RNASeq_Meth_Compare

HM Putnam

3/8/2020

```r
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: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 object is masked from 'package:base':
##
##     expand.grid
```

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

```
## Loading required package: GenomicRanges
```

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

```
## Loading required package: SummarizedExperiment
```

```
## Warning: package 'SummarizedExperiment' was built under R version 3.6.1
```

```
## 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

## 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
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::collapse()   masks IRanges::collapse()
## x dplyr::combine()    masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::count()      masks matrixStats::count()
## x dplyr::desc()       masks IRanges::desc()
## x tidyr::expand()     masks S4Vectors::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks S4Vectors::first()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()     masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()     masks S4Vectors::rename()
## x purrr::simplify()   masks DelayedArray::simplify()
## x dplyr::slice()      masks IRanges::slice()
```

```
library("pheatmap")
library("genefilter")
```

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

## The following object is masked from 'package:readr':
##
##     spec
## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

```
library("ggplot2")
library("RColorBrewer")
```

Load Data

```
#treatmentinfo
treatmentinfo <- read.csv("metadata/Ribodepleted_RNASeq_metadata.csv", header = TRUE, sep = ",")
print(treatmentinfo)
```

```
##   Sample.Number.Tube.Label CD.Index..            Species Plug.ID
## 1                        1          7  Pocillopora acuta PA_1041
## 2                        2          8  Pocillopora acuta PA_1471
## 3                        3         15  Pocillopora acuta PA_1637
## 4                        4         16 Montipora capitata MC_1101
## 5                        5         23 Montipora capitata MC_1548
## 6                        6         24 Montipora capitata MC_1628
##                  GTF
## 1 Sample1_merged2.gtf
## 2 Sample2_merged2.gtf
## 3 Sample3_merged2.gtf
## 4 Sample4_merged2.gtf
## 5 Sample5_merged2.gtf
## 6 Sample6_merged2.gtf
##                                              Genome.Assembly
## 1 http://ihpe.univ-perp.fr/telechargement/Data_to_download.rar
## 2 http://ihpe.univ-perp.fr/telechargement/Data_to_download.rar
## 3 http://ihpe.univ-perp.fr/telechargement/Data_to_download.rar
## 4                      http://cyanophora.rutgers.edu/montipora/
## 5                      http://cyanophora.rutgers.edu/montipora/
## 6                      http://cyanophora.rutgers.edu/montipora/
```

```
MC.treatmentinfo <- subset(treatmentinfo, Species =="Montipora capitata")
MC.gcount <- as.data.frame(read.csv("RNASeq/cleaned_reads/Mcap_gene_count_matrix.csv", row.names="gene_
head(MC.gcount)
```

```
##          MC_1101 MC_1548 MC_1628
## MSTRG.1        1       6      15
## g21533        16      39      40
## g21534        14      24      38
## g21535        22      34      42
## MSTRG.7        2       4       5
## MSTRG.8        7      28       6
```

```
#Ensure all sample IDs in colData are also in CountData and match their orders
rownames(MC.treatmentinfo) <- MC.treatmentinfo$Plug.ID
colnames(MC.gcount) <- MC.treatmentinfo$Plug.ID

MCAP.Data <- DESeqDataSetFromMatrix(countData = MC.gcount,
                              colData = MC.treatmentinfo,
                              design = ~1)
```

```
PA.treatmentinfo <- subset(treatmentinfo, Species =="Pocillopora acuta")
PA.gcount <- as.data.frame(read.csv("RNASeq/cleaned_reads/Pact_gene_count_matrix.csv", row.names="gene_
head(PA.gcount)
```

```
##              X1041 X1471 X1637
## g1               0    20     0
## g2               0     0     0
## g3               6     7     0
## MSTRG.42442     59    44    28
## MSTRG.42443    410   174   243
## MSTRG.42445   1127  1109  1065
```

```
#Ensure all sample IDs in colData are also in CountData and match their orders
rownames(PA.treatmentinfo) <- PA.treatmentinfo$Plug.ID
colnames(PA.gcount) <- PA.treatmentinfo$Plug.ID

PACT.Data <- DESeqDataSetFromMatrix(countData = PA.gcount,
                              colData = PA.treatmentinfo,
                              design = ~1)
```

Visualize gene count data

Log-transform the count data

First we are going to log-transform the data using a variance stabilizing transforamtion (vst). This is only for visualization purposes. Essentially, this is roughly similar to putting the data on the log2 scale. It will deal with the sampling variability of low counts by calculating within-group variability (if blind=FALSE). Importantly, it does not use the design to remove variation in the data, and so can be used to examine if there may be any variability do to technical factors such as extraction batch effects.

