Universitat Politècnica de València

Escola Tècnica Superior D'Ingenyeria Agronòmica I del Medi Natural

Grado en Biotecnología

Evaluation of sequencing technologies Single Cell in the field of systems biology.

Biotechnology Bachelor's Thesis
(Trabajo de Fin de Grado en Biotecnología)

Academical year 2016 – 2017

STUDENT: Diego Traver Larraz

TUTOR: Javier Forment Millet

EXTERNAL TUTOR: Jose Carbonell Caballero

Valencia, July 2017

CC BY NC ND

Title: Evaluation of sequencing technologies Single Cell in the field of systems biology.

Author: D. Diego Traver Larraz

Tutor: Prof. D. Javier Forment Millet

External Tutor: Prof. D. Jose Carbonell Caballero

License: Creative Commons, Non-Commercial (NC) and Non-Derivative Works (ND)

Location: Valencia, July 2017

Abstract

Systems biology enables us to make a mechanical approach of human illnesses, or in general of the phenotypic characteristics of the population. In this scene modelling and analysis of the changes in some parts from the system, like metabolic or signalling pathways, allows to describe how procedures of the cellular machinery are modified by alterations in single elements from the system like genes or metabolites. Nowadays, Next Generation Sequencing is being a significant improvement in genomic studies, as they enable to simultaneously check the activity of most of coding genes of the cell in a determined data. Thus, Single Cell technologies are an additional improvement as they allow to collect information from the activity from single cells. This is a meaningful difference in the study of illnesses like cancer or in other contexts where the interaction between different cell types presents in the same sample play an essential role. This work will evaluate the impact of those technologies in different computational models.

Keywords

Single Cell, Next Generation Sequencing, NGS, Systems Biology, Bioinformatic, Genomic.

Resumen

La biología de sistemas permite realizar un abordaje mecanístico de las enfermedades humanas, o en un caso general, de las características fenotípicas de la población. En este escenario, la modelización y el análisis de los cambios en partes esenciales del sistema, como rutas metabólicas o de señalización, permiten describir como el funcionamiento de la maquinaria celular es perturbado a partir de alteraciones en los elementos individuales del sistema como genes o metabolitos. En la actualidad, las tecnologías de secuenciación de nueva generación (NGS, Next Generation Sequencing) han supuesto una mejorar significativa en los estudios genómicos, ya que permiten evaluar de forma simultánea la actividad de la gran mayoría de genes codificantes de la célula en una muestra dada. En ese sentido, las tecnologías de Single Cell, constituyen además una mejora adicional, ya que permiten trazar la actividad de células individuales, lo que supone una mejora substancial en el estudio de enfermedades como el cáncer, o en contextos donde la interacción de los distintos tipos celulares presentes en una misma muestra juega un papel esencial. En este trabajo se evaluará el impacto de dichas tecnologías en diferentes modelos computacionales.

Palabras clave:

Single Cell, Next Generation Sequencing, NGS, Biología de Sistemas, Bioinformática, Genómica.

Resum

La biologia de sistemes permet realitzar un abordament mecanístic de les malalties humanes, o en una situació general, de les característiques fenotípiques de la població. En aquest escenari, la modelització i l'anàlisi dels canvis en parts essencials del sistema, com rutes metabòliques o de senyalització, permeten descriure com el funcionament de la maquinaria cel·lular es pertorbat a partir d'alteracions en els elements individuals del sistema com gens o metabòlits. A l'actualitat, les tecnologies de seqüenciació de nova generació (NGS, Next Generation Sequencing) han suposat una millora significativa dels estudis genòmics, perquè permeten avaluar de forma simultània l'activitat de la gran majoria de gens codificants de la cèl·lula en una mostra determinada. D'aquesta forma, les tecnologies de Single Cell, constitueixen a més a més una millora addicional, ja que permeten traçar l'activitat de les cèl·lules individuals, el que suposa una millora substancial en l'estudi de les malalties com el càncer, o en contextos on la interacció dels diferents tipus cel·lulars presents a una mateixa mostra juguen un paper essencial. A aquest treball avaluarà l'impacte de les esmentades tecnologies a diferents models computacionals.

Paraules Clau:

Single Cell, Next Generation Sequencing, NGS, Biologia de Sistemes, Bioinformàtica, Genòmica.

Acknowledgements

This thesis is not the work from last months, but the work of all the years of study that brought me here.
Thus, like if it was a cake, each person who contributed on my education should have a proportional recognition part of this.

I want say a warm *thank you* to all the people from the "Genomica de sistemas" laboratory at CIPF, specially to Paco and Rubén who helped me a lot with all the problems I had dealing with the data and my laptop respectively.

Thanks to my tutors, Javier and Jose, who were patient enough with this student who wanted to discover the world of bioinformatics and managed to survive somehow…

Thanks to those friends who have been always there joking, and giving the support needed in the right moment.

A big thanks to the *kool kidz.* All those hours spent together with the weirdest and darkest humour ever to relax ourselves before, meanwhile, and after the exams, presentations, labs… It didn't matter where in the world we were.

But above all,

specially an enormous thanks to my family.

Who always gave me, gives me, and will give me support in all my projects.

Index

Figure Index

Abbreviations

EXPRS – Expression

GEO – Gene Expression Omnibus

GNU – GNU is not Unix

NGS – Next Generation Sequencing

PCA – Principal Component Analysis

PCR – Polymerase Chain Reaction

RNA – Ribonucleic acid

SC – Single Cell

SCE – Single Cell Expression

Seq – Sequencing

1. Introduction

Bioinformatics is the union of two different fields; on one hand, all the concepts related to the physical-chemical properties of the molecules, and on the other hand the techniques from the disciplines of applied math and statistics (What is bioinformatics?)[22]. When the sequencing methods came out in the 70s, the need to process large quantity of data forced biologist to start working with computational techniques even if they were a bit reticent to used them (Caulfield, T. *et al.*)[1].

