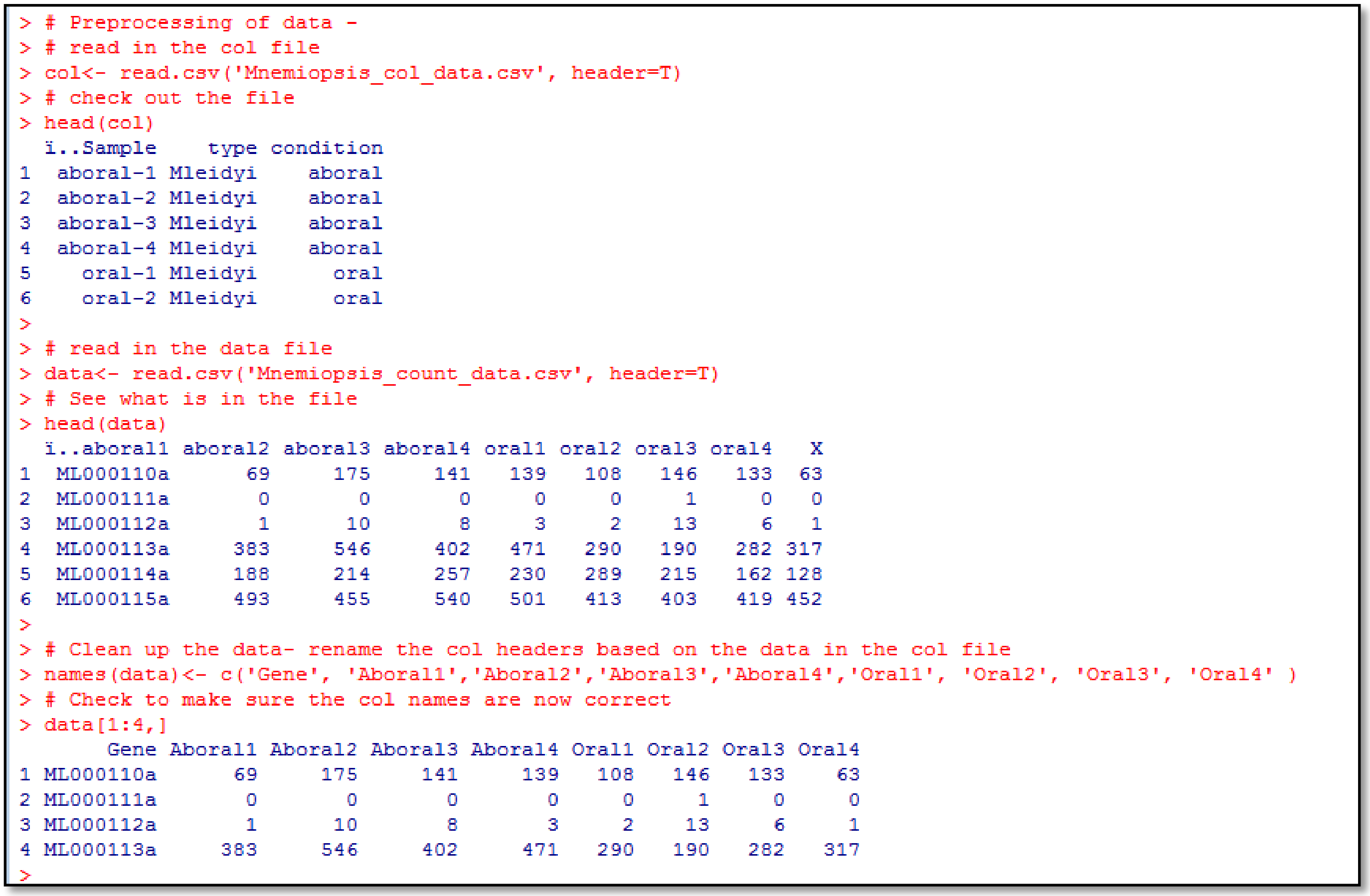
Dataset 2:

a) Mnemiopsis_col_data.csv b) Mnemiopsis_count_data.csv

This is gene expression data, the columns represent samples, whose information is in the col_data file. The count_data file contains counts for each gene (rows). The file, info_gene.txt contains information about the organism and some links to look up gene functions. It will be a good experience to learn to use the genome resources, as this is the kind of struggles most researchers go through when they start looking at genes.

First let's read in and clean up our data:

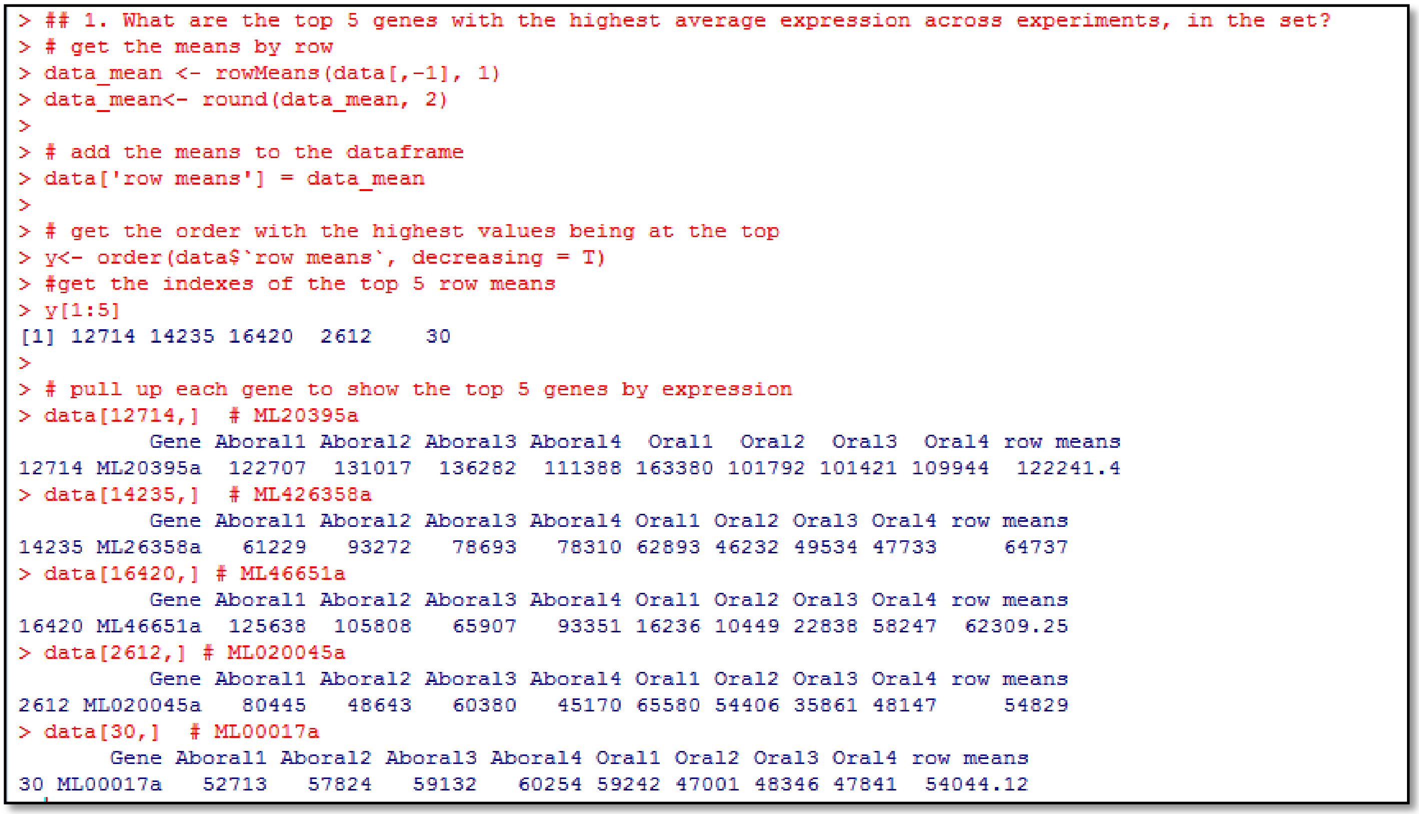


```
> # Preprocessing of data -
> # read in the col file
> col<- read.csv('Mnemiopsis_col_data.csv', header=T)
> # check out the file
> head(col)
  i..Sample    type condition
1  aboral-1 Mleidyi    aboral
2  aboral-2 Mleidyi    aboral
3  aboral-3 Mleidyi    aboral
4  aboral-4 Mleidyi    aboral
5    oral-1 Mleidyi      oral
6    oral-2 Mleidyi      oral
>
> # read in the data file
> data<- read.csv('Mnemiopsis_count_data.csv', header=T)
> # See what is in the file
> head(data)
  i..aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4   X
1  ML000110a      69     175     141   139   108   146   133  63
2  ML000111a       0       0       0     0     0     1     0   0
3  ML000112a       1      10       8     3     2    13     6   1
4  ML000113a     383     546     402   471   290   190   282 317
5  ML000114a     188     214     257   230   289   215   162 128
6  ML000115a     493     455     540   501   413   403   419 452
>
> # Clean up the data- rename the col headers based on the data in the col file
> names(data)<- c('Gene', 'Aboral1','Aboral2','Aboral3','Aboral4','Oral1', 'Oral2', 'Oral3', 'Oral4' )
> # Check to make sure the col names are now correct
> data[1:4,]
       Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4
1 ML000110a      69     175     141     139   108   146   133    63
2 ML000111a       0       0       0       0     0     1     0     0
3 ML000112a       1      10       8       3     2    13     6     1
4 ML000113a     383     546     402     471   290   190   282   317
>
```

1. What are the top 5 genes with the highest average expression (across experiments) in the set? What is their function?

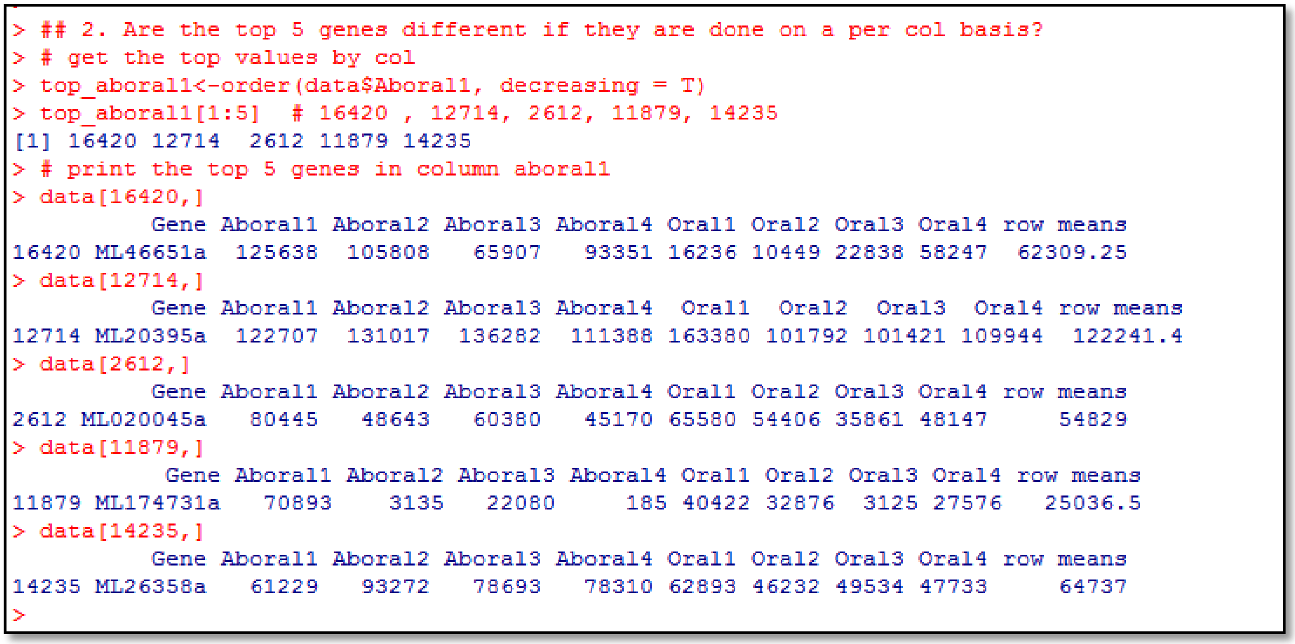
   Top 5 genes -

   - ***ML20395a* –** this gene is a protein coding gene – elongation factor 1- alpha – this is used in the larval and embryo development, gamete generation, growth regulation, locomotion, GTP binding, GTPase activity.
   - ***ML26358a* -** this gene is also a protein coding gene – Actin related protein – used in cytoskeleton organization, cytokinesis, embryo development, protein binding, and ATP binding.
   - ***ML46651a* -** this gene is also a protein coding gene – Membrane attack complex
   - ***ML020045a* –**also a protein coding gene - Beta-tubulin chain – used for microtubule-based processes and movements, structural constituents of the cytoskeleton, nucleotide binding, protein binding, GTP binding, and GTPase activity.
   - ***ML00017a* –** This gene is also a protein coding gene – Elongation factor 2 – used for translational elongation, embryo and larval development, growth, hermaphrodite genitalia development, oogenesis, regulation of translational elongation, GTP catabolic processes, GTP catabolic processes, nucleotide binding, GTP binding, GTPase activity



