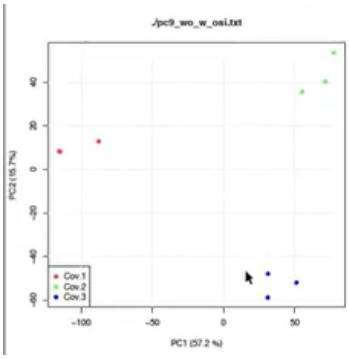
Data analysis – Prof. Calogero

Lesson 7

SOLUTIONS OF THE PREVIOUS LESSON'S EXERCISE

```
library(docker4seq)
pca(
  experiment.table = "./pc9_wo_w_osi.txt",
  type = "counts",
  covariatesInNames = T,
  samplesName = F,
  principal.components = c(1, 2),
  legend.position = "bottomleft",
  pdf = TRUE,
  output.folder = getwd()
)
```

I had to use the function pca, present in docker4seq. The file is made of counts, which are easier to manage in contrast to the tpm file. Cov1 = DMSO; Cov2 = chronic treatment; Cov3 = acute treatment. samplesName = F to see how the points are organized. Finally, only the 1st and the 2nd component were selected.



In the PDF, there are 3 red points (even though I see 2 of them because they are superimposed). What conclusions can I say about this file? The difference between treatment and non-treatment is very big, while the difference between the 2 treatments is small but visible. During this lesson we will go through the differential expression analysis, and we will evaluate which genes change their expression in the DMSO vs chronic or DMSO vs acute treatment.

```
counts2cpm <- function(file, sep = ","){
  tmp <- read.table(file, sep=sep, header=T, row.names=1)
  col.sum <- apply(tmp, 2, sum)
  tmp1 <- t(tmp)/col.sum
  tmp1 <- t(tmp1)
  tmp1 <- tmp1 * 1000000
  write.table(tmp1, "cpm.csv", sep=",",
              col.names=NA)
  write.table(log2(tmp1 + 1), "log2cpm.csv",
              sep=",", col.names=NA)
}
```

```
counts2cpm(file="saver_ctrl.csv", sep=",")
file.rename(from="log2CPM.csv", to="ctrl_log2CPM.csv")

counts2cpm(file="saver_osi.csv", sep=",")
file.rename(from="log2CPM.csv", to="osi_log2CPM.csv")
```

Next step: given a data reduction method, I had to evaluate the overall characteristics of 2 experiments of sc analysis = the untreated PC9 cell line and the treated cell line for 3 days with Osimertinib. Concept of sc is: first of all, I look at the structure of the untreated cell line → is it homogeneous or made of more than 1 sub-population? Then, after the treatment, I have to evaluate if something changes compared to the initial condition. The data given by the prof are already imputed: the critical problem about sc analyses is that they are 0-inflated, so the gene expression is not reliably represented because their expression is not tracked during the extraction of the RNA. This can be partially moderated with the use of statistical models, which try to obtain the expected behaviour of the expression data, looking at a gene per time; finally, they perform imputation to give the expected result. Imputation helps to improve the quality of sc data during these analyses. Given the imputation, the values are converted into log2cpm, in order to make data similar to the real expression values I am expecting: this step performs normalization for visualization purposes. Then, 2 software are used: tSNE (which focuses on the overall behaviour) and umap (which focuses on the global overview of data). In case of tSNE, the columns are the cells, and the basic parameters are used (perplexity = 30). Once the tsne.out results are generated, they are

```
tsne.out<- Rtsne(as.matrix(t(tmp)), pca=FALSE, perplexity=30, theta=0.0)
f=data.frame(x = as.numeric(tsne.out$Y[,1]),y = tsne.out$Y[,2])
#plotting tSNE
sp <- ggplot(f, aes(x=x,y=y)) + geom_point(pch=19, cex=0.3)
pdf("ctrl_noPCA.pdf")
print(sp)
dev.off()
```

a list (=a container of any kind of data. Each element of the list can be different). The first element of the list is a dataframe with 2 columns (x and y), which represent the space in which I can build my visualization. Actually, there is the possibility to change the number of dimensions I would like to represent: I might decide to use more than 1 column or to work with multiple dimensions. Step: data frame creation → ggplot on the data frame. Pch19 is the full dot; cex=0.3 indicates the dimension of the dot (1 is the standard size) → creation of a PDF → print the output of the ggplot → close. In this case, there is no separation among different cells, because I don't know if I have different cells. I simply plot the data in order to see how they look like.