In research, it is necessary to make tests to approach as much as possible to the living conditions. Thus, when using model organisms is not possible, systems biology approaches by mechanical modelling. At this moment, the era of the *Big Data* has made a revolution with the improvement of the technologies of the Next Generation Sequencing. Dealing with such a huge quantity of information has forced biologist to improve the way to work with it, and this turn into in the incorporation of new computational and statistical tools like the R programming language.

1.1 Single Cell RNA-sequencing

This expression defines the methodology of getting genomic data from a single isolated cell from a given sample. This procedure is interesting to take into account when single mutations on cells has an effect on the whole environment. This procedure is focused to investigate diseases as cancer where the resistance to a drug may cause a resistant tumour on the patient (Ellsworth, D. L. *et al.*)[14].

Before starting with the sequencing step, it is necessary to split the bulk sample into isolated single cells on the wells where the sequencing step will take place. Nowadays there are different procedures for isolation which may deal with different types of samples. In a huge number of cells, the most common method is to use a Fluorescent Activation Cell Sorting (FACS), where cells are isolated according to determined characteristics and deposited into wells (Herzenberg, L. A. *et al*)[2].

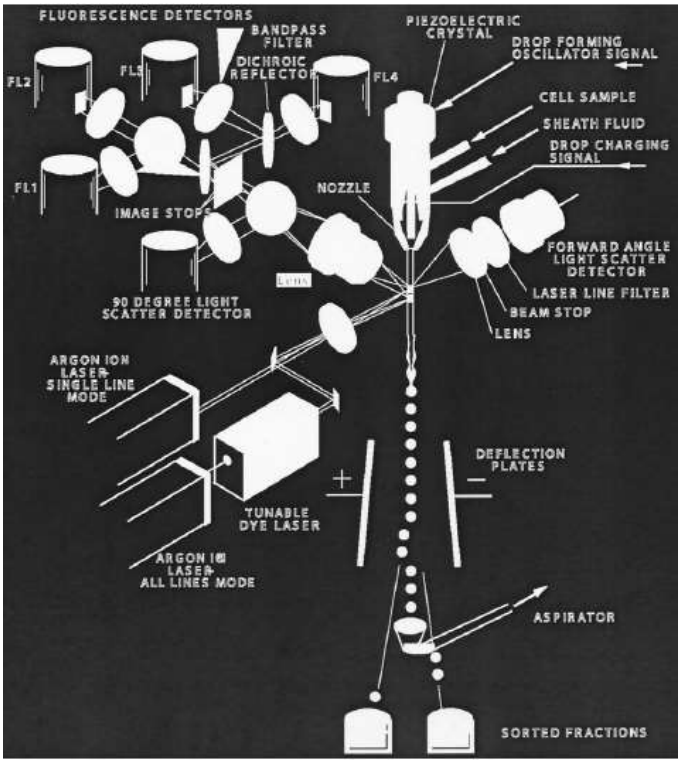

Image 1. Diagram from a Fluorescent Activation Cell Sorting (FACS) machine (Herzenberg, L. A. *et al*)[1].

If working with an important sample, sometimes is preferred to proceed making an isolation of the cells by manual cell picking with microcapillaries and tips from the pipettes. One tip is used to make the vacuum effect with the mouth and the other will aspire the cell.

Once the cells have been isolated, it is necessary to prepare them for the RNA-seq step. The following procedure which goes from the cell lysis to get the final transcripts after being amplify by the Polymerase Chain Reaction (PCR) were described previously by Picelli, S. *et al.*[8] The following image summarizes the protocol used for the creation of the transcripts library.

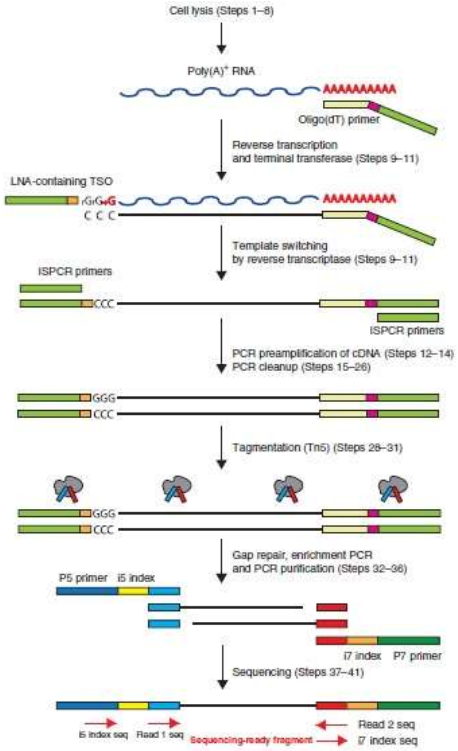

Image 2. Flowchart of Smart-seq2 protocol. Steps involving from the single cell lysis to the sequencing step (Picelli, S. *et al.)*[8].

1.2 The R language

R is a programming language which has been developed for statistical computing and graphics. It is a GNU project (it means it is an operative system of free software), which has similarities to the S language and environment. R was developed by John Chambers and colleagues at Bell Laboratories.
This language is considered to be a different implementation of S, even though most of S code runs without modification under R, some of its important characteristics are the key points to make R a different language from S.

This language provides a large range of statistical and graphical techniques, and is highly extensible. It has an integrated repertory of software facilities for data manipulation, calculation, graphical display, and storage capacity.

Defining R as an “environment” it is done on the porpoise to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as frequently is the situation with other data analysis software.

(The R-project)[19]

1.3 Biconductor

Bioconductor is a repository which provides tools for the analysis and comprehension of high-throughput genomic data. The packages of the repository are run under R language. This website hosts about 1383 software packages available to be downloaded for free use. All packages are being actualised time after time and new ones with better features are being uploaded.