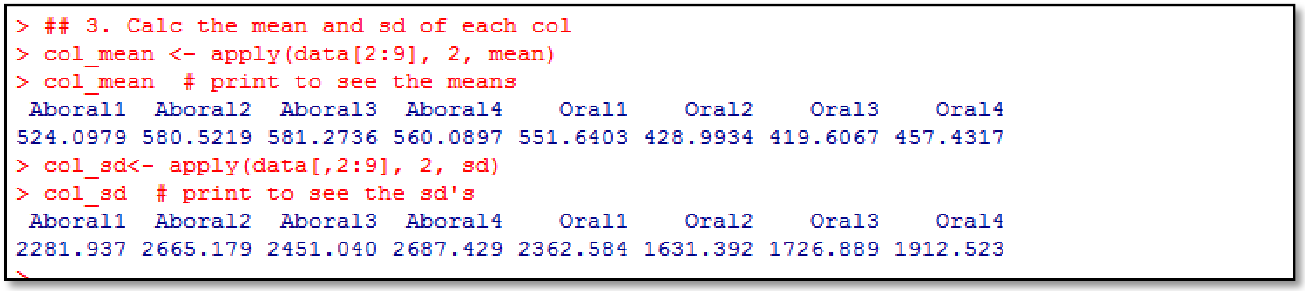
```
> ## 1. What are the top 5 genes with the highest average expression across experiments, in the set?
> # get the means by row
> data_mean <- rowMeans(data[,-1], 1)
> data_mean<- round(data_mean, 2)
>
> # add the means to the dataframe
> data['row means'] = data_mean
>
> # get the order with the highest values being at the top
> y<- order(data$`row means`, decreasing = T)
> #get the indexes of the top 5 row means
> y[1:5]
[1] 12714 14235 16420  2612    30
>
> # pull up each gene to show the top 5 genes by expression
> data[12714,]  # ML20395a
          Gene Aboral1 Aboral2 Aboral3 Aboral4  Oral1  Oral2  Oral3  Oral4 row means
12714 ML20395a  122707  131017  136282  111388 163380 101792 101421 109944  122241.4
> data[14235,]  # ML426358a
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
14235 ML26358a   61229   93272   78693   78310 62893 46232 49534 47733     64737
> data[16420,] # ML46651a
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
16420 ML46651a  125638  105808   65907   93351 16236 10449 22838 58247  62309.25
> data[2612,] # ML020045a
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
2612 ML020045a   80445   48643   60380   45170 65580 54406 35861 48147     54829
> data[30,]  # ML00017a
       Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
30 ML00017a   52713   57824   59132   60254 59242 47001 48346 47841  54044.12
```

2. Are the top 5 genes different if they are done on a per column basis?
   Yes, while many are the same, the top 5 genes fluctuate based on the column -
   - **Aboral1 top 5 -** *ML46651a, ML20395a , ML0200045a, ML174731a, ML26358a*
   - **Aboral2 top 5** - *ML20395a, ML46651a, ML26358a, ML01482a, ML034334a*
   - **Aboral3 top 5** - *ML20395a, ML01482a, ML26358a, ML46651a, ML034334a*
   - **Aboral4 top 5** - *ML01482a, ML20395a, ML034334a, ML46651a, ML034336a*
   - **Oral1 top 5 -** *ML20395a, ML020045a, ML04011a, ML26358a, ML00017a*
   - **Oral2 top 5** - *ML20395a, ML020045a, ML04011a, ML00017a, ML26358a*
   - **Oral3 top 5** - *ML20395a, ML004510a, ML26358a, ML00017a, ML04011a*
   - **Oral4 top 5** - *ML20395a, ML004510a, ML46651a, ML020045a, ML00017a*



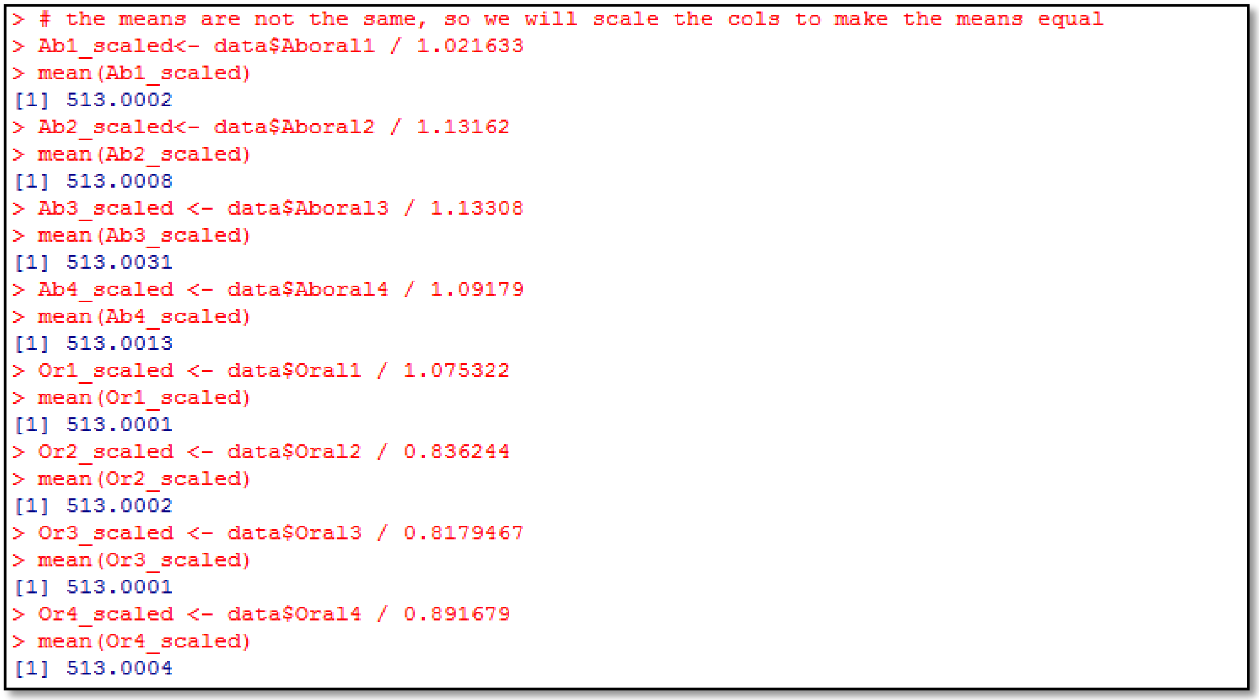
```
> ## 2. Are the top 5 genes different if they are done on a per col basis?
> # get the top values by col
> top_aboral1<-order(data$Aboral1, decreasing = T)
> top_aboral1[1:5]  # 16420 , 12714, 2612, 11879, 14235
[1] 16420 12714  2612 11879 14235
> # print the top 5 genes in column aboral1
> data[16420,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
16420 ML46651a  125638  105808   65907   93351 16236 10449 22838 58247  62309.25
> data[12714,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4  Oral1  Oral2  Oral3  Oral4 row means
12714 ML20395a  122707  131017  136282  111388 163380 101792 101421 109944  122241.4
> data[2612,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
2612 ML020045a   80445   48643   60380   45170 65580 54406 35861 48147     54829
> data[11879,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
11879 ML174731a   70893    3135   22080     185 40422 32876  3125 27576   25036.5
> data[14235,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means
14235 ML26358a   61229   93272   78693   78310 62893 46232 49534 47733     64737
>
```

3. Calculate mean and standard deviation of each column



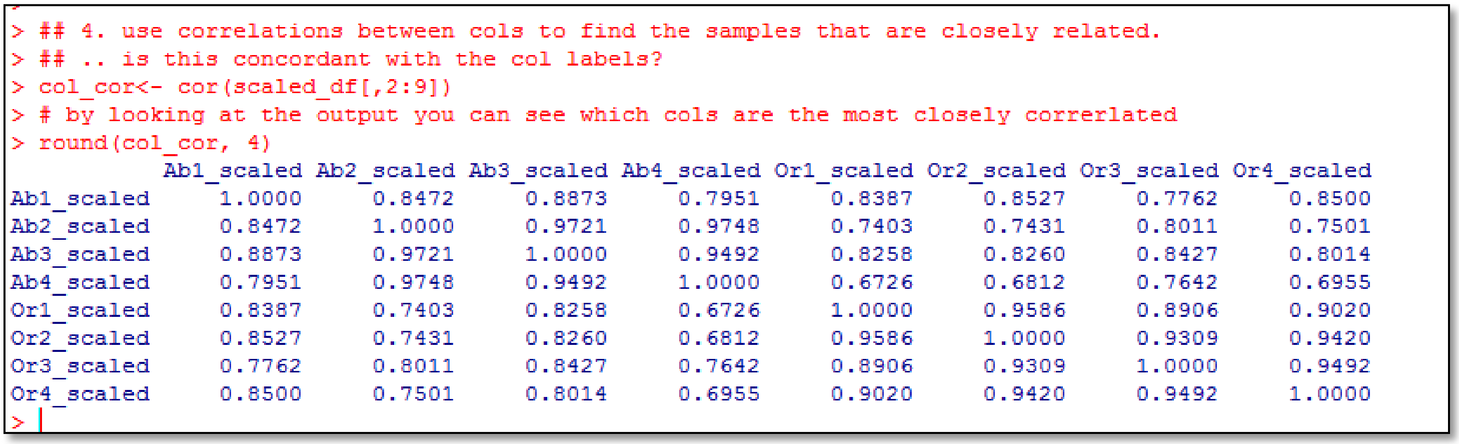
```
> ## 3. Calc the mean and sd of each col
> col_mean <- apply(data[2:9], 2, mean)
> col_mean  # print to see the means
 Aboral1  Aboral2  Aboral3  Aboral4    Oral1    Oral2    Oral3    Oral4 
524.0979 580.5219 581.2736 560.0897 551.6403 428.9934 419.6067 457.4317 
> col_sd<- apply(data[,2:9], 2, sd)
> col_sd  # print to see the sd's
 Aboral1  Aboral2  Aboral3  Aboral4    Oral1    Oral2    Oral3    Oral4 
2281.937 2665.179 2451.040 2687.429 2362.584 1631.392 1726.889 1912.523 
>
```

If the mean is different, then scale the columns so that they all have the same mean for the subsequent questions



```
> # the means are not the same, so we will scale the cols to make the means equal
> Ab1_scaled<- data$Aboral1 / 1.021633
> mean(Ab1_scaled)
[1] 513.0002
> Ab2_scaled<- data$Aboral2 / 1.13162
> mean(Ab2_scaled)
[1] 513.0008
> Ab3_scaled <- data$Aboral3 / 1.13308
> mean(Ab3_scaled)
[1] 513.0031
> Ab4_scaled <- data$Aboral4 / 1.09179
> mean(Ab4_scaled)
[1] 513.0013
> Or1_scaled <- data$Oral1 / 1.075322
> mean(Or1_scaled)
[1] 513.0001
> Or2_scaled <- data$Oral2 / 0.836244
> mean(Or2_scaled)
[1] 513.0002
> Or3_scaled <- data$Oral3 / 0.8179467
> mean(Or3_scaled)
[1] 513.0001
> Or4_scaled <- data$Oral4 / 0.891679
> mean(Or4_scaled)
[1] 513.0004
```

4. Use correlations between columns to find the samples that are closely related. Is this concordant with the column labels?

