Desarrollo de un proceso de análisis de datos RNA-seq, para el estudio de la expresión diferencial en distintos puntos en el tiempo

Alumna:

Miriam Magallón Lorenz

Máster en Bioinformática y Bioestadística

Área del trabajo final:

Bioinformática translacional, análisis de datos y genómica del cáncer

Tutor:

Bernat Gel

Fecha Entrega

5 de Junio del 2018

A) Creative Commons:

B) GNU Free Documentation License (GNU FDL)

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Desarrollo de un proceso de análisis de datos RNA-seq para el estudio de la expresión diferencial en distintos puntos en el tiempo (*time-course*).
Nombre del autor:	Miriam Magallón Lorenz
Nombre del consultor/a:	*Bernat Gel*
Nombre del PRA:	*Carlos Ventura*
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	Máster en Bioinformática y Bioestadística
Área del Trabajo Final:	Bioinformática translacional, análisis de datos y genómica del cáncer.
Idioma del trabajo:	Castellano
Palabras clave	RNA-seq, time-course, DESeq2

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

En el grupo de cáncer hereditario del Institut Germans Trias I Pujol (IGTP) se realizó un experimento *time-course* de diferenciación de células de Schwann a partir de un modelo no perecedero para tumores procedentes de neurofibromas plexiformes (PNF), causados por la doble inactivación del gen supresor tumoral *NF1* en estas células. El experimento *time-course* se llevó a cabo desde células pluripotentes (iPSC) hasta células de Schwann maduras (iPSC, cresta neural, día7, día14 y día30). De este experimento, se obtuvieron datos de RNA-seq que necesitaban ser analizados.

El objetivo de este proyecto fue desarrollar un proceso de análisis de datos de RNA-seq, para estudiar los datos procedentes del estudio de diferenciación en distintos puntos en el tiempo.

Con el uso de paquetes de R y Bioconductor , como *DESeq2,*se implementaron distintas funciones, tomando como referencia las preguntas planteadas por las

investigadoras, para responderlas de una manera más precisa y adecuada.

Como resultado, se obtuvieron unas funciones capaces de realizar un análisis de expresión diferencial clásico, así como responder a otras cuestiones que se habían planteado, como observar la expresión de un gen a lo largo del tiempo.

Para comprobar los métodos desarrollados se analizaron las muestras procedentes de fibroblastos (FiPS), utilizadas como control. Se comprobó que el proceso de diferenciación de células de Schwann se produjo de la manera esperada. Además, se encontraron nuevos posibles marcadores de los distintos estadios de diferenciación usando las funciones implementadas.

En conclusión, las distintas funciones implementadas funcionaban satisfactoriamente y se respondieron a las preguntas planteadas de manera apropiada.

Abstract (in English, 250 words or less):

Cancer Hereditary group from Germans Trias i Pujol Institute (IGTP) did a time course experiment to differentiate Schwann Cells from no perishable model for plexiforms neurofibromes tumours (PNF). These kinds of tumours are caused by double inactivation of *NF1* in these cells. The time course experiment was performed from induce pluripotent stem cells (iPSC) to mature Schwann cells (iPSC, neural crest, day7, day14, day30). We get RNA-seq data of the time points to analyse it.

The aim of this project was to develop an analysis process for RNA-seq data to study the differentiation of Schwann cells between time points.

We implement several functions using R and Bioconductor packages, such as DESeq2. We did it taking as reference the biological questions researchers have, in order to answer it as properly as possible.

Regarding to the implemented functions, we get some of them capable of analysing differential expressed genes, as well as, others to show specific gene

expression over time, giving the user more flexibility to answer specific researcher's questions.

In order to check this developed methodology, we analyse fibroblast samples (FiPS), which were used as control for the whole analysis. We prove that the differentiation process was taken place as expected. In addition, we found new possible markers for the different stages of differentiation process, using implemented functions.

In conclusion, the several functions implemented worked to answer researcher questions properly.

Índice

Lista de figuras

Lista de Tablas

1. Introducción

1.1. Contexto y justificación del Trabajo

1.1.1. Descripción general

En el grupo de Cáncer hereditario del Institut Germans Trias I Pujol (IGTP) se ha llevado a cabo un experimento *time-course* de diferenciación celular a partir del cual se han obtenido datos de RNA-seq. Estos datos necesitaban ser analizados para así sacar conclusiones.

Los datos proceden de un experimento de diferenciación de células de Schwann (SC), células glía del sistema nervioso periférico, en distintos puntos en el tiempo. Las células de Schwann tienen un papel muy importante en la neurofibromatosis[1].

La neurofibromatosis tipo I (NF1) es una enfermedad autosómica dominante causada por mutaciones germinales en el gen supresor tumoral *NF1,* que codifica para la neurofibromina, un regulador negativo de Ras. Esta enfermedad se caracteriza por tener diferentes manifestaciones clínicas como neurofibromas cutáneos o neurofibromas plexiformes (PNF), entre otras. Los neurofibromas plexiformes son tumores benignos de células de Schwann de la vaina del nervio periférico, que se desarrollan por la inactivación doble de *NF1,* y pueden progresar hacia un sarcoma de tejido blando maligno (tumores malignos de la vaina del nervio periférico (MPNSTs)), siendo la causa principal de mortalidad por NF1. En este contexto, las células PNF y MPNST comparten la mutación somática *NF1*[2–4].

Debido a que las células primarias procedentes de los PNF no se pueden mantener en cultivo durante mucho tiempo, para poder estudiar el proceso de diferenciación de las células de Schwann en este tipo de tumores, se desarrolló un modelo no perecedero para tumores benignos como son los PNF, ya que existe una carencia de este tipo de modelos procedentes de tumores benignos. Por ello, se generaron diferentes líneas de células madre pluripotentes inducidas (iPSC) isogénicas *NF1*$^{-/-}$ y *NF1* $^{+/-}$ a partir de unas células primarias derivadas de PNF. El mismo proceso de reprogramación se realizó con fibroblastos, *NF1* $^{+/+}$, procedentes de una biopsia de piel (FiPS), que se usaron

como células control para estudiar un proceso de diferenciación de las células de Schwann[5] (Explicado en el apartado de métodos).

El objetivo de este proyecto es desarrollar un proceso de análisis de datos de RNA-seq, para estudiar los datos procedentes del estudio de diferenciación de células de Schwann en distintos puntos en el tiempo.

Cabe destacar que el trabajo de análisis bioinformático se realiza en el laboratorio de investigación en el que se llevó a cabo el experimento, es decir, la bioinformática está en contacto directo con las investigadoras responsables del proyecto. Por tanto, el proceso de análisis de datos responderá directamente a las preguntas y necesidades planteadas por las investigadoras.

1.1.2. Justificación del trabajo

La tecnología de secuenciación masiva del transcriptoma, RNA-seq, permite, ente otras aplicaciones, observar la expresión diferencial de genes (DEG), así como descubrir SNVs (*Small Nucleotide Variants*), usando sólo un conjunto de datos [6]. Esta tecnología ha adquirido gran importancia en el estudio de la expresión génica diferencial. Los métodos computacionales y estadísticos utilizados para el análisis de estos datos son recientes y no se encuentran tan establecidos si los comparamos con los usados para el análisis de datos de microarrays.

El análisis de datos de RNA-seq se puede llevar a cabo usando distintas estrategias dependiendo de los objetivos y las cuestiones biológicas en el momento del diseño experimental. Es por ello, que cada análisis de RNA-seq puede tener diferentes métodos óptimos para el alineamiento y cuantificación de los transcritos, su normalización y su expresión diferencial[7,8]. En la Figura 1 se muestra una representación del proceso clásico de análisis de RNA-seq.

Actualmente, existen distintas formas para realizar el alineamiento. Una de ellas consiste en el alineamiento explícito de los *reads* sobre un genoma o transcriptoma de referencia, seguido de una cuantificación de los *reads* mapeados (ej. *Tophat-Cufflinks*[9]*, Tophat-HTSeq*[10], y *STAR-HTSeq*[11]). Por otro lado, los llamados "métodos de pseudoalineamiento", rompen los *reads* en *k-mers* antes de enfrentarlos a los transcritos (ej*. Salmon*[12] y *Kallisto*[13]),

consiguiendo una cuantificación directa sin el proceso previo de mapeado explícito. Este tipo de *software* proporciona una mayor resolución y velocidad de cálculo en comparación con aquellos que utilizan un alineamiento previo a la cuantificación de los *reads*[14,15].

En cuanto el análisis de la expresión diferencial, existen distintos paquetes de Bioconductor para llevarlo a cabo. Los métodos más utilizados son *edgeR*[16], *DESeq2*[17] *y lima Voom*[18], ya que son los recomendados para comparaciones entre muestras en las que se quiera conocer la expresión diferencial. *DESeq2* es capaz de realizar el análisis de datos *time-course* así como comparaciones *pair-wise*[7]. Normalmente, las comparaciones que se realizan para el estudio de la expresión diferencial de los genes, son realizadas entre dos condiciones (control y tratamiento). En el caso de datos procedentes de un *time-course*, los estudios estadísticos y comparativos entre distintos puntos en el tiempo comportan una mayor cantidad de contrastes, generando gran cantidad de datos. Esto dificulta el análisis y la interpretación por parte de las investigadoras[19].

Por último, para poder realizar una buena interpretación de los resultados, existen distintos paquetes de Bioconductor[20], que permiten hacer el análisis de enriquecimiento (*Gene Set Enrichment Análisis(GSEA)*), o también, software como *PANTHER*[21], con las que también puede encontrar información de interés de los genes diferencialmente expresados. Además con paquetes como *biomaRt*[22] se puede obtener la anotación referente a cualquier conjunto de genes.

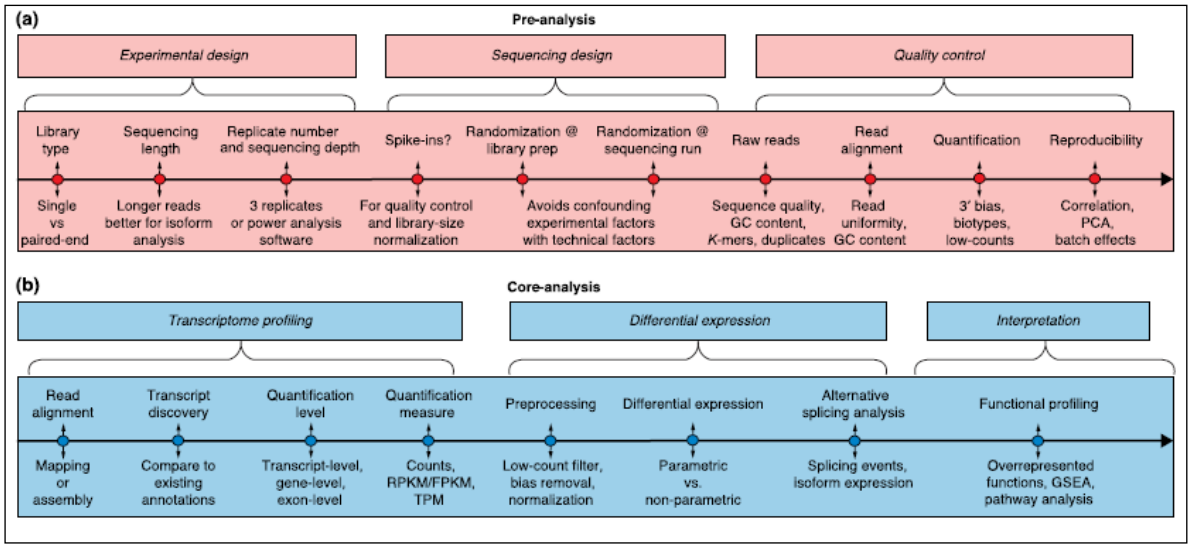


Figura 1. Proceso general de anàlisis de RNA-seq. [8]

En definitiva, debido a la amplia variedad de herramientas presentes para un análisis de datos de este tipo, lo que se pretende conseguir con este proyecto es desarrollar un proceso de análisis de datos de RNA-seq procedentes de un experimento *time-course,* que se ajuste exactamente a las necesidades dlas investigadoras y que facilite el proceso de exploración y análisis de los datos. Para ello, se ha desarrollado un conjunto de funciones que facilitan y automatizan el uso de métodos ya existentes para el análisis de la expresión diferencial (*Salmon* y paquetes de Bioconductor, como *DESeq2.),* y que permiten explorar losdatos y llegar de manera eficaz a conclusiones concretas.

1.2. Objetivos del Trabajo

Objetivos generales

El objetivo general de este proyecto es realizar la parte computacional del análisis de datos de RNA-seq procedentes de un experimento *time-course*.

Objetivos específicos

Los objetivos específicos son:

1. Diseñar y desarrollar un proceso de análisis de datos de RNA-seq.
2. Probar el proceso desarrollado para analizar datos de RNA-seq procedentes de un experimento *time-course.*

1.3. Enfoque y método seguido

En este proyecto el proceso de análisis se realizó a partir del pre-análisis. Es decir, los datos ya habían pasado un control de calidad previo, y se comprobó que las lecturas o *reads* eran de buena calidad.

El desarrollo de las distintas funciones surgió de la necesidad de dar respuesta a unas preguntas biológicas concretas. Debido a que los métodos existentes, como *DESeq2*, no facilitaban el análisis de los datos de la manera requerida, se implementaron funciones a partir de métodos ya existentes, con el fin de hacerlas más fáciles para el usuario, y con las que poder dar unos resultados óptimos de forma sencilla.

1.4. Tecnologías empleadas

En este apartado se explican los métodos empleados para implementar las distintas funciones, así como las herramientas de soporte que se utilizaron en este proyecto.

1.4.1. Herramientas soporte y lenguajes de programación

1.4.1.1. GitHub

Los diferentes scripts elaborados a lo largo del proyecto se depositaron en un repositorio de github, que utiliza el sistema de control de versiones. La existencia de este tipo de plataformas pretende cubrir dos problemáticas principales, la dificultad que supone la colaboración de distintas personas en un mismo proyecto, así como evitar la pérdida de información, siendo capaz de volver a versiones anteriores.

1.4.1.2. R 3.4 y Bioconductor 3.6

Para el desarrollo del proceso de análisis de los datos, se utilizó el software libre R-3.4 [23] a través de la interfaz RStudio[24]. Los paquetes que fueron usados para la realización del proyecto, provienen tanto de R como de Bioconductor (versión 3.6)[20]. R es un lenguaje de programación funcional orientado especialmente a la manipulación de datos, cálculos estadísticos y generación y visualización de gráficos[23]. Por su parte, RStudio es un entorno de desarrollo integrado (IDE) de R[24] Bioconductor es un software libre que utiliza el lenguaje estadístico de R y proporciona herramientas para el análisis y comprensión de datos genómicos de alto rendimiento (*high-throughput genomic data*)[20].

1.4.2. Salmon (Versión 0.9.1)

Salmon es un método para cuantificar la abundancia de los tránscritos a partir de los *reads* de una manera precisa y rápida. Tiene la capacidad de corregir los sesgos del contenido de fragmentos GC presentes en los transcritos lo que hace que aumente la precisión de los estimadores de abundancia. Esto hace que los resultados del análisis de la expresión diferencial sean fiables[12]. Esta herramienta combina un algoritmo de inferencia paralela y un sofisticado modelo de sesgo con un alineamiento mucho más rápido que el alineamiento clásico[12]. *Salmon* consiste en tres componentes: un alineamiento ligero, una fase inicial que estima los niveles expresión y los parámetros del modelo, y una fase encargada de refinar la expresión estimada. Además, proporciona la habilidad de estimar la abundancia incierta, debido a que realiza un muestreo aleatorio, y la ambigüedad introducida por el multialineamiento de los *reads*[12].

Todo lo que *Salmon* necesita es un fichero FASTA que contenga el transcriptoma de referencia, y un conjunto de ficheros FASTA/FASTQ que contengan las lecturas o *reads*. *Salmon* también tiene la capacidad de usar archivos que vengan de alineaciones ya calculadas en forma de archivos BAM/SAM, en vez de las lecturas crudas[25].

El método de *quasi-mapping* en el que está basado *Salmon* presenta dos fases:

- Fase de creación del índice con la que enfrentar los *reads*: este paso es independiente de las lecturas y solo necesita un conjunto de tránscritos de referencia para que este se llevara a cabo[25].
- Fase de cuantificación[12].

Como resultado se obtienen un fichero de cuantificación llamado *quant.sf.* Este archivo contiene las siguientes columnas[25]:

- *Name:* nombre del archivo FASTA que contenían los *reads.*
- *Length:* la longitud del transcrito de referencia en nucleótidos.
- *EffectiveLength:* Esta es la longitud efectiva calculada del transcrito de referencia. Toma en cuenta todos los factores que pueden afectar a la

probabilidad de mapear estos fragmentos de transcripción teniendo en cuenta el sesgo específico de secuencia y los fragmentos GC.

- *TPM:* es el estimador de la abundancia relativa de *Salmon* del transcrito en unidades de Transcrito Por Millón (*Transcripts Per Million (TPM))*. TPM es la medida de abundancia relativa recomendada para el análisis estadístico siguiente.
- *NumReads:* es el estimador de *Salmon* correspondiente al número de lecturas mapeadas a cada transcrito que fue cuantificado.

1.4.3. Paquete tximport

El paquete *tximport*[1,26] genera las matrices de cuentas a partir de los estimadores de abundancia y la longitud de los tránscritos que pueden ser usadas por métodos estadísticos como *DESeq2*, así como también permite realizar la normalización de las cuentas de dos maneras distintas a partir del parámetro *countsFromAbundance.* Las opciones son:

- Por el tamaño de la librería (*scaledTPM)*
- Por la media de la longitud de los tránscritos entre las muestras y el tamaño de la librería (*lengthScaledTPM)*.

Además, con el parámetro *dropInfReps* se indica si se quiere obtener una varianza única por tránscrito y por muestra.

[1] Viñeta *tximport:* https://bioconductor.org/packages/release/bioc/vignettes/tximport/inst/doc/tximport.html

1.4.4. DESeq2

El paquete *DESeq2*[2] proporciona métodos para estudiar la expresión diferencial con el uso de modelos lineales generalizados binomiales negativos. El uso de los modelos lineales proporciona flexibilidad para analizar diseños más complejos.

$$Kij \sim NB\ (\text{mean} = \mu_{ij}\ ,\ \text{dispersion}\ =\ \alpha_i)$$

$$\mu_{ij} = s_{ij} q_{ij}$$

$$log\ q_{ij} = \Sigma_r\ x_{jr}\ \beta_{ir}$$

El punto de partida del análisis de *DESeq2* es la matriz que contiene las cuentas (K), importada, por ejemplo, con *tximport*, donde cada fila representa un gen (i) y cada columna representa una muestra (j). La matriz de cuentas es modelada como indica la distribución bionmial negativa (NB), con media *μij* y dispersión *αi.* La media es la cantidad proporcional a la concentración de fragmentos de cDNA del gen en la muestra (q_{ij}), escalada con un factor de normalización (*size factor*) (s_{ij}). s_j, puede ser usado por todos los genes en una muestra, teniendo en cuenta las diferencias en la profundidad de la secuencia (*sequencing depth)* entre las muestras[17.]

La variabilidad entre grupos se modela por la dispersión del parámetro *αi,* el cual describe la variabilidad del factor vía:

$$\text{Var } K_{ij} = \mu_{ij} + \alpha_i \mu^2_{ij}.$$

Una estimación precisa de este parámetro es crucial para la inferencia sobre la expresión diferencial. Este paquete asume que los genes de medias similares de expresión tienen dispersión similar. En primer lugar estima secuencialmente una distribución previa para los valores de dispersión verdaderos sobre el modelo, y luego proporciona el máximo *a posteriori* (MAP). Además, *DESeq2* estima el ancho de la distribución previa con los datos, por ello, automáticamente controla la cantidad de reducción basada en las propiedades

[2] La viñeta del paquete DESeq2 se encuentra en el siguiente enlace con la información de las funciones más importantes:
https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

observadas de los datos. Este procedimiento de reducción ayuda a eliminar los falso positivos que pueden dar lugar a infraestimar la dispersión[17,27].

Otra dificultad que se puede encontrar en el análisis de datos *high-throughput sequencing* (HTS) es la gran variabilidad en los estimadores de *log2 fold change* (LFC) de genes con bajas cuentas. *DESeq2* mediante un procedimiento empírico de Bayes, consigue apaliar esta dificultad con la reducción hacia cero de los estimadores de LFC (*lfc shrink,)* de una manera en la que la reducción es mayor cuando la información disponible de un gen es pequeña, que puede ser debida a que las cuentas sean bajas, que la dispersión sea alta o que haya pocos grados de libertad. Para llevar este procedimiento a cabo, en primer lugar se ajusta el modelo GLM para obtener el *maximum-likelihood estimates* (MLE) para los LFCs y así centrar a distribución normal a 0 que había sido observada mediante MLE sobre estos genes. Esta distribución, es la distribución previa de los LFC. Se vuelve a ajustar un modelo GLM donde los MAP son los estimadores que se guardan como estimadores finales de los LFC. Además, se añade un error estándar de cada estimador. Estos estimadores de LFC reducidos y sus errores, son los que se usan con el test de Wald para estudiar la expresión diferencial de los genes[17,27].

Para el estudio de la expresión diferencial de los genes, *DESeq2* permite realizar dos tipos de test, *Likelyhood Ratio Test* (LRT) y el test de Wald. El test de Wald permite probar los coeficientes individuales o contrastes de coeficientes, sin la necesidad de un modelo reducido como LRT. Los p-valores obtenidos de la prueba de Wald son distintitos según las comparaciones que se quieran llevar a cabo, teniendo un p-valor específico para cada gen en esa comparación. Estos p-valores obtenidos se ajustan mediante el procedimiento FDR (*False Discovery Rate)* de Benjamini y Hochberg (BH) [17,27].

La hipótesis nula de LRT considera que los coeficientes de las variables del modelo completo frente al reducido son 0, por lo tanto, los p-valores obtenidos por el LRT representan las diferencias que existen entre el modelo completo frente al reducido, mostrando diferencias únicamente en el LFC de las distintas comparaciones.