(Bioconductor)[18]

2. Objectives

The main objectives from this bachelor's thesis have been to understand Single Cell RNA-sequencing technology, and to reproduce the results obtained by Li J. *et al*[11] where a pancreatic sample was sequenced and cells were classified according to its cellular type.
Thus, following different pipelines the idea was to demonstrate the powerful technology that Bioinformatics analysis gives to sample identification.

3. Materials and Methods

For this thesis, the methodology used was to follow the workflow for the analysis of Single Cell RNA-seq by A.T.L. Lun *et al.*[12]. Some modifications were done on the data processing as the data to deal with were different in some features.

All the scrips used in this project are available on the attachments section.

3.1 Computational resources

3.1.1 Hardware

For data analysis, the following platforms were used:

A) Computer with Xubuntu Operative System Version:16.04 with 6gb RAM and i7-6820HK processor.
B) Laptop with Windows 10 Operative System with 6gb RAM and i7 3680 processor.

3.1.2 Software

A) The version of R used in this bachelor's thesis was 3.4 "*You stupid darkness"*. (The R project)[19].
B) To work with the R code, the program used was RStudio. This program is available to be downloaded at its website for free use. (RStudio)[20].

3.2 Samples

The data used in this bachelor's thesis were published and it is available on the Gene Expression Omnibus database (accession no. GSE73727) Li J. *et al*[11]. The files used in data analysis were the counts files from the 72 sequenced cells from a pancreas islet.

3.3 Data processing

Many difficulties arise in working with Single Cell RNA-sequencing. The problem of this technique is to assume that cells in a given sample are the same when they might be quite different. Some of these characteristics which are assumed to be equal are cell size, cell cycle phase, content of RNA of each cell, etc (Bacher, R. & Kendziorski, C)[10].

Another problem of dealing with sequencing data is the large quantity of information to deal with. The first step is to make a quality control of the sample. Every step must have a check to verify if the information obtained is correct or should be discarded. One of the most common procedures to make a quality control is the use of spike-ins. This term comes from synthetic transcripts spiked into each cell's library at known concentrations; this procedure is used to

estimate relative differences in the RNA content to improve normalization (Bacher, R. & Kendziorski, C)[10].

Reducing the data to a smaller quantity is a must to be able to work with it without losing information by masking it. Once the check is done, the Transcripts Identifiers are changed to Genes Identifiers; thus, it will be easier to work with as the data are reduced by a factor of 5. After the conversion, it is needed to apply a normalization on the data. Working with huge matrices of counts in Single Cell covers up relevant data. This problem happens due to the large amplification step in the sequencing procedure which gives a great number of transcripts with the value of 1 count. There are different methodologies, in this situation quantiles function was applied.

3.4 Analysis of the samples

In order to analyse the data after being processed, two different analysis were performed, a Principal Components Analysis (PCA) and a Heatmap, that integrates a hierarchical clustering.

3.4.1 PCA

The Principal Components Analysis shows the variation of the data of a given sample. To represent this, orthogonal vectors are estimated representing the main variability sources of analysed data and their relation with original features (Rignér, M *et al.)*[3].

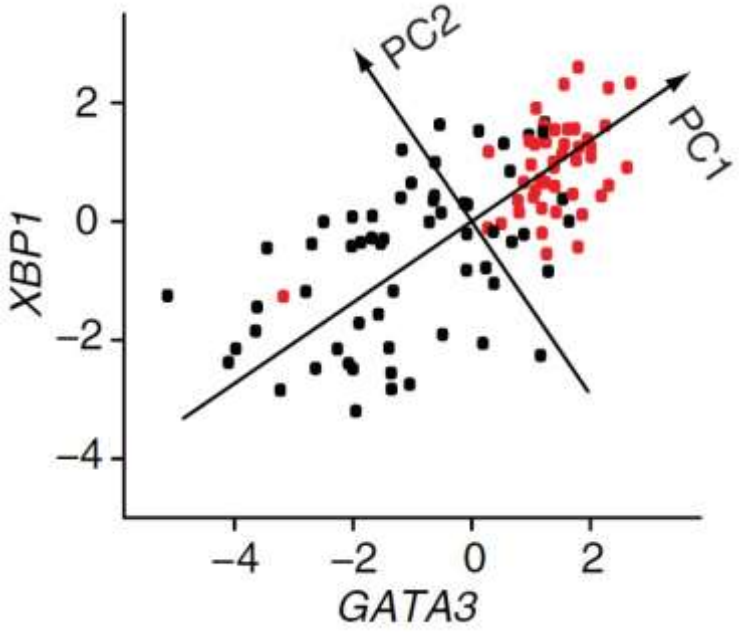

Image 3. PCA plot from Rignér, M *et al*. The image represents the expression of genes GATA3 and XBP1 of the analysed sample which were chosen as the PCA1 and PCA2 respectively sample (Rignér, M *et al.)*[3].

3.4.2 Heatmap

The heatmap is a representation of the data in a matrix where numbers are replaced by colours following a gradient according to the values.

Another interesting characteristic from the heatmap function is the dendrogram which can be plotted with the rows, the columns, or in both sides.