To do this we first need to calculate the size factors of our samples. This is a rough estimate of how many reads each sample contains compared to the others. In order to use VST (the faster log2 transforming process) to log-transform our data, the size factors need to be less than 4. Otherwise, there could be artifacts in our results.

```
MC.SF <- estimateSizeFactors(MCAP.Data) #estimate size factors to determine if we can use vst  to trans
print(sizeFactors(MC.SF)) #View size factors
```

```
##   MC_1101   MC_1548   MC_1628
## 0.7682701 1.2499099 1.0653326
```

```
PA.SF <- estimateSizeFactors(PACT.Data) #estimate size factors to determine if we can use vst  to trans
print(sizeFactors(PA.SF)) #View size factors
```

```
##   PA_1041   PA_1471   PA_1637
## 1.2447398 0.8206077 1.0000000
```

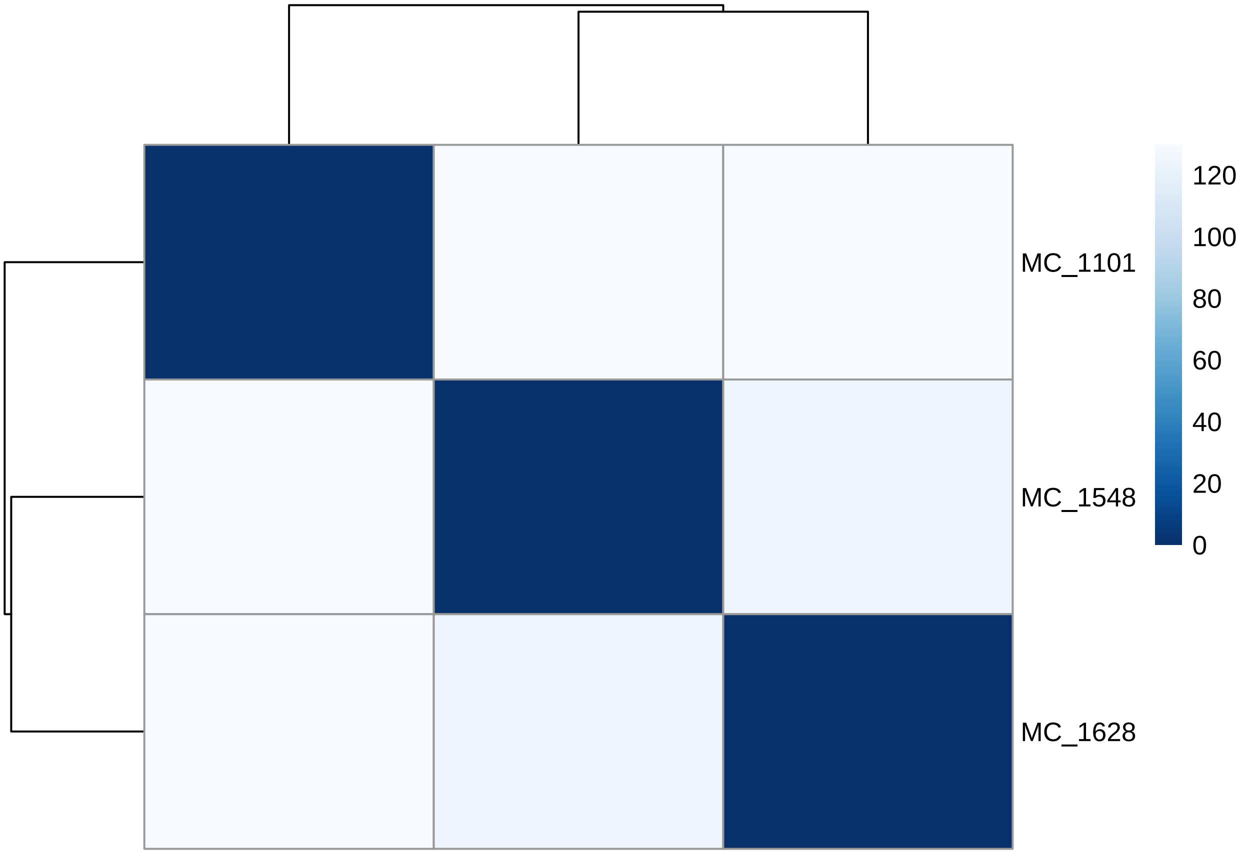
```
MC.trans.Data <- vst(MCAP.Data, blind=FALSE) #apply a variance stabilizing transforamtion to minimize e
```