DESeq2 ofrece una solución integral y general para el análisis a nivel genético de datos de RNA-seq. Los estimadores de reducción (*shrink estimates)* mejoran sustancialmente la estabilidad y reproducibilidad de los resultados del análisis. Esto permite que *DESeq2* ofrezca un rendimiento constante para una amplia gama de tipos de datos y lo hace aplicable para pequeños estudios con pocas réplicas, así como para grandes observaciones muestrales[17,27].

1.4.5. GOseq, GAGE y pathview.

En un estudio estándar de datos transcriptómicos, una vez obtenidos los genes diferencialmente expresados, se pueden realizar pruebas estadísticas para conseguir qué funciones moleculares, o vías de señalización, se encuentran más representados según el conjunto de genes diferencialmente expresados que se han obtenido. Este método se basa en clasificar el transcriptoma contemplando la medida de la expresión diferencial.

Los datos procedentes de un estudio de RNA-seq complican la aplicación directa de estos métodos debido a los sesgos que se presentan en la longitud de los genes. Es por ello que existen métodos específicos para el análisis de estos datos[8]. El método utilizado en este proyecto fue *GOseq*[28,3] para la obtención de los térmicos de *Gene Ontology* (GO)[29] de los genes diferencialmente expresados. Esta herramienta estima un efecto de sesgo en los resultados de expresión diferencial (como la longitud de los genes) y adapta la estadística hipergeométrica tradicional utilizada en la prueba de enriquecimiento funcional para explicar este sesgo.

Además, para conocer qué vías se encontraban diferencialmente representadas a partir de los genes diferencialmente expresados, se utilizó la herramienta *GAGE*[30,31] junto con *pathview*[32] con el uso de la base de datos KEGG[33].

[3] Viñeta GOseq:
http://bioconductor.org/packages/release/bioc/vignettes/goseq/inst/doc/goseq.pdf

Viñeta GAGE-pathview:
https://bioconductor.org/packages/release/bioc/vignettes/gage/inst/doc/RNA-seqWorkflow.pdf

1.5. Planificación del Trabajo

1.5.1. Tareas realizadas y planificación temporal.

Los datos analizados proceden de un estudio de análisis de diferenciación desde células iPS (*induced Pluripotent Stem Cells*) a células de Shwann, en distintos puntos en el tiempo. El primer paso consistió en hablar con las investigadoras para obtener un listado de preguntas concretas sobre los datos.

El diseño del proceso de análisis se basó en las preguntas planteadas por las investigadoras. Se crearon funciones con las que analizar los datos procedentes de un *time-course* y conseguir los genes diferencialmente expresados, así como para la obtención de gráficos. Para estudiar la expresión diferencial, se usó el paquete de *DESeq2*[17]*.* A partir de esta librería se generaron las distintas funciones adaptándolas a las necesidades que requiere el análisis de estos datos del tipo *time-course*, para así responder de una manera más concreta a las preguntas planteadas. Antes de la obtención de los DEG, se hizo un control de calidad de los datos crudos con los que se pretendía conocer el comportamiento de las muestras y cómo estas se relacionaban entre ellas Para ello, se hicieron dendrogramas, *heatmap*s y, sobre todo, gráficos en los que se representaban los componentes principales (PC).

Para la interpretación de los resultados se usaron distintas librerías como son *GOseq*[28] y *GAGE*[30]. Con estas librerías se adaptaron funciones para conseguir resultados del perfil funcional de los genes diferencialmente expresados.
Además, se implementaron funciones para observar de manera más visual la expresión de genes concretos en el tiempo y una para determinar qué genes se expresaban de manera similar a un gen de interés mediante el estudio de la correlación de su expresión a lo largo del tiempo.

Con el análisis de los datos se comprobó si las funciones actuaban de la manera esperada para la obtención de los resultados, y así poder encontrar fallos en ellas para mejorarlas.

Por último, se analizaron los datos del estudio diferencial de células de Schwann con el fin de responder a las preguntas planteadas por las investigadoras.

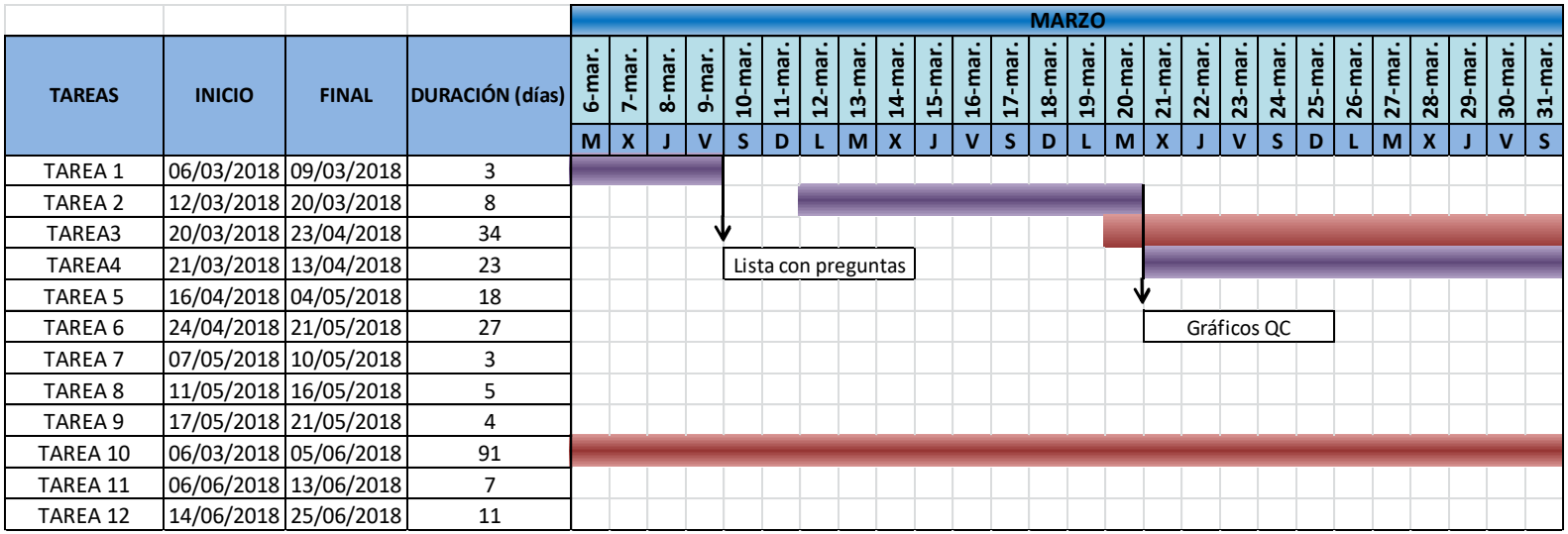
Las distintas tareas realizadas a lo largo del proyecto se encuentran en la Tabla 1, de una manera resumida, junto con su planificación temporal.

Tabla 1. Tareas realizadas a lo largo del proyecto

TAREAS	TAREAS	INICIO	FINAL	DURACIÓN (días)
Hablar con el investigador para concretar las preguntas	TAREA 1	06/03/2018	09/03/2018	3
Realizar un control de calidad previo al análisis	TAREA 2	12/03/2018	20/03/2018	8
PEC2: Desarrollo del Trabajo I	TAREA3	20/03/2018	23/04/2018	34
Diseñar la *pipeline* de análisis para responder las preguntas	TAREA4	21/03/2018	13/04/2018	23
Análisis de datos RNA-Seq del experimento de *time-course*	TAREA 5	16/04/2018	04/05/2018	18
PEC3: Desarrollo del Trabajo II	TAREA 6	24/04/2018	21/05/2018	27
Presentar resultados para refinar el análisis	TAREA 7	07/05/2018	10/05/2018	3
Análisis final de los datos	TAREA 8	11/05/2018	16/05/2018	5
Presentar resultados a los investigadores	TAREA 9	17/05/2018	21/05/2018	4
PEC4: Redacción de la memoria	TAREA 10	06/03/2018	05/06/2018	91
PEC5a: Presentación	TAREA 11	06/06/2018	13/06/2018	7
PEC5b: Exposición oral	TAREA 12	14/06/2018	25/06/2018	11

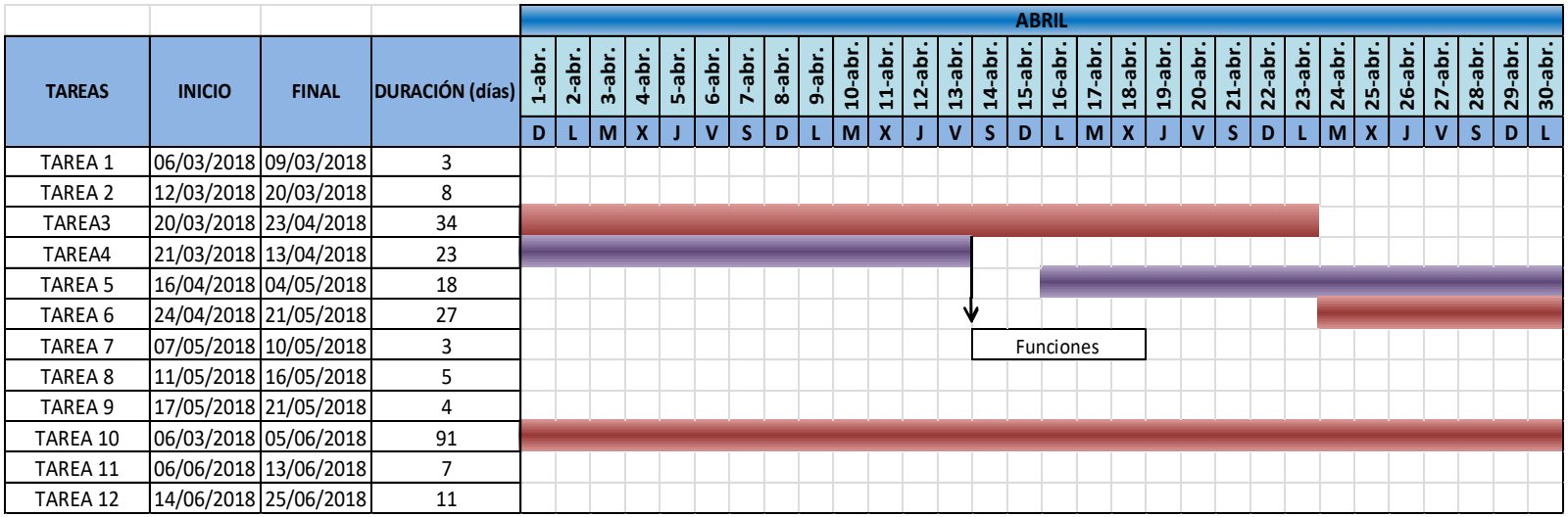
Tareas
PECs

En las siguientes figuras se muestra la planificación de proyecto de una manera más visual, utilizando el diagrama de Gantt.



TAREAS	INICIO	FINAL	DURACIÓN (días)
TAREA 1	06/03/2018	09/03/2018	3
TAREA 2	12/03/2018	20/03/2018	8
TAREA3	20/03/2018	23/04/2018	34
TAREA4	21/03/2018	13/04/2018	23
TAREA 5	16/04/2018	04/05/2018	18
TAREA 6	24/04/2018	21/05/2018	27
TAREA 7	07/05/2018	10/05/2018	3
TAREA 8	11/05/2018	16/05/2018	5
TAREA 9	17/05/2018	21/05/2018	4
TAREA 10	06/03/2018	05/06/2018	91
TAREA 11	06/06/2018	13/06/2018	7
TAREA 12	14/06/2018	25/06/2018	11

Figura 2. Calendario correspondiente al mes de Marzo en el que se incluyen las tareas y el diagrama de Gantt junto con los hitos asociados a cada tarea. El color lila corresponde a las tareas planteadas por el alumno, mientras que el color granate son las Pruebas de evaluación continua realizadas a lo largo del Trabajo Final de Master (TFM).



TAREAS	INICIO	FINAL	DURACIÓN (días)
TAREA 1	06/03/2018	09/03/2018	3
TAREA 2	12/03/2018	20/03/2018	8
TAREA3	20/03/2018	23/04/2018	34
TAREA4	21/03/2018	13/04/2018	23
TAREA 5	16/04/2018	04/05/2018	18
TAREA 6	24/04/2018	21/05/2018	27
TAREA 7	07/05/2018	10/05/2018	3
TAREA 8	11/05/2018	16/05/2018	5
TAREA 9	17/05/2018	21/05/2018	4
TAREA 10	06/03/2018	05/06/2018	91
TAREA 11	06/06/2018	13/06/2018	7
TAREA 12	14/06/2018	25/06/2018	11

Figura 3. Calendario correspondiente al mes de Abril en el que se incluyen las tareas y el diagrama de Gantt junto con los hitos asociados a cada tarea. El color lila corresponde a las tareas planteadas por el alumno, mientras que el color granate son las pruebas de evaluación continua realizadas a lo largo del Trabajo Final de Master (TFM).

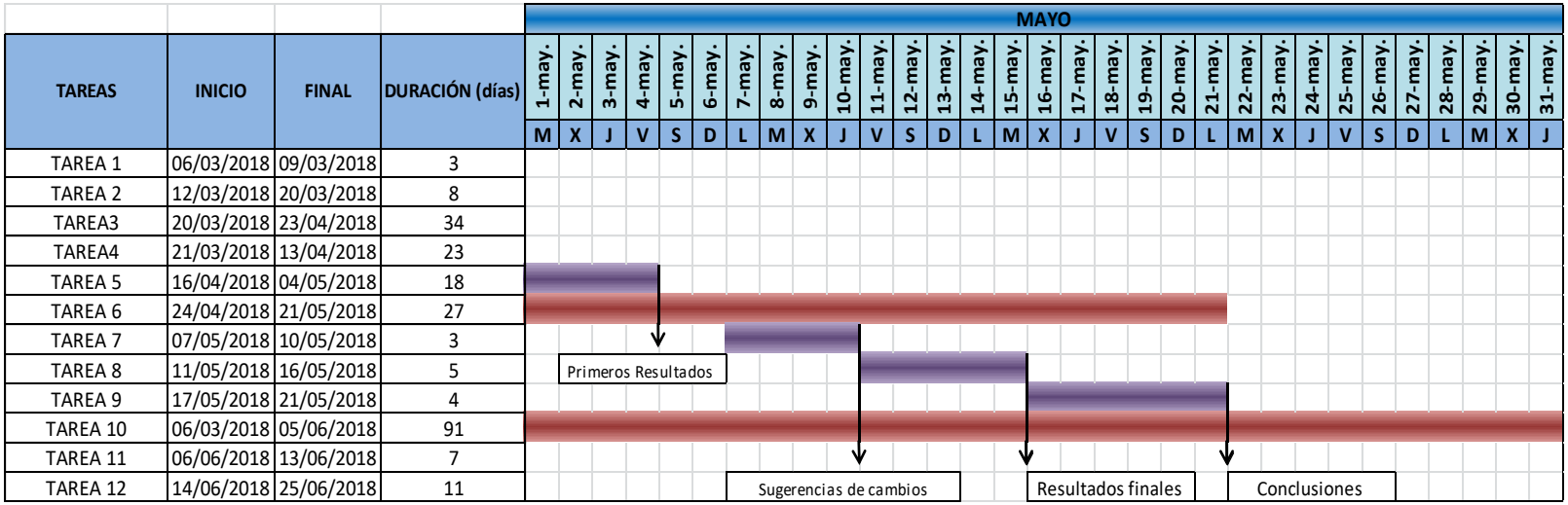


Figura 4. Calendario correspondiente al mes de Mayo en el que se incluyen las tareas y el diagrama de Gantt junto con los hitos asociados a cada tarea. El color lila corresponde a las tareas planteadas por el alumno, mientras que el color granate son las pruebas de evaluación continua realizadas a lo largo del Trabajo Final de Master (TFM).

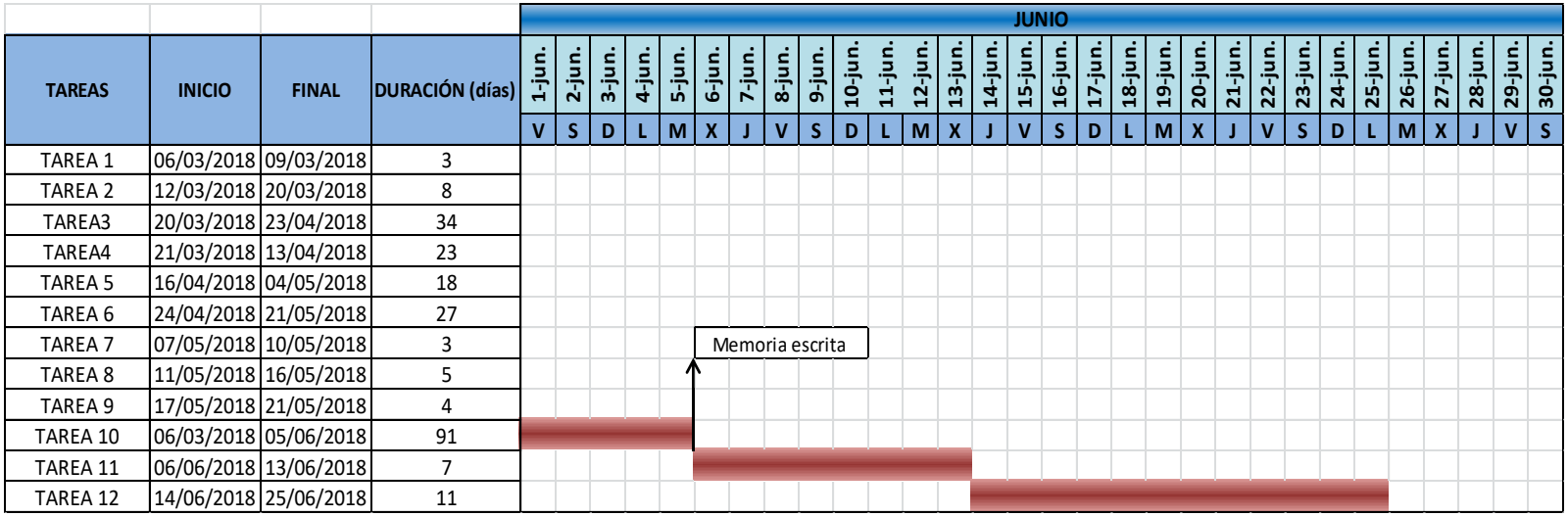


Figura 5.Calendario correspondiente al mes de Junio en el que se incluyen las tareas y el diagrama de Gantt junto con los hitos asociados a cada tarea. El color lila corresponde a las tareas planteadas por el alumno, mientras que el color granate son las pruebas de evaluación continua realizadas a lo largo del Trabajo Final de Master (TFM).

Los hitos marcan los estados intermedios del proyecto y permiten avanzar en las sucesivas etapas del proyecto.

Tabla 2. Hitos Asociados a las tareas realizadas.

TAREAS	Hito	INICIO	FINAL	DURACIÓN (días)
Hablar con las investigadoras para concretar las preguntas	Lista con preguntas	06/03/2018	09/03/2018	3
Realizar un control de calidad previo al análisis	Gráficos de control de calidad	12/03/2018	20/03/2018	8
Diseñar la *pipeline* de análisis para responder las preguntas	Funciones para analizar los datos	21/03/2018	13/04/2018	23
Análisis de datos RNA-seq del experimento de *time-course*	Primeros resultados	16/04/2018	04/05/2018	18
Presentar resultados para refinar el análisis	Sugerencias de cambios	07/05/2018	10/05/2018	3
Análisis final de los datos	Resultados finales	11/05/2018	16/05/2018	5
Presentar resultados a las investigadoras	Conclusiones	17/05/2018	21/05/2018	4

El hito asociado a la primera tarea era tener **una lista con las preguntas** que las investigadoras quería contestar para así poder realizar un buen diseño de los métodos de análisis.

Tras la realización del control de calidad se obtuvieron **gráficos** que nos permitían pensar cómo diseñar el proceso de análisis, conociendo el comportamiento que tenían los datos.

Se construyeron una serie **de funciones** útiles para el correcto análisis de los datos.

Una vez analizados los datos se obtuvieron unos **primeros resultados** que son necesarios para refinar el diseño.

El último hito corresponde a los **resultados finales** obtenidos, junto con las **conclusiones** que sacamos una vez comentados con las investigadoras.

1.6. Breve sumario de productos obtenidos

Documentos oficiales para la Universitat Oberta de Catalunya:

- Propuesta de Trabajo Final de Máster.
- Pruebas de evaluación continua 1, planificación del trabajo 2, Informe de seguimiento Fase I y el informe de seguimiento Fase II.
- Memoria Final.
- Informe de Autoevaluación.

Resultados del estudio:

- Distintas funciones para el análisis de datos RNA-seq procedentes de un *time-couse*.
- Gráficos e información de los genes diferencialmente expresados entre distintos puntos en el tiempo de las muestras procedentes de las FiPS, para comprobar el funcionamiento de las funciones desarrolladas.
- Conclusiones de la herramienta desarrollada así como conclusiones sobre los resultados del análisis que se llevó a cabo.

1.7. Breve descripción de los otros capítulos de la memoria

En los siguientes apartados se explicara a partir de qué herramientas se implementaron las funciones de este proyecto, así como un ejemplo de análisis con los datos procedentes del estudio de diferenciación celular desde célula iPSC a SC.

- **Materiales y métodos**: Las preguntas biológicas con las que se consiguieron las ideas de cómo construir las funciones del proyecto y, herramientas utilizadas para el desarrollo de las funciones.

- **Resultados**: este apartado contiene las funciones que se implementaron para conseguir responder de una manera más adecuada preguntas planteadas por las investigadoras y, un ejemplo de análisis que se puede realizar con ellas usando los datos procedentes del estudio de diferenciación de las células de Schwann.