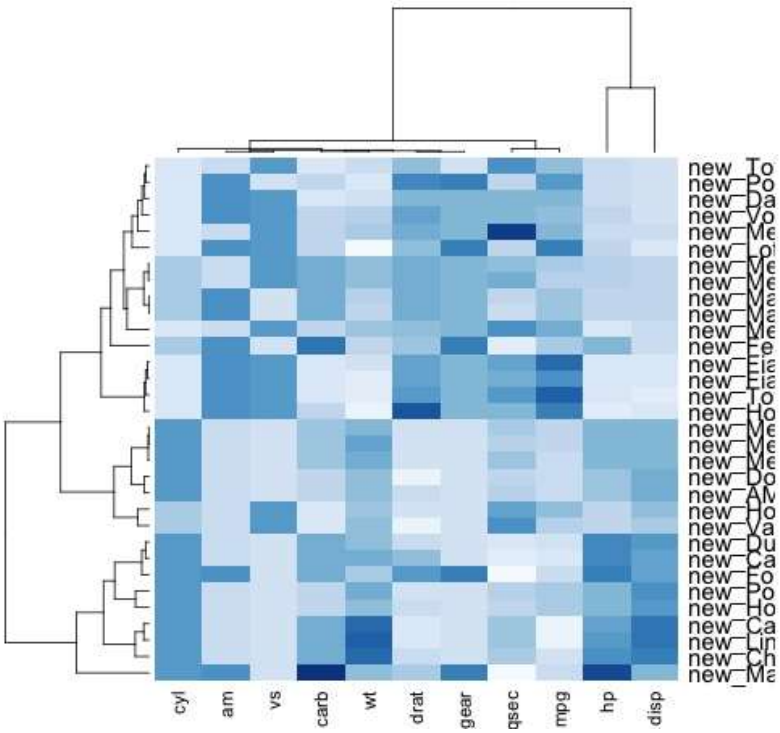

Image 4. Heatmap representing car models by rows and characteristics by columns. Dendrograms of columns and rows are both plotted (The heatmap function)[21].

4. Results and Discussion

The first step was to load all the downloaded data from the GEO (accession no. GSE73727) Li J. *et al*[11] into the program RStudio. Once all the files were loaded into the program, it was necessary to join them together to a unique file. For all this procedure was used the Script Load (attachments 7.1).

After creating a matrix with all the cells and transcripts, it is possible to start working with them. The first step was to check if all the cells were expressing transcripts or if any problem arose in the files downloaded. To checked it, matrices were check for null values.

As all the cells were sequenced properly, now it is time to see the expression of the cells. Huge mistakes are done when it is assumed that all the cells from the sample have same number of transcripts. When working with Single Cell data, differences in each cell are the key point of this analysis. If we plot a histogram of the expression transcripts, an enormous first column corresponding to the counts equal to 1 read of the matrix masks the rest of the data, as it is shown in the following graph:

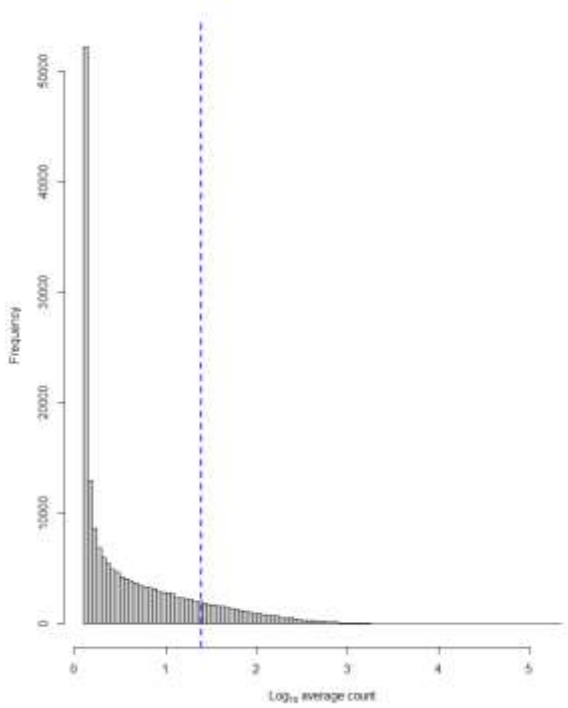

Image 6. Histogram of expressed transcripts from the cells. The blue line represents the threshold settled. The result obtained shows the counts equal to 1 read masking the ones with higher reads but in fewer number of transcripts.

To unmask those values, it is necessary to normalise the data, but before making a normalization, a quality control has to be done to know if all the cells are good examples for the analysis or some of them are not as good as thought.

The first quality control was made to prove which of them should be removed. Thus, a spike-ins control was done following the Script Workflow (attachments 7.2). After the check was done, 5 cells were removed from the initial sample of 72 leaving a final sample of 67 cells. Cells removed were number 23,46,47,49, and 60.

To normalise the data, quantile function was applied to the counts matrix. This function re-arranges the data to avoid outlines in the expression of each cell. After the normalisation, a histogram was plotted obtaining a result close to a gaussian distribution.

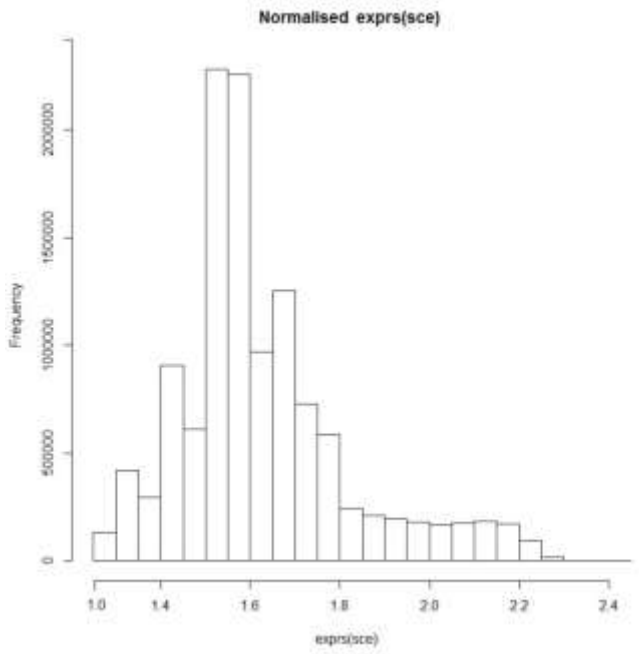

Image 7. Histogram of the normalized expression of the cells. The data has been modified applying the quantile function and then to better visualization the logarithm function was applied two times.