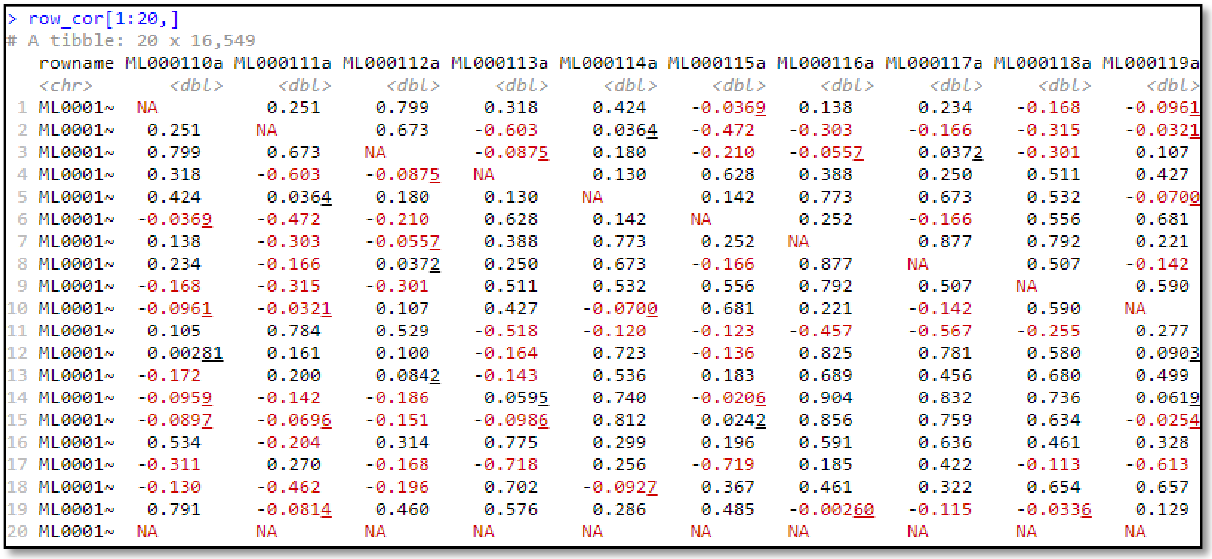


```
> ## 4. use correlations between cols to find the samples that are closely related.
> ## .. is this concordant with the col labels?
> col_cor<- cor(scaled_df[,2:9])
> # by looking at the output you can see which cols are the most closely correrlated
> round(col_cor, 4)
           Ab1_scaled Ab2_scaled Ab3_scaled Ab4_scaled Or1_scaled Or2_scaled Or3_scaled Or4_scaled
Ab1_scaled     1.0000     0.8472     0.8873     0.7951     0.8387     0.8527     0.7762     0.8500
Ab2_scaled     0.8472     1.0000     0.9721     0.9748     0.7403     0.7431     0.8011     0.7501
Ab3_scaled     0.8873     0.9721     1.0000     0.9492     0.8258     0.8260     0.8427     0.8014
Ab4_scaled     0.7951     0.9748     0.9492     1.0000     0.6726     0.6812     0.7642     0.6955
Or1_scaled     0.8387     0.7403     0.8258     0.6726     1.0000     0.9586     0.8906     0.9020
Or2_scaled     0.8527     0.7431     0.8260     0.6812     0.9586     1.0000     0.9309     0.9420
Or3_scaled     0.7762     0.8011     0.8427     0.7642     0.8906     0.9309     1.0000     0.9492
Or4_scaled     0.8500     0.7501     0.8014     0.6955     0.9020     0.9420     0.9492     1.0000
> |
```

Highest correlations between columns – The correlations do seem concordant with the column labels.

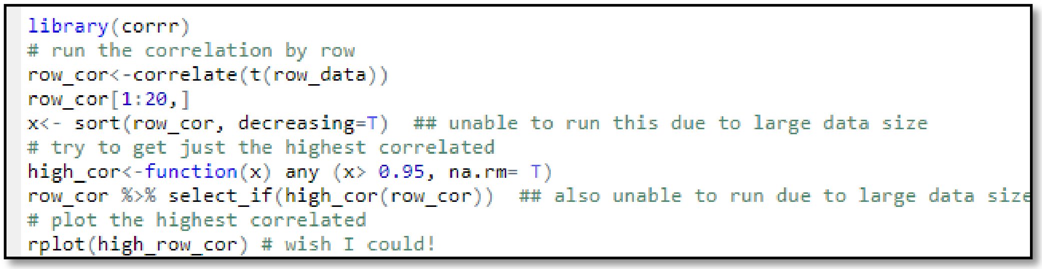
- Aboral1 = Aboral3
- Aboral2 = Aboral4
- Aboral3 = Aboral2
- Aboral4 = Aboral2
- Oral1 = Oral2
- Oral2 = Oral1
- Oral3 = Oral4
- Oral4 = Oral3

5. Use correlations between rows to find the closest pairs (top 5). Are these close because they vary a lot between the groups you found in question 2 or are they close because they don't vary much? **They are close because they do not vary much.**



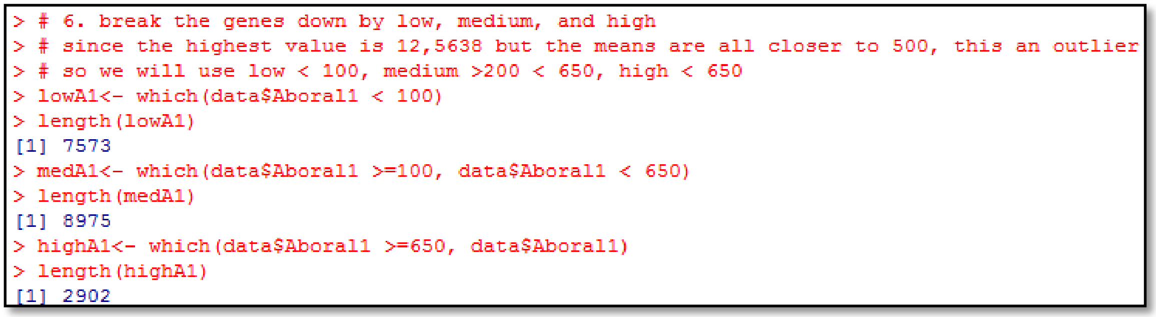
```
> row_cor[1:20,]
# A tibble: 20 x 16,549
   rowname ML000110a ML000111a ML000112a ML000113a ML000114a ML000115a ML000116a ML000117a ML000118a ML000119a
   <chr>       <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
 1 ML0001~  NA          0.251     0.799     0.318     0.424    -0.0369    0.138     0.234    -0.168    -0.0961
 2 ML0001~   0.251     NA         0.673    -0.603     0.0364   -0.472    -0.303    -0.166    -0.315    -0.0321
 3 ML0001~   0.799      0.673    NA        -0.0875    0.180    -0.210    -0.0557    0.0372   -0.301     0.107
 4 ML0001~   0.318     -0.603    -0.0875   NA         0.130     0.628     0.388     0.250     0.511     0.427
 5 ML0001~   0.424      0.0364    0.180     0.130    NA         0.142     0.773     0.673     0.532    -0.0700
 6 ML0001~  -0.0369    -0.472    -0.210     0.628     0.142    NA         0.252    -0.166     0.556     0.681
 7 ML0001~   0.138     -0.303    -0.0557    0.388     0.773     0.252    NA         0.877     0.792     0.221
 8 ML0001~   0.234     -0.166     0.0372    0.250     0.673    -0.166     0.877    NA         0.507    -0.142
 9 ML0001~  -0.168     -0.315    -0.301     0.511     0.532     0.556     0.792     0.507    NA         0.590
10 ML0001~  -0.0961    -0.0321    0.107     0.427    -0.0700    0.681     0.221    -0.142     0.590    NA
11 ML0001~   0.105      0.784     0.529    -0.518    -0.120    -0.123    -0.457    -0.567    -0.255     0.277
12 ML0001~   0.00281    0.161     0.100    -0.164     0.723    -0.136     0.825     0.781     0.580     0.0903
13 ML0001~  -0.172      0.200     0.0842   -0.143     0.536     0.183     0.689     0.456     0.680     0.499
14 ML0001~  -0.0959    -0.142    -0.186     0.0595    0.740    -0.0206    0.904     0.832     0.736     0.0619
15 ML0001~  -0.0897    -0.0696   -0.151    -0.0986    0.812     0.0242    0.856     0.759     0.634    -0.0254
16 ML0001~   0.534     -0.204     0.314     0.775     0.299     0.196     0.591     0.636     0.461     0.328
17 ML0001~  -0.311      0.270    -0.168    -0.718     0.256    -0.719     0.185     0.422    -0.113    -0.613
18 ML0001~  -0.130     -0.462    -0.196     0.702    -0.0927    0.367     0.461     0.322     0.654     0.657
19 ML0001~   0.791     -0.0814    0.460     0.576     0.286     0.485    -0.00260  -0.115    -0.0336    0.129
20 ML0001~  NA         NA        NA        NA        NA        NA        NA        NA        NA        NA
```

Note: I was not able to get the highest correlated due to the large size of the matrix (16548 x 16548)

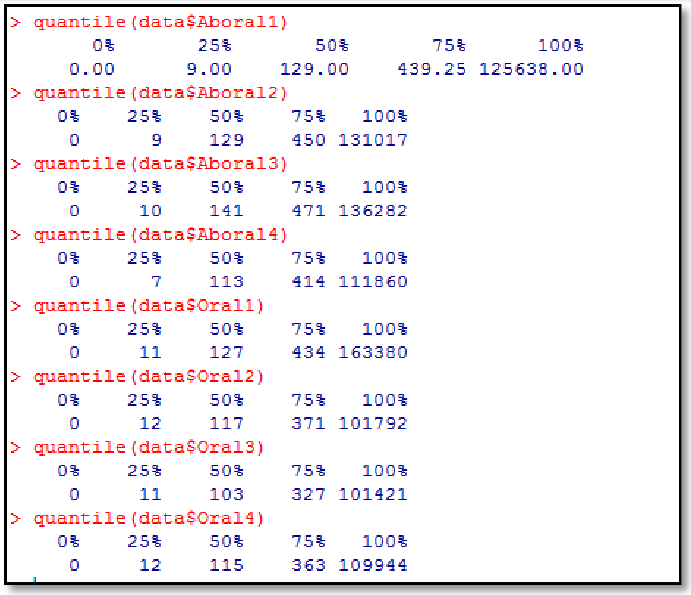


```
library(corrr)
# run the correlation by row
row_cor<-correlate(t(row_data))
row_cor[1:20,]
x<- sort(row_cor, decreasing=T)  ## unable to run this due to large data size
# try to get just the highest correlated
high_cor<-function(x) any (x> 0.95, na.rm= T)
row_cor %>% select_if(high_cor(row_cor))  ## also unable to run due to large data size
# plot the highest correlated
rplot(high_row_cor) # wish I could!
```

6. If you were forced to divide the genes in each column into high, medium and low count genes, how would you do this based on the data that you have? **See examples of both options below -**



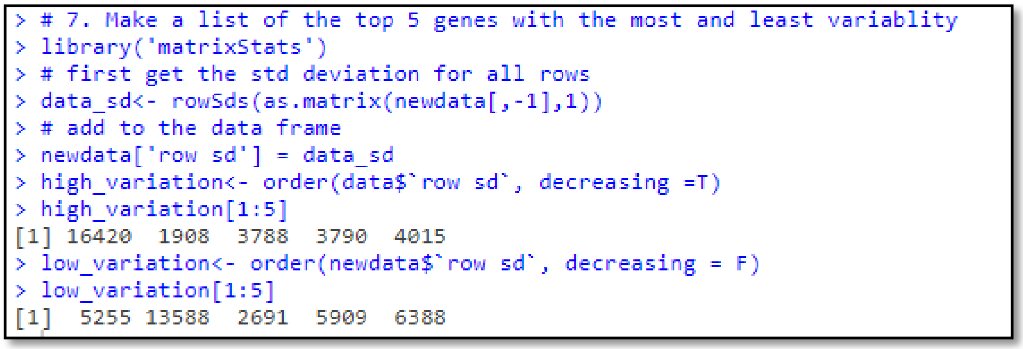
```
> # 6. break the genes down by low, medium, and high
> # since the highest value is 12,5638 but the means are all closer to 500, this an outlier
> # so we will use low < 100, medium >200 < 650, high < 650
> lowA1<- which(data$Aboral1 < 100)
> length(lowA1)
[1] 7573
> medA1<- which(data$Aboral1 >=100, data$Aboral1 < 650)
> length(medA1)
[1] 8975
> highA1<- which(data$Aboral1 >=650, data$Aboral1)
> length(highA1)
[1] 2902
```



