

# Treball BIO

*author*

*March 23, 2018*

## Contents

<b>Introducción</b>	<b>1</b>
<b>Objetivos</b>	<b>2</b>
<b>Materiales y métodos</b>	<b>2</b>
Naturaleza de los datos . . . . .	2
Diseño experimental . . . . .	2
Métodos utilizados en el análisis . . . . .	3
<b>Análisis de datos con R</b>	<b>4</b>
1. Directorios y opciones de trabajo . . . . .	4
2. Lectura de Datos . . . . .	4
3. Exploración, Control de Calidad y Normalización. . . . .	5
4. Selección de genes diferencialmente expresados. . . . .	10
5. Comparaciones múltiples. . . . .	13
6. Anotación de resultados. . . . .	14
7. Visualización de los perfiles de expresión . . . . .	15
<b>Bibliografía.</b>	<b>15</b>

## Introducción

Este documento pretende explicar el estudio sobre la expansión in vitro de **células beta** ( $\beta$ ): mostrar qué mueve a realizar este estudio, cuál es la hipótesis planteada en un principio, qué métodos biotecnológicos y estadísticos se han utilizado y cuáles han sido los resultados finales.

### Información previa.

El **páncreas** es un órgano que, entre muchas otras cosas, está formado por islotes de Langerhans. Los islotes pancreáticos son cúmulos de células, la mayoría de las cuales son células beta que producen la hormona de la **insulina**, secretada para regular la glucemia en sangre. La enfermedad de la **diabetes** se caracteriza por la destrucción de células beta y, por lo tanto, en un defecto en la regulación de la glucemia que causa insulino-dependencia. Las donaciones de páncreas son escasas por lo que el trasplante de islotes de células beta se ve limitado. De aquí nace el deseo de estudiar cómo expandir las células beta obtenidas a partir de islotes de páncreas cadavéricos. No obstante, las células beta tienen una capacidad de proliferación baja in vivo y los intentos de inducir la expansión de las células beta cultivadas in vitro siempre han dado como resultado la **desdiferenciación celular**: pérdida del fenotipo de las células beta y consecuentemente, su funcionalidad biológica (producir insulina).

## Objetivos

Los investigadores han demostrado mediante el rastreo de **linaje celular**<sup>1</sup> la existencia de células beta-derivadas supervivientes de la dediferenciación y la replicación de estas. También, **análisis epigenéticos**<sup>2</sup> de las células beta-derivadas han indicado que los genes se han mantenido a pesar de no haberse transcrito.

Se hipotetiza, pues, que la célula dediferenciada ha mantenido su epigenoma y que, por lo tanto, es el epigenoma el responsable de que la célula pueda seguir produciendo insulina. Esto significa que, si se consigue rediferenciar las células beta-derivadas, es decir, si se logra que vuelvan a producir insulina (recuperación del fenotipo) la expansión de células beta-derivadas podría generar suficientes células como para reponer las células beta humanas.

## Materiales y métodos

### Naturaleza de los datos

Los islotes de los donantes se disociaron en células individuales, y las células beta se marcaron, se expandieron y cultivaron. Algunas células se mantuvieron en un medio sin suero (**SFM**) mientras que otras fueron tratadas con un cóctel de rediferenciación (**RC**).

Además, para el caso de las células no tratadas, se aprovechó la capacidad de los virus para introducirse en células e incorporar su propio material genético al genoma del organismo infectado. Se utilizó un lentivirus (virus de infección lenta) al que se le introdujo un **plásmido**<sup>3</sup> con un gen particular expresamente para inhibir el gen SLUG de las células beta. Este gen codifica una proteína que reprime la transcripción de ciertas secuencias que hacen que la célula se dediferencie.

De esta manera, a partir de las células expandidas de los islotes donados, se obtuvieron células re-diferenciadas (RC) y células no diferenciadas (medio SFM infectadas con el virus).

### Diseño experimental

**Objetivo:** determinar si las células de islotes expandidas y re-diferenciadas consiguen más células beta que segreguen insulina.

**Variable respuesta:** intensidad del color para medir la expresión génica.

**Factor:** medio en el que se encuentran las células.

**Individuo objeto de estudio:** las células beta obtenidas a partir de islotes de páncreas cadavéricos.

En el apartado anterior se han mencionado dos condiciones que se aplicaron a las células cultivadas, pero se experimentó también con células que no fueron cultivadas ni expandidas, o sea que fueron estudiadas tan pronto como fueron obtenidas.

Así pues, la expresión génica fue estudiada en estos 3 grupos:

- Células de islotes no expandidas (4 donantes).
- Células de islotes expandidas y no diferenciadas (4 donantes).
- Células de islotes expandidas y re-diferenciadas (3 donantes).

Lo que significa que se tuvo un diseño ANOVA de 1 factor a 3 niveles y se utilizó un total de 11 microarrays (1 para cada donante).

---

<sup>1</sup>Técnica que permite conocer la ascendencia de una célula.

<sup>2</sup>Estudio de los factores que intervienen en la expresión génica (modifican la actividad del ADN) sin alterar su secuencia. Estos factores no se corresponden a los elementos clásicos de la genética.

<sup>3</sup>molécula de ADN que se replica y transmite independientemente del ADN cromosómico.