Once normalised, it is possible to see the expression of each cell plotting a box-plot:

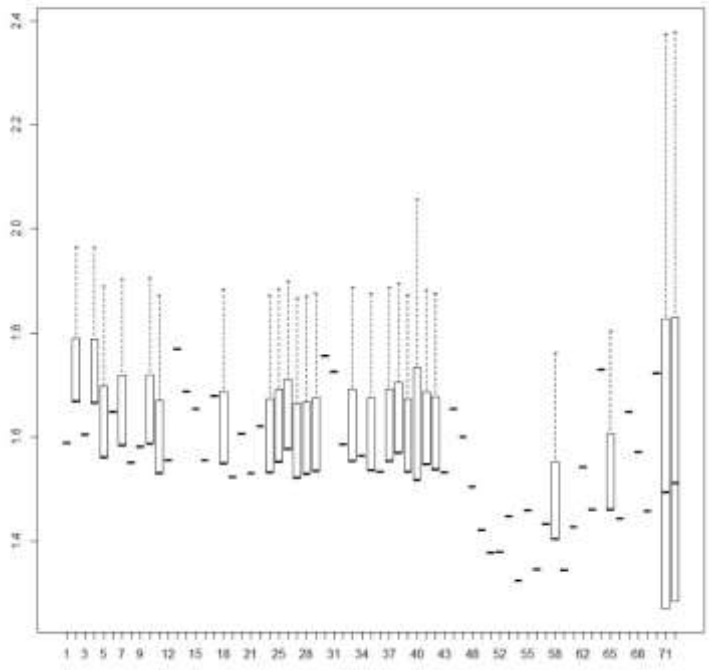

Image 8. Boxplot of the cells expression after the quality control check. Outliers were not plotted as they masked the plot.

To do further analysis, a conversion of the data from transcripts to genes is needed. Until this moment of the analysis, it was possible to work with matrices of transcripts of 180253 rows x 67 columns. For deeper comprehension in the analysis of the samples, these data have to be simplify as well as to reduce the time to process it. Right now, it requires a large time, and in order to have a better global view of the sample, a conversion from transcripts to genes was done. For this procedure was used the Script Transcripts to Genes (Attachment 7.3). After the conversion, the new dimension of the matrix is 35543 rows x 67 columns. This conversion helps to work with the data as the rows of the new data set have been reduced by a factor of 5 to the previous matrix data set.

Once the genes matrix has been obtained, the PCA is plotted using PCA 1 and PCA 2, with the following result:

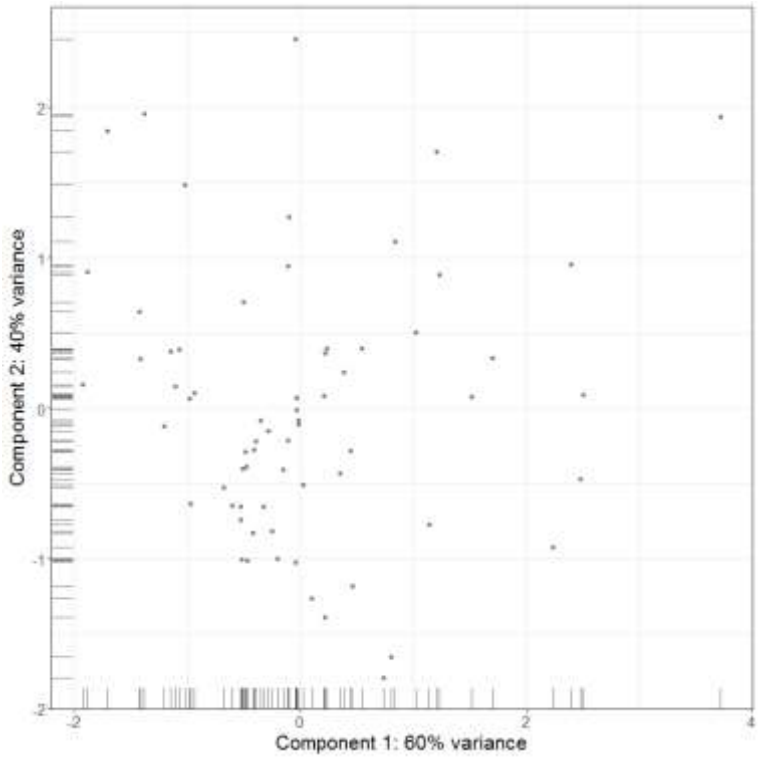

Image 9. PCA plot from the components 1 and 2 from the genes of the 67 cells. Data used to plot this graph were the genes which have expression on the cells.

The PCA shows a great variation in the sample data as expected. The variability observed verifies the heterogeneity of the sample as it has different cell types. To isolate the cells by its cellular type another analysis was performed, a heatmap.

A heatmap is a plot where the cells from the matrix are changed into a gradient of colours instead of showing the counts from the transcripts or the conversion of the genes. To plot this graph, it was used the Script Heatmap2 (attachment 7.4).