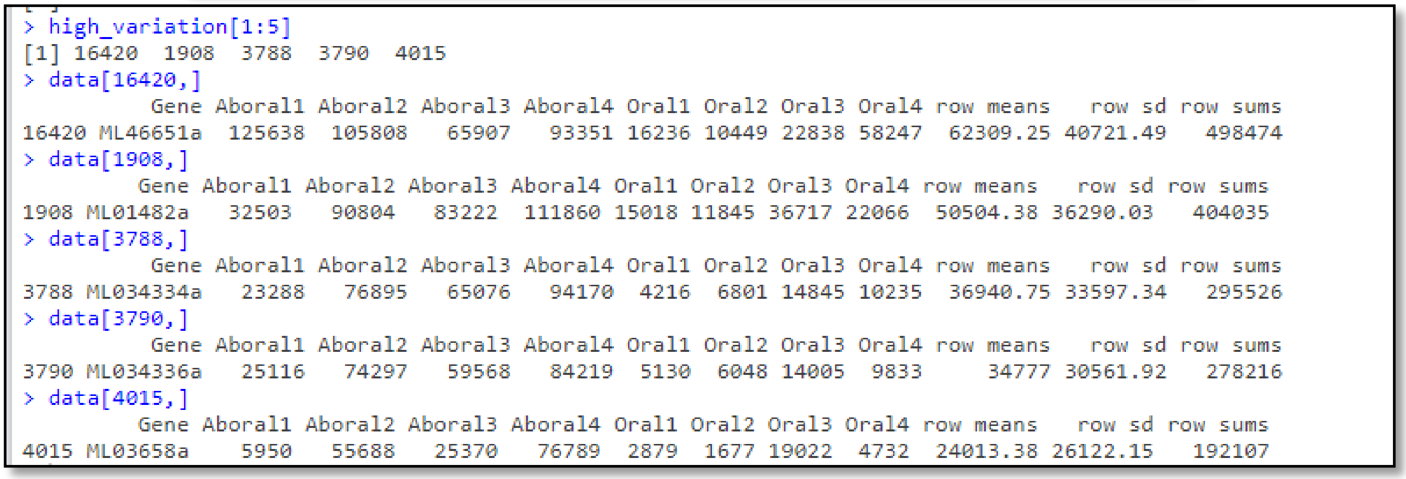
```
> quantile(data$Aboral1)
       0%       25%       50%       75%      100% 
     0.00      9.00    129.00    439.25 125638.00 
> quantile(data$Aboral2)
    0%    25%    50%    75%   100% 
     0      9    129    450 131017 
> quantile(data$Aboral3)
    0%    25%    50%    75%   100% 
     0     10    141    471 136282 
> quantile(data$Aboral4)
    0%    25%    50%    75%   100% 
     0      7    113    414 111860 
> quantile(data$Oral1)
    0%    25%    50%    75%   100% 
     0     11    127    434 163380 
> quantile(data$Oral2)
    0%    25%    50%    75%   100% 
     0     12    117    371 101792 
> quantile(data$Oral3)
    0%    25%    50%    75%   100% 
     0     11    103    327 101421 
> quantile(data$Oral4)
    0%    25%    50%    75%   100% 
     0     12    115    363 109944 
```

7. Make a list of the top 5 genes with most variability and top 5 genes with least variability (exclude genes that have low expression values. **The genes with the most variability have the highest standard deviations. The ones with the least variability have the lowest standard deviations (eliminating genes with a value < 5).**

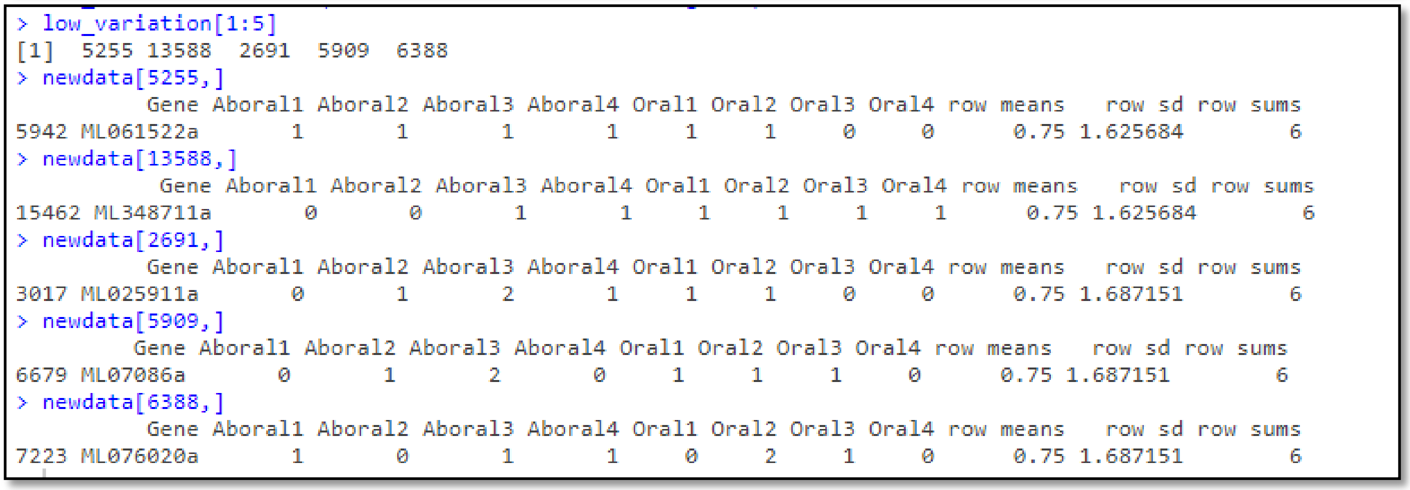
Top 5 with most variation – *ML46651a, ML01482a, ML034334a, ML0343336a, ML03658a*
Top 5 with least variation – *ML061522a, ML348711a, ML025911a, ML07086a, ML076020a*



```
> # 7. Make a list of the top 5 genes with the most and least variablity
> library('matrixStats')
> # first get the std deviation for all rows
> data_sd<- rowSds(as.matrix(newdata[,-1],1))
> # add to the data frame
> newdata['row sd'] = data_sd
> high_variation<- order(data$`row sd`, decreasing =T)
> high_variation[1:5]
[1] 16420  1908  3788  3790  4015
> low_variation<- order(newdata$`row sd`, decreasing = F)
> low_variation[1:5]
[1]  5255 13588  2691  5909  6388
```



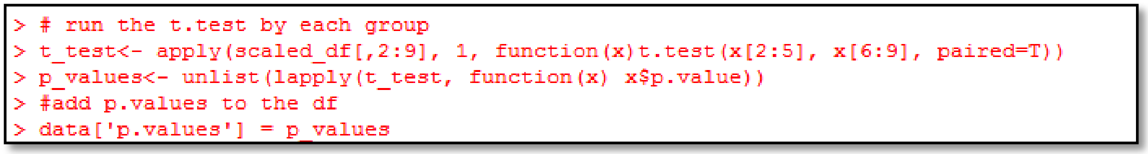
```
> high_variation[1:5]
[1] 16420  1908  3788  3790  4015
> data[16420,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
16420 ML46651a  125638  105808   65907   93351 16236 10449 22838 58247  62309.25 40721.49   498474
> data[1908,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
1908 ML01482a   32503   90804   83222  111860 15018 11845 36717 22066  50504.38 36290.03   404035
> data[3788,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
3788 ML034334a   23288   76895   65076   94170  4216  6801 14845 10235  36940.75 33597.34   295526
> data[3790,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
3790 ML034336a   25116   74297   59568   84219  5130  6048 14005  9833     34777 30561.92   278216
> data[4015,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
4015 ML03658a    5950   55688   25370   76789  2879  1677 19022  4732  24013.38 26122.15   192107
```



```
> low_variation[1:5]
[1]  5255 13588  2691  5909  6388
> newdata[5255,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
5942 ML061522a       1       1       1       1     1     1     0     0      0.75 1.625684        6
> newdata[13588,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
15462 ML348711a       0       0       1       1     1     1     1     1      0.75 1.625684        6
> newdata[2691,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
3017 ML025911a       0       1       2       1     1     1     0     0      0.75 1.687151        6
> newdata[5909,]
         Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
6679 ML07086a       0       1       2       0     1     1     1     0      0.75 1.687151        6
> newdata[6388,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   row sd row sums
7223 ML076020a       1       0       1       1     0     2     1     0      0.75 1.687151        6
```

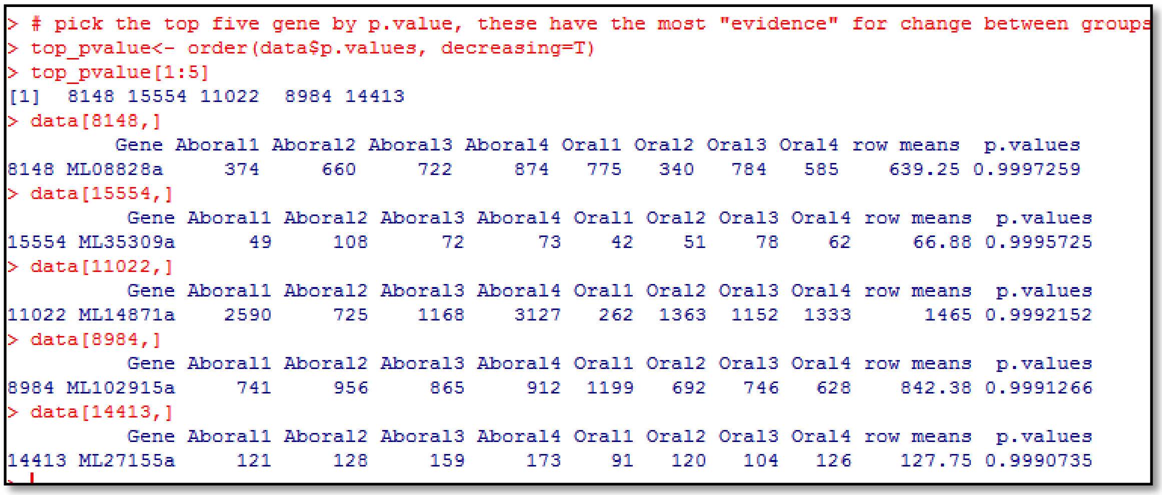
8. Using the labels of columns provided, find the top variable genes between The two groups using a t-test, list the 5 most up regulated and 5 most down regulated genes. What happens if you rank by p-value of the t-test ? would you exclude some of the high p-value genes for having low expression ?