PDF #1

The cells seem to be homogeneous and not so well separated.

PDF #2

The same analysis has been performed with the same parameters, but after the acute treatment: some cells start separating from the bulk, and there is a little extension from the large set of cells: something starts to change during a brief treatment. For next time, I will have to do a differential expression analysis → once I identify the differentially expressed genes, I have to select only the subset of genes that change during the chronic or acute treatment.

Instead of using all the genes for single cell, select only the subset of genes that are differentially expressed in the bulkRNA seq to get a better representation on the UMAP plot.

For bulkRNAseq dataset, you have to compare acute treatment (A→24h Osimertinib treatment) vs MOCK (M→only DMSO) and chronic treatment (C→21 days Osimertinib treatment) vs MOCK (M→only DMSO) . Comparing this two groups you find the list of differentially expressed genes (DE).

A(acute treatment)/M → DE_A (Differential expression genes for acute treatement)

C(chronic treatment)/M → DE_C (Differential expression genes for chronic treatment)

! We are not really interest in the intersection between the DE genes. We are interest to have the two independent groups (acute and chronic) and the group that get all together (combining acute and chronic).

Then you take the full list of genes from the single cell data treadted with osi (because are the only one in which you can see a difference). Looking at the image above to have a better clear concept. In the red circle you have all the DE genes in the acute treatment, in the blue circle you have all the DE genes following the chronic treatment and in the intersection you have the DE genes that are in common between the acute and the chronic treatment. If we start to select the only the DE genes in the response to the drug and we use these selected genes to perform UMAP on single cell dataset, we can grab a better separation of the cells that are acquiring resistance. In other words, we can better distinguish subpopulation of cells on single cell data. In practice, we want to use bulkRNAseq data to grab more information on the single-cell data. Before doing the comparison and the selection of the DE genes we have to solve one problem: for scRNAseq data the annotation of the genes is ensembleID:symbol while for bulkRNAseq data the annotation is symbol:ensembleID. To solve this problem we have to use the strsplit function to have the same annotation to perform intersection. The first element of the strsplit function is the element that has to be splitted (in our case is the rownames of the dataset that is a vector conaining the symbol:ensembleID) and the second element is the symbol used for the splitting (in our case is ":").

```
#strsplit function

strsplit(element_to_be_split, symbol_for_split)
```

⚠ This part written is only to solve the annotation problem!!!

```
1. #start with the bulkRNAseq file p9_wo_w_osi.txt and setwd(/wherethefileis)
2. #insert the file as a variable in R
x <- read.table("pc9_wo_w_osi.txt", header=TRUE, sep = "\t", row.names=1)
# with header=TRUE it considers the first line as a header (colum names)
# with row.names=1 it means that the 1st column of the table becomes the rowname of the dataframe
#these 2 options are conceptually the same but one for columns and the other for rows1
3. #to check the file to see if it's correct
head(x)
4. #use strsplit function to create a large list containing the symbol and the ensemble IDs separeted
y <- strsplit(row.names(x), ":")
#the strsplit function take the rownames of the dataframe and separate the name for the ":" character
5. #convert the list to a matrix
matrix <- matrix(unlist(y), ncol = 2, byrow = TRUE)
# at this point you have a list that as to be converted in  a matrix with the unlist function that use the number of colum=2 and unlist by row.
6. #use the paste function to create a vector with the new annotation as in the single cell dataset  ensembleID:symbol
z <- paste(matrix[,2], matrix[,1], sep=":")
#the function paste, paste the two colum of the matrix in the order that we want separeted by the character ":"
7. #put the new row names in the original dataframe (x) where we had all the data
row.names(x) <- z
```

Read the missing part from the sbobina.

UMAP

The file of umap of the control has a very similar behaviour of the tSne one. While umap gives me the global picture, tSNE focuses on the local features. The 2 driving forces of umap is the n_neighbour and the mon_distsnce between neighbours. In the file, there is a set of cells that separates from the others. I can ply with the 2 parameters to see if I can get a better separation among dots. for example, if I reduce the min_distance from 1 to 0.5, more set of cells are different from the bulk.

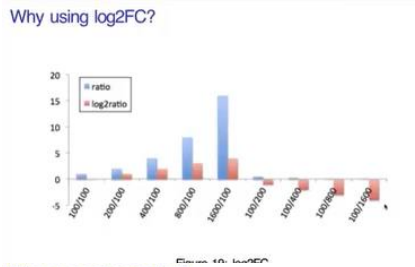
N_epochs = number of times I run my file to obtain the final results. Normally, 200 is too small if the data are very inhomogeneous.