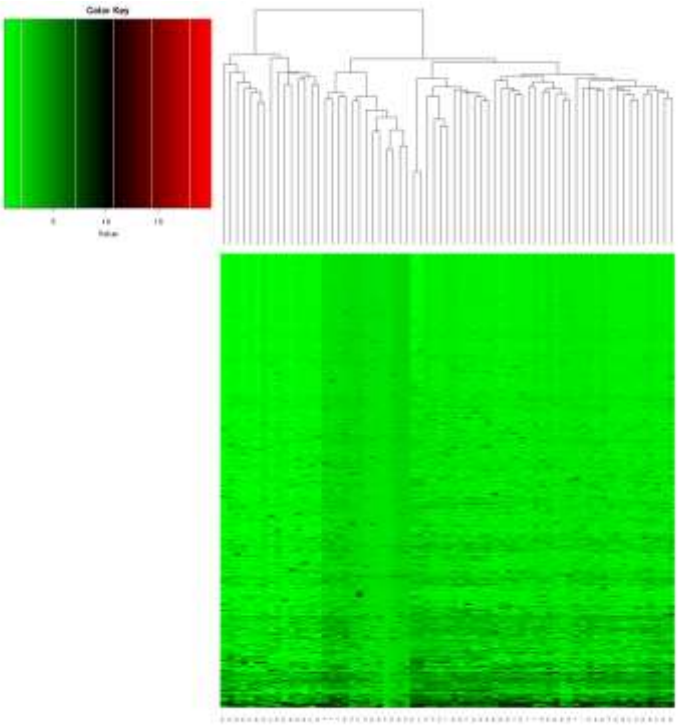

Image 10. Heatmap plot from the genes of the 67 cells. The gradient colour shows which genes are more expressed in each cell. The gradient colour goes from green to black and finally to red as more times the gene is expressed. This expression data frame was normalized by quantile function and for better visualization logarithm function was applied before plotting. Thus, the colour gradient is not showing a straight relation to the reads of the genes but logarithmic.

In this situation, the interesting function of the heatmap is not the coloured region, but the dendrogram plotted over the columns. This schematic hierarchy makes possible to classify the cells into the same cellular type due to the genes expressed. Making a horizontal cut along the dendrogram, it is possible to identify 6 different cellular types in the sample analysed.

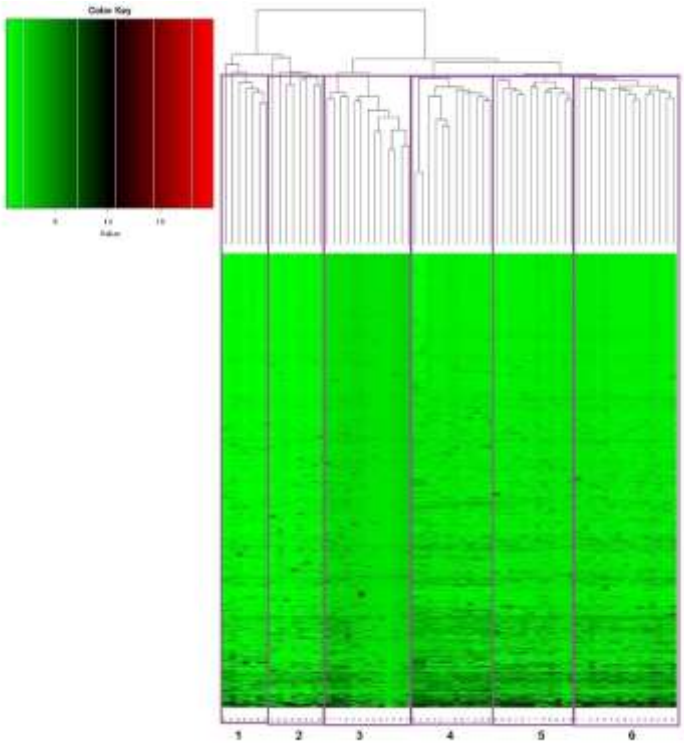

Image 11. Heatmap with the 6 different cell types which are present in the sample of the pancreatic islet.

According to the group of Li J. *et al.*[11], the cellular types identified corresponded to: alpha cells (expressing glucagon), beta cells (expressing insulin), delta cells (expressing somatostatin), pp cells (expressing pancreatic polypeptide), and pancreatic duct cells.

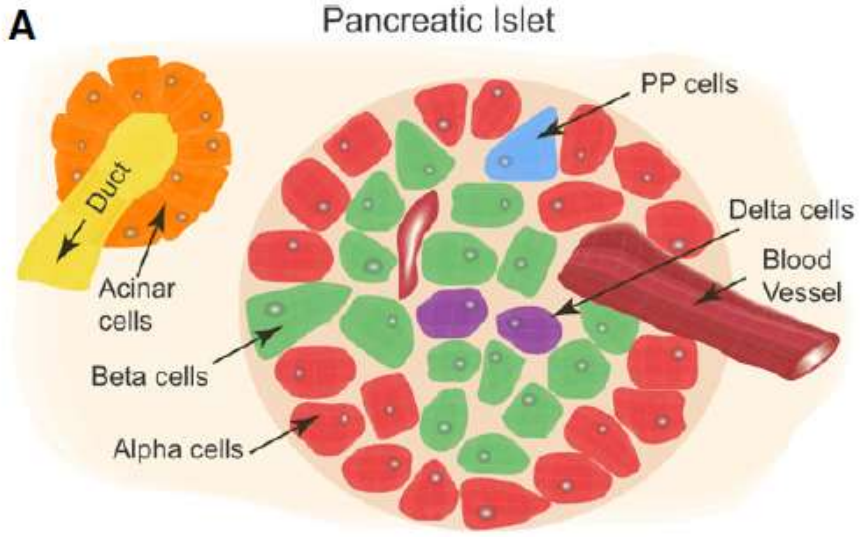

Image 12. Representation of a pancreatic islet. (Li J. *et al.*)[10].

5. Conclusions

This bachelor's thesis has focused on the great benefits that Bioinformatic technology helps on the identification of the cellular components of a given sample. Following a workflow established for Single Cell RNA-seq (A.T.L. Lun *et al*)[12], it was possible to identify 6 different cellular types in the 67 cells that passed the quality control check. The results obtained are similar to the ones published by Li J. *et al.*[11]. The important point of this technology is that enables us to detect in each cell the transcripts which are more expressed, and as a result of this to distinguish the different cellular types from a heterogenous sample.

Single Cell RNA-sequencing methodology is a powerful technique to classify with high precision the transcriptome of each cell of a determined sample. This type of analysis is focused to broaden the knowledge of some illnesses like cancer where single mutations have effects on the whole sample. This huge genetical variation as a result of the replication activity effects has given resistance to the illness. The aim of this technique is to find single mutations acquired by the cancer cells which they have obtained really fast due to their great replication activity and take advantage of it setting up a specialized treatment focused on those mutations. Moreover, this approach is quite interesting as the experimental methodology that rises the costs and the time needed for the identification of each cellular type decreases.