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.
```

```
head(assay(MC.trans.Data), 3) #view transformed gene count data
```

```
##          MC_1101  MC_1548  MC_1628
## MSTRG.1 4.658333 4.783567 5.117015
## g21533  5.329413 5.604211 5.746570
## g21534  5.250787 5.280831 5.706126
```

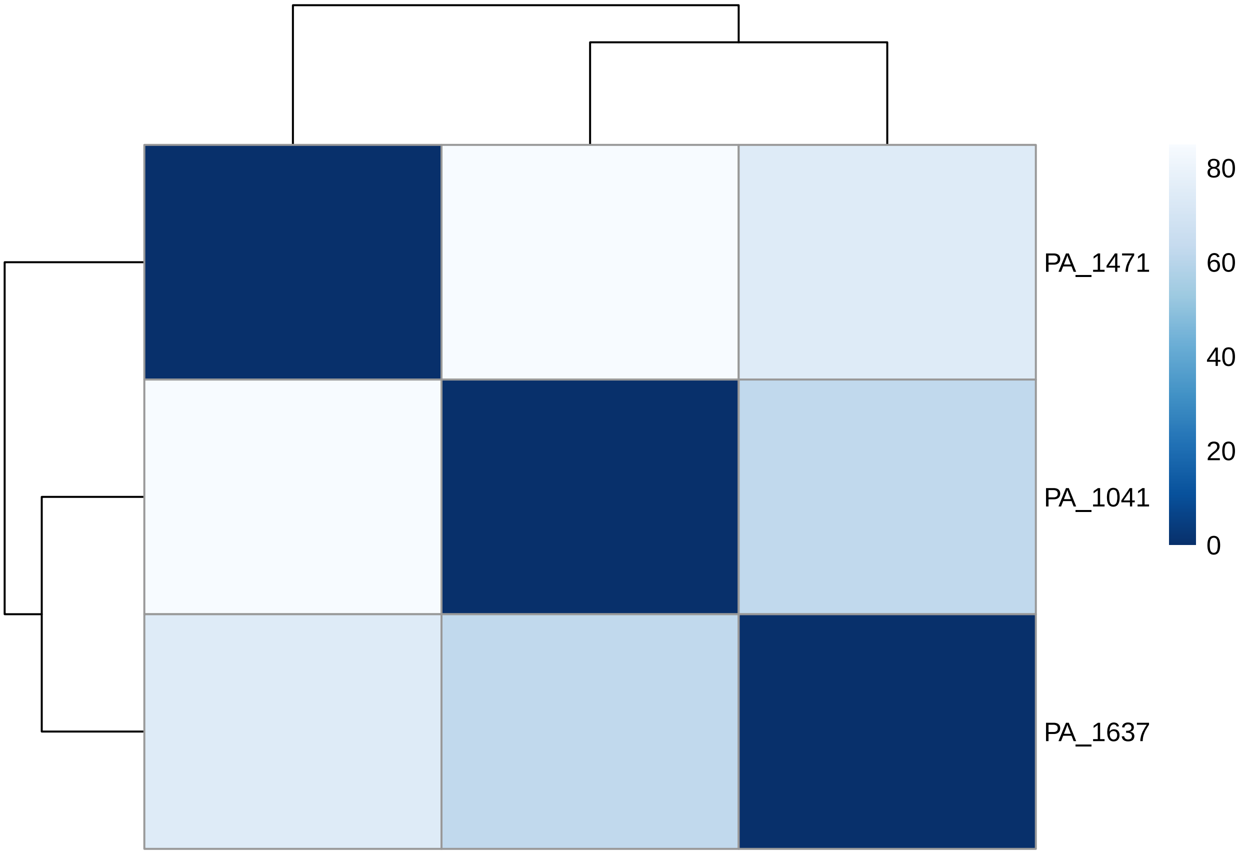
```
MC.gsampleDists<- dist(t(assay(MC.trans.Data))) #calculate distance matix
MC.gsampleDistMatrix <- as.matrix(MC.gsampleDists) #distance matrix
rownames(MC.gsampleDistMatrix) <- colnames(MC.trans.Data) #assign row names
colnames(MC.gsampleDistMatrix) <- NULL #assign col names
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")) )(255) #assign colors
pheatmap(MC.gsampleDistMatrix, #plot matrix
         clustering_distance_rows=MC.gsampleDists, #cluster rows
         clustering_distance_cols=MC.gsampleDists, #cluster columns
         col=colors) #set colors
```



```
PA.trans.Data <- vst(PACT.Data, blind=FALSE) #apply a variance stabilizing transforamtion to minimize e
head(assay(PA.trans.Data), 3) #view transformed gene count data
```