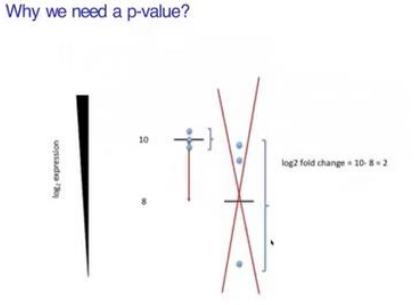
Changing the distance to 0.05 and num_neighbours = 10, another scheme is present. The command is: length(which(f$x > 20))

If I want to select values < -30 on the y axis and values < -50 in the x axis: length(intersect(which(f$y < -30), which (f$x < -50)))

With this command I select a rectangle having the features I want. Starting from the image I obtain, using the genes related to resistance, I can improve the separation from cells treated with Osimertinib: if I use all the genes, a lot of noise will be obtained; if I use the genes related to resistance, maybe I will obtain a better picture.

DIFFERENTIAL EXPRESSION





In bulk, a linear dimensionality reduction is more than enough, while in sc I need more sophisticated dimensionality reduction methods, such as umap (which is unlinear). I want to identify, on the bulk, which genes are differentially expressed comparing 2 conditions (given the level of expression of a gene in the condition a and b): are these expressions significantly different? I have to represent these changes as log2 fold change = expression of the gene in condition a/expression of the gene in the reference condition. If Er > Ea → 1-0; if Ea > Er → 1-infinite. Data can be represented in the log2 value; as an alternative, I can calculate the log2 of expression a – log2 of expression r → independently to that happens, upregulation will have a positive value and the downregulation will have a negative one. Another way data can be represented is |log2FC| >= 1, because there are many false positives below 1, and data are too noisy: the significant genes are hidden by the false positives. Is

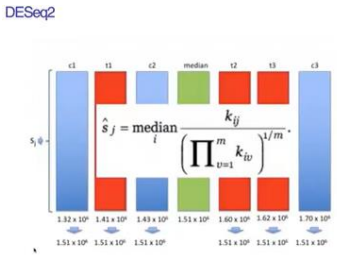
log2 fold-change enough for our analysis or do I need something else? In the slide there is a strong differential expression (2-fold), but in the next slide this is not true: this because the dispersion of data in a sample is so big that if I remove the dot present at the bottom of the picture, the big difference that I see is not so big, actually. This indicates that the dispersion of one sample is so big that the potential mean difference is not significant. To solve this problem, I have to associate a p-value to the FC one. There are 3 types of p-value:

- Adjusted p-value: 0.01 (1 out of 100 genes that are differentially expressed are there by chance)
- 0.05 (5 out of 100 genes that are differentially expressed are there by chance)
- 0.1 (10% false positives)

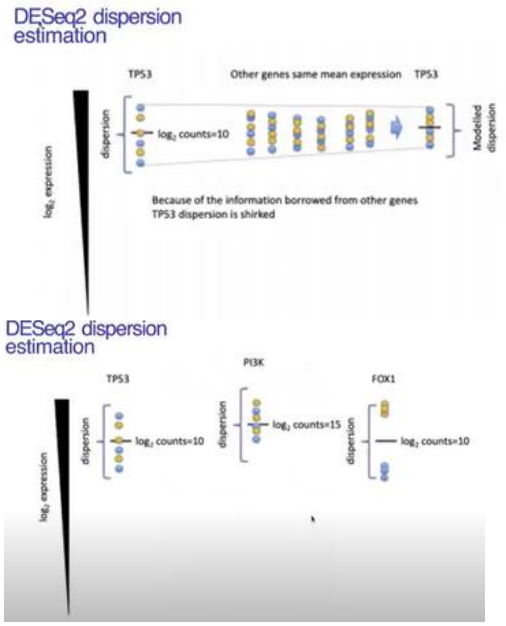
The choice of the p-value depends on which data I want to extract: noisy data → 10%; big difference among data → 0.01

Normally, the p-value and the log2 FC are put together to analyse data. The genes characterized by a 0.05 of p-value and a log2FC >= 1 are generally used to filter data. What do I have to do to find differential expression?