Nowadays Single Cell RNA-sequencing is under a stage of development where it is necessary to reduce the time between the analysis of the sample and the determination of the treatment for the patient. According to a study carried out by Ellsworth, D. L. *et al*[14], the time needed it is about 55 days with a group integrated by specialists of each area (Bioinformatics, Medics, Pharmaceutics, etc). In order to improve the benefits from this technique is essential to reduce this time, thus it will be shown in a better response to treatments on the patients.

6. Bibliography

1.
Caulfield, T., Gold, E. R. & Cho, M. K. Patenting human genetic material: refocusing the debate. *Nature Reviews Genetics* 1, 227–231 (2000).

2.
Herzenberg, L. A. *et al.* The history and future of the fluorescence activated cell sorter and flow cytometry: a view from Stanford. *Clin. Chem.* 48, 1819–1827 (2002).

3.
Ringnér, M. What is principal component analysis? *Nature Biotechnology* 26, 303–304 (2008).

4.
Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* 6, 377–382 (2009).

5.
Navin, N. *et al.* Tumour evolution inferred by single-cell sequencing. *Nature* 472, 90–94 (2011).

6.
Ramsköld, D. *et al.* Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nature Biotechnology* 30, 777–782 (2012).

7.
Patel, A. P. *et al.* Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* 344, 1396–1401 (2014).

8.
Picelli, S. *et al.* Full-length RNA-seq from single cells using Smart-seq2. *Nature Protocols* 9, 171–181 (2014).

9.
Akrap, N. *et al.* Identification of Distinct Breast Cancer Stem Cell Populations Based on Single-Cell Analyses of Functionally Enriched Stem and Progenitor Pools. *Stem Cell Reports* 6, 121–136 (2016).

10.
Bacher, R. & Kendziorski, C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology* 17, (2016).

11.
Li, J. *et al.* Single-cell transcriptomes reveal characteristic features of human pancreatic islet cell types. *EMBO reports* 17, 178–187 (2016).

12.
Lun, A. T. L., McCarthy, D. J. & Marioni, J. C. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research* 5, 2122 (2016).

13.
Yu, P. & Lin, W. Single-cell Transcriptome Study as Big Data. *Genomics, Proteomics & Bioinformatics* 14, 21–30 (2016).

14.
Ellsworth, D. L. *et al.* Single-cell sequencing and tumorigenesis: improved understanding of tumor evolution and metastasis. *Clinical and Translational Medicine* 6, (2017).

15.
Knouse, K. A., Wu, J. & Hendricks, A. Detection of Copy Number Alterations Using Single Cell Sequencing. *Journal of Visualized Experiments* (2017). doi:10.3791/55143

16.
Tung, P.-Y. *et al.* Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports* 7, 39921 (2017).

17.
Wang, J. & Song, Y. Single cell sequencing: a distinct new field. *Clinical and Translational Medicine* 6, (2017).

18.
Bioconductor. *Repository* Available at: http://bioconductor.org/. (Accessed: 1st May 2017)

19.
R-project. Available at: https://www.r-project.org/about.html. (Accessed: 1st May 2017)

20.
RStudio. *RStudio* Available at: https://www.rstudio.com/. (Accessed: 1st May 2017)

21.
The heatmap function. *R-graph-gallery* Available at: http://www.r-graph-gallery.com/215-the-heatmap-function/. (Accessed: 3rd July 2017)

22.
What is Bioinformatics? *Bioinformatics Department from Yale University* Available at: http://bioinfo.mbb.yale.edu/what-is-it/. (Accessed: 7th July 2017)

7. Attachments

7.1 Script Load

```
a <- read.table("D:/Diego/Universidad/Grado/2016_2017_Cuarto/Cuatri
B/TFG/Papers/Austria (pancreatic)/GSM1901455_sample_1_counts.tsv",
header=T, sep="\t", stringsAsFactors=F)

fichs <-
list.files("D:/Diego/Universidad/Grado/2016_2017_Cuarto/Cuatri
B/TFG/Papers/Austria (pancreatic)/",pattern =
"counts.tsv$",full.names = T)

exp <- mat.or.vec(nc=length(fichs),nr=nrow(a))
rownames(exp) <- a$ensT
colnames(exp) <- b(^"GSM", $"counts.tsv")

for(i in 1:length(fichs)){
  cat("leyendo muestra",i,"\n")
  sample_exp <- read.table(fichs[i], header=T, sep="\t",
stringsAsFactors=F)
  exp[sample_exp$ensT,i] <- sample_exp$count

}

hist((exp),30)
hist(log(exp),30)

## Heat diagram, requiere limpieza, imposible de vis.
heatdiagram(stat = exp, coef = exp, names = colnames(exp),orientation
= "portrait",low = "blue",high = "red")

##Normalizamos el archivo sce por medio del uso de log y se plotea un
hist y un box plot.
hist(log(log(exprs(sce)+1)+1)+1)
boxplot(log(log(exprs(sce)+1)+1)+1)
##El boxplot tiene demasiados puntos fuera, se debe volver a
normalizar, esto enmascara expresion celulas en baja proporcion

##Normalizacion quantiles
exprssce_logX2<- log(log(exprs(sce)+1)+1)+1

bar <- normalize.quantiles(foo)
plot(density(bar[,1]), xlab='reads', main='after')
for(i in 2:100) lines(density(bar[,i]))
```