First run the t-test to get the p-values to find the highest p-values.



```
> # run the t.test by each group
> t_test<- apply(scaled_df[,2:9], 1, function(x)t.test(x[2:5], x[6:9], paired=T))
> p_values<- unlist(lapply(t_test, function(x) x$p.value))
> #add p.values to the df
> data['p.values'] = p_values
```

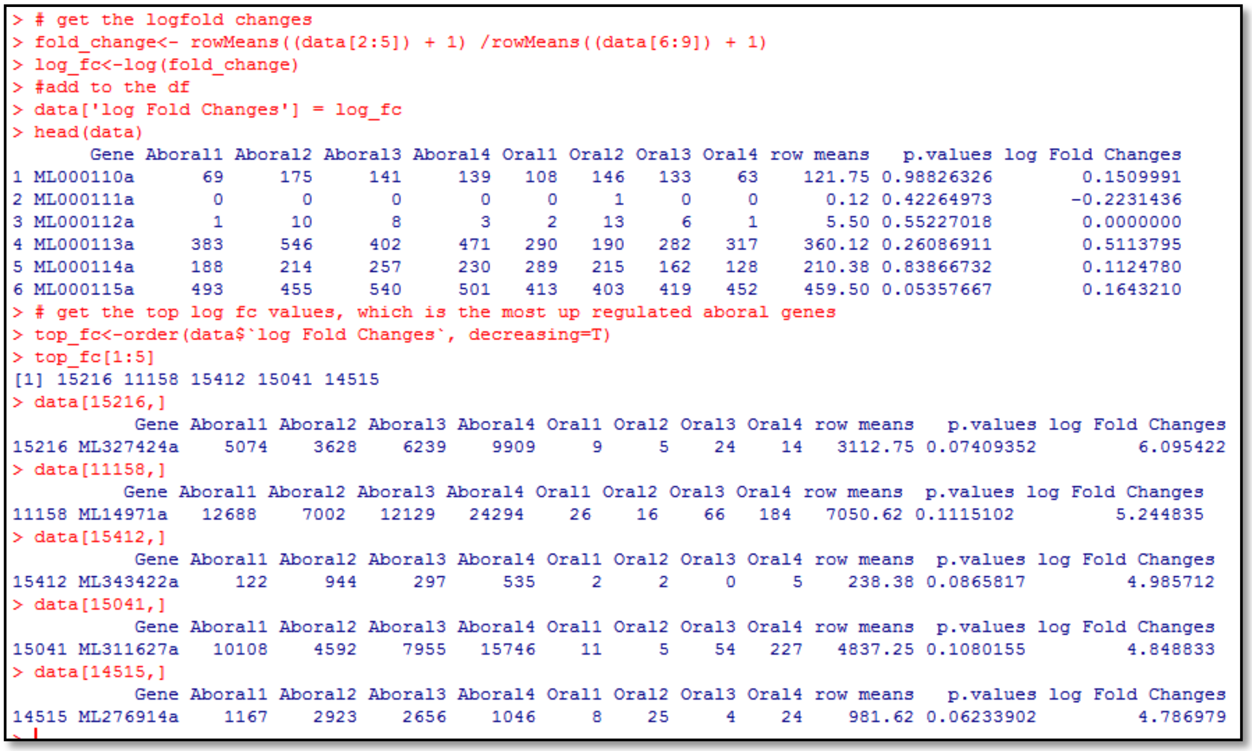
Top 5 most evidence for change between the groups : *ML08828a, ML35309a, ML14871a, ML102915a, ML27155a*



```
> # pick the top five gene by p.value, these have the most "evidence" for change between groups
> top_pvalue<- order(data$p.values, decreasing=T)
> top_pvalue[1:5]
[1]  8148 15554 11022  8984 14413
> data[8148,]
         Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values
8148 ML08828a     374     660     722     874   775   340   784   585    639.25 0.9997259
> data[15554,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values
15554 ML35309a      49     108      72      73    42    51    78    62     66.88 0.9995725
> data[11022,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values
11022 ML14871a    2590     725    1168    3127   262  1363  1152  1333      1465 0.9992152
> data[8984,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values
8984 ML102915a     741     956     865     912  1199   692   746   628    842.38 0.9991266
> data[14413,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values
14413 ML27155a     121     128     159     173    91   120   104   126    127.75 0.9990735
>
```

Top 5 most up regulated genes –

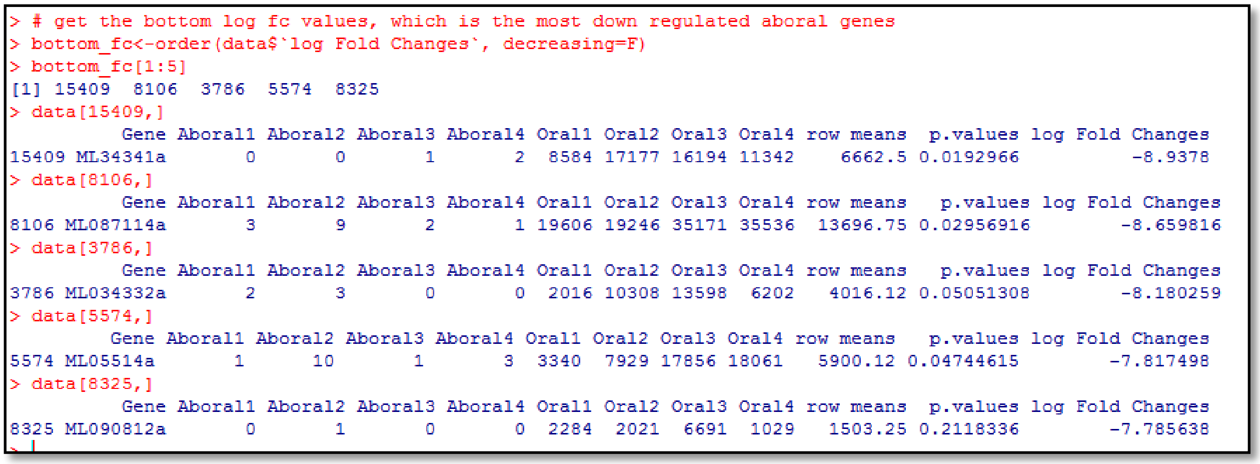
- **ML327424a, ML14971a, ML343422a, ML311627a, ML276914a**



```
> # get the logfold changes
> fold_change<- rowMeans((data[2:5]) + 1) /rowMeans((data[6:9]) + 1)
> log_fc<-log(fold_change)
> #add to the df
> data['log Fold Changes'] = log_fc
> head(data)
       Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
1 ML000110a      69     175     141     139   108   146   133    63    121.75 0.98826326        0.1509991
2 ML000111a       0       0       0       0     0     1     0     0      0.12 0.42264973       -0.2231436
3 ML000112a       1      10       8       3     2    13     6     1      5.50 0.55227018        0.0000000
4 ML000113a     383     546     402     471   290   190   282   317    360.12 0.26086911        0.5113795
5 ML000114a     188     214     257     230   289   215   162   128    210.38 0.83866732        0.1124780
6 ML000115a     493     455     540     501   413   403   419   452    459.50 0.05357667        0.1643210
> # get the top log fc values, which is the most up regulated aboral genes
> top_fc<-order(data$`log Fold Changes`, decreasing=T)
> top_fc[1:5]
[1] 15216 11158 15412 15041 14515
> data[15216,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
15216 ML327424a    5074    3628    6239    9909     9     5    24    14   3112.75 0.07409352         6.095422
> data[11158,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values log Fold Changes
11158 ML14971a   12688    7002   12129   24294    26    16    66   184   7050.62 0.1115102         5.244835
> data[15412,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values log Fold Changes
15412 ML343422a     122     944     297     535     2     2     0     5    238.38 0.0865817         4.985712
> data[15041,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values log Fold Changes
15041 ML311627a   10108    4592    7955   15746    11     5    54   227   4837.25 0.1080155         4.848833
> data[14515,]
           Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
14515 ML276914a    1167    2923    2656    1046     8    25     4    24    981.62 0.06233902         4.786979
>
```