2. Materiales y métodos

El proceso clásico de análisis de datos RNA-seq presenta los siguientes pasos para obtener los genes diferencialmente expresados entre distintas condiciones:

- Establecer un diseño experimental adecuado para poder analizar de la manera correcta los datos crudos.
- Realizar el alineamiento y cuantificación de los tránscritos presentes en las muestras.
- Importar los datos para realizar el estudio diferencial.
- Control de calidad de los datos.
- Hacer el análisis de expresión diferencial de las muestras.
- Gene Set Enrichment Analysis (GSEA)
- Anotación de los genes obtenidos

Sin embargo, en este proyecto, el modo en el que se analizaron los datos, se basó en las preguntas planteadas por las investigadoras. Estas preguntas fueron utilizadas para implementar nuevas funciones que facilitaran el análisis de los datos, con el fin de responder de una manera adecuada a las cuestiones biológicas planteadas por parte de las investigadoras. Cabe destacar que algunas de las funciones generadas permiten una mayor flexibilidad al usuario saliendo del esquema general de un análisis de expresión diferencial.

En los siguientes apartados se explicarán la procedencia de los datos que fueron analizados y las preguntas que se plantearon las investigadoras, así como el modo en el que se utilizaron las tecnologías explicadas en el apartado “Tecnologías empleadas” de la introducción, para desarrollar las distintas funciones.

2.1. Datos analizados y diseño experimental

Los datos con los que se hizo un estudio de expresión diferencial de SC y con los que se probaron las funciones generadas, provienen del grupo de Cáncer Hereditario del Instituo Germans Trias i Pujol (IGTP).

Debido a la cadencia de modelos no perecederos para tumores benignos, como son los PNFs, en el laboratorio de Cáncer Hereditaro del IGTP se generaron diferentes líneas de células madre inducias (iPSC) isogénicas (*NF1*$^{+/-}$ y *NF1*$^{-/-}$), a partir SC primarias procedentes de tumores plexiformes. Se utilizaron fibroblastos *NF1* $^{+/+}$, procedentes de una biopsia de piel como células control, y se siguió el mismo proceso de reprogramación a iPSC (FiPS) que con las células primarias procedentes de PNF. Una vez conseguidas unas líneas celulares imperecederas para mantenerlas en cultivo, se comenzó el estudio de diferenciación de SC *NF1*$^{-/-}$ así como de las células control[5].

Se dirigió la transformación de las iPSC a células de la cresta neural (NC) (precursoras de las células de Schwann entre otros tipos celulares[4,34]), con medios de cultivo celular especiales. El estudio de diferenciación se llevó a cabo durante 30 días, partiendo de NC (día 0). Las células fueron mantenidas en cultivo y se recogieron en tres ocasiones: a los 7 días de cultivo, 14 días de cultivo y 30 días de cultivo. Para comprobar que este proceso de diferenciación se estaba llevando a cabo, se hizo una *RT*-qPCR para ver la expresión de genes específicos en cada uno de los estadios. Según la bibliografía, en esos periodos de diferenciación, las células se deberían encontrar en los siguientes puntos: células de Schwann precursoras, células de Schwann inmaduras y células de Schwann maduras, respectivamente[4].

Con RNA extraído en los mismos puntos en el tiempo, se prepararon librerías *unstranded* para RNA-seq siguiendo los protocolos estándar de Illumina y se secuenciaron en secuenciadores Illumina HiSeq con reads de 2x 150bp. Con estos datos se pretende conseguir marcadores más específicos de cada punto de diferenciación y hacer un estudio de expresión diferencial entre las células de Schwann con un genotipo *NF1*$^{-/-}$ frente a las células control con un genotipo *NF1*$^{+/+}$ en los distintos estadios de diferenciación, con el fin de conocer qué papel desempeña la neurofibromina en este proceso de diferenciación.

Sin embargo, para comprobar la metodología de las funciones desarrolladas, en este proyecto sólo se analizaron los datos procedentes del estudio de diferenciación de las células control (FiPS) en los distintos puntos en el tiempo. El análisis se realizó con un total de 12 muestras con 3 réplicas por cada punto en el tiempo, excepto para el punto NC, o tiempo 0, en la que sólo se tenían dos réplicas, y las iPSC, donde solo se contaba con una. La información, asociada a estas muestras, se encuentra remarcada en la tabla 3.

Cabe destacar que un estudio de diferenciación en el tiempo puede realizarse de dos maneras distintas según lo que se quiera observar:

- La diferencia de expresión a lo largo del tiempo (diferencias entre el comienzo del experimento y el final).
- La diferencia de expresión entre los distintos puntos del tiempo.

En este experimento el tipo de análisis que se llevó a cabo fue el segundo, ya que se pretende observar las diferencias de expresión entre cada punto en el tiempo, asociado con el estadio de diferenciación de las células de Schwann.

En la tabla 3 se muestra de una manera más clara el número de contrastes que se realizaron para conseguir los DEG entre los distintos puntos en el tiempo de los datos procedentes de las células FiPS.

Tabla 3. Contrastes que se llevaron a cabo para el análisis de expresión diferencial de las células FiPS en los distintos puntos del tiempo. Los nombres de las columnas corresponden a las muestras que se establecieron como referencia (*ref_group)* en los distintos contrastes, mientras que el nombre de las filas hace referencia a los grupos de los que se quieren obtener la expresión diferencial (*query_group).* PSC: iPSC, NC: cresta neural.

	PSC	NC	day7	day14
NC	PSC-NC	-	-	-
day7	PSC-day7	NC-day7	-	-
day14	PSC-day14	NC-day14	day7-day14	-
day30	PSC-day30	NC-day30	day7-day30	day14-day30

Tabla 4. Tabla que recopila la información de las muestras necesaria para realizar los contrastes y conseguir los genes diferencialmente expresados.sample.name: nombre de las muestras (Exp: experimento, 5PNFiPs: tumor plexiforme 5, 3PNFiPs: tumor plexiforme 3, FiPS: células procedentes de fibroblastos, PSC: pluripotent stem cells, NC: células de la cresta neural (Neural Crest), MM: células con genotipo $NF1^{-/-}$, 7;14;30: días del proceso de diferenciación.). Time.char: puntos en el tiempo en forma de carácter para realizar el contraste entre los distintos puntos (PSC: iPSC, NC: cresta neural, day7;14;30: los días del estudio) . Cell.type: tipo de célula. Genotype: genotipo (PP: NF1 positivas, MM: NF1 negativas). Time: puntos en el tiempo en forma numérica (-1: iPSC, 0:NC, 7:day7, 14:day14, 30: day30). Exp: número del experimento del que provienen los datos. Genotype.time: información para realizar el contraste de las diferencias de expresión entre genotipos en los distintos puntos en el tiempo del experimento. Sample.time: información de la muestra y el punto temporal en el que se sitúa.

sample.name	time.char	cell.type	genotype	time	Exp	genotype.time	Sample.time
FiPS_PSC	PSC	FiPS	PP	-1	EX	PP_PSC	FiPS_PSC
EX15_FiPS_NC	NC	FiPS	PP	0	EX15	PP_NC	FiPS_NC
EX15_FiPS_7	day7	FiPS	PP	7	EX15	PP_7	FiPS_7
EX15_FiPS_14	day14	FiPS	PP	14	EX15	PP_14	FiPS_14
EX15_FiPS_30	day30	FiPS	PP	30	EX15	PP_30	FiPS_30
EX16_FiPS_NC	NC	FiPS	PP	0	EX16	PP_NC	FiPS_NC
EX16_FiPS_7	day7	FiPS	PP	7	EX16	PP_7	FiPS_7
EX16_FiPS_14	day14	FiPS	PP	14	EX16	PP_14	FiPS_14
EX16_FiPS_30	day30	FiPS	PP	30	EX16	PP_30	FiPS_30
EX18_FiPS_7	day7	FiPS	PP	7	EX18	PP_7	FiPS_7
EX18_FiPS_14	day14	FiPS	PP	14	EX18	PP_14	FiPS_14
EX18_FiPS_30	day30	FiPS	PP	30	EX18	PP_30	FiPS_30
3PNF_SiPSsv_MM_PSC	PSC	3MM	MM	-1	EX	MM_PSC	3PNFiPS_PSC
EXP11_3PNFiPs_MM_NC	NC	3MM	MM	0	EX11	MM_NC	3PNFiPS_NC
EXP11_3PNFiPs_MM_7	day7	3MM	MM	7	EX11	MM_7	3PNFiPS_7
EXP11_3PNFiPs_MM_14	day14	3MM	MM	14	EX11	MM_14	3PNFiPS_14
EXP11_3PNFiPs_MM_30	day30	3MM	MM	30	EX11	MM_30	3PNFiPS_30
5PNF_TDiPSsv_MM_PSC	PSC	5MM	MM	-1	EX	MM_PSC	5PNFiPS_PSC
EXP13_5PNFiPS_NC	NC	5MM	MM	0	EX13	MM_NC	5PNFiPS_NC
EXP10_5PNFiPS_7	day7	5MM	MM	7	EX10	MM_7	5PNFiPS_7
EXP10_5PNFiPS_14	day14	5MM	MM	14	EX10	MM_14	5PNFiPS_14
EXP10_5PNFiPS_30	day30	5MM	MM	30	EX10	MM_30	5PNFiPS_30
EXP13_EXP14_3PNFiPS_5PNFiPS_NC	NC	pool	MM	0	EX13	MM_NC	pool_NC
EXP13_EXP14_3PNFiPS_5PNFiPS_7	day7	pool	MM	7	EX13	MM_7	pool_7
EXP13_EXP14_3PNFiPS_5PNFiPS_14	day14	pool	MM	14	EX13	MM_14	pool_14

2.2. Preguntas biológicas planteadas

En primer lugar se obtuvo una lista de preguntas a partir de las cuales se diseñaron las distintas funciones que podían dividirse en dos puntos clave:

1. Comparación de la expresión de las células FiPS en los distintos puntos del tiempo y descubrir nuevos marcadores específicos de cada estadio. (Analizada en este proyecto)
2. Comparación de la expresión por genotipo (*NF1*$^{+/+}$ y *NF1*$^{-/-}$) en el tiempo para entender el papel de la neurofibromina en el proceso de diferenciación.

La lista de preguntas fue la siguiente:

- ¿Qué genes se encuentran diferencialmente expresados en dos puntos en el tiempo?
- ¿Qué función tienen los genes diferencialmente expresados?
- Conociendo la expresión de un gen, ¿qué genes se comportan de la misma manera o de forma inversa a él?
- Si se comportan de la misma manera, ¿pertenecen a la misma vía? ¿Tienen la misma función? ¿Se trata del mismo componente celular?
- ¿Existen genes afectados entre dos puntos en el tiempo de una vía concreta?

A partir de estas preguntas, se procedió a diseñar las distintas funciones para dar una respuesta más adecuada. En los siguientes apartados se explican los materiales utilizados para la implementación de las funciones basadas en las preguntas planteadas por las investigadoras. Las funciones que se construyeron serán explicadas en el apartado de resultados.

2.3. Paquetes de R y software usados para el proceso de análisis

2.3.1. Alineamiento y cuantificación

La herramienta que se utilizó en este proyecto para el alineamiento y la cuantificación de los tránscritos, fue *Salmon.*

Salmon necesita un fichero FASTA que contenga el transcriptoma de referencia, y un conjunto de ficheros FASTA/FASTQ que contengan las lecturas o *reads* que, en este proyecto proceden de los datos descritos en el apartado anterior.

En este proyecto el transcriptoma de referencia escogido fue el *hg19*. El archivo FASTA fue descargado del *Genome Browser* de la Universidad Santa Cruz de California (UCSC). El link de la descarga es el siguiente: http://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/refMrna.fa.gz

2.3.2. Importación de los datos

La importación de los datos obtenidos con *Salmon* se realizó con la función generada a partir del paquete *tximport.* Se importaron la longitud de los tránsctritos, los estimadores de la abundancia y se exportaron las matrices de cuentas.

En este proyecto, la normalización de todas las muestras de las que se querían estudiar la expresión diferencial, se hizo en este punto. Para la normalización se utiliz´´o el método *lengthScaledTPM,* que tiene en cuenta tanto el tamaño de la librería como la longitud de los tránscritos. Se normalizaron todas las muestras a la vez.

2.3.3. Análisis de expresión diferencial

Para el desarrollo de las funciones que obtenían la expresión diferencial, se usaron algunos métodos *DESeq2.* A continuación se nombran brevemente las funciones de *DESeq2* utilizadas, con una breve explicación de cada una de ellas y de su cometido en este proyecto.

Una vez son importados los datos*,* la siguiente función utilizada fue *DESeqDataSetFromTximport().* Este método de *DESeq2* se utilizó para generar una nueva función con la que construir un objeto *DESeqDataSet* tras importar las abundancias de transcritos con la función desarrollada a partir de *tximport.*

Cuando se obtiene el objeto *DESeqDataSet*, ya es posible proceder al análisis de expresión diferencial de los datos. El siguiente paso a realizar es la estimación de los parámetros. Esto se consigue con la función *DESeq().* Esta función realiza los tres pasos clave para realizar el estudio de expresión diferencial, los cuales son:

1. Estimar el tamaño del factor *sj* mediante la función *estimateSizeFactors.*
2. Estimar la dispersión *αi* mediante *estimateDispersions*
3. Ajustar el modelo GLM siguiendo la Binomial negativa y los estadísticos de Wald mediante *nbinomWaldTest,* si se realiza esta prueba.

El LRT puede realizarse especificando en la función *DESeq() test="LRT.* Tanto "LRT" como la prueba de "Wald" necesitan añadir en esta función una fórmula diseño a partir de la cual se realizaran los contrastes. En un modelo con más de una condición a contrastar, el test que se debe realizar, es el LRT. Un ejemplo de la fórmula diseño que se debería añadir sería ~ *time* + *genotype*.

Con la estimación de los parámetros anteriores, ya se puede proceder a realizar el estudio de expresión diferencial. La función *results()* recibe el objeto *DESeqDataSet y* es la encargada de obtener los resultados en forma de tabla en los que se detalla el *log2 fold change*, p-valor y los p-valores ajustados. Por defecto, esta función realiza el test de "Wald". Mediante la variable *name* o *contrast* el usuario puede decidir de qué comparación se quieren observar los resultados en la tabla generada por esta función. El objeto obtenido es un *DESeqResults.* Si el test escogido en la función *DESeq()* es "LRT", el usuario

debe añadir un modelo reducido de la fórmula diseño en *results(),* teniendo claro qué tipo de contraste quiere hacer. Continuando con el ejemplo de la fórmula anterior, el modelo reducido podría ser ~ *time*.

Con la prueba de "Wald", los genes diferencialmente expresados se obtienen en referencia a los contrastes realizados. Sin embargo, los DEG obtenidos por "LRT", son aquellos que son diferentes a lo largo de todo el proceso ya que se analizan las diferencias entre el modelo completo y el reducido.

El paquete *DESeq2* también incluye una función con la que reducir el efecto del tamaño (estimadores de LFC) siendo útil para la visualización y obtener un *ranking* adecuado de los genes. La reducción de LFC se consigue con la función *lfcShrink().* Esta función, consigue el *Maximo a posteriori* (MAP) de los *log2 fold change*. Para ello, se le proporciona el objeto *DESeqDataSet* junto con el nombre (*name)* o el número de coeficiente *(coef)* que se quiere reducir. Esto es un punto a tener en cuenta, ya que los resultados que se muestran por pantalla cuando se realizan tanto la función *results()* como la función *lfcshrink(),* son únicamente los referentes al último contraste de la condición analizada.

lfcshrink() permite realizar distintos tipos de reducción de LFC. Con la opción *type* se pueden elegir:

- *normal* es el original del paquete y es una distribución adaptativa Normal de manera previa (usado en este proyecto).
- *apeglm* es el estimador previo t adaptativo a la reducción del paquete *apeglm*[35].
- *ashr*[36] es un estimador del paquete *ashr*. *DESeq*2 utiliza esta función para ajustar una mezcla de la distribución Norma para formar la previa.

Estos tres pasos de análisis anteriores (*DESeq(), results() y lfcshrink())* fueron implementados en una única función para facilitar al usuario que pudiera usar estas funciones a la vez de una forma sencilla.

En resumen, estos métodos de *DESeq2* son las piezas claves de las funciones generadas, con las que se consiguen los genes diferencialmente expresados entre distintas condiciones. Sin embargo, a pesar de que *DESeq2* ya tiene

establecido un método de análisis para los genes diferencialmente expresados, la manera en la que se debían usar para responder a las preguntas establecidas por las investigadoras, no era todo lo flexible que se requería. Es por ello, se desarrollaron nuevas funciones a partir de estos métodos, cuya idea surgió de las preguntas de las investigadoras.

2.3.4. Control de calidad previo al análisis de los genes diferencialmente expresados.

El control de calidad previo al análisis de la expresión diferencial es necesario para encontrar posibles *outliers* en las muestras y entre las réplicas. Normalmente, el control de calidad se realiza mediante la observación del comportamiento de las muestras en distintos tipos de gráficos. Se realizaron funciones para eliminar aquellos genes cuyo nivel de expresión en el conjunto de muestras no era superior a 5 *read counts.* Además, se hizo una transformación logarítmica de los datos mediante la función *rlog()* para poder visualizar los datos gráficamente. La diferencia del comportamiento de las muestras, previos al filtro que pasaron y después de este, se observó mediante histogramas y gráficos de densidad.

Para comprobar la asociación de las distintas muestras y, si se comportaban de la manera esperada, se representaron las muestras en dendrogramas, *heatmaps* y en gráficos de componentes principales (PCA-plot) y MA-plots.

Con el fin de comprobar si el experimento biológico había funcionado de la manera adecuada, se estudió la expresión de genes que habían sido analizados previamente mediante *RT*-qPCR.

2.3.5. Gene Set Enrichment Analysis (GSEA)

En este proyecto se utilizaron *GOseq*[28] y *GAGE*[30,31] junto con *pathview*[32], con el uso de la base de datos KEGG[33]. Algunos de los métodos presentes en sus viñetas fueron usados para construir las nuevas funciones y obtener los resultados de una manera más sencilla y directa.

2.3.6. Anotación de los genes

Además de realizar un GSEA, se construyó una función para anotar los genes diferencialmente expresados, o aquellos que interesaban al investigador, y así conocer su funcionalidad por medio de los GO terms. Para ello, se utilizó el paquete de R *biomaRt* [22]. Este paquete utiliza el servicio web de BioMart para proporcionar los mapeos moleculares correspondientes.

3. Resultados

Debido a que este proyecto se realizó en contacto directo con las investigadoras, se implementaron funciones usando métodos ya existentes, como *DESeq2* para responder de una manera más precisa las cuestiones biológicas que se habían planteado. Con estas funciones se analizaron los datos del experimento de diferenciación de las células control (FiPS) del proceso de diferenciación de iPS a SC en el tiempo, con el fin de comprobar su funcionalidad y se responder a las preguntas relacionadas con la expresión diferencial entre los distintos puntos en el tiempo de las muestras. El análisis de las células FiPS se realizó para comprobar que la diferenciación se estaba llevando a cabo y, descubrir nuevos posibles marcadores específicos de cada punto del estudio en el tiempo (NC, día7, día 14 y día30).

3.1. Funciones desarrolladas.

Una vez plateadas las preguntas, se desarrollaron las distintas funciones, con las que se generó un paquete de R, RNAseqHelper[4]. La explicación del papel de cada función de manera detallada se encuentra en dicho paquete. A continuación se nombran algunas de las funciones clave que se fueron generando a lo largo de este proyecto.

3.1.1. Función salmonAlignment()

La función *salmonAlignment()* da al usuario la capacidad de hacer el alineamiento mediante *Salmon,* con el uso de una única función y, pudiendo llamar a esta herramienta desde R. Esta función está diseñada para realizar el alineamiento con una librería tipo *paired-end* ya que son el tipo de librerías que se preparan cotidianamente en el laboratorio de estancia. El parámetro que indica el tipo de librería utilizada es *libtype*, que por defecto es “IU”(*Inward Unstranded*), que indica que el alineamiento se realiza con los finales de las hebras en sentido inverso, mientras que, si fuera “OU” (*Outward Unstranded*),

[4] El código de las funciones desarrolladas se adjuntan con la memora final. En el anexo1 se detalla la lista de los archivos que se adjuntan.

el tipo de alineamiento partiría de un punto en común e iría en direcciones contrarias.

```
salmonAlignment(sample_names, file_1_suffix, file_2_suffix, fastqdir,
transcripts_index, output_suffix, output_quants_dir, libtype = "IU", threads = "4",
verbose = TRUE)
```

Es necesario proporcionar el nombre de las muestras a analizar que debe coincidir con el nombre de los ficheros FASTQ. Los parámetros *file_1_suffix y file_2_suffix* son necesarios para diferenciar los ficheros generados por las librerías *paired-end.* También se debe introducir el directorio en el que se encuentran, mediante el parámetro *fastqdir*. El índice con el que alinear las muestras, se tiene que indicar en *transcripts_index.* Debe generarse antes de realizar este alineamiento con el siguiente comando en la terminal:

```
salmon index -t (reference transcriptome directory) –i (index name) --type quasi -k 31
```

Se debe indicar el directorio de salida en *output_quants_dir,* y el sufijo asociado al fichero (normalmente se usa *quant*). El parámetro *threads* (por defecto es “4”) indica el número de *subprocesos* que se quieren utilizar para ejecutar *salmonAlignment()*.

Como resultado de esta función, se obtiene un fichero por muestra alineada llamado *quand.sf,* guardados en un directorio *salmon_quant* con el nombre de cada muestra. Este archivo contiene las cuantificaciones realizadas por *Salmon* que deben ser importadas por la función *importQuantData().*

3.1.2. Función importQuantData()

Esta función es la encargada de importar la longitud de los tránscritos y los estimadores de abundancia obtenidos por *Salmon* (TPM), así como exportar las matrices de las cuentas. De esta manera, los datos pueden ser usados por el usuario para realizar el análisis de expresión diferencial.

```
importQuantData(sample_names, output_suffix, txdb, txdb_keytype, txdb_columns,
output_quants_dir, output_tximp_dir,  type = "salmon", dropInfReps = TRUE,
countsFromAbundance ="lengthScaledTPM", verbose = TRUE, ...)
```

La función *importQuantData()* necesita del nombre de las muestras introducido en el parámetro *sample_names.* Además, se añade el sufijo que se puso en la función anterior mediante el parámetro *output_suffix*. También, se tienen que proporcionar los directorios donde se encuentran los ficheros obtenidos por *Salmon* (*output_quants_dir*), y el directorio en el que se quiere guardar el archivo que se genere a partir de esta función (*outpunt_tximp_dir*).