7.2 Script Workflow

```
##https://www.bioconductor.org/help/workflows/simpleSingleCell/

##
##library(R.utils)
##gunzip("GSE61533_HTSEQ_count_results.xls.gz", remove=FALSE,
overwrite=TRUE)
##library(readxl)
##all.counts <-
as.data.frame(read_excel('GSE61533_HTSEQ_count_results.xls',
sheet=1))
##rownames(all.counts) <- all.counts$ID
##all.counts <- all.counts[,-1]

##Empezamos aqui
library(scater)
sce<- newSCESet(countData=exp)
dim(sce)

##
is.spike <- grepl("^ERCC", rownames(sce))

##
sce <- calculateQCMetrics(sce, feature_controls=list(ERCC=is.spike))
head(colnames(pData(sce)))

##
library(BiocParallel)
library(scran)
setSpike(sce)<- "ERCC"

##
par(mfrow=c(1,2))
hist(sce$total_counts/1e6, xlab="Library sizes (millions)", main="",
     breaks=20, col="grey80", ylab="Number of cells")
hist(sce$total_features, xlab="Number of expressed transcripts",
main="",
     breaks=20, col="grey80", ylab="Number of cells")

##No hay features asue...
libsize.drop <- isOutlier(sce$total_counts, nmads=3, type="lower",
log=TRUE)
feature.drop <- isOutlier(sce$total_features, nmads=3, type="lower",
log=TRUE)

##
par(mfrow=c(1,2))
hist(sce$pct_counts_feature_controls_ERCC, xlab="ERCC proportion
(%)",
     ylab="Number of cells", breaks=20, main="", col="grey80")

##
spike.drop <- isOutlier(sce$pct_counts_feature_controls_ERCC,
nmads=3, type="higher")


##
sce <- sce[,!(libsize.drop | feature.drop | spike.drop)]
data.frame(ByLibSize=sum(libsize.drop), ByFeature=sum(feature.drop),
BySpike=sum(spike.drop), Remaining=ncol(sce))
```

```
##
fontsize <- theme(axis.text=element_text(size=12),
axis.title=element_text(size=16))
plotPCA(sce, pca_data_input="pdata") + fontsize

##Saltamos el apartado que hace referencia al ciclo de fase celular.

##Ahora procede a eliminar genes(en nuestro caso transcr) de baja
expresion
ave.counts <- calcAverage(sce)
##En el ejemplo ponen el treshold en 1, pero en nuestro artlo dicen
que lo pusieron en 25.
keep <- ave.counts >= 25
sum(keep)

##Ahora comprueba que el treshold puesto sea adecuado.
hist(log10(ave.counts), breaks=100, main="", col="grey80",
     xlab=expression(Log[10]~"average count"))
##en log10(1) cambiar por el 25 que hemos seleccionado anteriormente
abline(v=log10(25), col="blue", lwd=2, lty=2)

##
plotQC(sce, type = "highest-expression", n=50) + fontsize

##En este paso restringe aquellas cel. que tengan transc que se
expresen en al menos 10 cel.
##En nuestro caso al tratarse de un SC-RNAseq vamos a tener
transcritos siempre.
##Es inutil este paso.
numcells <- nexprs(sce, byrow=TRUE)
alt.keep <- numcells >= 10
sum(alt.keep)

##Por lo que hacer el smoothscatter tmb es intil.
smoothScatter(log10(ave.counts), numcells,
xlab=expression(Log[10]~"average count"),
              ylab="Number of expressing cells")
is.ercc <- isSpike(sce, type="ERCC")
points(log10(ave.counts[is.ercc]), numcells[is.ercc], col="red",
pch=16, cex=0.5)

sce <- sce[keep,]

##Normalizacion sesgos especc celulares
library(DESeq2)

##Estima los factores
estimateSizeFactorsForMatrix()

sce <- computeSumFactors(sce, sizes=seq(20, 80, 5))
summary(sizeFactors(sce))
```

7.3 Script Transcripts to Genes

```
library(limma)

dat <- as.data.frame(exprs(sce))
class(dat)
dat[,"ids"] <- rownames(dat)
dim(sce) #180253 x 67
head(dat)
colnames(dat)

head(mart_export_CSV_)
equiv <- as.data.frame(mart_export_CSV_)
class(equiv[,1])
head(equiv)
dim(equiv) #218207 x 2
colnames(equiv) <- c("geneID", "transcriptID")

ind1 <- equiv[, 2] %in% rownames(sce)
table(ind1)

total <- merge(equiv, dat, by.y= "ids", by.x = "transcriptID", all.y
= T)
head(equiv)
head(dat)
head(total)
dim(total)
table(duplicated(total$geneID))

ind2 <- total$geneID == "ENSG00000004059"
table(ind2)
total[ind2==T,]

ind3 <- is.na(total$geneID)
table(ind3)
total2 <- total[!ind3,]
dim(total2)
colnames(total2)
mat <- as.matrix(total2[, 3:69])
rownames(mat) <- total2$geneID
dim(mat)
mat[1:5,1:5]
mat.final <- avereps(mat)
dim(mat.final)
head(mat.final)
boxplot(mat.final)
summary(mat.final)
save.image("prep.heatmap.RData")
```

7.4 Script Heatmap2

```
library(gplots)
#cc <- c(rep("blue",15), rep("cyan",11)) #assigning colors to arrays

png (filename = "s2_3_669.png",  width = 480 ,   height = 2*480,
pointsize = 30)
h6 <- heatmap.2(datos2,
          #col=topo.colors(150),
          #col=cm.colors(255), #here I choose the two colors
          #col= rainbow(4, start = 0.3, end = 0.1),
          col= greenred(75),
          #col= terrain.colors(4),
          #topo.colors(4),
          #cm.colors(4),
          #col=topo.colors(150), #If I have the topographical colours
installed (yellow-green-blue), you can do this
          #col="heat.colors",
          scale="none", #our data is already normalized, so we do not
apply
          key=TRUE,   #including color key
          keysize = 2,
          symkey=FALSE,
          density.info="none", #sometimes is interesting to
          #main= "heatmap3", xlab="samples", ylab="genes",
          cexRow=0.5, #to reduce the size of names for rows
          cexCol=0.4,
          labRow = "",
          trace="none")
dev.off ()
```