## Métodos utilizados en el análisis

### Análisis qPCR

El análisis qPCR es un método que permite la amplificación específica del ADN y la cuantificación de la expresión génica de una muestra en tiempo real. Este análisis se realiza con un **termociclador**<sup>4</sup> y un **espectrofluorómetro**<sup>5</sup>, este último se encarga de definir la fluorescencia producida en el tubo de amplificación en todo momento, con el fin de combinar la amplificación del DNA con su detección. La presencia del fluoróforo es persistente en la reacción y la señal obtenida al final de cada ciclo de amplificación es representada en una gráfica frente al número de ciclos, por lo tanto al final se obtiene una curva que representa el intervalo del proceso.

Se suele basar en la comparación entre un tratamiento y un control a partir de tests que asumen normalidad. Para cuantificar los datos, se suele relacionar la señal obtenida con el contenido de ADN, para ello se utiliza la curva de calibrado, esto se hace cuando todos tienen una misma eficiencia de amplificación. Otra manera de cuantificar los datos es expresar el cambio en los niveles de expresión de ARNm.

Los valores de fluorescencia se expresan como logaritmos para así estudiar la fase exponencial de amplificación.

### Análisis de inmunofluorescencia

La inmunofluorescencia es una técnica común de laboratorio basada en el uso de unos anticuerpos específicos los cuales están químicamente combinados con colorantes fluorescentes. Se usa para evaluar células en suspensión, cultivadas, en tejidos y en microarrays. Existen dos maneras de realizar el análisis:

La inmunofluorescencia **directa** consiste en ligar un fluoróforo (componente que hace que una molécula sea fluorescente) al anticuerpo de la molécula que buscamos. De esta manera, ambos componentes se enlazarán a la molécula, que podrá ser detectada con microscopio. Se trata de un método rápido aunque el hecho de que el anticuerpo solo encuentre un número limitado de moléculas, lo hace poco sensible.

Por otro lado, la **indirecta**, se basa en la misma idea que la directa pero consta de dos anticuerpos: el anticuerpo primario, que se une a la molécula objetivo, y el secundario, unido al fluoróforo. Es un proceso más complejo y lento, pero más flexible y preciso gracias al número de distintos anticuerpos secundarios y las técnicas de detección que se pueden usar en el anticuerpo primario.

### Tipos de microarrays utilizados

Los microarrays utilizados fueron *Affymetrix GeneChip Human Gene 1.0 STE Arrays*, microarrays de expresión de oligonucleótidos sintetizados in situ (1 color). Este tipo de microarrays tiene las siguientes características:

- Cada chip contiene muestras de un solo tipo.
- Las sondas se sintetizan directamente sobre el chip.
- Cada gen está representado por un grupo de sondas cortas .

El análisis de microarrays se realizó en archivos CEL utilizando PartekH Genomics Suite Tm, un software mundialmente utilizado para la visualización y el análisis de datos genómicos.

Cabe destacar que:

- Los datos se normalizaron con el método multiaverage.
- La eliminación del efecto Batch fue aplicada para las diferentes muestras.
- El Clustering, se hizo con Partek Genomics Suite software con Correlación de disimilitud de Pearson.

---

<sup>4</sup>Aparato usado en biología molecular que permite realizar los ciclos de temperaturas necesarios para una reacción de amplificación de ADN.

<sup>5</sup>Dispositivo de laboratorio utilizado para medir los parámetros de la fluorescencia: su intensidad y la distribución de longitudes de onda del espectro de emisión después de la excitación por un cierto espectro de luz.

# Análisis de datos con R

Se analizará una muestra de 11 células de personas. Estas 11 muestras se han distribuido en 3 grupos .

## 1. Directorios y opciones de trabajo

```
workingDir <- "C:/Users/Gemma/Documents/Analisis"
dataDir <- file.path(workingDir, "DADES2")
resultsDir <- file.path(workingDir, "resultados")
setwd(workingDir)
```

### Paquetes

```
options(width=80)
options(digits=5)
source("http://bioconductor.org/biocLite.R")
if (!(require(mouse4302.db))){
  biocLite("mouse4302.db")
}
if (!(require(annaffy))){
  biocLite("annaffy")
}
if (!(require(affy))){
  biocLite("affy")
}
if (!(require(gplots))){
  biocLite("gplots")
}
if (!(require(limma))){
  biocLite("limma")
}
if (!(require(GOstats))){
  biocLite("GOstats")
}
}
if(!(require(org.Mm.eg.db))){
  biocLite("org.Mm.eg.db")
}
if(!(require(Biobase))){
  biocLite("Biobase")
}
if(!(require(genefilter))){
  biocLite("genefilter")
}
```

## 2. Lectura de Datos

```
sampleInfo <- read.AnnotatedDataFrame(file.path(dataDir, "targets.txt"),
  header = TRUE, row.names = 1, sep="\t")
```

```
## Warning in readLines(filename): incomplete final line found on 'C:/Users/Gemma/
## Documents/Analisis/DADES2/targets.txt'
```

```
fileNames <- rownames(pData(sampleInfo))
rawData <- read.affybatch(filenamees=file.path(dataDir,fileNames),
                        phenoData=sampleInfo)
```

```
## Warning:
##
## The affy package can process data from the Gene ST 1.x series of arrays,
## but you should consider using either the oligo or xps packages, which are specifically
## designed for these arrays.
```

### 3. Exploración, Control de Calidad y Normalización.

#### 3.1. Exploración de los datos.

- 3.1.1. Gráfico de degradación.

```
info <- data.frame(grupo=pData(rawData)$grupo)
sampleNames <- unlist(lapply(strsplit(rownames(pData(rawData))),"\\."),
                      function(x)x[1]))
hist(rawData, main="Signal distribution", col=info$grupo, lty=1:nrow(info))
```