1) Finding a good data distribution that allows to fit my statistical model. The most used distribution used in biological tests is the t-test. However, because of the way data are distributed in sc (which is more Poisson-like), I have to use a negative binomial distribution, which combines the Poisson distribution with the normal one. I take in account the way I am sub-sampling the data from the RNAseq analysis and, at the same time, I take in account the biological variability of the replicates.
2) Normalization → with respect to the previous types of data (like microarrays), in which normalization was done on the data before the analysis, in this case I use a statistical test that uses this type of information: I have to use the row data, in order to conserve the variability. Then, the statistical model performs normalization and detects the differentially expressed genes. I give the list of my sample (for example: 3 DMSO + 3 acute treatment samples).
   Normalization = I have to normalize the differences related to the different sampling present in my experimental settings. When I do sequencing, not all the libraries have the same size → the overall numbers I collect are not the same → normalization tries to scale the data in order to obtain the same overall counts for each sample. Example: in an experiment I have 9 million counts, while in another experiment I have 11 million counts, the final counts for each sample will be around 10. However, in this step, I do not adjust the count distribution between samples: the distribution of each sample is different. If I have the mean expression of genes, they are realigned in order to be the same to each other. Another very important thing is that I assume that most of the genes are not differentially expressed (this is the strongest and important assumption): in an experiment, for example, I inactivate histone acetylases → all the regions that, in chromatin, are opened, they will close most of the regions that were previously open, now they are closed. This experiment cannot be associated to a sc analysis, because the modifications observed in cells are too big. Instead, differential expression works only if I work on a small subset of elements (a few genes changing). We will focus on how to do differential expression with DESeq2, which is a package of Bioconductor, that must be installed on RStudio (it will take some time). Then, using the vignette of DESeq2, do the differential expression analysis in the DMSO vs acute or DMSO vs chronic treatment. What does DESeq do to perform normalization? DESeq takes the mean of each gene's counts across all the samples (all the reads get the mean value) and divides them by the geometrical mean referred to the gene in analysis → I have a list of ratios, that are the parameters that must be applied to each gene to convert data in geometrical means → I take all the values and make the geometrical mean of these values.



Example: I have 6 samples, that I order according to the full number of reads that characterizes each sample → I calculate the geometrical mean for all the genes (median column) → I calculate, for each gene, the ratio, obtaining the scaling factor that I should apply to gene 1 to make it identical to the median. If I have the scaling factor of each gene (that allows me to convert the count into the median value), I can take this value and calculate the mean of it: this value will be divided for all the values of all the genes present in sample 1. Then, I do the same step for t1 and so on. At the end, the overall size of each sample is going to be identical to the median because of the scaling procedure I did; I eliminate the effects of the different sampling that were coming from the library size. Formula: expression of gene "I" in the sample "j" divided by the geometrical mean of the corresponding gene in all the samples. Once I collected these values, I calculate the median value of gene "I" and multiply it for the ratio → I obtain the sample's normalization value for the experiment. To summarize, in this step I take out the technical variability related to the different sampling I did.