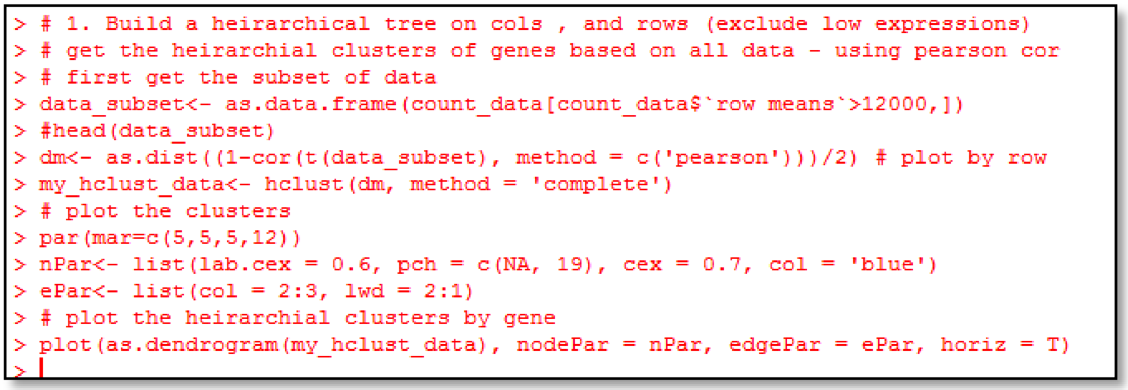
Top 5 most down regulated genes –

- **ML34341a, ML087114a, ML34332a, ML05514a, ML090812a**

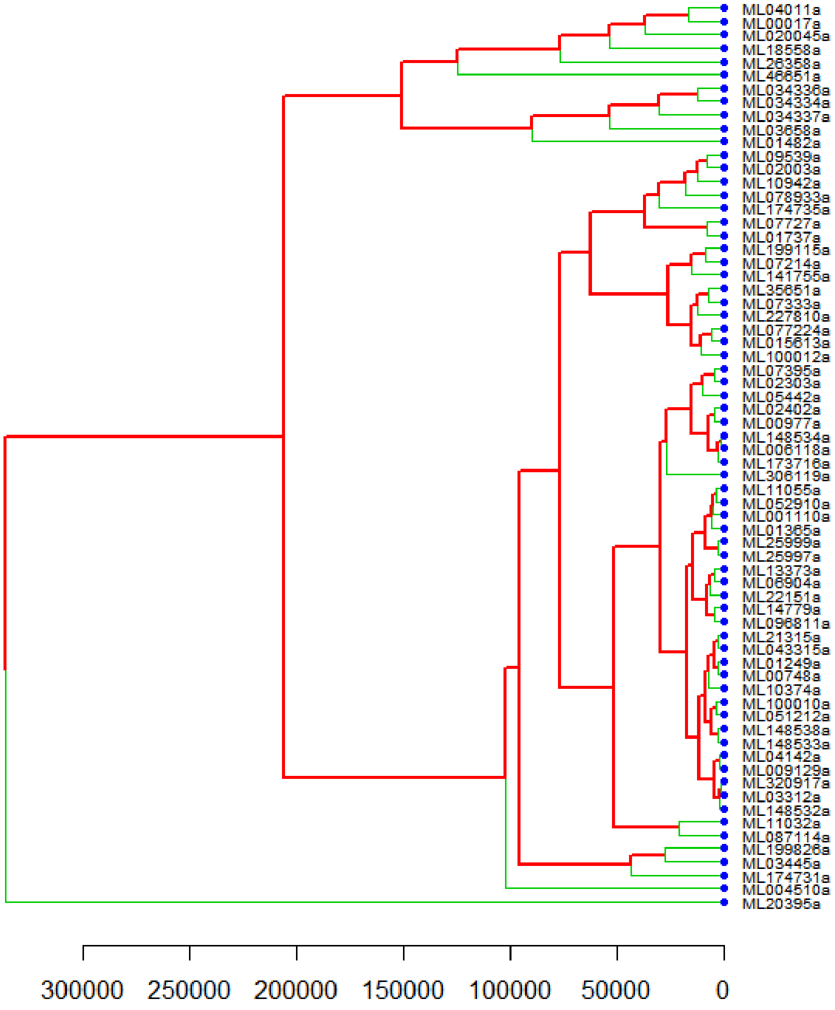


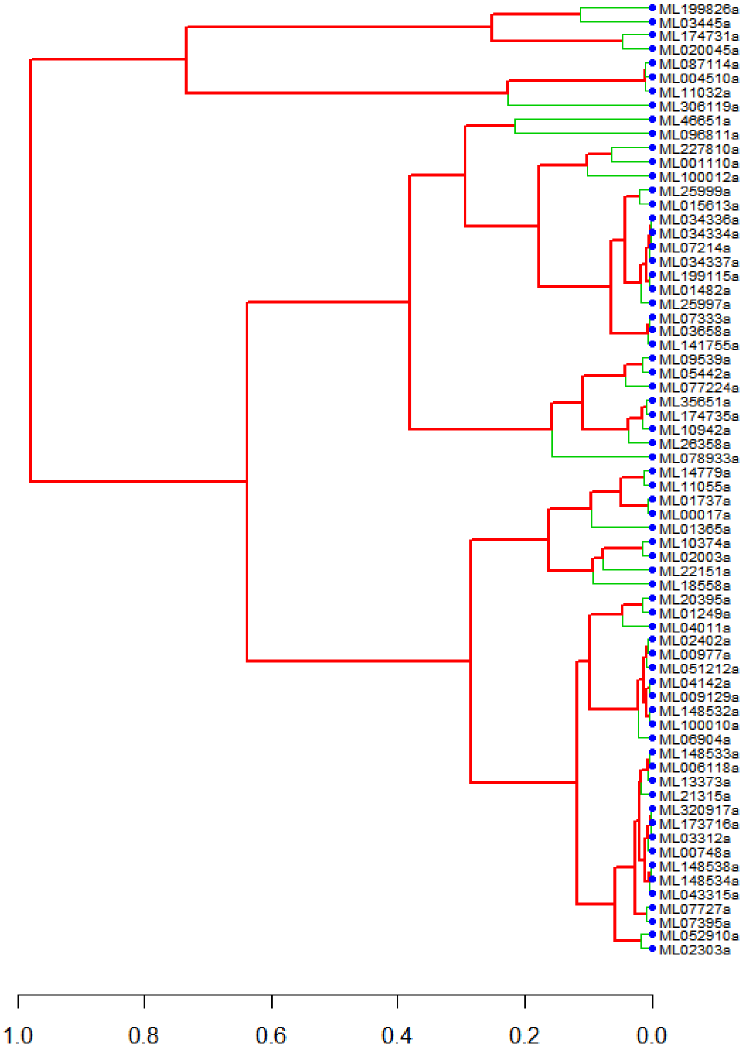
```
> # get the bottom log fc values, which is the most down regulated aboral genes
> bottom_fc<-order(data$`log Fold Changes`, decreasing=F)
> bottom_fc[1:5]
[1] 15409  8106  3786  5574  8325
> data[15409,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values log Fold Changes
15409 ML34341a       0       0       1       2  8584 17177 16194 11342    6662.5 0.0192966          -8.9378
> data[8106,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
8106 ML087114a       3       9       2       1 19606 19246 35171 35536  13696.75 0.02956916        -8.659816
> data[3786,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
3786 ML034332a       2       3       0       0  2016 10308 13598  6202   4016.12 0.05051308        -8.180259
> data[5574,]
         Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means   p.values log Fold Changes
5574 ML05514a       1      10       1       3  3340  7929 17856 18061   5900.12 0.04744615        -7.817498
> data[8325,]
          Gene Aboral1 Aboral2 Aboral3 Aboral4 Oral1 Oral2 Oral3 Oral4 row means  p.values log Fold Changes
8325 ML090812a       0       1       0       0  2284  2021  6691  1029   1503.25 0.2118336        -7.785638
>
```

1. Build hierarchical trees based on the columns and for the rows (exclude rows that are "low" expression)
   See hierarchical tree based on rows (Euclidean distance and Pearson correlation distance)

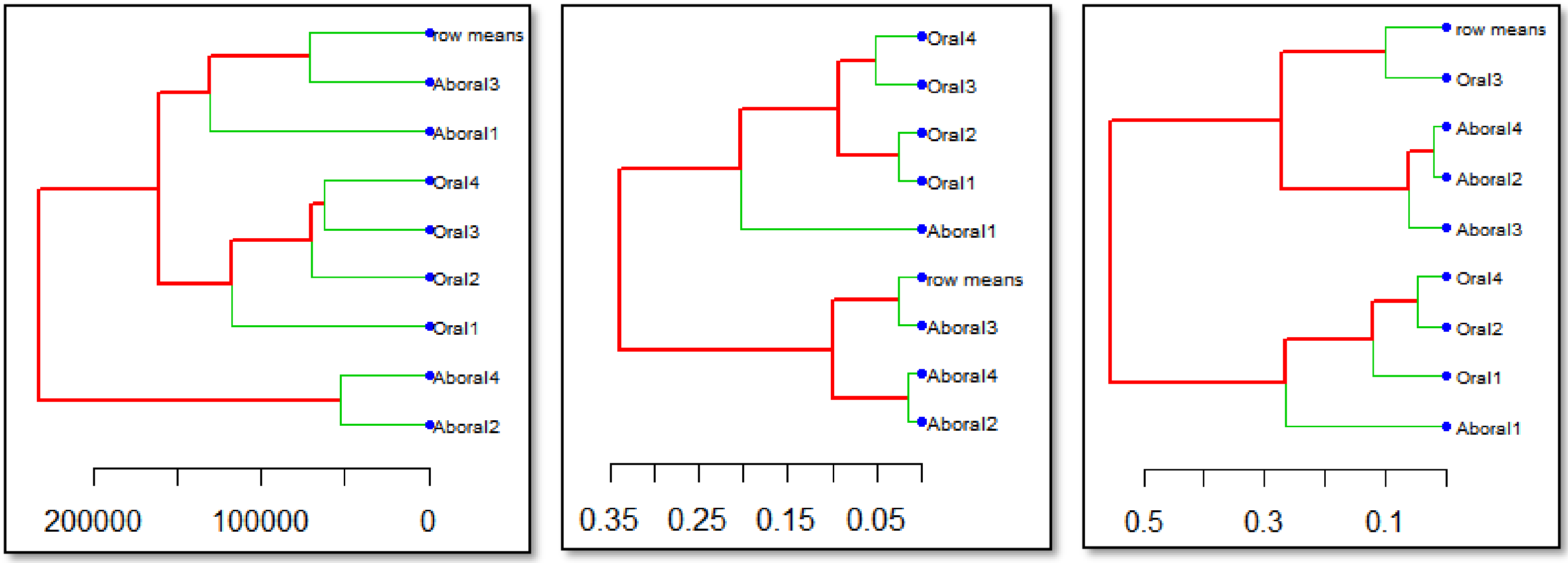


```
> # 1. Build a heirarchical tree on cols , and rows (exclude low expressions)
> # get the heirarchial clusters of genes based on all data - using pearson cor
> # first get the subset of data
> data_subset<- as.data.frame(count_data[count_data$`row means`>12000,])
> #head(data_subset)
> dm<- as.dist((1-cor(t(data_subset), method = c('pearson')))/2) # plot by row
> my_hclust_data<- hclust(dm, method = 'complete')
> # plot the clusters
> par(mar=c(5,5,5,12))
> nPar<- list(lab.cex = 0.6, pch = c(NA, 19), cex = 0.7, col = 'blue')
> ePar<- list(col = 2:3, lwd = 2:1)
> # plot the heirarchial clusters by gene
> plot(as.dendrogram(my_hclust_data), nodePar = nPar, edgePar = ePar, horiz = T)
> 
```

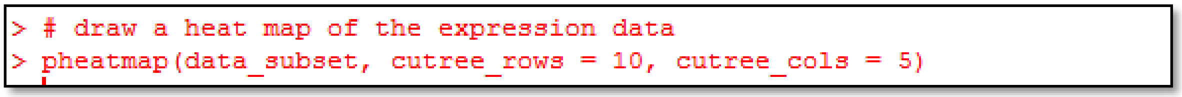




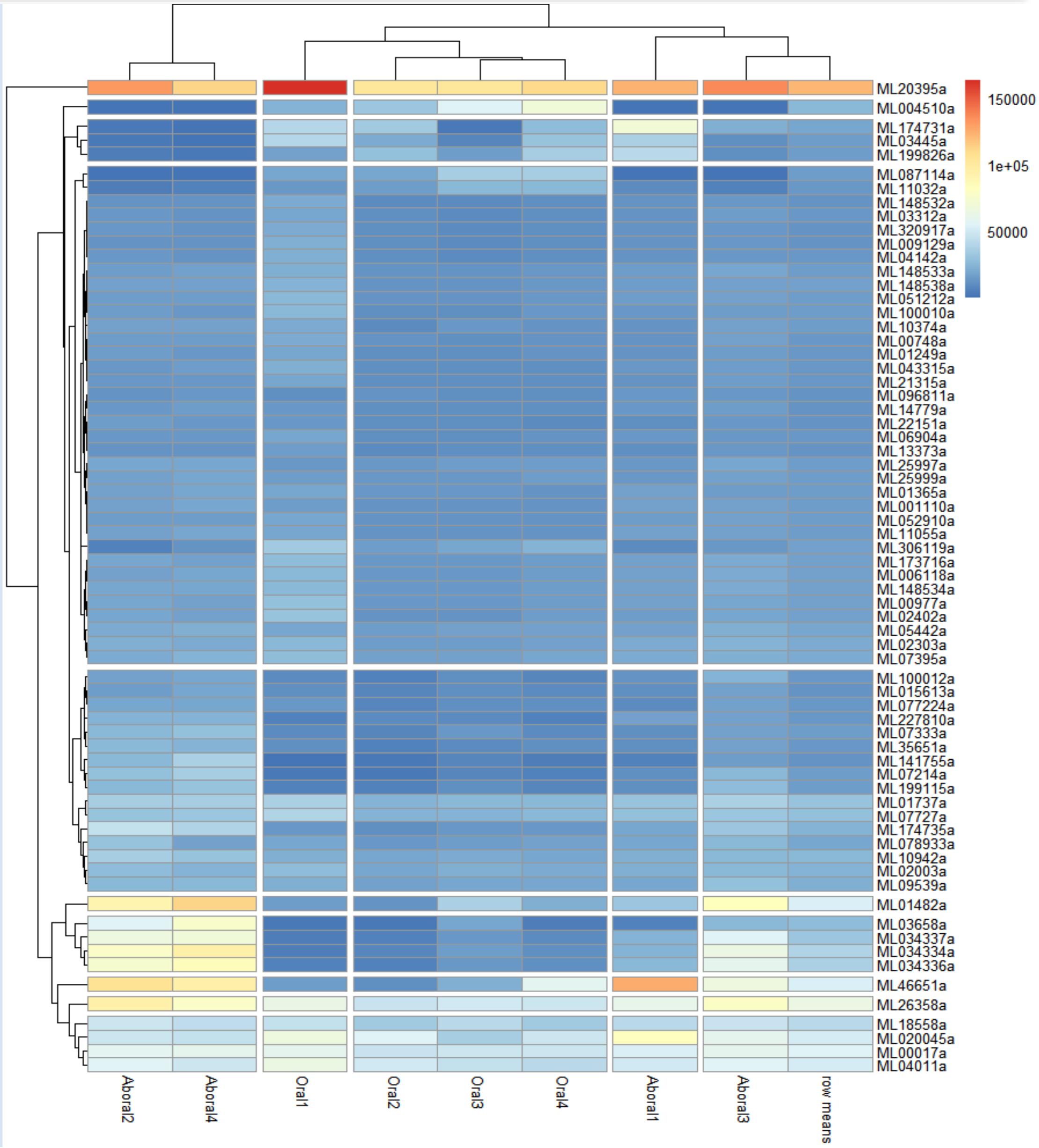
Hierarchical tree based on columns (Euclidean, Pearson, Spearman):



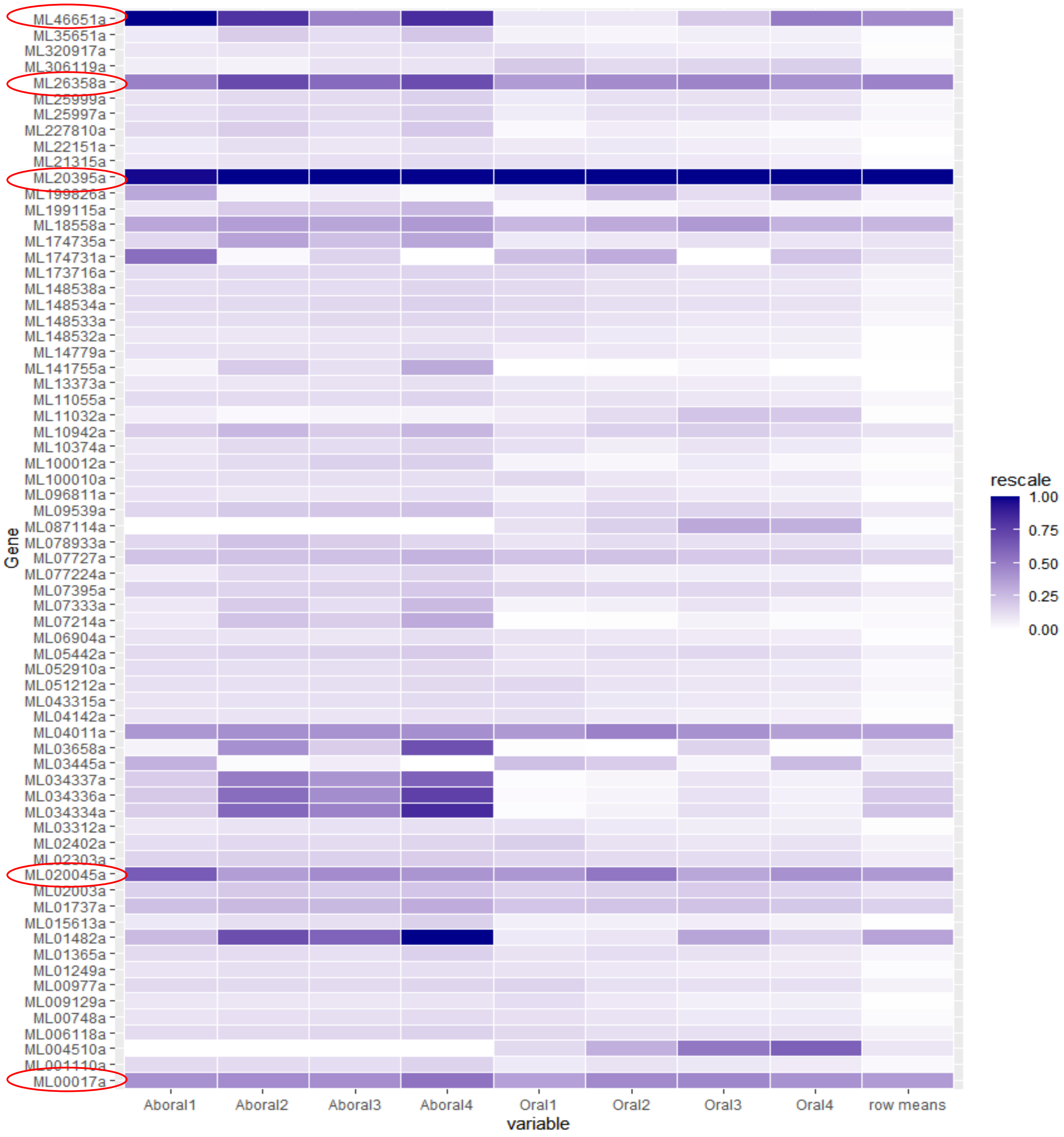
2. Draw a heat map of the expression data.