Con esta función se puede cambiar la anotación de los tránscritos, cuya anotación inicial será diferente dependiendo del transcriptoma de referencia utilizado. En el parámetro *txdb* se añade la base de datos que se quiere usar para la anotación (librería *AnnotationDb),* como por ejemplo *OrgDb*(utilizada en este proyecto). Con el uso de las funciones *keys()*, *keytypes()* y *columns()*, se pueden conocer por qué tipo de anotaciones puede cambiarse el nombre de los transcritos. En *txdb_keytype* se debe indicar el tipo de anotación que presentan los tránscritos según el transcriptoma de referencia utilizado en el alineamiento con *Salmon*. En *txdb_columns* se introduce el tipo de anotación que el usuario quiere que tengan los tránscritos. Esta anotación debe coincidir 1:1 con la que se indica en *txdb_keytype.* En este proyecto se utilizaron los siguientes parámetros en lo que a anotación respecta:

- *txdb:* se usó la librería *org.Hs.eg.db.*
- txdb_keytype: REFSEQ.
- txdb_columns: SYMBOL.

El parámetro *countsFromAbundance* es encargado de decir de qué manera escalar los estimadores de la abundancia, el cual puede realizarse de las siguientes formas: por el tamaño de la librería (*scaledTPM),* dividiendo la media de la longitud de los tránscritos entre las muestras y el tamaño de la librería (*lengthScaledTPM)*, o no usar ningún ajuste de los estimadores *(no)*. Por defecto esta función ajusta por *lengthScaledTPM*. De esta manera se normaliza la expresión de los tránscritos. Para el análisis de los datos el ajuste fue el que se estableció en la función por defecto.

Además, con *dropInfReps* se indicar si se quiere obtener una varianza única por tránscrito y por muestra, si *dropInfReps =TRUE (*por defecto).

El objeto de salida de esta función es una lista que contiene los estimadores de abundancia (*abundance)*, la longitud de los tránscritos (*length*), las cuentas (*counts*) y el tipo de *countsFromAbundance* realizado.

3.1.3. Función selectFromTximport()

selectFromTximport() es una función con la que se pretenden seleccionar aquellas muestras de las que se quiera realizar el contraste en un momento dado. Esta función es útil si en la función *importQuantData()* se usa el parámetro *countsFromAbundance*, normalizando las cuentas de los tránscritos. De esta manera, la normalización se hace conjunta con todas las muestras que se quieren analizar, aunque en los contrastes no se utilicen todas ellas. Es una manera de poder comparar los resultados entre las muestras aun analizándolas en momentos distintos.

En esta función sólo es necesario pasarle la lista obtenida por *importQuantData()* y la lista del nombre de las muestras que debe coincidir con el nombre que contienen las columnas de las distintas matrices de la lista.

```
selectDataFromTximport(tximport, sample_names)
```

Se obtiene una lista de matrices, con el mismo contenido que la obtenida por la función anterior, con las muestras de las que se quiere hacer el estudio de expresión diferencial.

3.1.4. Función getFilteredDDS()

Debido a la necesidad de eliminar el ruido de fondo y los genes que no se habían expresado, se construyó la función *getFilteredDDS(),* que contiene la función *DESeqFromTximport()*. Además, esta función permite filtrar los genes que tienen cuentas bajas a lo largo de las muestras así como importar las cuentas de la lista generada por *selectDataFromTximport() o importQuantData()*. Para ello, se deben utilizar los parámetros *filter_min_reads* y *filter_min_samples*. Además, se añade la información de las muestras (*samples_df*) en el que se encuentran las variables que detallan los contrastes que se quieren realizar en el análisis de expresión diferencial.

```
getFilteredDDS(tximport, samples_group, samples_df, filter_min_reads = 5,
filter_min_samples = 1, test = "Wald", interaction = FALSE, verbose = TRUE)
```

En el parámetro *samples_group* se debe especificar el nombre de las variables, o condiciones, de *samples_df* que se quieren contrastar. Otros parámetros importantes que se tienen que tener en cuenta son *test* e *interaction*. El parámetro *test* indica el tipo de prueba que se quiere llevar a cabo, determinando el tipo de contraste que se quiere hacer. Puede tomar dos valores, “Wald” o “LRT”. Si el tipo de prueba es “Wald”, el parámetro *samples_group* solo puede adquirir un valor, mientras que si es “LRT” puede adquirir dos.

Una de las razones por las que se desarrolló esta función es para que el usuario no tuviera que pensar en cómo construir la fórmula del diseño que precisa la herramienta *DESeq2* para realizar el contraste y obtener los DEG. En el caso de “Wald” el diseño es sencillo, ya que sólo se puede pasar una única condición con los grupos correspondientes. Sin embargo, cuando se trata del test “LRT” y se estudia más de una condición, el diseño experimental puede variar. Si se quiere estudiar la interacción de dos condiciones, esta función tiene la capacidad de, pasándole el nombre de las variables en *samples_group*, y poniendo *TRUE* en el parámetro *interaction,* la función genera el diseño que el usuario precisa para realizar el estudio de expresión diferencial.

El objeto de salida de esta función es del tipo *DESeqDataSet* el cual contiene información de las cuentas y el *data frame* con la información de las muestras, entre otras.

En conclusión, esta función facilita al usuario conseguir el filtrado de las muestras así como la fórmula necesaria para la realización de los contrastes, indicando la información de lo que se quiere analizar y dando el archivo *tximport* generado en la función anterior.

3.1.5. Función runDESeq()

La función *runDESeq()* es la función encargada de realizar el análisis diferencial de los datos y obtener los genes diferencialmente expresados. Es considerada la función clave del análisis.

Esta función se generó con el fin de facilitarle al máximo al usuario la obtención de los genes diferencialmente expresados, sin tener que pensar en cómo generar el modelo reducido de la formula diseño si se usa el test “LRT” y con el fin de poder adquirir todos los contrastes realizados por *DESeq2* de una manera más automática, ya que *runDESeq()* abarca en una única función los pasos que el paquete *DESeq2* lleva a cabo con las funciones *DESeq(), results() y lfcshrink().* Es decir, estima los parámetros y obtiene los resultados en una única función.

runDESeq() es capaz de conseguir los DEG especificando el nombre de las muestras (*sample_names*) que se quieren comparar, el nivel de referencia, o muestra control, del contraste (*ref_group*), el *DESeqDataSet* obtenido previamente por la función *getFilteredDDS() (filtered_dds)*, el *data frame* que contiene la información de las muestras(*samples_df*) y del contraste que se quiere realizar, así como el nombre de la variable, dentro del *data frame,* que contiene el nombre de las muestras(*sname_variable*).

```
runDESeq (sample_names, ref_group, filtered_dds, samples_df, sname_variable,
type_lfcShrink = "normal", maxpvalue = 0.05, maxlfc = 0, order_by = "padj", test =
"Wald", studied_condition = NULL, interaction = FALSE,  verbose = TRUE, ...)
```

Si se ha especificado en el parámetro *test* la prueba “LRT”, se tiene que tener en cuenta que para realizar el contraste, es necesario generar un modelo

reducido de la fórmula diseño establecida en *getFilteredDDS().* Con esta prueba, se consiguen como resultado aquellos genes diferencialmente expresados a lo largo de todo el proceso, comparando el diseño de la fórmula establecida con el modelo reducido. Debido a esta comparación general, los p-valores de los genes son los mismos en todos los contrastes realizados. Lo que varía es su *log2FoldChange* dependiendo de los distintos contrastes. Este tipo de análisis es útil en un estudio de expresión diferencial a lo largo del tiempo, y no entre los distintos puntos del tiempo, como se requiere para el análisis de los datos usados para este proyecto, tal y como se ha comentado anteriormente en el apartado de materiales y métodos.

Si el usuario quiere analizar la interacción (*interaction = TRUE)* de dos condiciones especificadas previamente indicadas en *getFilteredDDS()*, *runDESeq()* se encarga de generar el modelo reducido correspondiente para que se obtengan los DEG. Si se quieren analizar dos condiciones con "LRT" pero sin interacción, se debe usar el parámetro *studied_condition* en el que se tiene que dar el nombre de la condición de la cual se quieren obtener los resultados. Por ejemplo, si se están estudiando las condiciones genotipo y tiempo sin interacción y, se quieren obtener los genes diferencialmente expresados en el tiempo en los distintos genotipos, *studied_condition* deberá ser igual a tiempo.

Con el test de "Wald" no es necesario generar ningún tipo de reducción de la fórmula de diseño, además, el parámetro *interaction* no tiene efecto en esta prueba.

Con esta fórmula también se utiliza la función *lfcShrink()* para así eliminar los falsos positivos y poder generar un ranking de los genes de una manera más precisa.

runDESeq() permite realizar dos tipos de *lfcShrink:* el "normal", que se encuentra establecido por defecto y *"apeglm"*. Si el número de muestras y réplicas es pequeño, el "*normal*" es el que se recomienda. Si por el contrario, se tiene un gran número de muestras, se recomienda el *lfcShrink "apeglm".* Este tipo de *lfcShrink* utiliza una distribución previa *heavy-tailed Cauchy* para estimar el efecto de los tamaños, dando como resultado menos sesgos que el

normal y consiguiendo reducir la varianza al igual que el método establecido por defecto.

Como resultado se obtiene un objeto *DESeqResults* que contiene los DEG ordenados de menor p-valor a mayor o por su *log2FoldChange*, según el valor establecido por el usuario en el parámetro *ordered_by.* Además, con *maxpvalue* puedes indicar cuál es el nivel de significación para la obtención de los genes diferencialmente expresados (por defecto es de 0.05).

Un ejemplo del tipo de salida que se obtiene se encuentra en la figura 15. Los valores que se muestran en las distintas columnas son:

- *baseMean:* representa la media de las cuentas normalizadas en todas las muestras
- *log2FoldChanges:* la estimación del tamaño del efecto. Explica cuanto ha cambiado la expresión de un gen en las distintas condiciones contrastadas (ej. Tratameinto vs control).
- *lfcSE:* el error estándar de *log2FoldChange*
- *stat:* el estadístico de Wald o LRT
- *p-value:* p-valor obtenido
- *p-adj:* p-valor ajustado por BH FDR.

3.1.6. Función getTopGenes()

La función *getTopGenes()* se construyó con el fin de seleccionar de manera rápida y sencilla aquellos genes que estuvieran diferencialmente expresados, según el p-valor ajustado dado por el usuario y los resultados obtenidos a partir de la función anterior.

```
getTopGenesLFC(deseq_results, padj_value)
```

3.1.7. Función getGOTermsfromDEG()

La función *getGOTermsfromDEG()* se basa en el *GSEA* realizado por las funciones del paquete *GOseq* cuya información está detallada en su viñeta[37]. El motivo de la implementación de esta función con el uso de este paquetes se hizo con el fin de condensar las funciones de *GOseq* en una sola, y conseguir los *GO terms* significativos con los genes diferencialmente expresados obtenidos de la función anterior, así como una lista de los genes que contienen cada *GO term.*

```
getGOTermsfromDEG(deseq_results, ref_genome, id_type, updown, base_filename,
base_directory, split_files = FALSE, maxpadj_value = 0.05, minlfc_value = 0, verbose
= TRUE, ...)
```

Esta función genera distintos archivos que contienen los *GO terms* significativos. El usuario debe proporcional el objeto *DESeqResults* en el parámetro *deseq_results*. Con el fin de seleccionar los genes diferencialmente expresados, se tiene que añadir el *maxpadj_value* (por defecto 0.05). Además se puede indicar el *minlfc_value,* que corresponde al mínimo valor de LFC por el que se quieren conseguir los DEG. Así mismo, debe ser especificado el genoma de referencia (*ref_genome)*, el tipo de identificador de los genes (*id_type)*.

La información obtenida se clasifica según su función biológica (*Molecular Function (MF), Cell Component (CC)* o *Biological Process (BP))* de manera conjunta o separada, indicando *TRUE* o *FALSE* en el parámetro *split_files*. Otra característica a destacar de *getGOTermsfromDEG()* es la posibilidad de conseguir el GSEA de todos los genes diferencialmente expresados("*all"*), sólo los que se encuentras sobre expresados(*"up")*, o aquellos que se encuentran infra expresados ("*down*"), especificándolo en el parámetro *updown.*

Como resultado de la ejecución, se obtienen distintos archivos. El primero de ellos contiene los *GO terms* y los genes que se encuentran asociados a ese término. El segundo guarda la información estadística de cada *GO term*, mientras que el último contiene la información de qué significa cada *GO term* obtenido, ordenados de más a menos significativo. Estos archivos se guardan en el directorio que se especifique en el parámetro *base_directory* y con el

nombre que se introduzca en *base_filename*. Cabe señalar que, si s*plit_files* es *TRUE*, en el directorio que se haya establecido, se crean tres carpetas (*MF, CC, BP*) donde se clasifican los ficheros generados según su función biológica.

3.1.8. Función getKEGGpathways()

La función *getKEGGpathways()* es encargada de hacer un GSEA y conseguir aquellas vías que están significativamente representadas según los genes diferencialmente expresados obtenidos. Esta función agrupa los distintos métodos del paquete *gage* y *pathview* para conseguir un listado de las vías más representadas así como una representación gráfica de estas usando la base de datos *KEGG*.

```
getKEGGpathways(deseq_results, base_filename, base_directory, statistic = "q.val",
pvalue = 0.05, verbose = TRUE)
```

Esta función necesita el objeto *DESeqResults*, el valor de sigunificacion por el que se quieren obtener las vías significativas (*pvalue)* y el estadístico por el que se quiere obtener este valor (*statistic)* que por defecto es *q.value,* ya que con este valor eliminamos los falsos positivos de las vías representadas significativamente.

Los resultados de esta función son diferentes archivos *png* con las vías significativamente afectadas por los DEG. Estos archivos se guarda en el directorio establecido por el usuario en *base_directory* con el nombre que se quiera dado en *base_filename.*

En la figura 6, se muestra un ejemplo del tipo de imágenes que se obtienen, donde se ven los genes característicos de una vía en concreto que se encuentran expresados o infra expresados.

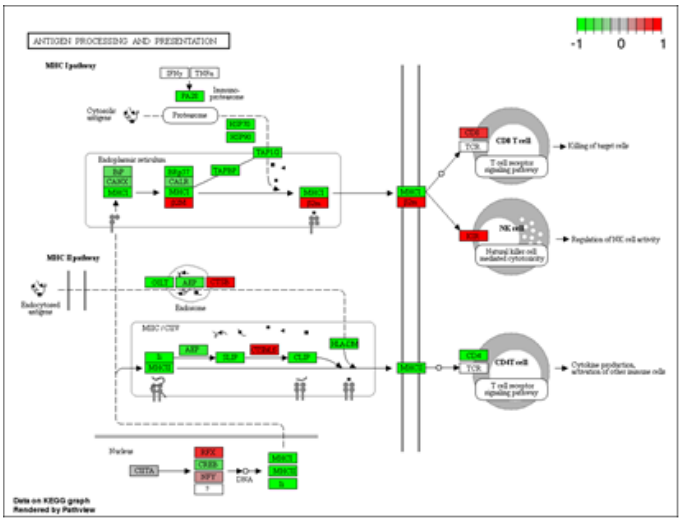


Figura 6. Ejemplo de una vía significativamente representada según el repertorio de genes diferencialmente expresados que hayan sido obtenidos. En rojo se representan los genes que están sobre expresados y que pertenecen a esta vía, mientras que el color verde corresponde a los genes que se encuentran infra expresados en esta vía.

3.1.9. Función getGroupSpecificGenes()

Esta función se generó para conseguir aquellos genes diferencialmente expresados comunes a un estadio celular, por ejemplo, determinar qué genes se encuentran diferenciados a día siete, en los diferentes contrastes realizados (Tabla3).

```
getGroupSpecificGenes(select_genes, query_group, updown_markers)
```

Para obtener los genes comunes con *getGroupSpecificGenes()* se debe introducir una lista de *data frames* que contengan los genes diferencialmente expresados de todos los contrastes realizados, como el que se llevó a cabo en el análisis de los datos del estudio de diferenciación de las células de Schwann.

Esta función es encargada de seleccionar los genes específicos del grupo de interés, introducido en el parámetro *query_group*. Además tiene la capacidad de conseguir por separado los genes sobre expresados en el grupo de interés, los infra expresados, o todos juntos, especificando en *updown_markers* entre "*up*", "*down*", o "*all*".

Como resultado da un vector de caracteres con aquellos genes diferencialmente expresados específicos, por ejemplo, de día siete.

Esta función es útil para conseguir posibles marcadores específicos de cada estadio de diferenciación de SC.

3.1.10. Función getBioMartGOAnnotation()

La función *getBioMartGOAnnotaiton()* proporciona una anotación de los términos de *Gene Ontology,* dando información de código *GO,* su significado y función biológica. Esta función utiliza la anotación de BioMart a través del paquete *biomaRt* [22]*,* para conseguir la información para anotar los genes.

```
getBioMartGOAnnotation (genes, gene_id = 'hgnc_symbol', ref_genome =
hsapiens_gene_ensembl" ,  verbose = TRUE )
```

Para conseguir la anotación de los *GO terms* es necesario especificar los genes que se quieren anotar (*genes)*, especificar el identificador de tus genes según *ensembl.* Para conocer qué tipo de identificación tiene los genes se puede usar la función *listFilter()* del paquete *biomaRt*. Por defecto es *"hgnc_symbol"*. También se tiene que especificar el genoma de referencia que, mediante la función *listDataSet()* del mismo paquete puede obtenerse (por defecto está establecido *"hsapiens_gene_ensembl"*.

Como resultado se obtiene una lista de *data frames* que clasifican los *GO terms* dependiendo de la información biológica que sea (MF, CC, BP).

Con esta función, por ejemplo, se puede conseguir la anotación de los genes específicos de cada estadio, pudiendo conseguir genes expresados en la membrana celular, que podría servir como posibles marcadores de un estadio de diferenciación concreto.

3.1.11. Función MAplot() para el control de calidad.

Esta función es útil para mirar en un primer momento el comportamiento de los datos, sobre todo, para comprobar si dos réplicas se comportan de manera similar.

```
MAplot (pseudocounts, dataframe_position, base_filename)
```

La función *MAplot()* se construyó con el fin de hacer su representación de una manera más automática. Genera unos gráficos a partir de la trasformación

logarítmica des cuentas *(*parámetro *pseudocounts)* y la posición que tienen las muestras que quieren ser analizadas en el *data frame* que contenga la información de las muestras (*dataframe_position)*. Los gráficos representan en el eje de las x la media del logaritmo de las cuentas, denominados como *A-values*, y en el eje de la y, el valor promedio de expresión de un gene entre las réplicas (M).

En la figura 7 vemos una representación del tipo de gráfico que genera esta función.

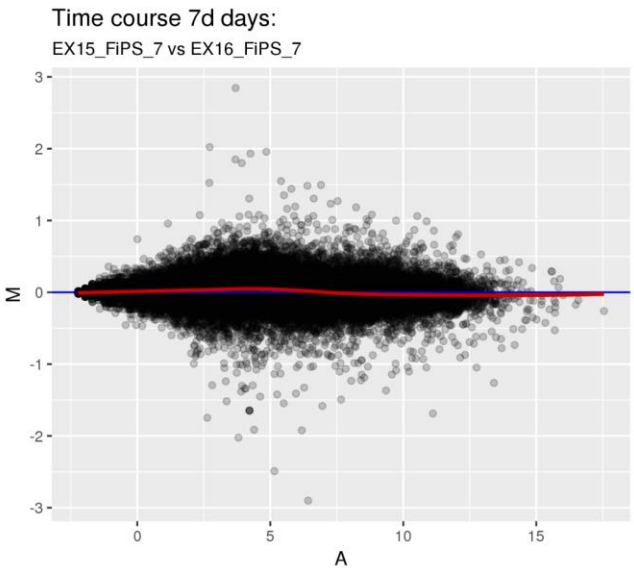


Figura 7. MA-plot representando las muestras del experimento 15 y 16 a día 7 de las FiPS. En el eje de las x se representa la media del logaritmo de las cuentas, denominados como A-values, mientras que en el eje de las y se representa el valor promedio de expresión de un gen en las réplicas (M).

3.1.12. Función plotDEGMA()

plotDEGMA() es una función encargada de representar los genes diferencialmente expresados dentro del conjunto de genes estudiados tras el uso de la función *runDESeq().*

Contiene la función *plotMA()* de *DESeq2.* El motivo de esta función fue para conseguir que el usuario pudiera obtener un MA-plot con los genes más expresados indicando el mínimo de parámetros posibles.

El objeto producido por la función *runDESeq()* se introduce en el parámetro *deseq_results.* En el eje de abscisas se representa la media de expresión de

un gen sobre todas las muestras, mientras que en el de ordenadas se representa el LFC obtenido de ese gen entre las dos condiciones que se hayan estudiado (ej. tratamiento y control). Cada gen se representa con un punto, siendo los que se encuentran de color rojo los diferencialmente expresados, según el p-valor establecido por el usuario.

```
plotDEGMA(deseq_results, selected_genes, number_genes, title, ...)
```

Esta función también da la oportunidad de marcar con un círculo azul y el nombre de los genes, aquellos que se encuentren más diferencialmente expresados. Para conseguir este resultado, el usuario debe introducir el *data frame* generado tras la selección de los genes con la función *getTopGenesLFC().* Además, puede decidir cuantos quiere remarcar con el uso del parámetro *number_genes.*

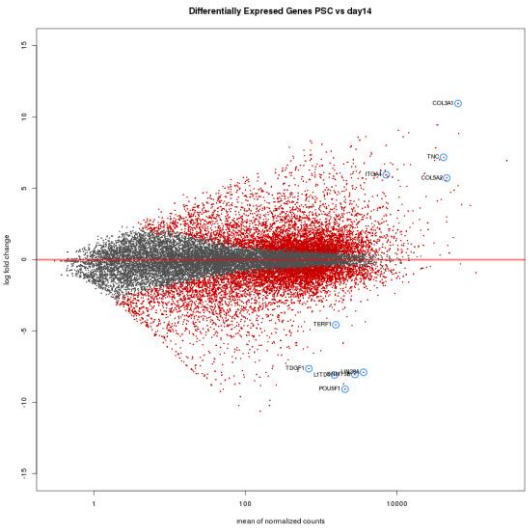


Figura 8. MA-plot para ver aquellos genes que se encuentran diferencialmente expresados. Cada punto es un gen. En el eje de abcisas se representan la expresión media de un gen en las distintas muestras, mientras que en el de ordenadas se representan los valores LFC de los genes. En rojo se representan los genes que están diferencialmente expresados. Rodeados en azul y con el nombre del gene, se resaltan aquellos que están más expresados.