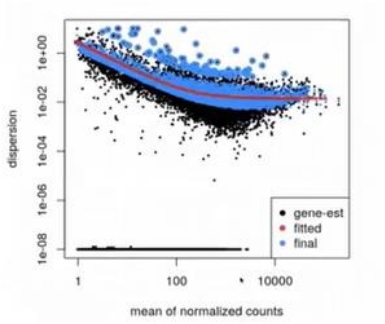


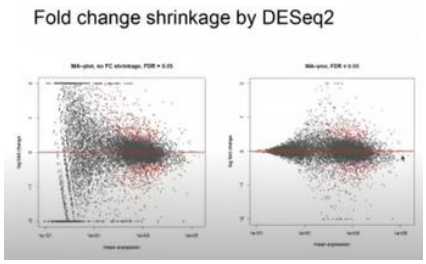
3) Dispersion estimation is the most important thing that must be calculated to evaluate what genes are most differentially expressed: if a gene is differentially expressed I expect that it changes very much from one sample to another, so it gives a big difference between the control and the treated sample. The idea is the following: I try to estimate the dispersion for the genes that are not differentially expressed, so that the differentially expressed ones are considered as "outliers" with respect to the rest. Dispersion is the square of the gene's biological coefficient variation, and it contains 2 information: the sample-to-sample variation (the one I am interested in. REMEMBER that most of the genes are not differentially expressed) and the uncertainty of the measurement of the expression (due to the system's noise). Genes that do not have enough information are discarded: DESeq shrinks this set of differentially expressed genes without considering the one that are associated to few counts. Example: the mean expression of my read in the control is 1 read; the gene expression in the treated sample is 2 reads. This seems to be a good scheme, but I have a few material for my gene → differences are not reliable and this gene is removed from my analysis. The gene's expression will be equal to NA (not available) when DESeq is run. This figure represents 3 scenarios: no differential expression in 3 cases and a differential expression. Let's analyse the example of p53. 3 replicates per condition are present → there is uncertainty in detecting the expression and the dispersion (which is part of the game). However, all the genes having the same expression share a very similar uncertainty: if an expression's log2FC value is =10, I can take other genes having this same expression and see how they are distributed. Since most of the genes are not differentially expressed and the dispersion depends on the noise in the measurement, if I borrow the information from other genes that have the same differential expression, I should improve the representation of p53 dispersion, shrinking it. The modelled dispersion tells me that, since all the genes that are characterized by the same expression level of p53 have a smaller dispersion than p53, p53 was not measured correctly, so I have to shrink its expression: in summary, I borrow the information derived from other genes having the same expression of the one I am interested in, assuming that most of the genes are not differentially expressed.

Opposite situation: PI3K. The average expression is 15, and I see that most of the genes have a bigger dispersion → I have underestimated the expression of my sample. Borrowing the information from the others, I will expand the dispersion for PI3K.

The dispersion is shrinked or expanded according to the values that I get from the other results.

What happens when I have differentially expressed genes? FOX1 has the same average expression of p53 → I collect the same set of genes I collected for p53. However, the dispersion due to differential expression is too big to be similar to what I am expecting to have the same expression of p53 in my gene → I cannot shrink it, and I have to consider it as an outlier: I won't do any normalization. All my outliers are the differentially expressed genes. Why do I need a few differentially expressed genes? Because if most of the genes are differentially expressed, during normalization the variance will shrink, and differentially expression data will be lost. Dispersion is plotted against mean normalized counts. The black dots are the real values of dispersion of each gene, the blue dots represent the shrinked dispersion, while the red curve represents the expected trend for the dispersion. As I can see, dispersion is big for low expression, then it stabilizes when the level of expression is around 100 counts. The blue dots circled by black are the outliers: their expression is > than the expected expression of the ther differentially expressed genes. I try to use this information to extract them from the noise. Dispersion is the most important one because it is the way I label differentially expressed genes. Now, I have to evaluate if these genes are sufficiently different to be considered really differentially expressed. Big dispersion → no fitting in the noise. Some of the real differentially expressed genes are lost because there is too much noise that cannot be solved with the number of samples I have. If, instead of 3 replicates, I work with 50 replicates, dispersion shrinks and I will not see some of the differentially expressed genes because they are masked by the noise.



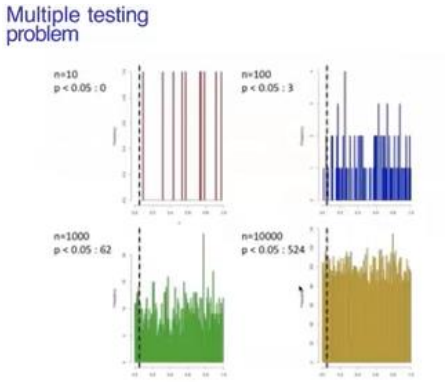


4) Statistical changing with DESeq2: it shrinks the log2FC (it takes out the genes characterized by a few reads and whose variation is not robust enough).

5) Log fold change estimation: in a normal experiment, the log2FC looks like this. The black dots present at the bottom are false data; DESeq removes these genes because they are not significant, while it leaves the others, on which it does the differential expression analysis. Red dots = different p-value from the expected average dispersion of data. The statistics used by DESeq is the Wald test significance (similar to a t-test, although I estimate something different): I do a sort of transformation of data into z-score (=distance between the data and their expected mean: up-modulation → + delta and vice versa). I estimate if the z-score is far away in a normal distribution of the z-score I have: I evaluate if the z-score of my differentially expressed genes drops on the tail of the normal distribution of the z-score. In this case, I don't use anymore the mean value. This provides me a row p-value, which indicates if a gene, given a threshold, is differentially expressed or not.

Correction by Cook's distance: if a sample has a different mean value, this sample is removed.