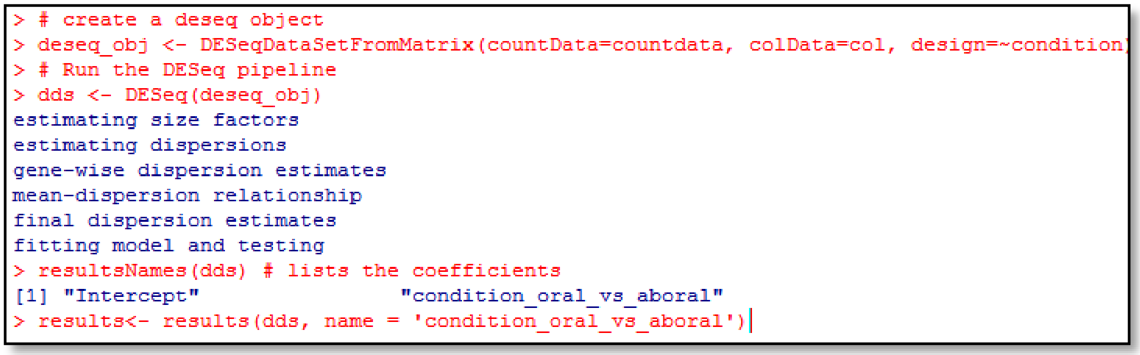
```
> # draw a heat map of the expression data
> pheatmap(data_subset, cutree_rows = 10, cutree_cols = 5)
```



Another example of a heatmap with one color and rescaled values : Notice the top 5 genes based on expression from the midterm (circled in red) match up visually here as being highly expressed

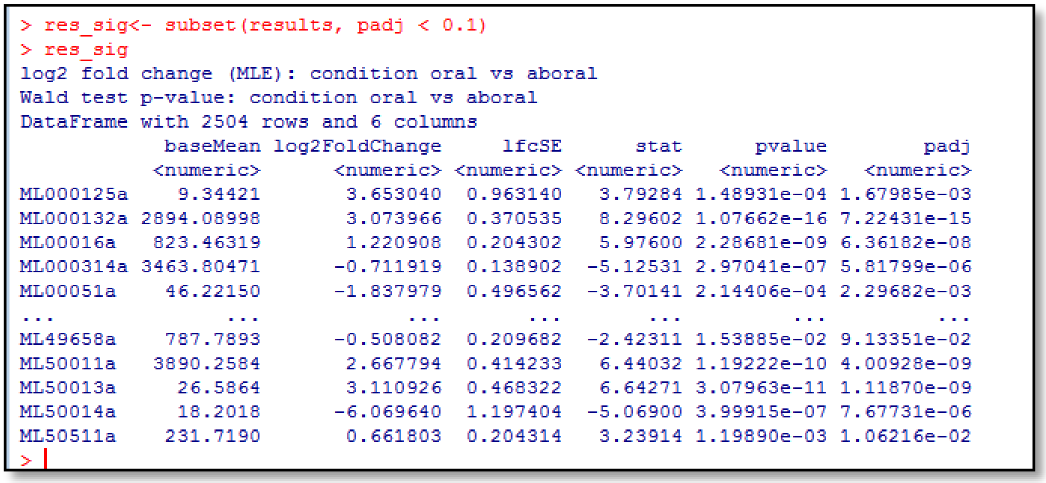


3. Use DESeq2 to analyze this data, which are the most significantly changing genes in this dataset?
A positive log2 fold change for a comparison of A vs B means that gene expression in A is larger in comparison to B.



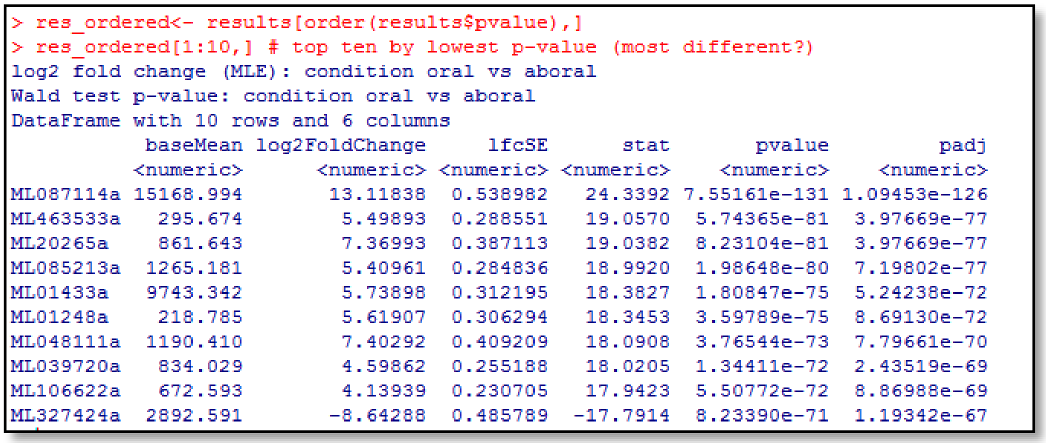
```
> # create a deseq object
> deseq_obj <- DESeqDataSetFromMatrix(countData=countdata, colData=col, design=~condition)
> # Run the DESeq pipeline
> dds <- DESeq(deseq_obj)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
> resultsNames(dds) # lists the coefficients
[1] "Intercept"                "condition_oral_vs_aboral"
> results<- results(dds, name = 'condition_oral_vs_aboral')
```

Extract the genes of significance -



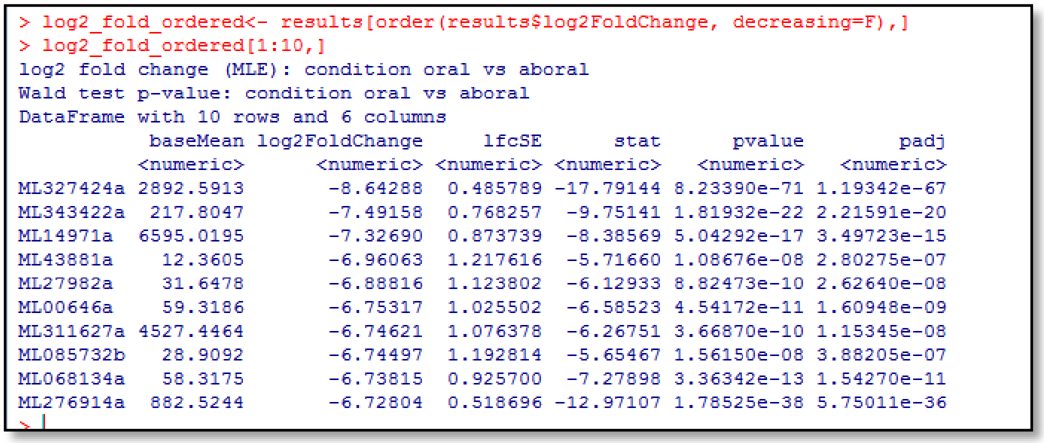
```
> res_sig<- subset(results, padj < 0.1)
> res_sig
log2 fold change (MLE): condition oral vs aboral
Wald test p-value: condition oral vs aboral
DataFrame with 2504 rows and 6 columns
            baseMean log2FoldChange     lfcSE      stat      pvalue        padj
           <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
ML000125a    9.34421       3.653040  0.963140   3.79284 1.48931e-04 1.67985e-03
ML000132a 2894.08998       3.073966  0.370535   8.29602 1.07662e-16 7.22431e-15
ML00016a   823.46319       1.220908  0.204302   5.97600 2.28681e-09 6.36182e-08
ML000314a 3463.80471      -0.711919  0.138902  -5.12531 2.97041e-07 5.81799e-06
ML00051a    46.22150      -1.837979  0.496562  -3.70141 2.14406e-04 2.29682e-03
...              ...            ...       ...       ...         ...         ...
ML49658a    787.7893      -0.508082  0.209682  -2.42311 1.53885e-02 9.13351e-02
ML50011a   3890.2584       2.667794  0.414233   6.44032 1.19222e-10 4.00928e-09
ML50013a     26.5864       3.110926  0.468322   6.64271 3.07963e-11 1.11870e-09
ML50014a     18.2018      -6.069640  1.197404  -5.06900 3.99915e-07 7.67731e-06
ML50511a    231.7190       0.661803  0.204314   3.23914 1.19890e-03 1.06216e-02
> 
```

Highest p-value between groups (largest difference)



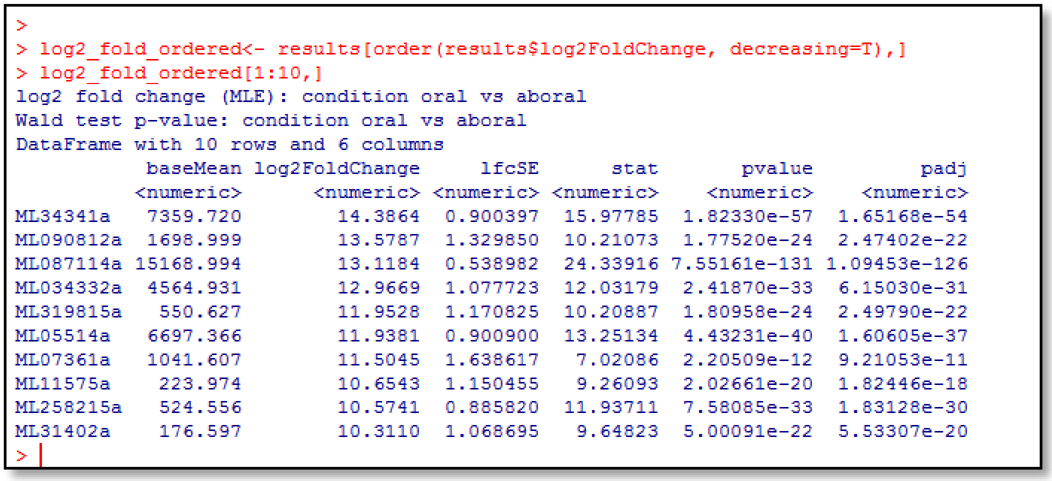
```
> res_ordered<- results[order(results$pvalue),]
> res_ordered[1:10,] # top ten by lowest p-value (most different?)
log2 fold change (MLE): condition oral vs aboral
Wald test p-value: condition oral vs aboral
DataFrame with 10 rows and 6 columns
           baseMean log2FoldChange     lfcSE      stat       pvalue         padj
          <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric>
ML087114a 15168.994       13.11838  0.538982   24.3392 7.55161e-131 1.09453e-126
ML463533a   295.674        5.49893  0.288551   19.0570  5.74365e-81  3.97669e-77
ML20265a    861.643        7.36993  0.387113   19.0382  8.23104e-81  3.97669e-77
ML085213a  1265.181        5.40961  0.284836   18.9920  1.98648e-80  7.19802e-77
ML01433a   9743.342        5.73898  0.312195   18.3827  1.80847e-75  5.24238e-72
ML01248a    218.785        5.61907  0.306294   18.3453  3.59789e-75  8.69130e-72
ML048111a  1190.410        7.40292  0.409209   18.0908  3.76544e-73  7.79661e-70
ML039720a   834.029        4.59862  0.255188   18.0205  1.34411e-72  2.43519e-69
ML106622a   672.593        4.13939  0.230705   17.9423  5.50772e-72  8.86988e-69
ML327424a  2892.591       -8.64288  0.485789  -17.7914  8.23390e-71  1.19342e-67
```

