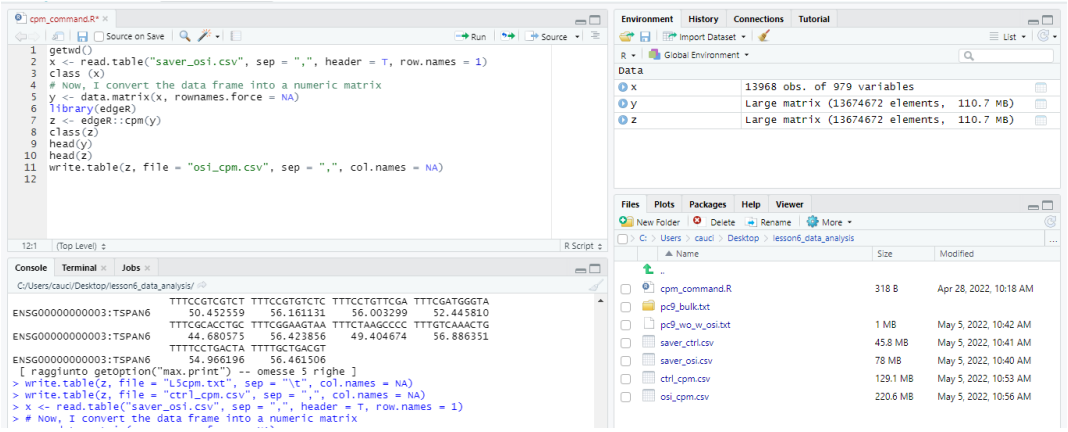
Exercise 6

Exercise

- Install on your R docker image Rtsne and umap packages
- Convert PC9 ctrl and Osimertinib treated cells in cpm.
- Run umap and Rtsne changing the parameter indicated in the tsne_umap tutorial provided to you:
  - umap min_dist
  - tSne perplexity

I install the 2 packages firstly in the R program present in my desktop, then I will do that also on the docker. Then, I download the .zip files present on Moodle, and I unzip them. In the folder "lesson6_data_analysis" I convert the control and osi files into cpm, using the same command line I used for lesson 5 but changing the separator. To do so, edgeR is needed.

The final output will be ctrl_cpm.csv and osi_cpm.csv files.



! The files provided by the professor are already saver-imputed: imputation aims to address the sparsity = the large number of zeroes observed in sc data. SAVER (single-cell analysis via expression recovery) is an expression recovery method for UMI-based scRNAseq data borrowing information across genes and cells to provide accurate expression estimates for all the genes. For this exercise, SAVER must not be applied because it has been already done by the professor.

Now, the SAVER-imputed files must be converted into log2cpm: this can be done by adapting the script on the tutorial present on Moodle.

counts2cpm.R*

Source on Save

```r
# Start from a SAVER imputed .csv file
# Create the function counts2cpm
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)
}
# Recall the function and run it on your ctrl file
counts2cpm(file = "saver_ctrl.csv", sep = ",")
file.rename(from = "log2cpm.csv", to = "ctrl_log2cpm.csv")
```

The function is the same for both the ctrl and osi file. The "cpm.csv" file that is created from this function in my folder is not useful for this exercise.

To do the last point of the exercise, I need to install umap and tSNE with the command install.packages("umap") and install.packages("ggplot2")

umap.R*

```
install.packages("umap")
install.packages("ggplot2")
library("umap")
library("ggplot2")

# Apply the umap function to the ctrl_log2cpm.csv file
ctrl <- read.table("ctrl_log2cpm.csv", sep=",", header=T, row.names=1)
ctrl.umap <- umap(t(ctrl), random_state=111, n_epochs = 1000)
f=data.frame(x=as.numeric(ctrl.umap$layout[,1]),y=as.numeric(ctrl.umap$layout[,2]))

# Plotting umap
sp <- ggplot(f, aes(x=x,y=y)) + geom_point(pch=19, cex=0.3)
pdf("ctrlUMAP.pdf")
print(sp)
dev.off()

# Apply the same function to the osi_log2cpm.csv
osi <- read.table("osi_log2cpm.csv", sep=",", header=T, row.names=1)
osi.umap <- umap(t(osi), random_state=111, n_epochs = 1000)
f=data.frame(x=as.numeric(osi.umap$layout[,1]),y=as.numeric(osi.umap$layout[,2]))

# Plotting umap
sp <- ggplot(f, aes(x=x,y=y)) + geom_point(pch=19, cex=0.3)
pdf("osiUMAP.pdf")
print(sp)
dev.off()
```

To run tSNE:

tsne.R*

```
install.packages("Rtsne")
library("Rtsne")
library("ggplot2")

#CTRL samples
tmp <- read.table("ctrl_log2cpm.csv", sep=",", header=T, row.names=1)
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 the tsne
sp <- ggplot(f, aes(x=x,y=y)) + geom_point(pch=19, cex=0.3)
pdf("ctrl_tsne_noPCA.pdf")
print(sp)
dev.off()
#OSI samples
tmp <- read.table("osi_log2cpm.csv", sep=",", header=T, row.names=1)
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 the tsne
sp <- ggplot(f, aes(x=x,y=y)) + geom_point(pch=19, cex=0.3)
pdf("osi_tsne_noPCA.pdf")
print(sp)
dev.off()
```