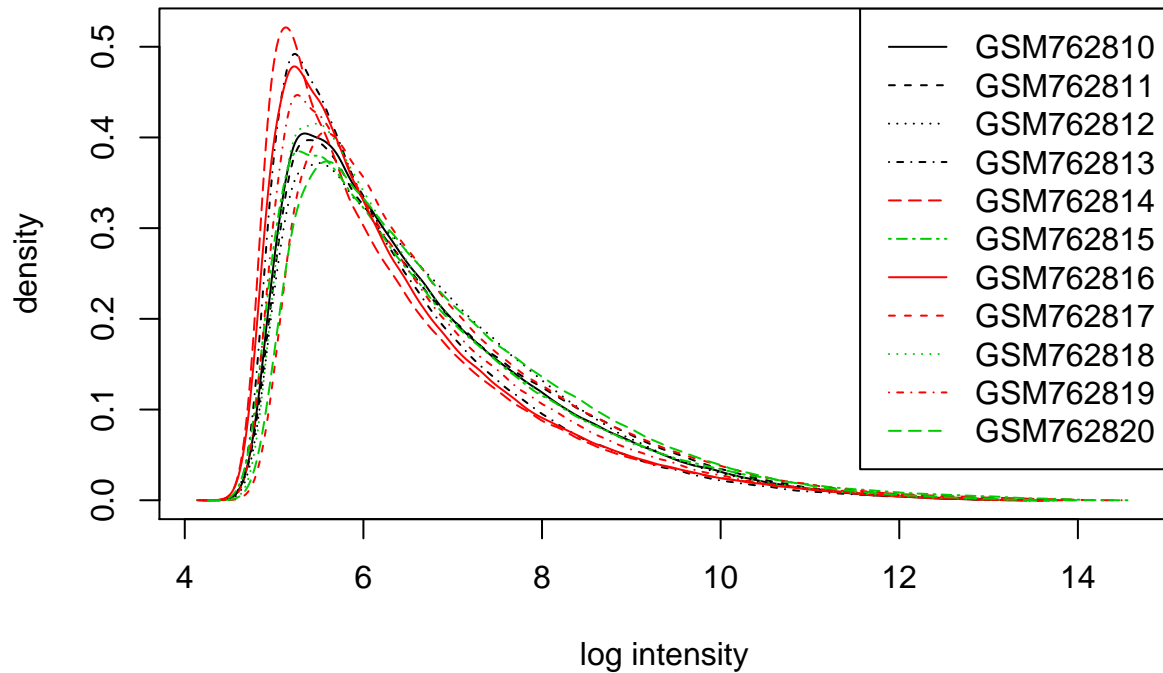
```
## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when
## loading 'hugene10stv1cdf'
```

```
## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'hugene10stv1cdf'
```

```
##
```

```
legend (x="topright", legend=sampleNames , col=info$grupo, lty=1:nrow(info)) #No queda be el legend
```

## Signal distribution



Este gráfico muestra cómo ha sido el proceso de hibridación de las muestras. Se puede observar que todas las líneas son paralelas, lo que significa que tienen una calidad parecida.

- 3.1.2. Cluster jerárquico y dendograma.

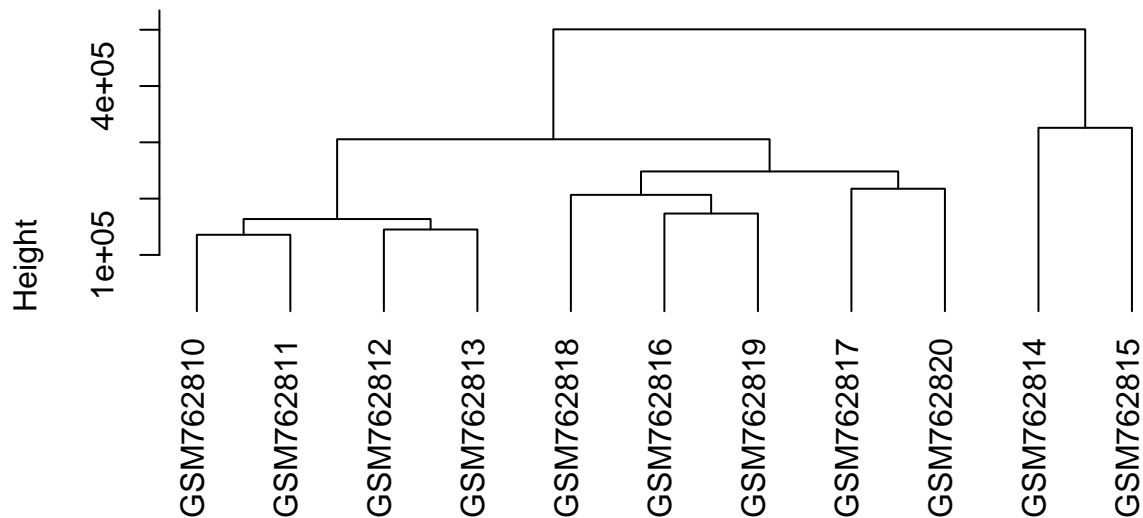
Se realiza para ver si las muestras se agrupan por condiciones experimentales.

```
deg<-AffyRNAdeg(rawData, log.it=T)
summaryAffyRNAdeg(deg)
```

```
##          GSM762810.CEL GSM762811.CEL GSM762812.CEL GSM762813.CEL GSM762814.CEL
## slope          0.0248          0.0244          0.0198          0.0277          0.043
## pvalue          0.6660          0.6890          0.7090          0.6030          0.374
##          GSM762815.CEL GSM762816.CEL GSM762817.CEL GSM762818.CEL GSM762819.CEL
## slope          0.0499          0.0491          0.0467          0.0466          0.0379
## pvalue          0.3150          0.3130          0.4170          0.4180          0.4970
##          GSM762820.CEL
## slope           0.052
## pvalue           0.395
```

```
clust.euclid.average <- hclust(dist(t(exprs(rawData))),method="average")
plot(clust.euclid.average, labels=sampleNames, main="Hierarchical clustering of samples", hang=-1)
```