Most up-regulated genes

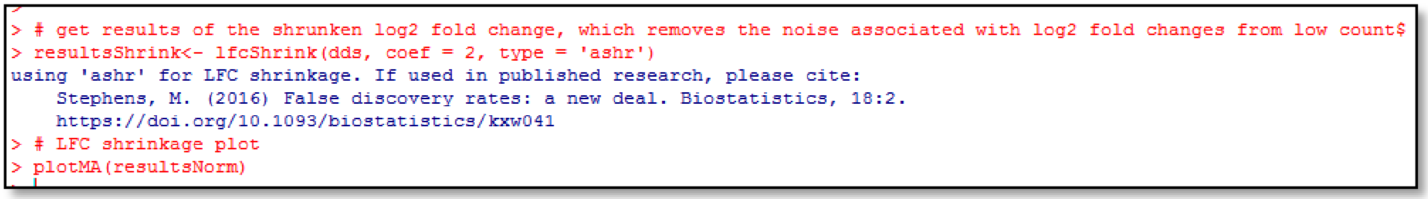


```
> log2_fold_ordered<- results[order(results$log2FoldChange, decreasing=F),]
> log2_fold_ordered[1:10,]
log2 fold change (MLE): condition oral vs aboral
Wald test p-value: condition oral vs aboral
DataFrame with 10 rows and 6 columns
           baseMean log2FoldChange     lfcSE      stat      pvalue        padj
          <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
ML327424a 2892.5913       -8.64288  0.485789 -17.79144 8.23390e-71 1.19342e-67
ML343422a  217.8047       -7.49158  0.768257  -9.75141 1.81932e-22 2.21591e-20
ML14971a  6595.0195       -7.32690  0.873739  -8.38569 5.04292e-17 3.49723e-15
ML43881a    12.3605       -6.96063  1.217616  -5.71660 1.08676e-08 2.80275e-07
ML27982a    31.6478       -6.88816  1.123802  -6.12933 8.82473e-10 2.62640e-08
ML00646a    59.3186       -6.75317  1.025502  -6.58523 4.54172e-11 1.60948e-09
ML311627a 4527.4464       -6.74621  1.076378  -6.26751 3.66870e-10 1.15345e-08
ML085732b   28.9092       -6.74497  1.192814  -5.65467 1.56150e-08 3.88205e-07
ML068134a   58.3175       -6.73815  0.925700  -7.27898 3.36342e-13 1.54270e-11
ML276914a  882.5244       -6.72804  0.518696 -12.97107 1.78525e-38 5.75011e-36
> 
```

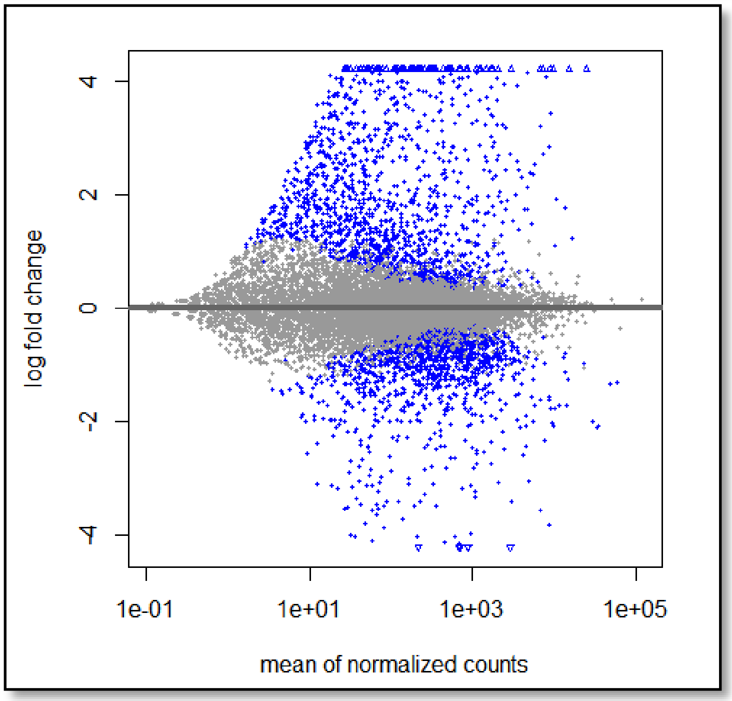
Genes with the highest log fold2 changes – biggest changes between the groups

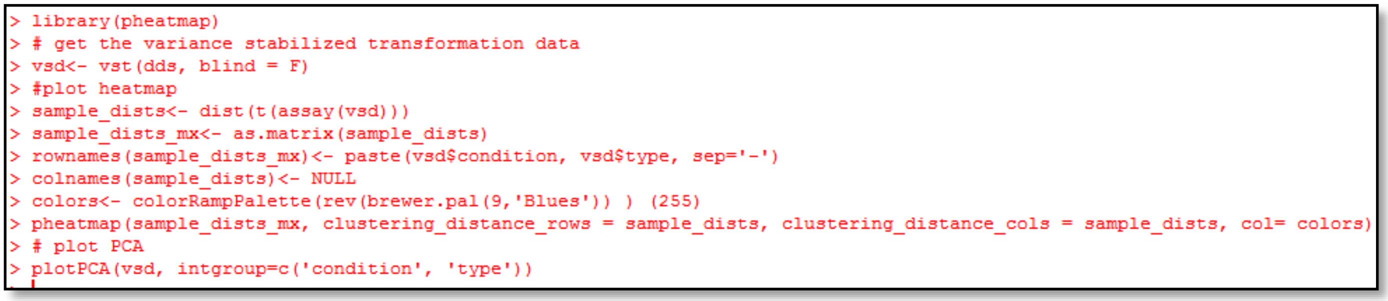


```
>
> log2_fold_ordered<- results[order(results$log2FoldChange, decreasing=T),]
> log2_fold_ordered[1:10,]
log2 fold change (MLE): condition oral vs aboral
Wald test p-value: condition oral vs aboral
DataFrame with 10 rows and 6 columns
           baseMean log2FoldChange     lfcSE      stat       pvalue         padj
          <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric>
ML34341a   7359.720        14.3864  0.900397  15.97785  1.82330e-57  1.65168e-54
ML090812a  1698.999        13.5787  1.329850  10.21073  1.77520e-24  2.47402e-22
ML087114a 15168.994        13.1184  0.538982  24.33916 7.55161e-131 1.09453e-126
ML034332a  4564.931        12.9669  1.077723  12.03179  2.41870e-33  6.15030e-31
ML319815a   550.627        11.9528  1.170825  10.20887  1.80958e-24  2.49790e-22
ML05514a   6697.366        11.9381  0.900900  13.25134  4.43231e-40  1.60605e-37
ML07361a   1041.607        11.5045  1.638617   7.02086  2.20509e-12  9.21053e-11
ML11575a    223.974        10.6543  1.150455   9.26093  2.02661e-20  1.82446e-18
ML258215a   524.556        10.5741  0.885820  11.93711  7.58085e-33  1.83128e-30
ML31402a    176.597        10.3110  1.068695   9.64823  5.00091e-22  5.53307e-20
> 
```

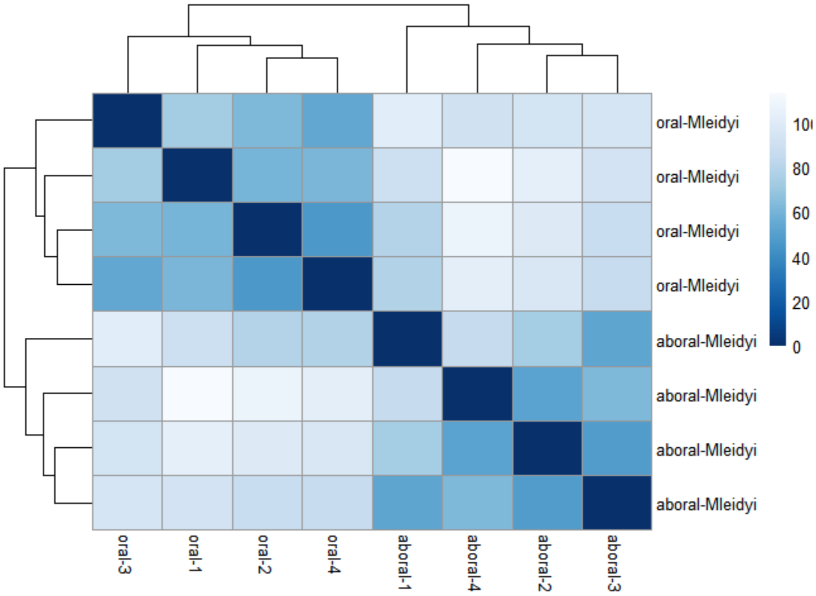


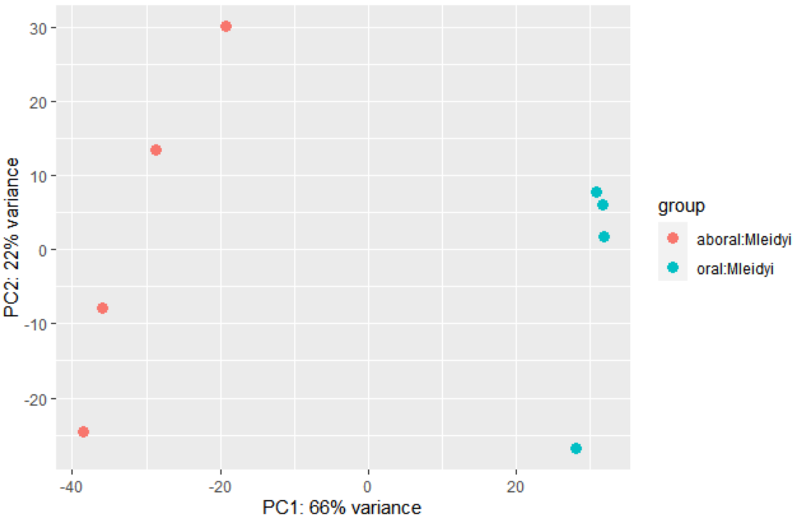
```
> # get results of the shrunken log2 fold change, which removes the noise associated with log2 fold changes from low count$
> resultsShrink<- lfcShrink(dds, coef = 2, type = 'ashr')
using 'ashr' for LFC shrinkage. If used in published research, please cite:
    Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
    https://doi.org/10.1093/biostatistics/kxw041
> # LFC shrinkage plot
> plotMA(resultsNorm)
```





```
> library(pheatmap)
> # get the variance stabilized transformation data
> vsd<- vst(dds, blind = F)
> #plot heatmap
> sample_dists<- dist(t(assay(vsd)))
> sample_dists_mx<- as.matrix(sample_dists)
> rownames(sample_dists_mx)<- paste(vsd$condition, vsd$type, sep='-')
> colnames(sample_dists)<- NULL
> colors<- colorRampPalette(rev(brewer.pal(9,'Blues')) ) (255)
> pheatmap(sample_dists_mx, clustering_distance_rows = sample_dists, clustering_distance_cols = sample_dists, col= colors)
> # plot PCA
> plotPCA(vsd, intgroup=c('condition', 'type'))
```


oral-Mleidyi
oral-Mleidyi
oral-Mleidyi
oral-Mleidyi
aboral-Mleidyi
aboral-Mleidyi
aboral-Mleidyi
aboral-Mleidyi
10(
oral-3
oral-1
oral-2
oral-4
aboral-1
aboral-4
aboral-2
aboral-3


PC2: 22% variance
PC1: 66% variance
group
aboral:Mleidyi
oral:Mleidyi