6) Multiple testing correction: I am using ~20000 genes, and I test them to evaluate if their p-value indicates their differential expression. P-value's distribution is uniform: each value has the same probability of another value to come out. The reason why it comes out is due to how many times I sample my data: if I sample my data 10 times and my threshold is 0.05, there is no sample, among 10 samples, having this value. If I sample 100 times → 3 samples have a $p < 0.05$. n = 1000 times → 62; 100000 times → 524 times $p < 0.05$. 20000 genes → 1000 times $p < 0.05$.



Problem: I have to find a way to eliminate this part of the unform distribution. I accept a certain number of false negatives contaminating my data, and if I redo the experiment I will allow the false negatives to drop in a region of the graph. These high-throughput experiments provide potential targets that I have to validate, in order to apply the biological question "are these targets important for that? In this case, the resistance to the drug allows me to see cells starting to change even if the treatment of the cell lasts short. Can I use some biological information coming from the bulk analysis to see if I manage to separate the sub-populations of cells even at the beginning of the treatment? In this experiment I'm not interested in the single genes, but in the effect that resistance (determined by many genes) does on my single cell. Then, if one of the genes is there by chance, the effect is little.

Here there is a sample, in which there are no changes and genes are uniformly present (their expression follows a normal distribution). Some of them has a different expression, even though it is part of the set → tail of the distribution.

I have 2 sets of samples made of non-differentially expressed genes; the 2 sets are treated with water that has been distilled in two different days; for each culture there are 3 replicates. The water I'm adding gives no effect, so the differences I observe are casual. The genes coming from experiment 1 and 2 stay in the mean of the expression level. In both experiments I'm going to have outliers, related to genes present in the tail because of a low expression or by chance. Even if the 2 datasets are not differentially expressed, some genes result to be → I have to eliminate them when I look at the correction of the p-value.

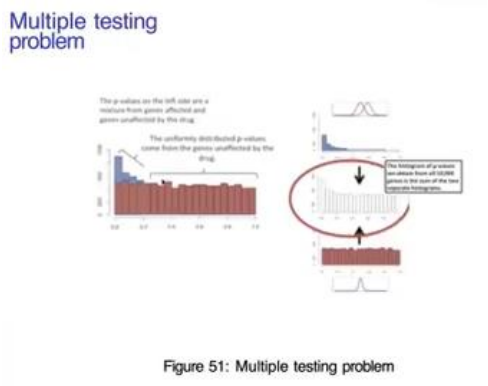


Figure 51: Multiple testing problem

One method that is used to do so is the Benjamini Opper test: if I combine the distribution of genes that are present by chance + the distribution of genes that are differentially expressed, I get a situation like this one; I'm interested in the blue part → there is a way to identify the region that divides the 2 groups. I take the p-value and I rank from the smallest to the highest → I associate the ranking to my genes → the adjusted p-value is going to be associated to the highest gene expression, to which it is identical. For the second gene, I take the normal p-value divided by the rank one, and I multiply it to the current p-value. Example: number of total p-value = 10; number of ranks = 9 → 10/9 → 0,83/(10/9) = 0,90 → my p-value increases. If I do this calculation for all the samples, the last p-value is not significant anymore. With this system, simply ordering the p-value from the smallest to the highest and considering the ranking of the data, I can identify the threshold that separates the subset of the true positives with respect to all the others. The values present in the red square are the false negatives that I can only correct increasing the sample size of my experiment. Combining the adjusted p-value + the log2FC decided as threshold, I can obtain a list of genes that are associated to the parameter I'm investigating (ex: acute treatment or chronic treatment).

Exercise (the solutions will be provided in lesson 8 and 9).

Exercise

- Install in your Rstudio Bioconductor DESeq2 package.
- Following DESeq2 vignette: detect the genes called differential expressed comparing:
  - DMSO versus acute osi,
    - Thresholds: $|logFC| \geq 1$ & adj-p-val ≤ 0.05
  - DMSO versus cronic osi
    - Thresholds: $|logFC| \geq 1$ & adj-p-val ≤ 0.05
- Filter osi_log2CPM.csv using DE from acute osi and plot UMAP
  - There is any difference with respect to what can be observed using all genes?
- Filter osi_log2CPM.csv using DE from cronic osi and plot UMAP
  - There is any difference with respect to what can be observed using all genes?
- Filter osi_log2CPM.csv using combined DE from cronic/acute osi and plot UMAP
  - There is any difference with respect to what can be observed using all genes?