En la figura 8, se puede ver un ejemplo del estilo de gráfico que genera esta función para observar los genes diferencialmente expresados de un contraste entre dos grupos muestras.

3.1.13. Función get*Heatmap*PlotDEG()

Con esta función se representan y se realiza un *clustering* de los DEG según su expresión y de las muestras según la similitud de la expresión de los genes representados. En el eje de abscisas se representan las muestras, siendo cada columna una muestra específica, mientras que en el eje de ordenadas se representan los genes, siendo cada fila un gen específico. Mediante los colores rojo, azul y blanco, representa la expresión de los distintos genes en las muestras. El color rojo es indicativo de los genes sobre expresados en una muestra concreta, mientras que el color azul hace referencia de los infra expresados y el color blanco representaría una expresión estándar, con un LFC de valor 0.

```
getHeatmapPlotDEG(dds, deg_genes, num_genes, margins = c(20, 20),...)
```

Para conseguir esta representación gráfica, se necesita el objeto *DESeqDataSet* (se pasa por el parámetro *dds)*, que se tiene que conseguir con la función *DESeq()* del paquete *DESeq2,* al que se le tiene que pasar el objeto obtenido de la función *getFilteredDDS().* Los genes que se quieren representar se pueden pasar en forma de vector de caracteres, o pasando un *data frame* donde el nombre de los genes se encuentre en las filas del objeto. Además da la posibilidad de representar un número de genes específico del total de los genes que contiene el objeto *DESeqDataSet.*

getHetmapPlotDEG() se encarga de realizar una transformación logarítmica de las cuentas que contiene el objeto pasado por el parámetro *dds* y, posteriormente, hace un *subset* de los genes que se encuentran en este objeto para así representar los genes que el usuario precisa.

Los ejemplos de este tipo de gráficos se muestran en el apartado de análisis de los datos.

3.1.14. Funciones para representar la expresión de los genes en las distintas muestras.

Además de conseguir aquellos genes que se encuentran diferencialmente expresados, las investigadoras también estaba interesado en observar la expresión de genes específicos. Es por ello que se construyeron las siguientes funciones para representar la expresión de los genes en los distintos puntos en el tiempo.

3.1.14.1. Función generationGeneMatrix()

Esta función da una matriz con la información de la expresión de un gene en las distintas muestras del estudio.

```
generationGeneMatrix (dds, gene,  sname_variable, contrast_group, normalize = TRUE)
```

Es la primera función para conseguir la representación de un gene en concreto en distintos puntos en el tiempo. La matriz que genera tienen como nombre de filas, el nombre de las muestras y, los puntos en el tiempo que se quieran observar, como nombre de columnas.

Tabla 5. Estructura de la matriz que se genera tras la ejecución de la función *generationGeneMatrix()* correspondiente a la expresión de *ADGRG6* en los distintos puntos en el tiempo (-1: iPSC, 0:NC, 7: día7, 14:día14, 30:día30).

	-1	*0*	*7*	*14*	*30*
FiPS_PSC	***181***	***NA***	***NA***	***NA***	***NA***
EX15_FiPS_NC	***NA***	***1483***	***NA***	***NA***	***NA***
EX15_FiPS_7	***NA***	***NA***	***5566***	***NA***	***NA***
EX15_FiPS_14	***NA***	***NA***	***NA***	***2941***	***NA***
EX15_FiPS_30	***NA***	***NA***	***NA***	***NA***	***5811***
EX16_FiPS_NC	***NA***	***2541***	***NA***	***NA***	***NA***
EX16_FiPS_7	***NA***	***NA***	***3276***	***NA***	***NA***
EX16_FiPS_14	***NA***	***NA***	***NA***	***1873***	***NA***
EX16_FiPS_30	***NA***	***NA***	***NA***	***NA***	***3794***
EX18_FiPS_7	***NA***	***NA***	***2140***	***NA***	***NA***
EX18_FiPS_14	***NA***	***NA***	***NA***	***1929***	***NA***
EX18_FiPS_30	***NA***	***NA***	***NA***	***NA***	***4784***

3.1.14.2. Función plotLimsAndLabels()

Esta es la segunda función para conseguir representar la expresión de un gen en concreto. *plotLimsAndLabels()* es la encargada de generar un gráfico vacío pero que contenga el límite del eje de abscisas y ordenadas así como el título del gráfico. Para ello, se le tiene que pasar la matriz generada en la función anterior (*gene_matrix)* y el nombre del gen para que construya el título del gráfico. Esta función se llama con el fin de usar la función *plotPointsAndMeans()* que será la encargada de aportar la información de la expresión.

```
plotLimsAndLabels(gene_matrix, gene, xlab = "Days", ylab = "Expression level(TPM)" ,
title = "Expression", ...)
```

El eje de ordenadas representa la expresión en TPMs mientras que el eje de abscisas representa los distintos puntos en el tiempo.

3.1.14.3. Función plotPointsAndMeans()

Esta es la última función a la que hay que llamar para conseguir la representación de la expresión de los datos. Aunque para ello, se debe haber representado previamente el gráfico en blanco con la función *plotLimsAndLabels()*.

```
plotPointsAndMeans(gene_matrix, legend_names, legend_title, legend_color,legend_locus,
legend_size = 0.5, points_col = "black", means_col = "black", line_col = "black",
plot_points= TRUE, plot_means=TRUE, plot_line=TRUE, means_length = 0.25, pch = 16,
col = "black", plot_legend = TRUE,...)
```

Para representar la información es necesario la matriz generada con la función *generationGeneMatrix()*, que contiene la información de la expresión de un gen específico en las muestras.

plotPointsAndMeans() es encargada de representar la media de expresión de las muestras que se encuentran en un mismo punto en el tiempo en forma de líneas horizontales. El usuario puede decidir si estas líneas se representan (*plot_means),* su longitud (*means_length)* y su color (col = “black”). También representa, en forma de puntos, la expresión específica de cada muestra. A los puntos se les puede especificar el color (*points_col)* así como también se

puede decidir si se representan o no (*plot_points).* Además, puede representar una línea que una las medias y, de esta forma, genere una idea del perfil de expresión de un gen en los distintos puntos. Este parámetro también es opcional y el usuario puede decidir si representarlo o no (*plot_line),* y su color *(line_col).* Si se quieren representar distintas muestras con distintos colores, se puede añadir una leyenda (*plot_legend)*, y su localización en el gráfico (*legend_locus)* y su tamaño (*legend_size)*. La leyenda se tiene que generar "manualmente", siendo el usuario el encargado de añadir el nombre de las muestras que se representan en el gráfico *(legend_names),* el título que tiene (*legend_title)* y el color que tienen cada una de las muestras *(legend_color*).

A continuación se muestra un ejemplo de cómo sería un gráfico generado por estas funciones.

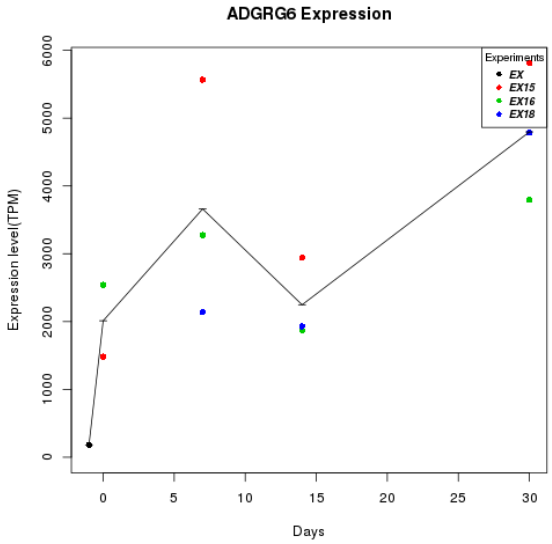


Figura 9. Representación de la expresión el gen *ADGRG6* en FiPS en los distintos puntos del tiempo.

3.1.15. Funciones para conocer los genes con expresión similar a uno dado.

Otro interés que había por parte de las investigadoras era conocer qué genes se expresaban de la misma forma o de manera contraria a uno dado. Para ello se generaron una serie de funciones con las que conseguir una matriz correlación de la expresión de todos los genes y, a partir de ahí, obtener los genes que tuvieran una expresión similar basándose en su correlación de expresión.

3.1.15.1. Función getGeneCorMatrix()

Esta función es la encargada de generar la matriz de correlación de la expresión de unos genes dados entre todas las muestras. Para conseguirla, se obtiene la expresión media entre las réplicas. Con la media de la expresión de las réplicas calculadas, se realiza la correlación de Pearson para conseguir la correlación de la expresión.

```
getGeneCorMatrix (dds, selected_genes, sample_names, sample_group_levels,
samples_group, samples_df, sname_variable, base_directory, base_filename,
verbose = TRUE, save = TRUE)
```

Para conseguir la matriz de correlación se necesita el objeto *DESeqDataSet* (*dds)* los genes de los que se quiera obtener la correlación (*selected_genes)*, el nombre de las muestras *(sample_names)*, el nombre de los niveles *(sample_group_level)* de la condición (s*ample_group)*. El *data frame* donde está la información con los datos de las muestras *(samples_df)*, el nombre de la variable donde se encuentra el nombre de las muestras en el *data frame (sname_varialble)* y, el directorio y el nombre en el que se quiere guardar la matriz de correlación.

Se tiene que tener en cuenta que según el número de genes de los que se quiere conseguir la matriz de correlación, el proceso de obtención será costoso y la matriz que se obtiene es pesada. Es por ello por lo que la matriz se guarda en un formato *.RData*.

3.1.15.2. Función getCorrelatedGenes()

Una vez tenemos la matriz de correlación de expresión generada, con la función *getCorrelatedGenes()* se puede conseguir los genes que se expresan de manera similar o contraria a un gen dado.

```
getCorrelatedGenes(gene, cor_matrix,  corlevel = 0.9)
```

Se debe añadir la matriz de correlación obtenida con la función anterior (*cor_matrix*), el gen del que se quieren conocer los genes de expresión similar a él (*gene*), y el nivel de correlación (*corlevel*) que por defecto es 0.9.

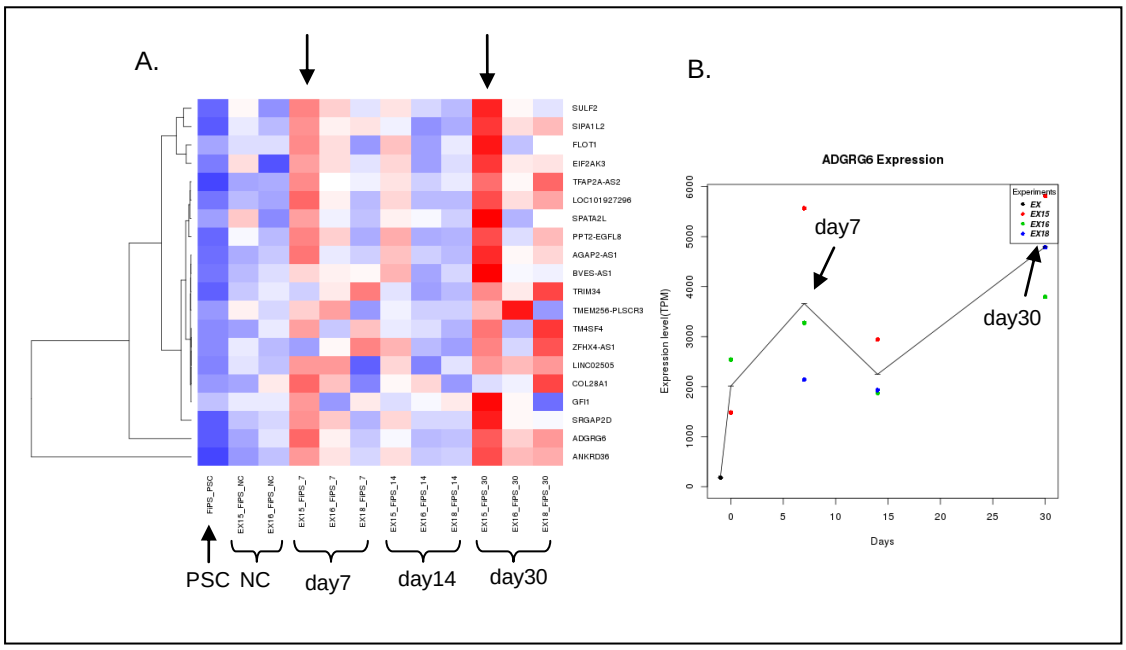


Figura 10. Ejemplo del tipo de resultado que se podría obtener tras obtener los genes con expresión similar a un gen en concreto. En este caso, el gen elegido es *ADGRG6*. A. *Heatmap: gráfico de la izquierda.* B. Gráfico de expresión

En la figura 10 se muestra un ejemplo del tipo de resultado que se podría obtener tras conseguir la lista de los genes que tienen una expresión más similar a uno en concreto a lo largo del tiempo. En este caso, los resultados corresponden a los 20 genes con una expresión más similar a la expresión de *ADGR6*. Los picos del gráfico de expresión (figura 10B) corresponden a aquellos genes que presentan un color rojo en la figura 10A (marcado con una flecha). Cabe destacar que el experimento 15 (EXP15), representado en color rojo en la figura 10B, presenta mayor expresión que el resto de experimentos, tal y como se observa en el *heatmap (*marcadas con la flecha).

3.2. Análisis de los datos.

Para verificar el correcto funcionamiento de los métodos explicados en el apartado anterior, se analizaron los datos procedentes de las células control (FiPS) del experimento de diferenciación celular, desde iPSC hacia células de Schwann. Con el análisis de estos datos se pretendía responder a las preguntas relacionadas con la expresión de las células FiPS en los distintos puntos del tiempo con el fin de descubrir nuevos marcadores específicos de cada estadio de diferenciación.

Para responder a las preguntas relacionadas con estas muestras, se construyó un proceso de análisis cuyo código se adjuntará con la memoria final (Anexo 1). Se trata de un ejemplo del método que se puede realizar para analizar unos datos procedentes de un proceso de diferenciación celular a lo largo del tiempo, utilizando las funciones generadas comentadas en el apartado anterior.

3.2.1. Importación de los datos

La importación con *importQuantData()* se hizo del conjunto de las muestras, para conseguir la normalización, por *lengthScaledTPM,* del todas ellas. De esta manera, los resultados obtenidos de los distintos contrastes se pueden comparar. Tras la importación de los datos, se hizo una selección de las muestras procedentes de las FiPS con la función selectFromTximport().

3.2.2. Control de calidad

Se realizó un control de calidad previo para observar el comportamiento de las muestras en su conjunto. Las muestras necesitaban ser filtradas para eliminar aquellos genes que no se encontraban expresados o, presentaban bajo número de cuentas en la mayoría de las muestras estudiadas (que en todas muestras menos en una, encontráramos un valor menor que el indicado en *filter_min_sample),* usando la función *getFilteredDDS().*

Con esta función se eliminaron los genes que tenían una expresión menor de 5 *read counts* en menos de una muestra del total analizadas. Una vez conseguido el objeto *DESeqDataSet,* se realizó la función *DESeq()* del paquete *DESeq2* para estudiar el comportamiento de las muestras mediante *heatmaps*, graficos de PCA y dendrogramas. Se realizaron dos controles de calidad. El primero de ellos se realizó con todas las muestras y, el segundo, únicamente con las muestras procedentes de las células FiPS.

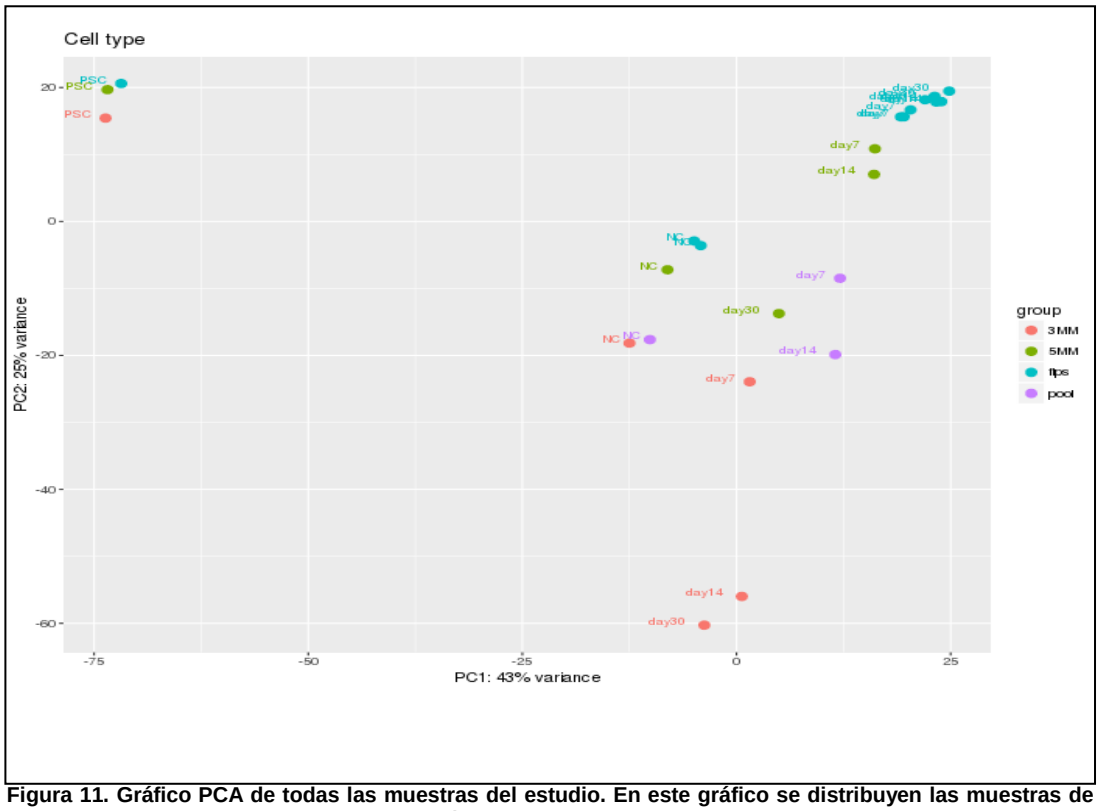


Figura 11. Gráfico PCA de todas las muestras del estudio. En este gráfico se distribuyen las muestras de acuerdo a varianza que presenta la expresión entre ellas. La varianza se ve representada por medio de las componentes principales que se encuentran en el eje x (PC1: 43%) y el eje y(PC2:25%). Antes de representarlas, se realizó la transformación logarítmica de las cuentas de las distintas muestras, para poder tener en cuenta las pequeñas diferencias entre ellas. 3MM: muestras *NF1*$^{-/-}$ procedente de un PNF del paciente 3 del cual se hizo la reprogramación a iPSC , y a partir del cual se realizó el estudio de diferenciación en distintos puntos en el tiempo. 5MM: muestras *NF1*$^{-/-}$ procedente de un PNF del paciente 5 del cual se hizo la reprogramación a iPSC, y a partir del cual se realizó el estudio de diferenciación en distintos puntos en el tiempo. FiPS: muestras de las células control (*NF1*$^{+/+}$), procedentes de fibroblastos a partir de los cuales se realizó la reprogramación a iPSC, y a partir del cual se llevó a cabo el estudio de diferenciación en distintos puntos en el tiempo. pool: muestras que contienen RNA-seq de las muestras 3MM y 5MM en los distintos puntos en el tiempo (cresta neural(NC), siete días (day7), catorce días (day14)). PSC hace referencias a las células iPSC.

En la figura 11 se muestra un gráfico de componentes principales. En este tipo de gráficos se puede comprobar cómo se distribuyen las muestras de acuerdo a varianza en su expresión. La varianza se ve representada por medio de las componentes principales que se encuentran en el eje x (PC1: 43%) y el eje y (PC2:25%). Antes de representarlas, se realizó la transformación logarítmica de las cuentas de las distintas muestras, para poder tener en cuenta las pequeñas diferencias de expresión entre ellas. La distribución de las muestras era la esperada. El componente principal uno (PC1) parece explicar la varianza entre los distintos estadios celulares, llegando a explicar un 43% del total. Se puede ver como en la esquina superior izquierda se sitúan las iPSC, bajo el nombre de PSC, de todas las muestras analizadas, tanto las $NF1^{+/+}$ como las procedentes de PNFs con un genotipo $NF1^{-/-}$. Lo mismo ocurre con el estadio de cresta neural (NC) donde se puede ver que todas las células, sin tener en cuenta el genotipo, se distribuyen por la zona central. También, se puede intuir que, a medida que el estudio de diferenciación avanza, las células $NF1^{-/-}$ presentan un proceso de diferenciación más irregular. Esto no se debe confundir con una mala calidad de las muestras sino que puede ser debido a que las células de las que proceden estos datos, no tienen neurofibromina y es posible que, su diferenciación irregular se debiera a esto.

Cabe destacar que las células control (FiPS), se agrupan en la esquina derecha del gráfico una vez que el proceso de diferenciación pasa del estadio de NC, pero siguen mostrando diferencias con las muestras procedentes de PNF (3MM, 5MM y pool).

El comportamiento de las muestras también comprobó por medio de dendrogramas(figura 12). Este tipo de representación permite apreciar claramente las relaciones de agrupación entre los datos aplicando algoritmos de *clustering*. Observando las sucesivas subdivisiones, se puede apreciar la distancia entre los datos según las relaciones establecidas.