## Hierarchical clustering of samples



```
dist(t(exprs(rawData)))
hclust (*, "average")
```

Se observa cómo las muestras se han agrupado, según las siguientes condiciones: no cultivadas (GSM762810 - GSM762813), cultivadas tratadas (GSM762818 - GSM762820) y cultivadas no tratadas (GSM762814- GSM762815).

### 3.2. Control de calidad.

- Ajuste PLM.

EL paquete `affyPLM` realiza un control de calidad basado en “probe-level models” (PLM).

```
stopifnot(require(affyPLM))
```

```
## Loading required package: affyPLM
```

```
## Loading required package: gcrma
```

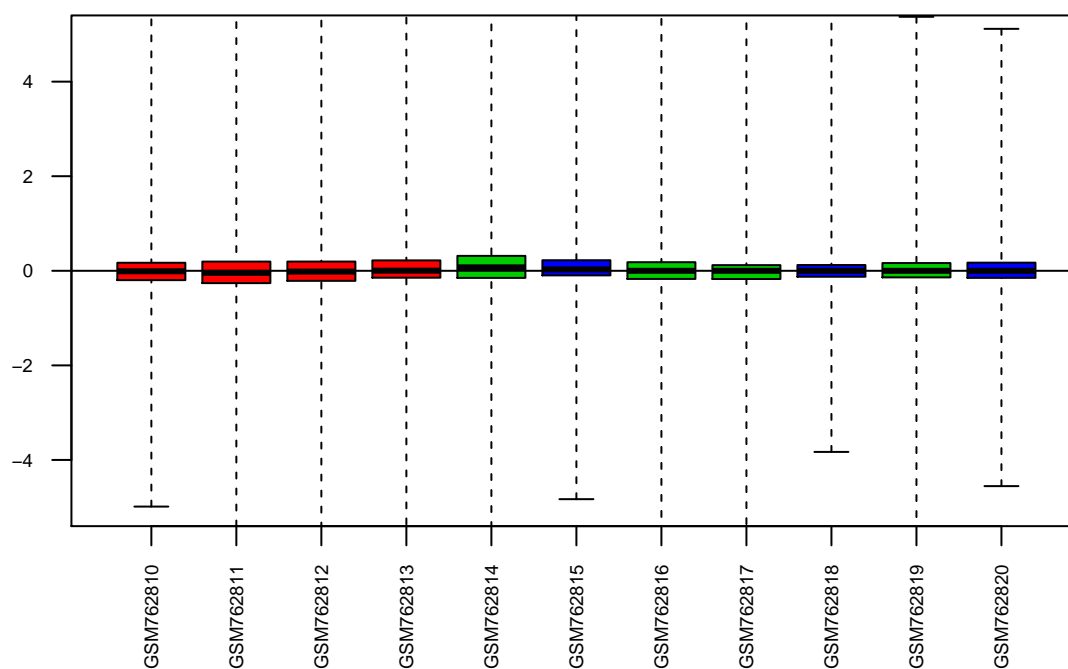
```
## Loading required package: preprocessCore
```

```
Pset <- fitPLM(rawData)
```

- Gráfico de expresiones relativas.

```
RLE(Pset, main = "Relative Log Expression", names = sampleNames, las = 2, col = info$grupo + 1, cex.axis = 0.8)
```