```
##     PA_1041  PA_1471  PA_1637
## g1 6.805986 7.473336 6.805986
## g2 6.805986 6.805986 6.805986
## g3 7.104890 7.203071 6.805986
```

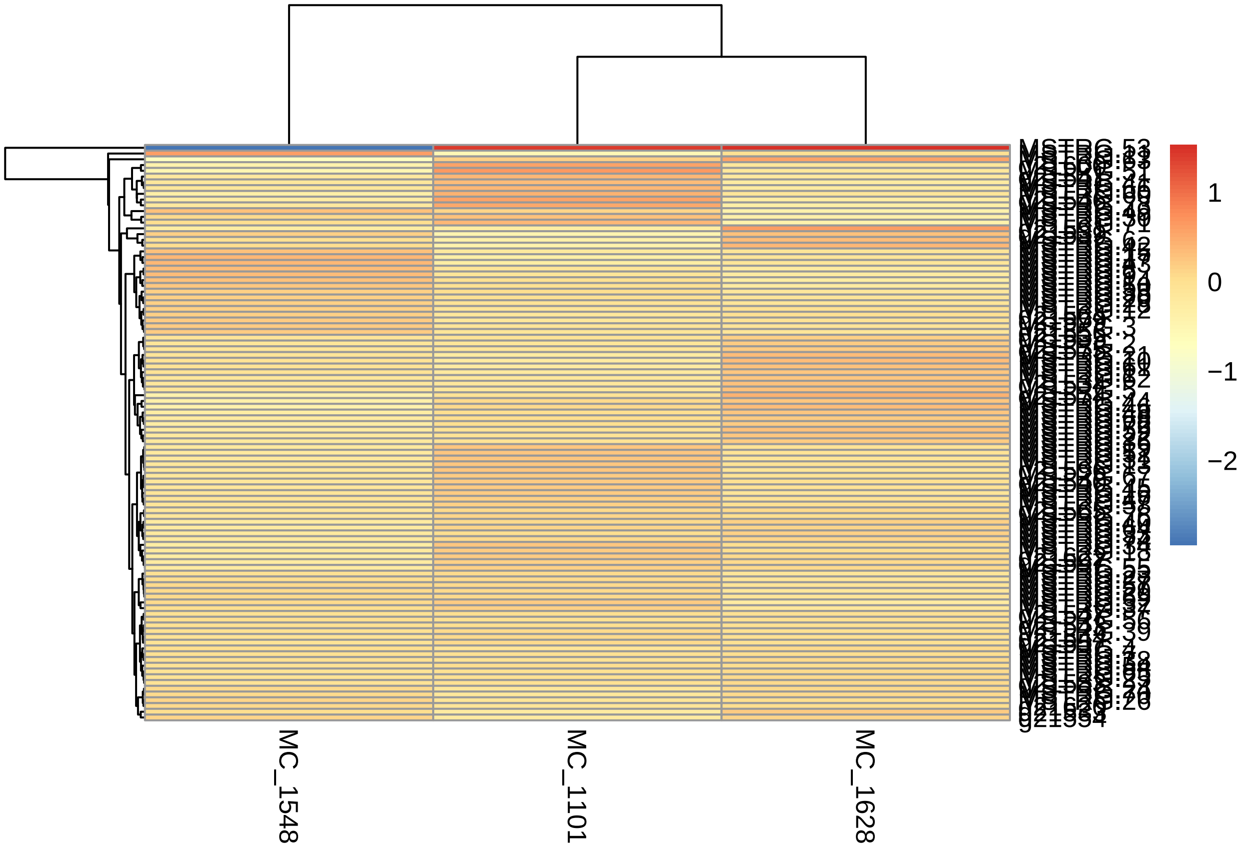
```
PA.gsampleDists<- dist(t(assay(PA.trans.Data))) #calculate distance matix
PA.gsampleDistMatrix <- as.matrix(PA.gsampleDists) #distance matrix
rownames(PA.gsampleDistMatrix) <- colnames(PA.trans.Data) #assign row names
colnames(PA.gsampleDistMatrix) <- NULL #assign col names
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")) )(255) #assign colors
pheatmap(PA.gsampleDistMatrix, #plot matrix
         clustering_distance_rows=PA.gsampleDists, #cluster rows
         clustering_distance_cols=PA.gsampleDists, #cluster columns
         col=colors) #set colors
```



Mcap

Plot a heatmap of a subset of genes to visualize outliers

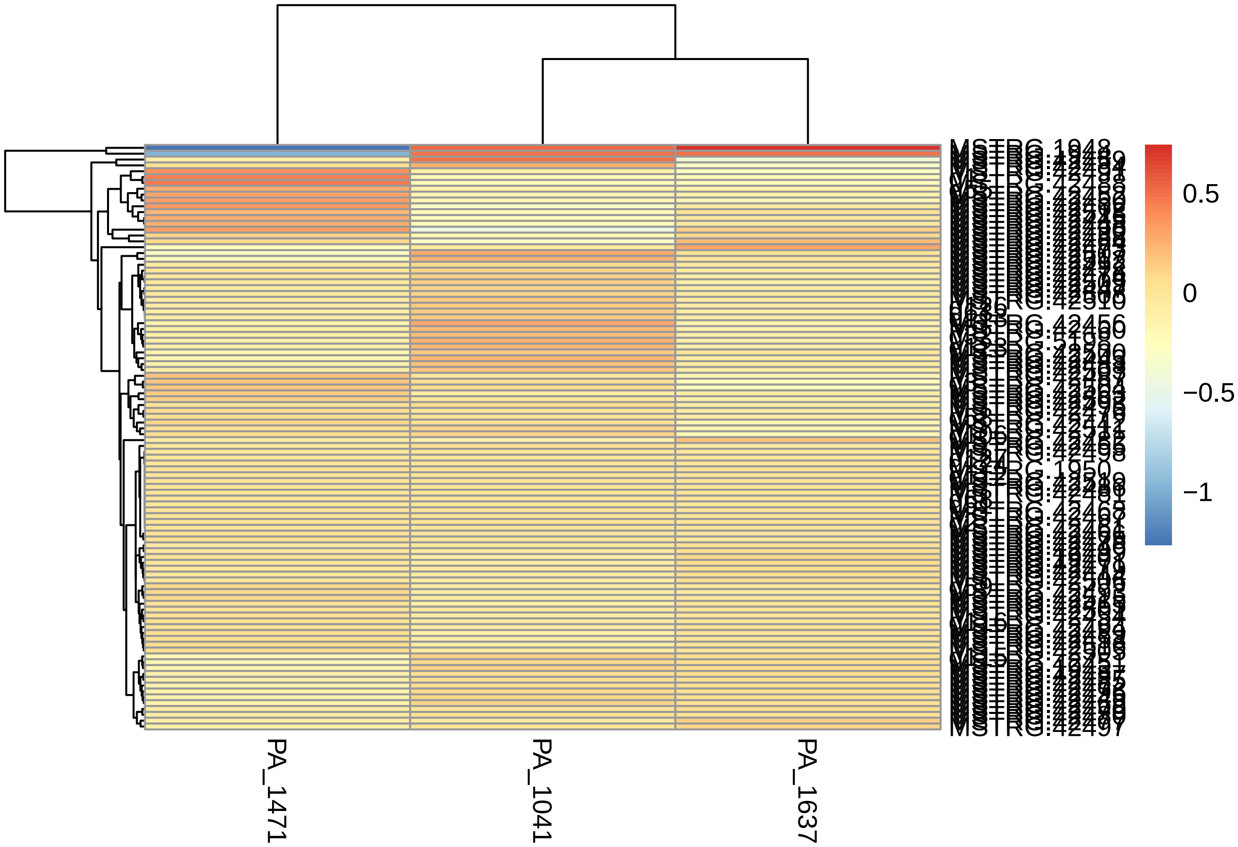
```
MC.mat <- assay(MC.trans.Data[c(1:100)]) #make an expression object
MC.mat <- MC.mat - rowMeans(MC.mat) #difference in expression compared to average across all samples
#pdf("MC.GeneExp.Corr.pdf")
pheatmap(MC.mat,
         clustering_distance_rows="euclidean", clustering_method = "average",
         show_rownames =TRUE,
         show_colnames =TRUE,
         cluster_cols = TRUE)
```



```
#dev.off()
```

Pact

```
PA.mat <- assay(PA.trans.Data[c(1:100)]) #make an expression object
PA.mat <- PA.mat - rowMeans(PA.mat) #difference in expression compared to average across all samples
#pdf("PA.GeneExp.Corr.pdf")
pheatmap(PA.mat,
         clustering_distance_rows="euclidean", clustering_method = "average",
         show_rownames =TRUE,
         show_colnames =TRUE,
         cluster_cols = TRUE)
```


MSTRG.1948
MSTRG.42510
MSTRG.42456
MSTRG.1950
MSTRG.42465
MSTRG.42497
PA_1471
PA_1041
PA_1637
0.5
0
−0.5
−1

```
#dev.off()
```