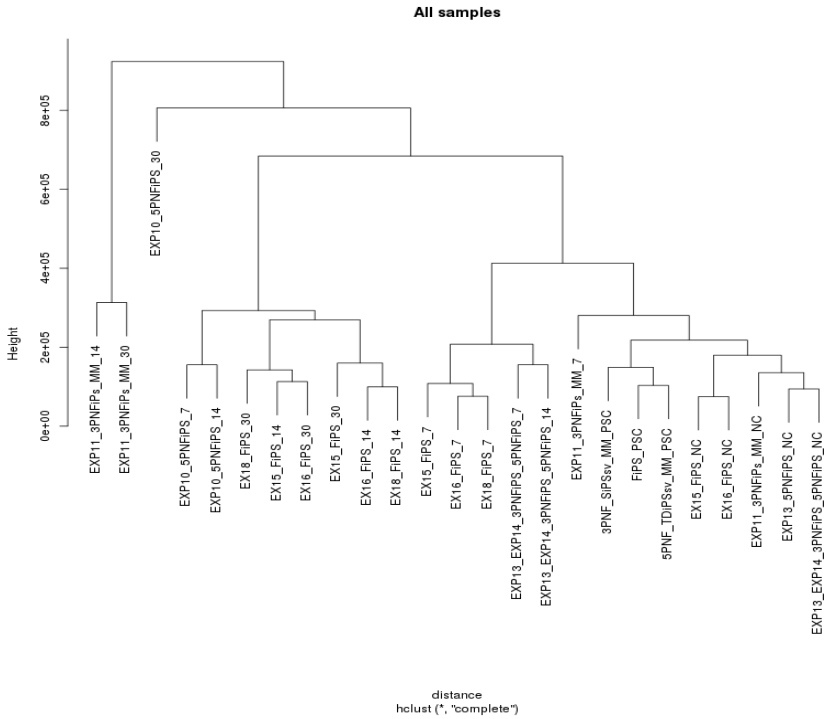


Figura 12. Dendrograma del conjunto de las muestras. Se representa de manera jerárquica el conjunto de las muestras usando la distancia Euclidiana. En este dendrograma se asocian las muestras sin realizar una transformación logarítmica de la expresión de sus tránscritos. Por lo tanto, se tienen en cuenta las grandes diferencias entre las distintas muestras a la hora de la agrupación. Para anotar a cada una de las muestras se utilizó el nombre que se les atribuyó. Se encuentra detallado en la tabla ajo el nombre de *sample.names.* Exp: experimento, 5PNF: tumor plexiforme 5, 3PNF: tumor plexiforme 3, FiPS: células procedentes de fibroblastos, PSC*: pluripotent stem cells,* NC: células de la cresta neural (*Neural Crest*), MM: células con genotipo *NF1*$^{-/-}$, 7;14;30: días del proceso de diferenciación.

Mediante la agrupación de las muestras con el uso del dendrograma, tampoco se obtuvo como resultado ninguna asociación atípica. Sin embargo, se pudo ver que, la muestra correspondiente con el experimento 10 del tumor 5PNF a los 30 días de diferenciación (*EXP10_5PNFiPS_30)* presentaba un perfil transcriptómico distinto al resto de las muestras.

Con el uso de los *heatmap*s se representó de una manera más visual las distancias entre las muestras. Los colores que se observan en este gráfico no es una escala real de asociación, únicamente ayudan al usuario a observar cómo se relacionan las muestras. En la figura13 se muestra el comportamiento del conjunto de las muestras. Cabe señalar que, en este tipo de gráficos se realiza la transformación logarítmica de las muestras para conseguir que se tenga en cuenta las pequeñas diferencias entre las muestras.

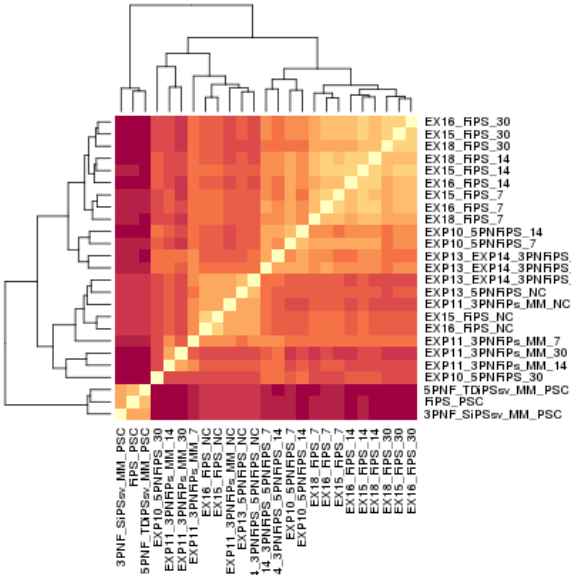


Figura 13. *Heatmap* representativo de la distancia entre las muestras. El nombre de las muestras se encuentra detallado en la tabla 4. El color que se observa en este gráfico explica la asociación entre las distintas muestras. Exp: experimento, 5PNF: tumor plexiforme 5, 3PNF: tumor plexiforme 3, FiPS: células procedentes de fibroblastos, PSC*: *pluripotent stem cells,* NC: células de la cresta neural (*Neural Crest*), MM: células con genotipo *NF1*$^{-/-}$, 7;14;30: días del proceso de diferenciación.

Con este gráfico se pudo comprobar que las muestras también se comportaban de la manera esperada, por lo que se procedió a centrar el análisis en las muestras procedentes de las células FiPS para encontrar los DEG entre los distintos puntos en el tiempo.

Los gráficos correspondientes al segundo control de calidad centrado en las muestras de las FiPS, se encuentran en la figura14. Mediante los gráficos de componentes principales, dendrograma y *heatmap*, se pudo concluir que no había presencia de muestras que pudieran distorsionar el estudio de expresión diferencial.

En la figura14 A., referente a la representación de los componentes principales, se observa una distribución más espaciada en lo que respecta al PC1, comparado con los que se observaba en la figura 11. En este gráfico la componente uno llega a explicar un 63% de la varianza. Este componente haría referencia a las diferencias de expresión que se observarían por estadio celular. En el dendrograma se observa cierta discrepancia en el orden de la asociación de la muestra referente al experimento 15, tanto a 14 como 30 días. Sin embargo, cuando observamos la asociación de las muestras por medio del

heatmap, esta discrepancia se desvanece. Esto es debido a que, el *heatmap* tiene en cuenta las pequeñas diferencias entre las muestras debido a la transformación logarítmica, mientras que en el dendrograma, al no llevarse a cabo dicha transformación, solo se tienen en cuenta las grandes diferencias de expresión y, por tanto, la manera de agruparse es distinta. No obstante, en ninguno de los gráficos de control de calidad se apreciaron posibles *outliers.*

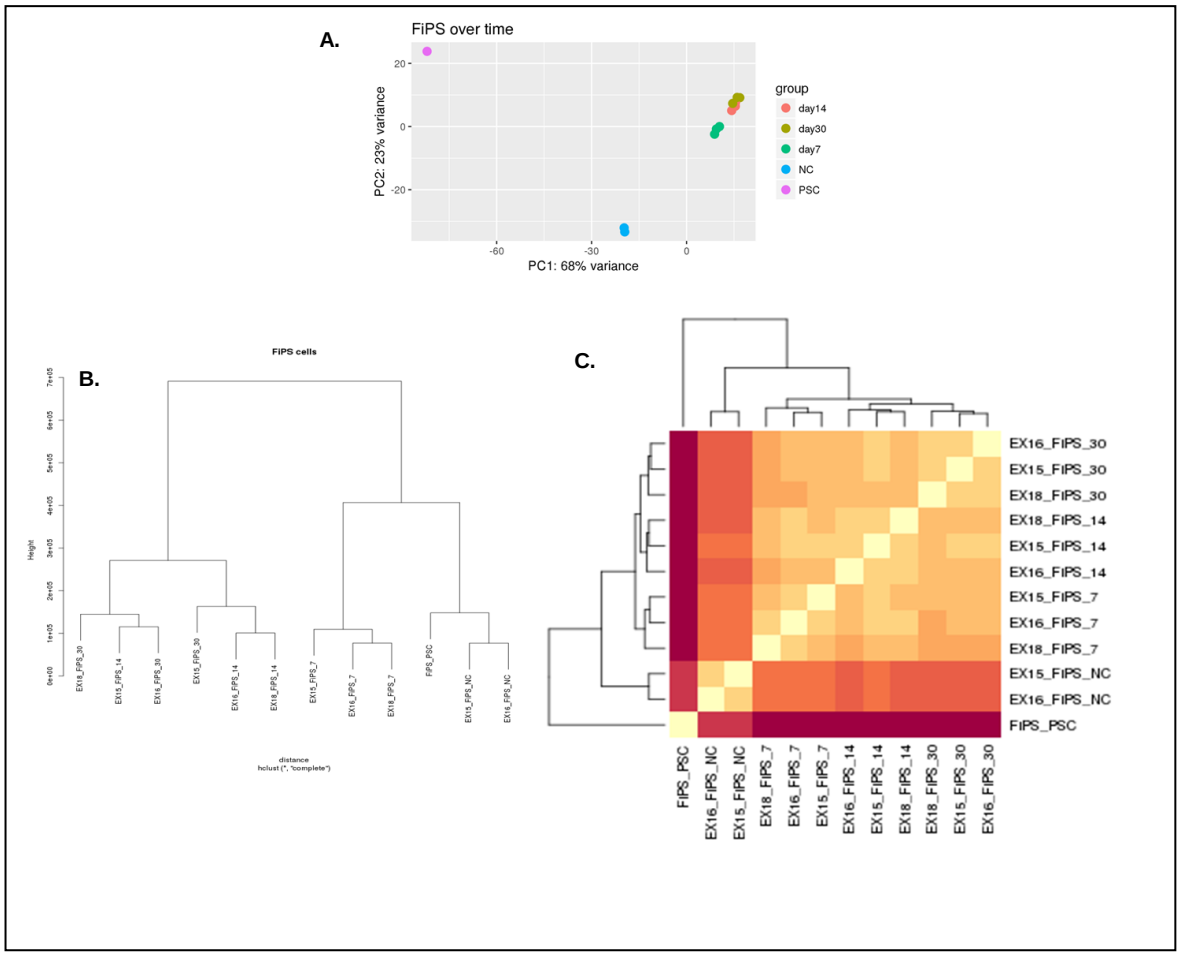


Figura 14. Figuras representativas del control de calidad de las muestras procedentes de las células control (FiPS). A. Gráfico PCA de las muestras FiPS. En este gráfico se distribuyen las muestras de acuerdo a varianza que presenta la expresión entre ellas. La varianza se ve representada por medio de las componentes principales que se encuentran en el eje x (PC1: 68%) y el eje y(PC2:23%). Antes de representarlas, se realizó la transformación logarítmica de las cuentas de las distintas muestras, para poder tener en cuenta las pequeñas diferencias entre ellas. *day14*: muestras a los 14 días de diferenciación. *day30*: muestras a los 30 días de diferenciación. *day7:* muestras a los 7 días de diferenciación. NC: Estadio de cresta neural. PSC: células iPSC. B. Dendrograma de las muestras procedentes de FiPS. Se representa de manera jerárquica el conjunto de las muestras usando la distancia Euclidiana. En este dendrograma se asocian las muestras sin realizar una transformación logarítmica de la expresión de sus tránscritos. Por lo tanto, se tienen en cuenta las grandes diferencias entre las distintas muestras a la hora de la agrupación. C. Heatmap representativo de la distancia entre las muestras. El nombre de las muestras se encuentra detallado en la tabla 4. El color que se observa en este gráfico explica la asociación entre las distintas muestras. Para anotar a cada una de las muestras en el dendrograma y en el heatmap, se utilizó el nombre que se les atribuyó. Se encuentra detallado en la tabla 4 del apartado materiales y métodos, bajo el nombre de *sample.names*. Exp: experimento, 5PNF: tumor plexiforme 5, 3PNF: tumor plexiforme 3, FiPS: células procedentes de fibroblastos, PSC: pluripotent stem cells, NC: células de la cresta neural (Neural Crest), MM: células con genotipo $NF1^{-/-}$ 7;14;30: días del proceso de diferenciación.

Además de realizar el típico control de calidad de datos de RNA-seq, se hicieron gráficos de expresión usando las funciones desarrolladas (*generationGeneMatrix(), plotLimsAndLabels(), plotPointsAndMeans()*). Los genes que se representaron ya habían sido estudiados previamente por las investigadoras para comprobar el proceso de diferenciación de las células de Schwann en los distintos puntos en el tiempo con el uso de *RT-q*PCR. Los genes que se utilizaron como marcadores de cada estadio de diferenciación de las SC, se detallan en el artículo de Jessen & Mirsky en 2005.

Estos marcadores se miraron para comprobar si el experimento biológico se había llevado a cabo de la manera correcta, mirando si la expresión de los genes analizados por *RT*-qPCR era semejante a la que se observaba en los datos procedentes de RNA-seq de este experimento. La expresión de los genes representada por los gráficos de la figura16, presentaban el mismo perfil de expresión que la analizada por *RT*-qPCR (figura 15). En conclusión, el experimento biológico se había llevado a cabo de la manera correcta.

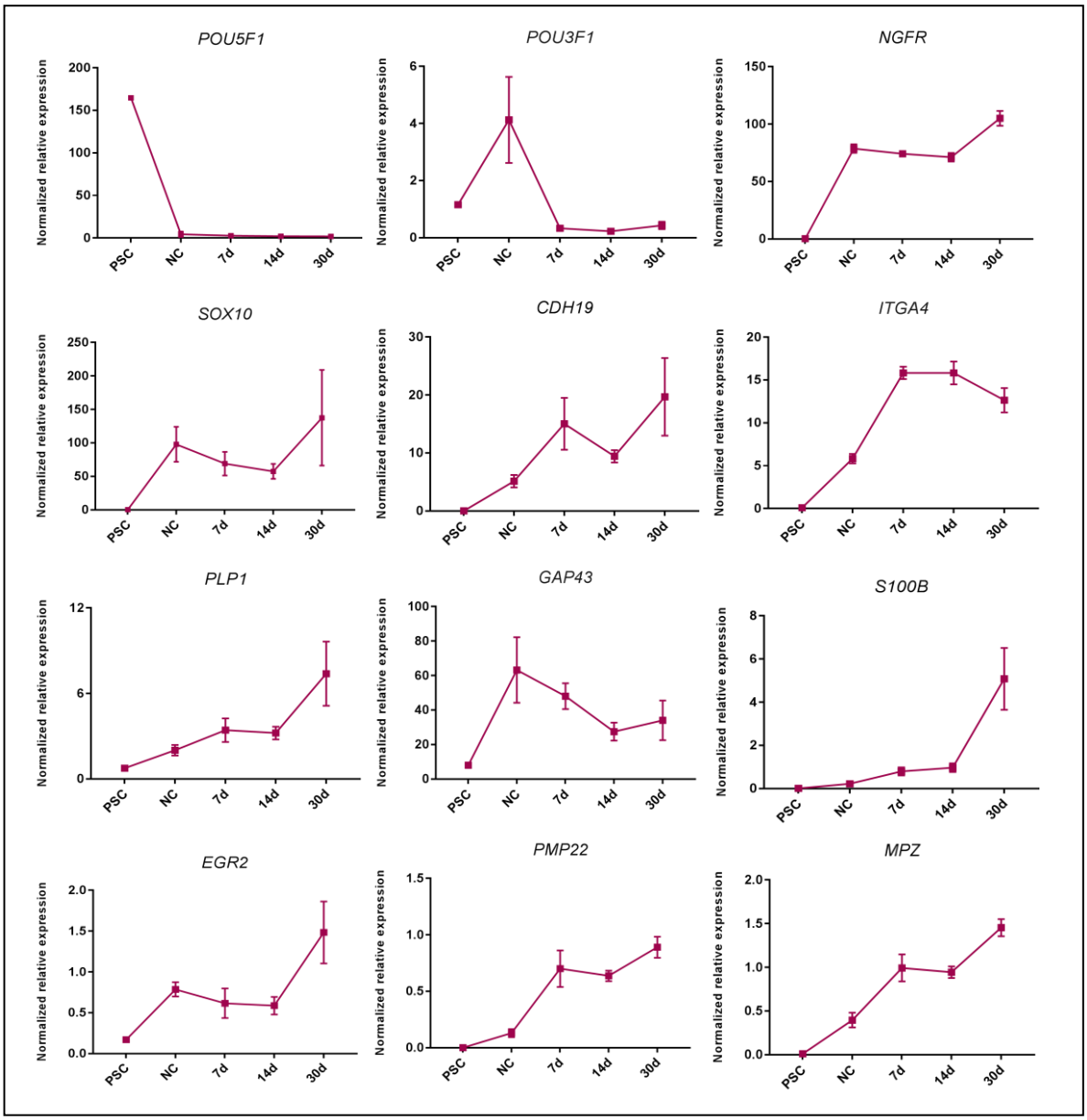


Figura 15. Marcadores de expresión analizados por *RT*-qPCR de las células FiPS en el proceso de diferenciación hacia células de Schwann. En el eje de las x se representan los distintos puntos en el tiempo del estuido de diferenciación. En el eje de las y se representa la expresión relativa normalizada de estos genes con EP300 y TBP(*topisomerase binding protein*). Se tratan de resultados obtenidos previamente en el laboratorio en el que se llevó a cabo el proyecto.

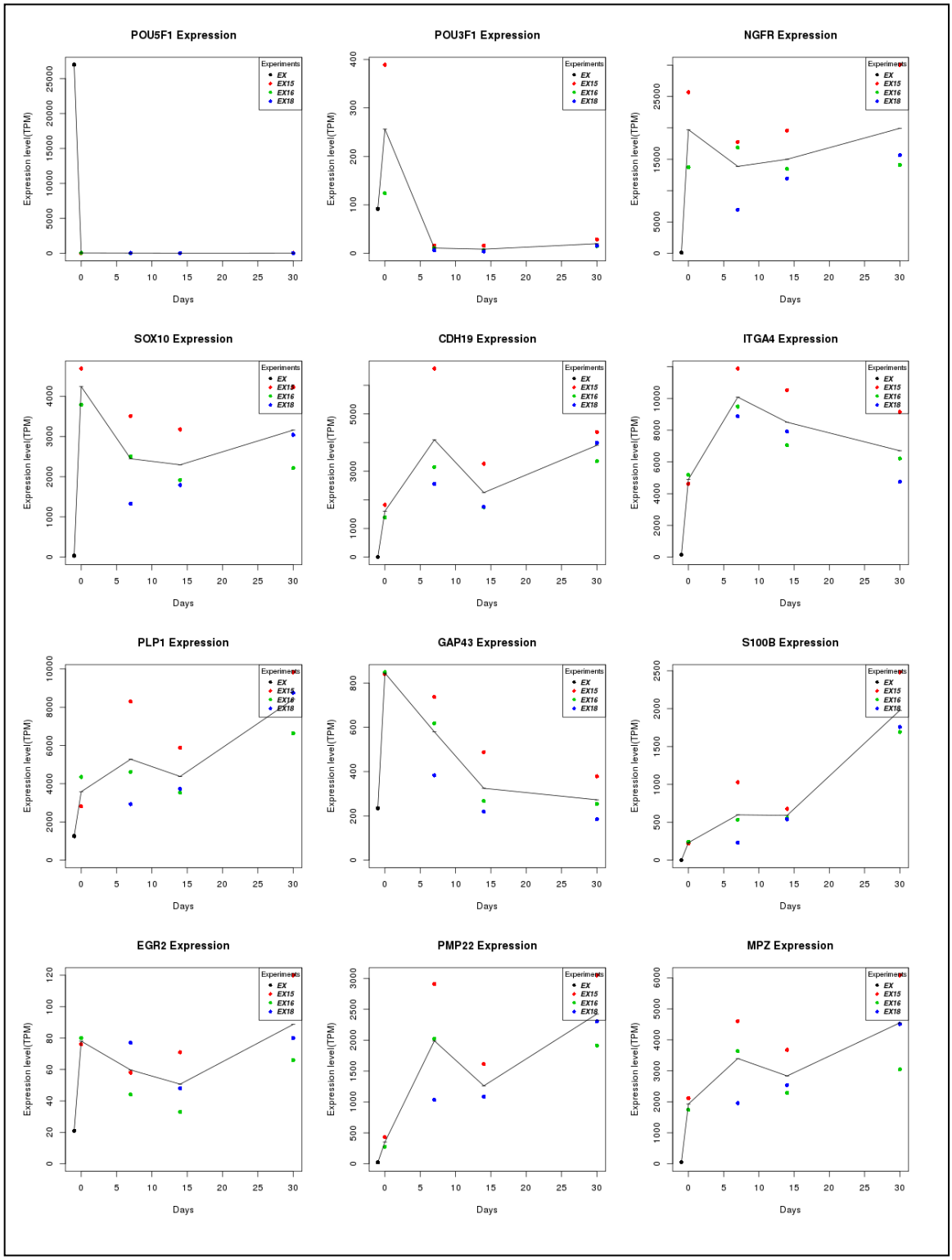


Figura 16. Graficos de expresión analizdos por RNA-seq de los genes de los genes analizados por *RT*-qPCR que se muestran el la figura 15. Cada muestra corresponden a un punto distinto con un color diferente, selecionado por el número del experimento al que corresponde cada una. La linea negra, une la expresión media de los distintos genes en cada uno de los puntos, trazando el perfil de expresión. En el eje de abscisas, se representa de forma numérica los distintos puntos en el tiempo del estudio. En el eje de ordenadas se representa el nivel de expresión en TPM .

3.2.3. Expresión diferencial

Una vez comprobada la calidad de las muestras, se procedió al análisis de la expresión diferencial de los datos procedentes de las células FiPS para responder dos preguntas principalmente:

1. Qué genes se expresaban de manera diferencial entre los distintos puntos en el tiempo.
2. Qué genes eran específicos de cada estadio de diferenciación establecido con el fin de encontrar marcadores específicos de cada punto.