## Relative Log Expression

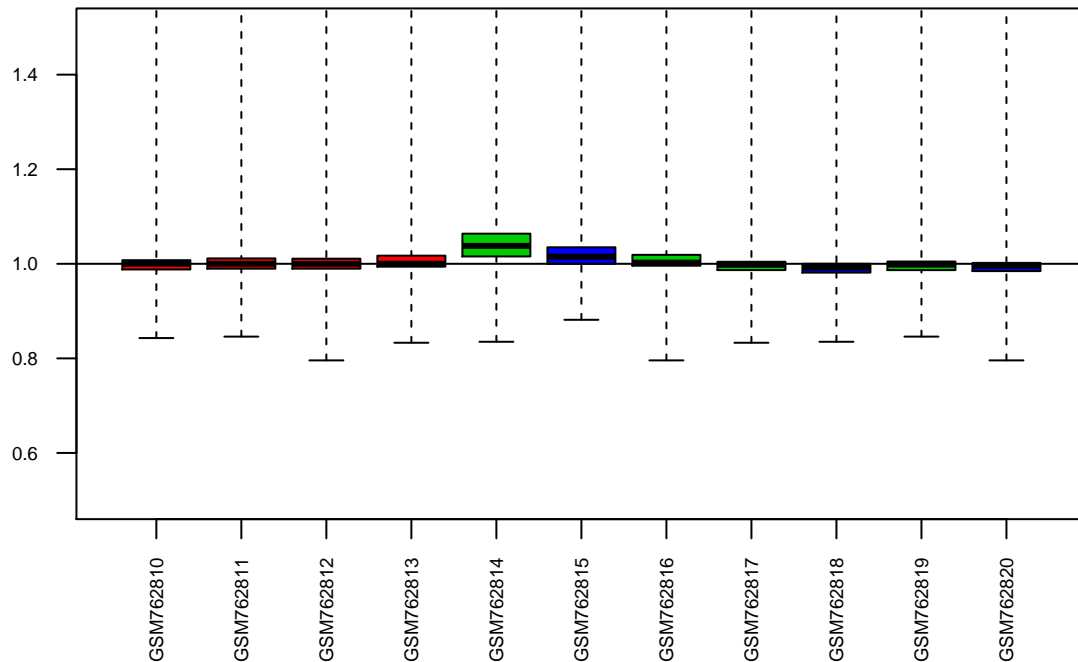


- Gráfico con los errores estandarizados.

```
NUSE(Pset, main = "Normalized Unscaled Standard Errors", las = 2, names = sampleNames, las = 2, col = i
```



## Normalized Unscaled Standard Errors



En ambos gráficos los datos están centrados y son relativamente simétricos. Esto sugiere que no se han dado problemas en los microarrays por lo que los datos son de calidad.

### 3.3. Normalización y filtraje.

Se procede a realizar un filtraje no específico (`nsFilter`) que permite eliminar aquellos genes que no consituyen ninguna aportación a la selección de genes diferencialmente expresados, ya sea porque tienen señales muy bajas o porque varían muy poco entre las diferentes condiciones.

El procesado mediante RMA se realiza en tres etapas:

- 3.3.1. Corrección de fondo.

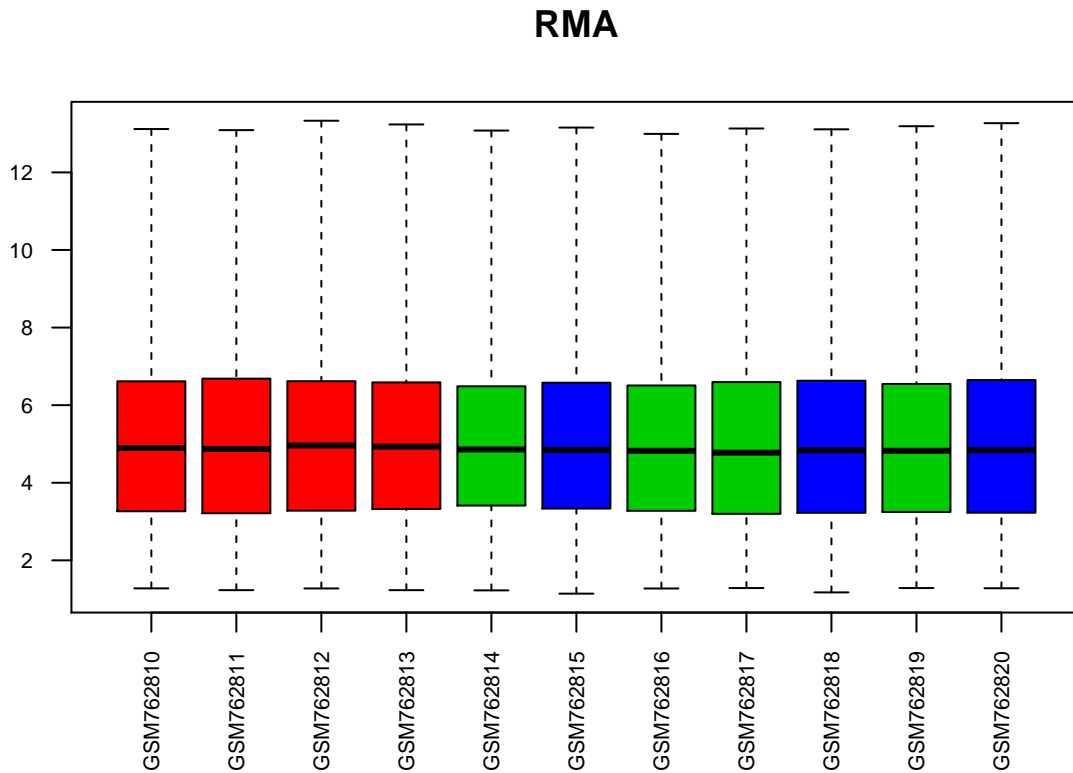
```
stopifnot(require(affy))
normalize <- T
if (normalize) {
  eset_rma <- rma(rawData)
  save(eset_rma, file = file.path(dataDir, "normalized.Rda"))
} else {
  load(file = file.path(dataDir, "normalized.Rda"))
}
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

- 3.3.2 Normalización.

La normalización es necesaria para que se puedan comparar los valores de los arrays entre ellos.

```
boxplot(eset_rma, main = "RMA", names = sampleNames, cex.axis = 0.7, col = info$grupo + 1, las = 2)
```



El boxplot de los valores normalizados nos muestra que los valores de los arrays ya están en una escala en la que se pueden comparar (obsérvese cómo los datos están centrados).

- 3.3.3 Filtrado.

