.   :1.000   Max.   :1.00
##  NA's   :28280   NA's   :39572
```