Los contrastes que se realizaron para responder a la primera pregunta son los que se encuentran en la tabla 3 del apartado Materiales y Métodos.

Para responder a la primera pregunta planteada, una vez los datos fueron filtrados, se usó la función *runDESeq()* con la que se obtuvieron los genes diferencialmente expresados de cada uno de los contrastes, usando la prueba estadística de "Wald" y realizando un *lfcShrink normal* . Se sabe que el método *apeglm* utiliza una distribución previa *heavy-tailed Cauchy* para estimar el efecto de los tamaños, dando como resultado menos sesgos que el *normal* y consiguiendo reducir la varianza al igual que el método establecido por defecto. Esto lo consigue, sin dejar de considerar aquello genes con gran expresión a la vez que tiene en cuenta a los genes más expresados. En el caso del método *normal,* puede perder genes con señal suficiente o puede realizar una contracción excesiva de los LFC que son grandes realmente, generando algunos "falsos negativos". Sin embargo, el motivo por el cual se utilizó el método *normal* se debe a que la cantidad de réplicas por muestra no era lo suficientemente grande como para conseguir resultados factibles utilizando el método *apeglm.*

```
log2 fold change (MAP): design day7 vs NC
Wald test p-value: design day7 vs NC
DataFrame with 100 rows and 6 columns
            baseMean log2FoldChange     lfcSE      stat        pvalue          padj
           <numeric>      <numeric> <numeric> <numeric>     <numeric>     <numeric>
LIN28A      3618.201      -6.688721 0.1558435 -42.79337  0.000000e+00  0.000000e+00
RUNX1       2443.217       6.655130 0.2230905  29.38135 9.506564e-190 9.880647e-186
PMEPA1      4728.639       6.252527 0.2115689  29.30048 1.022630e-188 7.085803e-185
OLFML2A     6534.818       5.155135 0.1790330  28.72719 1.746159e-181 9.074353e-178
ITGB3       3872.104       4.920515 0.1721746  28.49971 1.181140e-178 4.910470e-175
...              ...            ...       ...       ...          ...           ...
HIST1H2BD  1091.4197      -2.236090 0.1426718 -15.67551  2.224341e-55  4.816394e-53
HIST1H2BE   427.2187      -2.871347 0.1831159 -15.67439  2.264052e-55  4.851841e-53
PLXDC2     1296.9000       3.097018 0.1977587  15.66235  2.736031e-55  5.803457e-53
CENPV       328.5662      -2.919236 0.1864561 -15.65885  2.890761e-55  6.069722e-53
COL5A1    36443.5260       3.524522 0.2259654  15.58873  8.683736e-55  1.805088e-52
```

Figura 17. Ejemplo del data frame obenido con la función runDESeq() donde los p-valoes ajustados (padj) fueron ordenados en modo creciente. Este ejemplo corresponde al análsis realizado con las muestras FiPS tras 7 días de diferenciación frente a aquellas que parten del día o, cresta neural (NC) que actua como nivel de referencia. Este resultado sólo incluia los 100 primeros genes. Los valores que se tienen que considerar de esta tabla son el p-valor ajustado, (padj), y el log2FoldChange(LFC) cuyo valor da información de cuan expresado esta un gen en el contraste que se realiza. Si el valor de LFC es negativo, quiere decir que ese gen se encuentra infra expresado en el punto a contrastar del análisis (en este caso day7), mientras que, al tratarse de un contraste entre dos grupos de muestras, esto significa que esos valores negativos de LFC en esos genes, también indican que ese gen está sobre expresado en el grupo de referencia. baseMean: representa la media de las cuentas normalizadas en todas las muestras. log2FoldChanges: la estimación del tamaño del efecto. lfcSE: el error estándar de log2FoldChange. stat: el estadístico de Wald p-value: p-valor obtenido. p-adj: p-valor ajustado por BH FDR.

Una vez conseguidos los genes diferencialmente expresados, se seleccionaron aquellos que presentaban un p-valor ajustado menor de 0.05 y, se ordenaron según el valor absoluto de LFC de manera decreciente con la función *getTopGenesLFC()*[5].

Con el uso de la función *plotMADEG()* se representan aquellos genes que se encuentran diferencialmente expresados y con un LFC absoluto grande en un gráfico MA. En estos gráficos se representa la expresión media de un gene en las distintas muestras (eje x) frente al LFC obtenido (eje y). En color rojo se muestran aquellos genes diferencialmente expresados según el p-valor ajustado que delimite el usuario. Así mismo, el usuario puede remarcar con círculos azules aquellos genes más y menos expresados según su LFC, en el grupo que se quiera contrastar frente al de referencia. En la figura 18, figura 19 y figura 20, se muestran los *MA-plots* obtenidos de los distintos contrastes

[5] Los archivos con los genes diferencialmente expresados se adjuntaran junto con la memora del TFM. En el apartado Anexo1 se muestra una lista con los archivos que van a ser adjuntados.

realizados. Estas figuras se dividen en tres grupos según su grupo de referencia con el que se realizó el contraste. En la figura 18, el grupo de referencia era PSC (iPSC). En la figura 19, el grupo de referencia era NC, mientras que en la figura 20 a y 20 b, el grupo de referencia es el día 7 (day7), y en la figura 20 c es el día 14.

Cabe destacar que en los gráficos presentes en la figura 19 y 20, el número de genes diferencialmente expresados (puntos de color rojo) era menor que en la figura 18. Esto puede ser debido a que la similitud entre estas muestras es mayor que la que se observa entre PSC y el resto, ya que, como se ve en la figura 18, la cantidad de genes diferencialmente expresados es mayor.

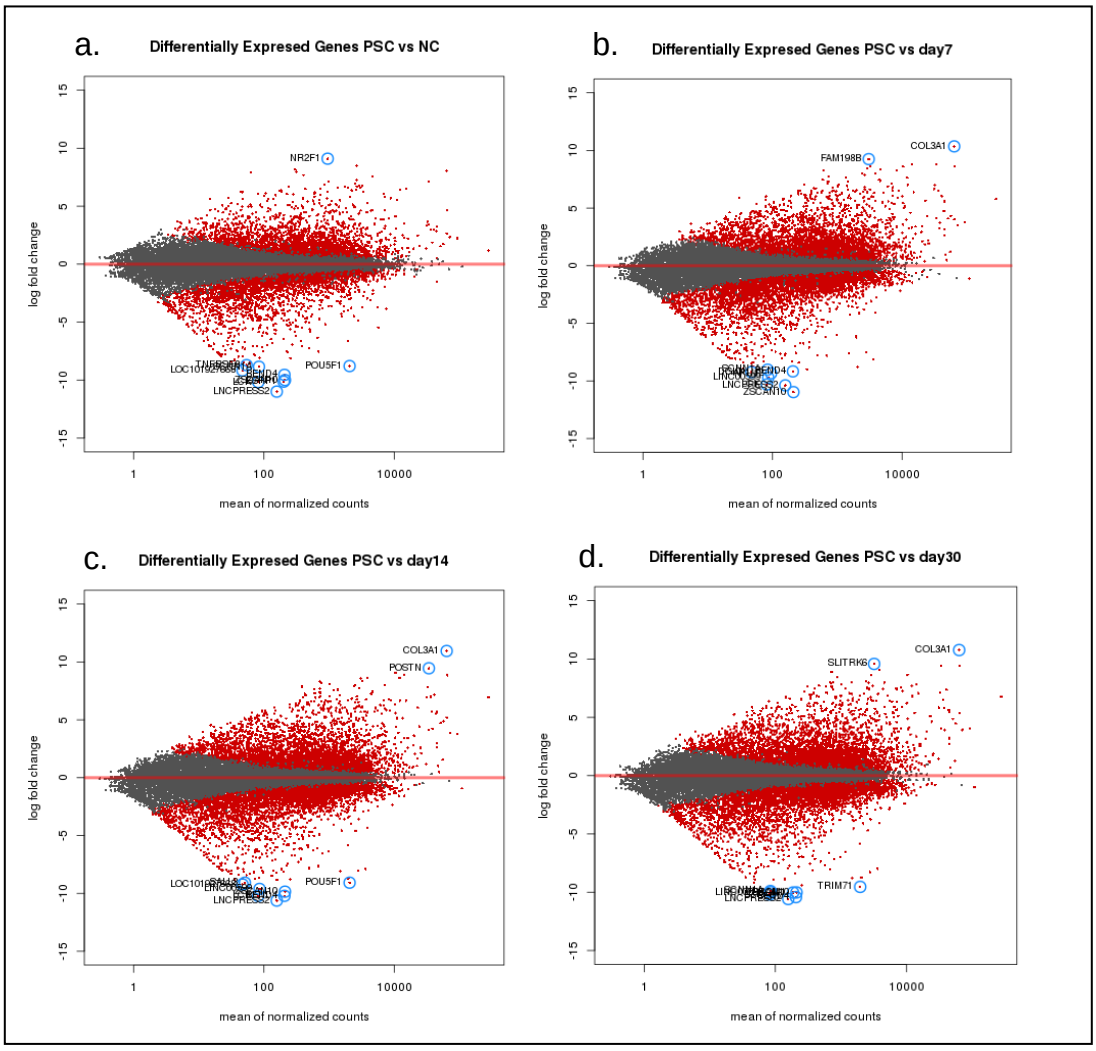


Figura 18. MA-plots de los genes diferencialmente expresados usando como nivel de referencia PSC (iPSC) frente al resto de los puntos analizados. En **el eje de abcisas se representan la expresión media de un gen en las distintas muestras, mientras que en el de ordenadas se representan los valores LFC de los genes. En rojo se representan los genes que están diferencialmente expresados. En azul y con el símbolo de su gen, se representan aquellos genes diferencialmente expresados que tiene mayor valor absoluto de su *log2 fold change.* Como se trata de un análisis entre dos tipos de muestras, los genes que se encuentran en la parte de debajo de la línea roja, se encuentran sobre representados en el grupo de referencia (PSC), mientras que los que se encuentran en la parte de arriba de esta línea, están más expresados en las muestras contrastadas.**

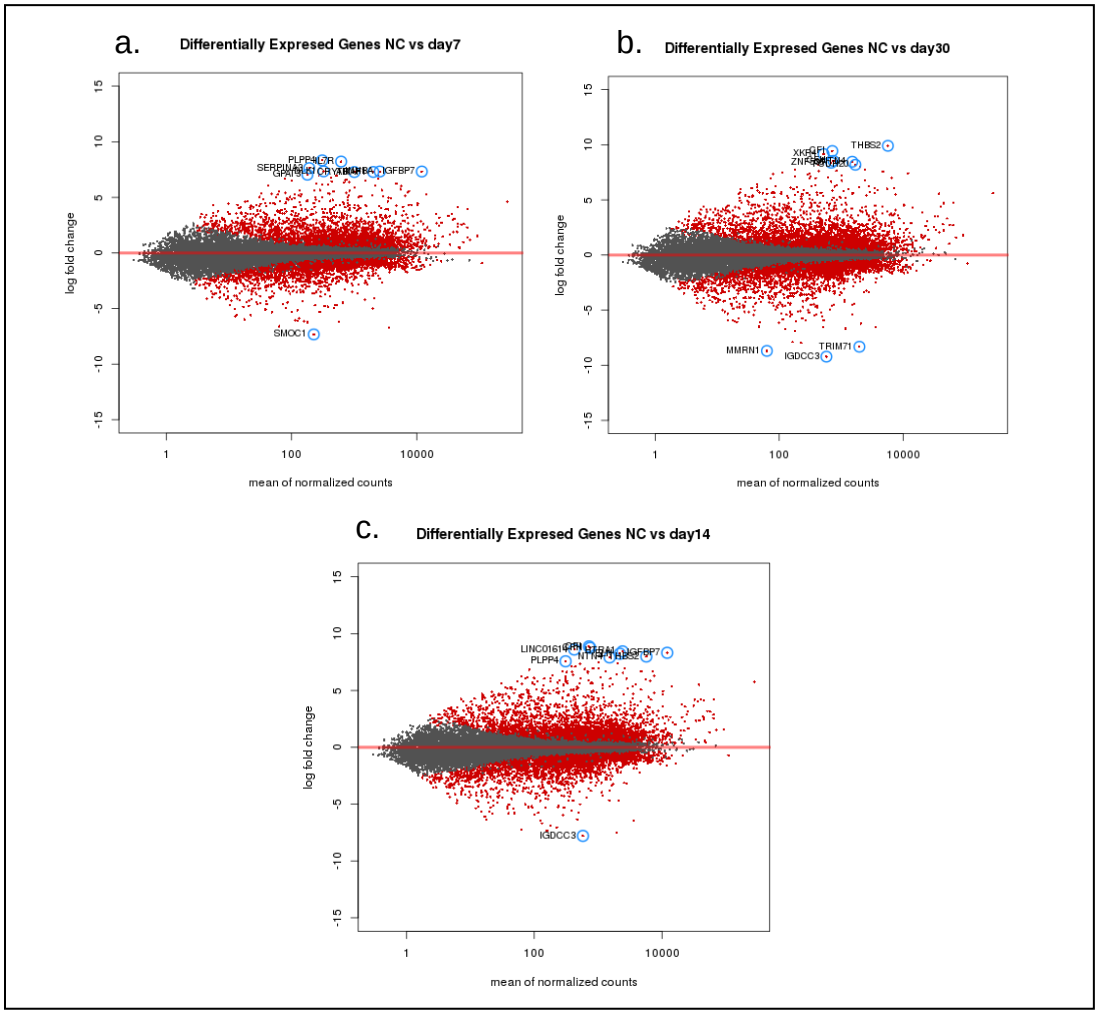


Figura 19. MA-plots de los genes diferencialmente expresados usando como nivel de referencia NC frente al resto de los puntos analizados. En **el eje de abcisas se representan la expresión media de un gen en las distintas muestras, mientras que en el de ordenadas se representan los valores LFC de los genes. En rojo se representan los genes que están diferencialmente expresados. En azul y con el símbolo de su gen, se representan aquellos genes diferencialmente expresados que tiene mayor valor absoluto de su *log2 fold change.* Como se trata de un análisis entre dos tipos de muestras, los genes que se encuentran en la parte de debajo de la línea roja, se encuentran sobre representados en el grupo de referencia (NC), mientras que los que se encuentran en la parte de arriba de esta línea, están más expresados en las muestras contrastadas.**

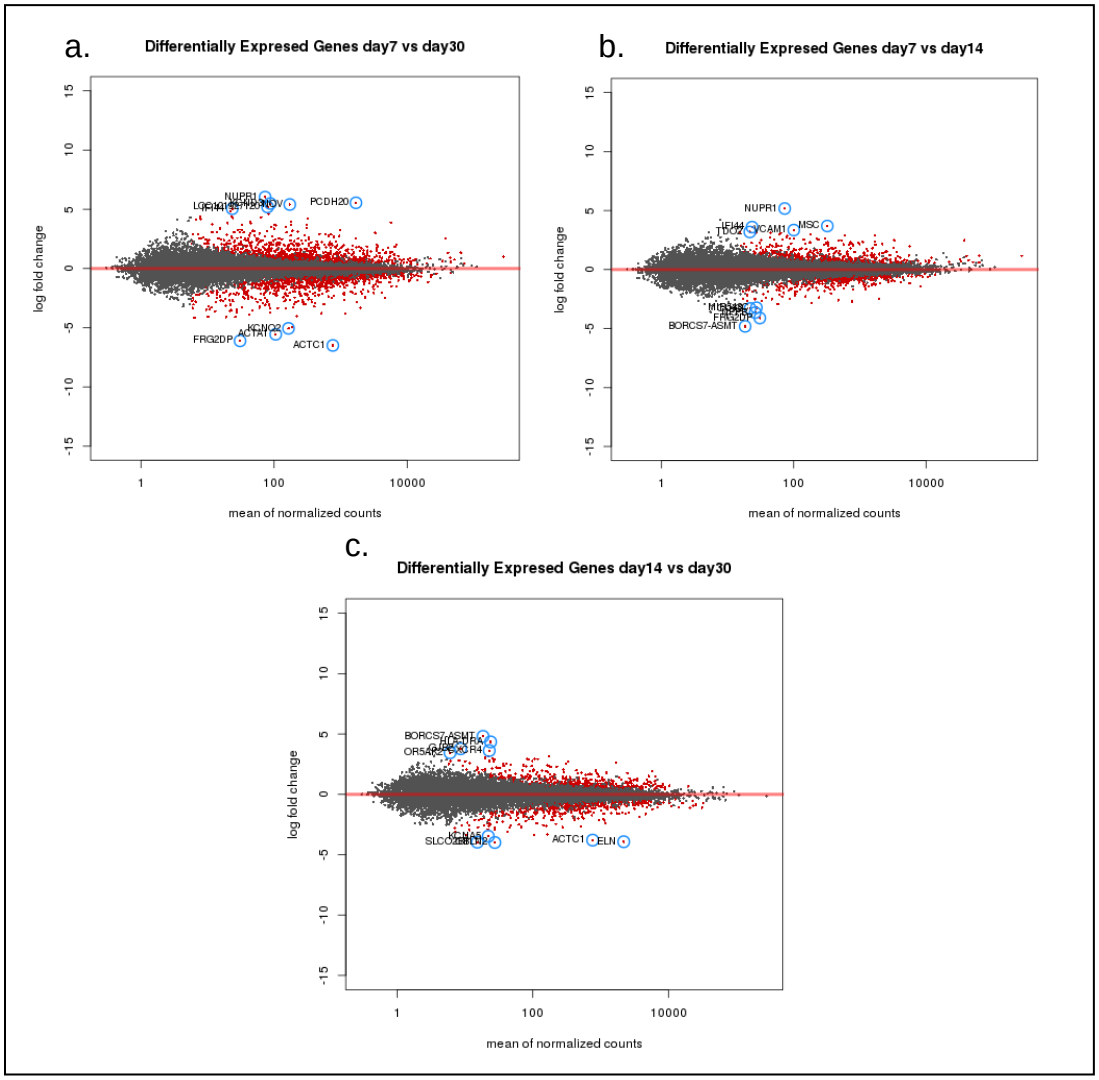


Figura 20. MA-plots de los genes diferencialmente expresados usando como nivel de referencia (day7 y day14) frente al resto de los puntos analizados. En **el eje de abcisas se representan la expresión media de un gen en las distintas muestras, mientras que en el de ordenadas se representan los valores LFC de los genes. En rojo se representan los genes que están diferencialmente expresados. En azul y con el símbolo de su gen, se representan aquellos genes diferencialmente expresados que tiene mayor valor absoluto de su *log2 fold change.* Como se trata de un análisis entre dos tipos de muestras, los genes que se encuentran en la parte de debajo de la línea roja, se encuentran sobre representados en el grupo de referencia, mientras que los que se encuentran en la parte de arriba de esta línea, están más expresados en las muestras contrastadas.**

3.2.4. Selección de genes específicos

Con el fin de responder a la pregunta planteada por las investigadoras para saber los genes que eran específicos de cada estadio de diferenciación, con el fin de encontrar posibles marcadores específicos de cada punto, se utilizó la función *getGroupSpecificGenes()*[6]. Se consiguieron aquellos genes específicos tanto sobre expresados como infra expresados de cada punto de diferenciación.

Estos genes se utilizaron para hacer distintos *heatmaps* (figura *21*) y poder comprobar de una manera más visual cómo se expresaban los posibles marcadores en las distintas muestras de las células FiPS en los distintos puntos en el tiempo. De este modo, se consiguieron posibles marcadores para este modelo *in vitro* de diferenciación de las células de Schwann desde iPSC hasta los 30 días.

En la figura 21 se representan los distintos *heatmap*s con los 100 primeros genes de la lista de los genes comunes en cada punto específico en el tiempo. En color rojo vemos aquellos genes que se encuentran sobre expresados mientras que en color azul vemos aquellos que están infra expresados. La figura21 A y 21 B representan los genes comunes diferencialmente expresados en el punto de diferenciación de día 0 o NC, y día 7. En las dos imágenes de debajo, se representan los genes comunes del punto día 14 y día 30 respectivamente.

Una vez se consiguen esta lista de los genes comunes, se puede realizar la anotación con GO para conocer a qué función biológica están asociados.