```
# stopifnot(require(genefilter))
# stopifnot(require(mouse4302.db))
# filtered <- nsFilter(eset_rma, require.entrez = TRUE, remove.dupEntrez = TRUE, var.func = IQR, var.cu
```

Informe de los resultados del filtrado:

```
# names(filtered)
# class(filtered$eset)
# print(filtered$filter.log)
# eset_filtered <- filtered$eset
```

## 4. Selección de genes diferencialmente expresados.

- 4.1. Matriz de diseño.

Como ya se ha mencionado con anterioridad, el diseño se ha modelizado como un modelo de un factor con tres niveles:

Nombre Archivo	Condiciones
GSM762810.CEL	No cultivada
GSM762811.CEL	No cultivada
GSM762812.CEL	No cultivada
GSM762813.CEL	No cultivada
GSM762814.CEL	Cultivada, No tratada
GSM762815.CEL	Cultivada, Tratada RC
GSM762816.CEL	Cultivada, No tratada
GSM762817.CEL	Cultivada, No tratada
GSM762818.CEL	Cultivada, Tratada RC
GSM762819.CEL	Cultivada, No tratada
GSM762820.CEL	Cultivada, Tratada RC

La matriz de diseño será:

```
require(Biobase)
if (!exists("eset_rma")) load(file.path(dataDir, "normalized.rda"))
targets <- pData(eset_rma)
stopifnot(require(limma))

treat <- paste(targets$islet, targets$treat, sep = ".")
lev <- factor(treat, levels = unique(treat))
design <- model.matrix(~0 + lev)
colnames(design) <- levels(lev)
rownames(design) <- rownames(targets)
print(design)
```

```
##          UnculturedIslet.untreated expandedislet.untreated
## GSM762810.CEL                1                0
## GSM762811.CEL                1                0
## GSM762812.CEL                1                0
## GSM762813.CEL                1                0
## GSM762814.CEL                0                1
## GSM762815.CEL                0                0
## GSM762816.CEL                0                1
## GSM762817.CEL                0                1
## GSM762818.CEL                0                0
## GSM762819.CEL                0                1
## GSM762820.CEL                0                0
##          expandedislet.RCtreatment
## GSM762810.CEL                0
## GSM762811.CEL                0
## GSM762812.CEL                0
## GSM762813.CEL                0
## GSM762814.CEL                0
## GSM762815.CEL                1
## GSM762816.CEL                0
## GSM762817.CEL                0
## GSM762818.CEL                1
## GSM762819.CEL                0
## GSM762820.CEL                1
## attr(,"assign")
## [1] 1 1 1
```

```
## attr("contrasts")
## attr("contrasts")$lev
## [1] "contr.treatment"
```

- 4.2. Contrastes.

Con el modelo lineal defino a través de la matriz de diseño formulamos las preguntas de interés como contrastes, es decir, como comparaciones entre los parámetros del modelo. Como se tiene un factor a tres niveles, debemos estudiar tres comparaciones dos a dos.

En este caso interesa estudiar:

- Efecto de la expansión.
- Efecto de la re-diferenciación (en células expandidas).
- Efecto de la re-diferenciación i expansión.

```
require(limma)
cont.matrix <- makeContrasts(
  Expansion.in.Untreated = (UnculturedIslet.untreated - expandedislet.untreated),
  Treated.in.Expansion = (expandedislet.untreated - expandedislet.RCtreatment),
  TreatedAndExpansion = (UnculturedIslet.untreated - expandedislet.RCtreatment), levels = design)
cont.matrix
```

```
##                      Contrasts
## Levels                Expansion.in.Untreated Treated.in.Expansion
##  UnculturedIslet.untreated                1                0
##  expandedislet.untreated                 -1                1
##  expandedislet.RCtreatment                0               -1
##                      Contrasts
## Levels                TreatedAndExpansion
##  UnculturedIslet.untreated                1
##  expandedislet.untreated                 0
##  expandedislet.RCtreatment               -1
```

- 4.3. Estimación del modelo y selección de genes.

Usamos el paquete limma para estimar el modelo, estimar los contrastes y realizar las pruebas de significación de cada comparación para determinar que genes pueden considerarse diferencialmente expresados.

Este método utiliza modelos de Bayes empíricos que combinan la información de toda la matriz de datos y de cada gen individual para obtener estimaciones de error mejoradas.

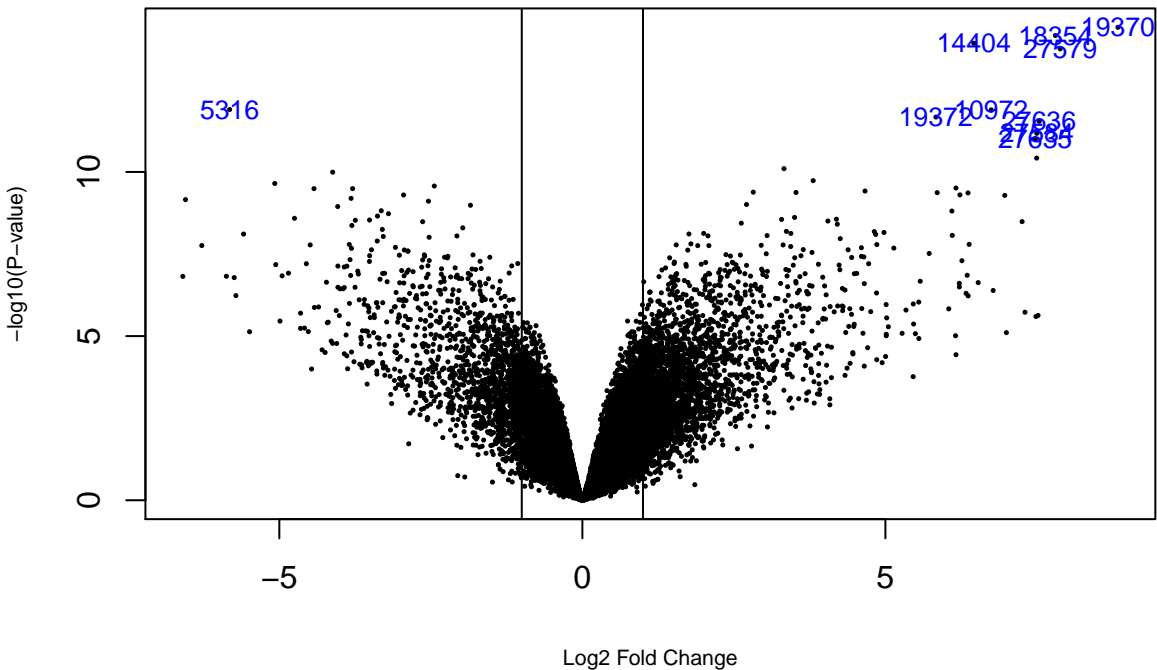
El análisis proporciona los estadísticos de test habituales como “Fold-change” t-moderados o p-valores ajustados que se utilizan para ordenar los genes de mas a menos diferencialmente expresados.

```
require(limma)
fit <- lmFit(eset_rma, design) #estimación del modelo
fit.main <- contrasts.fit(fit, cont.matrix)
fit.main <- eBayes(fit.main)
```

Para poder visualizar los resultados usamos un volcano plot que representa en abscisas los cambios de expresión del gen en escala logarítmica y en ordenadas el logaritmo negativo del p-valor.

```
coefnum = 1
opt <- par(cex.lab = 0.7)
volcanoplot(fit.main, coef = coefnum, highlight = 10, names = fit.main$ID, main = paste("Differentially
  colnames(cont.matrix)[coefnum], sep = "\n"))
abline(v = c(-1, 1))
```

## Differentially expressed genes Expansion.in.Untreated



```
par(opt)
```

Los puntos que se encuentran hacia la parte superior de la trama y que están muy lejos, ya sea hacia la izquierda o hacia la derecha representan genes que muestran cambios de gran magnitud en su expresión, así como una gran significación estadística.

## 5. Comparaciones múltiples.

La función `decideTests` selecciona los genes con un p-valor ajustado inferior a 0.01 y logaritmo de Fold Change mayor o igual a 2. Este es un buen criterio ya que combina la significación estadística y la significación biológica.

```
res <- decideTests(fit.main, method = "separate", adjust.method = "fdr", p.value = 0.01,
  lfc = 2)
print(summary(res))
```

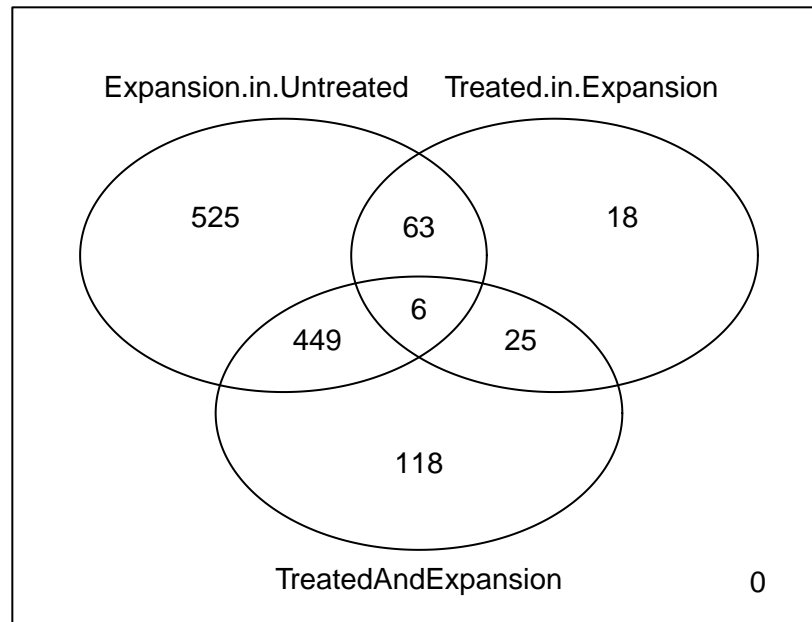
```
##          Expansion.in.Untreated Treated.in.Expansion TreatedAndExpansion
## Down                392                46                228
## NotSig              31278             32209             31723
## Up                   651                66                370
```

El diagrama de Venn permite visualizar la tabla anterior:

```
sum.res.rows <- apply(abs(res), 1, sum)
res.selected <- res[sum.res.rows != 0, ]

vennDiagram(res.selected[, 1:3], main = "Genes in common", cex = 0.9)
```

## Genes in common



## 6. Anotación de resultados.

La Anotación de resultados consiste en añadir a los indicadores del array otros más informativos de distintos aspectos biológicos.

El paquete `annaffy` permite de forma muy simple generar una tabla de anotaciones con hiperenlaces a las bases de datos para cada anotación seleccionada.

```
stopifnot(require(annaffy))
genesSelected <- rownames(res.selected)
at <- aafTableAnn(genesSelected, "mouse4302.db")
```

```
## Warning in chkPkgs(chip): The mouse4302.db package does not appear to contain
## annotation data.
```

```
## Warning in rsqLite_fetch(res@ptr, n = n): Don't need to call dbFetch() for
## statements, only for queries
```

```
## Warning in rsqLite_fetch(res@ptr, n = n): Don't need to call dbFetch() for
## statements, only for queries
```

La instrucción `saveHTML` crea una tabla con las anotaciones disponibles para los genes seleccionados en la sección de comparaciones múltiples.

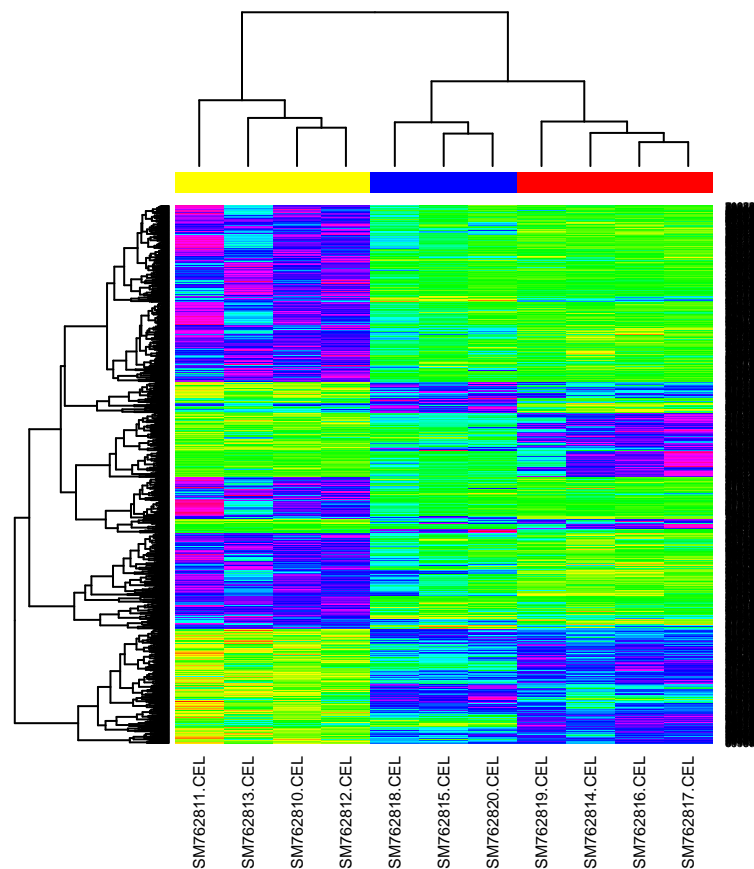
```
saveHTML(at, file.path(resultsDir, "anotations.html"), "Annotations for selected genes")
```

## 7. Visualización de los perfiles de expresión

Tras seleccionar los genes diferencialmente expresados podemos visualizar las expresiones de cada gen agrupándolas para destacar los genes que se encuentran up o down regulados simultáneamente constituyendo “perfiles de expresión”.

Para la visualización usamos el mapa de color o Heatmaps.

```
probeNames <- rownames(res)
probeNames.selected <- probeNames[sum.res.rows != 0]
exprs2cluster <- exprs(eset_rma)[probeNames.selected, ]
color.map <- function(grupo) {
  switch(grupo, "yellow", "red", "blue", "pink")
}
grupColors <- unlist(lapply(pData(eset_rma)$grupo, color.map))
heatmap(exprs2cluster, col = rainbow(100), ColSideColors = grupColors, cexCol = 0.7)
```



## Bibliografía.

### Artículo de la investigación:

Russ, H., Sintov, E., Anker-Kitai, L., Friedman, O., Lenz, A., Toren, G., Farhy, C., Pasmanik-Chor, M., Oron-Karni, V., Ravassard, P. and Efrat, S. (2011). Insulin-Producing Cells Generated from Dedifferentiated Human Pancreatic Beta Cells Expanded In Vitro. Disponible en: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0025566>

**Análisis qPCR:**

Biocompare.com. Quantitative PCR (qPCR) | Biocompare.com. Disponible en: <https://www.biocompare.com/PCR-Real-Time-PCR/7217-Real-Time-PCR/>

Es.wikipedia.org. (2017). PCR en tiempo real. Disponible en: [https://es.wikipedia.org/wiki/PCR\\_en\\_tiempo\\_real](https://es.wikipedia.org/wiki/PCR_en_tiempo_real).

Biología Molecular. Cuantificación de Ácidos Nucleicos. Disponible en: <https://biologiamolecularinteractiva.wordpress.com/about/cuantificacion-de-acidos-nucleicos/>.

**Análisis inmunofluorescencia:**

Eurodiagnostica.com. Immunofluorescence technique. Disponible en: <http://www.eurodiagnostica.com/index.php?headId=3&pageId=3&catId=10>.

**Análisis de datos con R:**

Sánchez, A. (2009). Ejemplo de Análisis de Datos de Microarrays con R y Bioconductor. [online] Ub.edu. Disponible en: [http://www.ub.edu/stat/docencia/bioinformatica/microarrays/ADM/labs/estrogen/Ejemplo\\_Estrogen.pdf](http://www.ub.edu/stat/docencia/bioinformatica/microarrays/ADM/labs/estrogen/Ejemplo_Estrogen.pdf).