[6] Los archivos con los genes específicos de cada punto en el tiempo junto con su anotación, se adjuntaran con el TFM (anexo1)

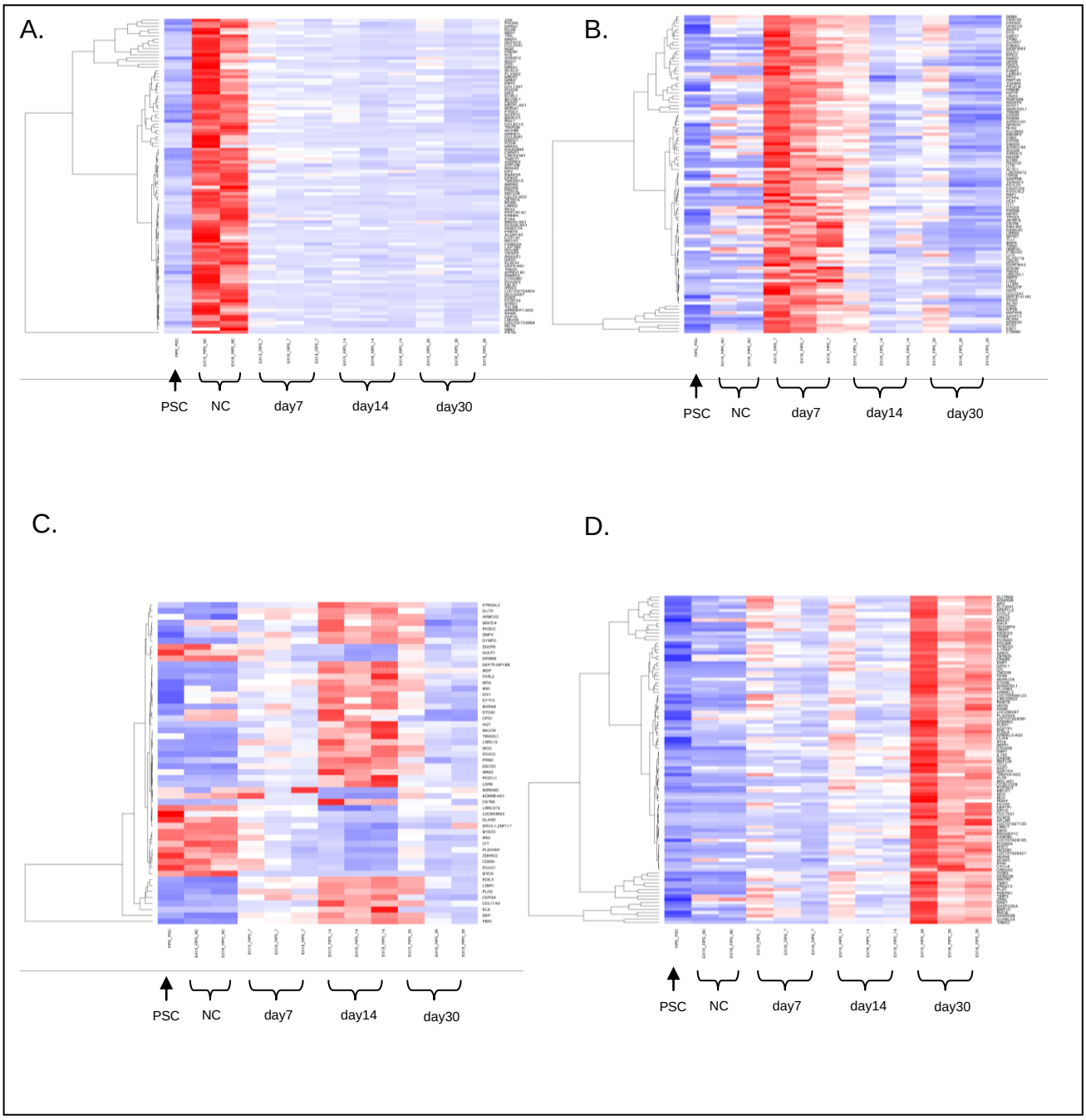


Figura 21. Heatmaps que representan los genes diferencialmente expresados que son específicos de cada estacio diferencial. En el eje de las x se representan las muestras ordenadas de la manera establecida por el usuario. En el eje de las y se representan, en el lado derecho los genes diferencialmente expresados y específicos de un punto del tiempo concreto del estudio de diferenciación de las células de Schwann y, en el lado izquierdo, se realiza un *clustering* jerarquico que asocia a los genes según su perfil de expresión en las distintas muestras. En color rojo se representan aquellos genes que se encuentran sobre expresados, en color azul los que se encuentran infraexpresados, mientras que en color blanco aquellos que tienen una expresión "standard". A. Heatmap en el que se representan los 100 primeros genes que son comunes a NC y se encuentran sobre expresados en esta condición en comparación con el resto de los tiempos. B. Heatmap en el que se representan los 100 primeros genes que son comunes a día 7 (*day7*) y se encuentran sobre expresados en esta condición en comparación con el resto de los tiempos. C. Heatmap en el que se representan los 100 primeros genes que son comunes a día 14 (*day14)* y se encuentran tanto sobre expresados como infra expresados, si comparamos esta condición con el resto de los tiempos. D. Heatmap en el que se representan los 100 primeros genes que son comunes a día 30 (*day30)* y se encuentran sobre expresados en esta condición en comparación con el resto de los tiempos. PSC: iPSC, NC: cresta neural.

3.2.5. GSEA y anotación

Una vez conseguidos los genes diferencialmente expresados y aquellos genes específicos de cada estadio, se realizaron dos procesos. El análisis de los genes diferencialmente expresados mediante GSEA para conocer que *GO terms* se encontraban más representados y, las vías más representadas con los DEG, así como la anotación de aquellos genes específicos de cada estadio de diferenciación para conocer que funcionalidad biológica tenían.

El *gene set enrichment analysis*[7] se llevó a cabo con el uso de dos funciones: *getGOtermsFromDEG()* y *getKEGGpathways().*

Para conseguir aquellos térmicos GO más representados se utilizó *getGOtermsFromDEG().* Con esta función, para cada contraste que se realizó, se consiguieron los términos más representativos de todas las funciones biológicas que agrupan los términos de GO (MF, BP, CC). Un resultado interesante a resaltar es el resultado del GSEA para el contraste entre el día 14 frente al día 30. Se obtuvieron los resultados mostrados en la figura 22. Estos términos de GO referentes al proceso biológico (BP) al que se asocian los genes diferencialmente expresados, son característicos de las células del sistema nervioso periférico, como lo son las células de Schwann. Esto indicaba que, el modelo *in vitro* de diferenciación de estas células se estaba llevando a cabo de la manera esperada.

[7] Se adjuntaran los resultados referentes al proceso biológico (BP) a los 30 días de diferenciación. La lista con los archivos que serán adjuntados se encuentra en el anexo1.

```
--------------------------------------
GOID: GO:0007272
Term: ensheathment of neurons
Ontology: BP
Definition: The process in which glial cells envelop neuronal cell bodies and/or
    axons to form an insulating layer. This can take the form of myelinating or
    non-myelinating ensheathment.
Synonym: ionic insulation of neurons by glial cells
--------------------------------------
GOID: GO:0008366
Term: axon ensheathment
Ontology: BP
Definition: Any process in which the axon of a neuron is insulated, and that
    insulation maintained, thereby preventing dispersion of the electrical
    signal.
Synonym: cellular axon ensheathment
Synonym: cellular nerve ensheathment
Synonym: nerve ensheathment
Synonym: GO:0042553
Secondary: GO:0042553
--------------------------------------
```

Figura 22. Ejemplo del contenido que tiene uno de los archivos obtenidos a partir de la función *getGOTermsFromDEG().* Este ejemplo corresponde a la función molecular (MF) que tiene asociada los genes diferencialmente expresados obtenidos del contraste de día 14 (referencia) frente a día 30 de aquellos genes que se encuentran sobre expresados.

Sin embargo, no se observaron vías significativamente representadas con la función *getKEGGpathways()* que utiliza el paquete de análisis (*gage).* Podría deberse a que el paquete y el método utilizado para conseguirlo no era el adecuado, o porque realmente, no se encontraban vías significativamente expresadas con los genes diferencialmente expresados. Además, aunque KEGG contiene información sobre las vías de señalización bien establecida, revisadas por grandes expertos, no es la base de datos más completa.

Con la función *getBioMartGOAnnotation()* se anotaron los genes específicos de cada estadio de diferenciación de las células control. La información recogida con esta función tiene la forma que se observa en la tabla 6.

Tabla 6. Ejemplo de anotación asociada con los términos de Componente Celular al que corresponde el gen *IL7R*.

hgnc_symbol	CC.entrezgene	CC.go_id	CC.name_1006	CC.namespace_1003	condition
IL7R	3575	GO:0016020	membrane	cellular_component	up
IL7R	3575	GO:0016021	integral component of membrane	cellular_component	up
IL7R	3575	GO:0005886	plasma membrane	cellular_component	up
IL7R	3575	GO:0005576	extracellular region	cellular_component	up
IL7R	3575	GO:0030665	clathrin-coated vesicle membrane	cellular_component	up
IL7R	3575	GO:0009897	external side of plasma membrane	cellular_component	up

La anotación de estos genes es muy útil para conseguir marcadores de membrana específicos de cada estadio de diferenciación. Para ello, el término de GO que hace referencia a los componentes celulares (CC) es el adecuado para conseguir posibles genes específicos que estén asociados al término "*membrane*", "*receptor*", o terminología similar. Las investigadoras es capaz, en este momento y con esta información, de seleccionar aquellos genes que más le interesen. Una vez seleccionados, la bioinformática, mediante los gráficos de expresión de estos genes, le puede confirmar que los genes que fueron seleccionados son específicos de cada punto de diferenciación en el tiempo.

Algunos de los genes que codificaban para receptores de membrana y eran específicos de un punto en concreto se encuentran detallados en la tabla 7:

Tabla 7. Tabla que recoge algunos genes específicos de los distintos estadios de diferenciación que tienene como característica común que codifican para proteínas de membrana.

Tiempo	Genes
NC	*LRP2, NTRK1*
Día 7	*IL7R*
Día14	*LGR5*
Día30	*PCDH20, IL1R1*

Para comprobar que eran específicos, se representaron con los gráficos de expresión que fueron desarrollados (figura 23). Con este tipo de gráfico se puede ver la expresión de los genes escogidos para cada punto de diferenciación en el que se recogieron las células, de una manera más representativa. Todos los genes que se seleccionaron tenían la característica de ser encargados de expresar proteínas de membrana útiles para la selección de las células en cada punto en concreto, y se expresaban específicamente en cada estadio de diferenciación.

En conclusión, con la obtención de estos resultados tras el análisis de los genes diferencialmente expresados, la representación de la expresión de genes específicos y su anotación, las investigadoras consigue una información útil procedente de unos datos RNA-seq, con los que poder sacar conclusiones acordes a las preguntas planteadas al comienzo del experimento.

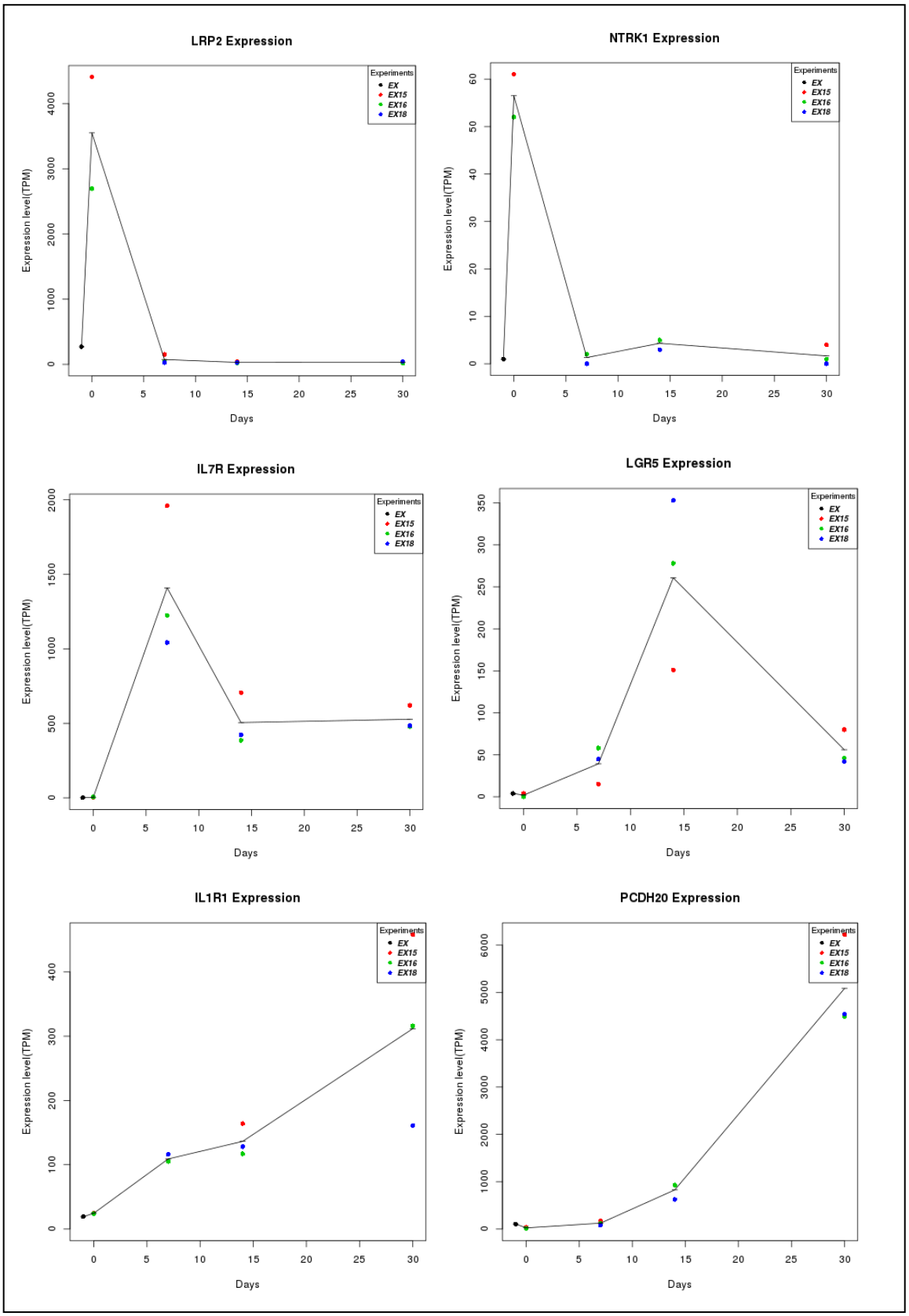


Figura 23. Graficos de expresión de los genes seleccionados como posibles parcadores de los diferentes puntos del tiempo. Cada muestra corresponden a un punto distinto, diferenciado por el número del experimento al que corresponde cada una. La linea negra, une la expresión media de los distintos genes en cada uno de los puntos, trazando el perfil de expresión. En el eje de abscisas, se representa de forma numérica los distintos puntos en el tiempo del estudio. En el eje de ordenadas se representa el nivel de expresión en TPM . *LRP2* y *NTRK1* son genes sobreexpresados específicamente en NC. *IL7R* es un gene sobreexpresado específicamente a día 7. *LGR5* es un gene sobreexpresado específicamente a día 14. *IL1R1* y *PCDH20* son genes sobreexpresados específicamente en el día 30 de diferenciación.

3.2.6. Genes estudiados con un patrón de expresión similar.

Otra manera posible de obtener marcadores es por medio del estudio de aquellos genes que presentan una expresión similar a marcadores que se nombran en el artículo de Jessen & Mirsky de 2005. Para conseguir aquellos genes que tenían una expresión similar, se utilizaron las funciones *getGeneCorMatrix()* y *getCorrelatedGenes()*. En la figura 14 se encuentran representados la expresión de RNA-seq para los genes utilizados como marcadores de un punto de diferenciación en concreto.

Una vez se obtiene por medio de la función *getGeneCorMatrix()* la matriz de correlación de la expresión, con el uso de *getCorrelatedGenes()* se obtiene la lista de los genes más correlacionados.

Tabla 8. Resultado que se obtiene tras ejecutar la función *getCorrelatedGenes()*. Ejemplo de los genes que tienen una expresión similar a *MPZ* y *POU3F1* a lo largo del tiempo tras realizar la matriz de correlación de la expresión de estos genes en las distintas muestrás. Se muestran los 10 primeros genes con la expresión más similar, teniendo un valor de correlación superior a 0.9.

MPZ	Genes	POU3F1	genes
1,000	MPZ	1,000	POU3F1
0,999	MIR4534	**1,000**	RNU12
0,999	LINC00342	0,999	FGF5
0,998	GFI1	0,999	XPNPEP3
0,998	BVES-AS1	0,999	LOC100130587
0,996	**SRGAP2D**	0,999	**TNFAIP8L1**
0,996	TFAP2A-AS2	0,999	PPIL6
0,996	TMX4	0,999	LOC105377849
0,996	ANKRD36	0,999	NTSR2
0,996	ARID5A	0,999	SEPSECS

La tabla 8 muestra un ejemplo del objeto de salida de la función *getCorrelatedGenes().* En este resultado se estableció un nivel de correlación superior a 0.9. De todos los genes con ese nivel de correlación, en esta tabla se muestran únicamente los 10 primeros genes con un perfil de expresión más similar a lo largo del tiempo, de los genes *MPZ* y *POU3F1*. En el apartado de anexos (Anexo 2), se encuentra el resto de genes con aquellos que presentan un perfil de expresión similar a ellos.

Para comprobar que presentan el mismo perfil de expresión, se representaron los genes de referencia *(MPZ* o *POU3F1*) y algunos genes con los que su perfil de expresión era más similar (*SRGAP2D* y *TNFAIP8L1*). Con estos

gráficos se podría elegir posibles marcadores, así como hacer un control de calidad de aquellos que, a pesar de su correlación, el perfil de expresión no es una réplica en todas las muestras. Esto es causado porque la correlación se realiza con la media de expresión entre las réplicas por cada uno de los puntos de estudio.

En la figura 24 se representa el perfil de expresión de los genes de referencia y aquellos que con *getCorrelatedGenes()* tenían una expresión similar. Se puede apreciar que, aunque el nivel de expresión (TPM) que tienen los genes obtenidos, no es la misma que la de *MPZ* y *POU3F1*, su perfil de expresión a lo largo del tiempo es prácticamente igual en todas las muestras.

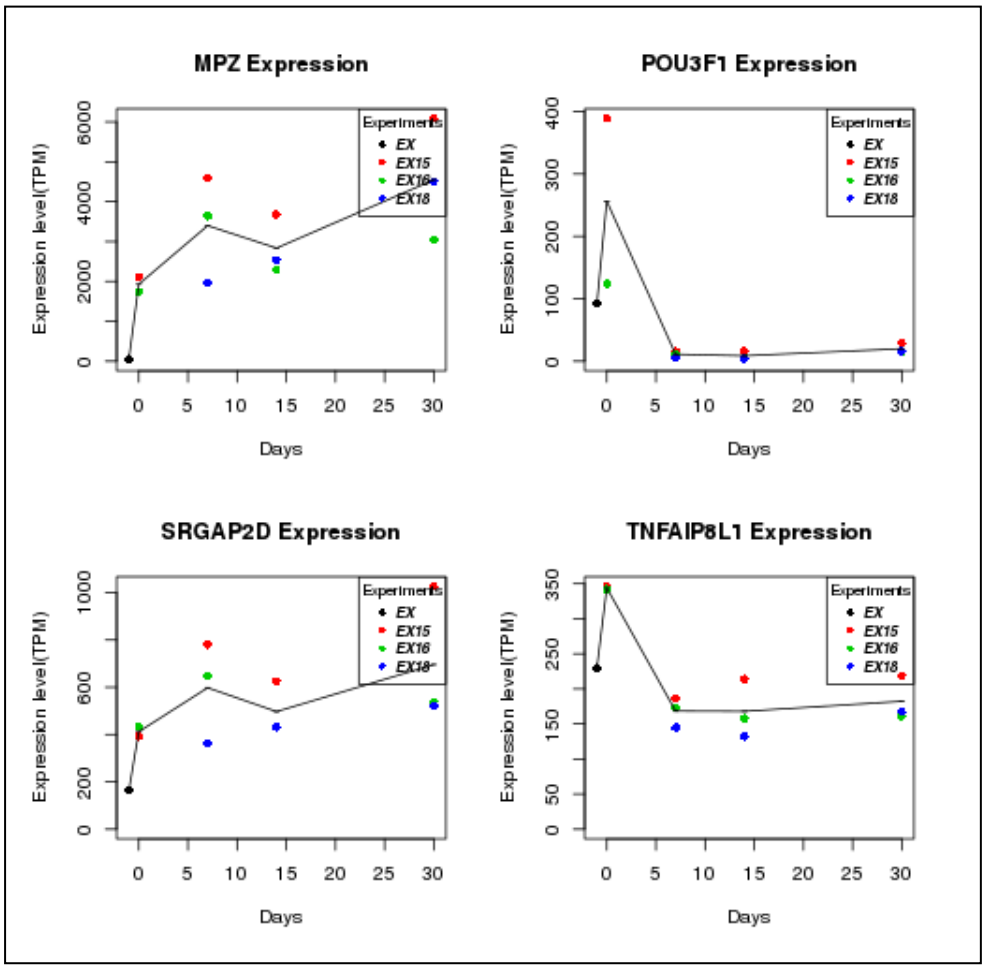


Figura 24. Graficos de expresión de los genes seleccionados como posibles marcadores de los diferentes puntos del tiempo. Cada muestra corresponden a un punto distinto, diferenciado por el número del experimento al que corresponde cada una. La linea negra, une la expresión media de los distintos genes en cada uno de los puntos, trazando el perfil de expresión. En el eje de abscisas, se representa de forma numérica los distintos puntos en el tiempo del estudio. En el eje de ordenadas se representa el nivel de expresión en TPM

4. Conclusiones

4.1. Conclusiones del estudio

- Los métodos desarrollados permiten realizar un análisis de expresión diferencial clásico.

- Para el desarrollo de las funciones se ha tenido en cuenta el tipo de análisis estadístico que se ha realizado en este proyecto (prueba de Wald), pero además, estos métodos permiten gran versatilidad relacionada con el tipo de test y el tipo de contrastes que el usuario quiera hacer.

- Las funciones desarrolladas dan al usuario la flexibilidad de realizar distintos análisis para responder a otras preguntas biológicas, como por ejemplo, observar la expresión de un gen a lo largo del tiempo.

- El modelo experimental de diferenciación desde iPSC hasta células de Schwann madura, se obtuvo de la manera esperada.

- Los métodos desarrollados permiten encontrar otros posibles marcadores específicos de cada estadio de diferenciación, para poder monitorizar el proceso.

4.2. Planificación y metodología

La planificación que se estableció al comienzo del proyecto se ha cumplido, cumpliéndose todos los objetivos propuestos. Se ha desarrollado un método de ayuda para analizar datos de RNA-seq de un experimento *time-course* con mayor facilidad y flexibilidad. Además, se añadieron nuevas funciones en la segunda fase de desarrollo del proyecto debido a que las investigadoras querían conocer qué genes eran específicos de cada estadio de diferenciación. Sin embargo, esto no supuso ningún cambio en el plan de trabajo, ya que en el plan inicial se contemplaba esta posibilidad en la tarea de presentación de resultados previos.

4.3. Caminos futuros, aspectos a mejorar

El proceso que se ha desarrollado funciona correctamente para responder a las preguntas planteadas por las investigadoras.

Una vez comprobada la utilidad de las funciones implementadas con las muestras procedentes de las células control, el siguiente paso sería analizar el resto de muestras, procedentes de PNF con un genotipo *NF1*$^{-/-}$, para sacar más conclusiones biológicas relacionadas con este experimento de diferenciación en las células Schwann.

Se podrían añadir nuevas funcionalidades que facilitaran la interpretación de los resultados a las investigadoras, como por ejemplo, obtener los términos de GO y observarlos mediante gráficos, para que los resultados puedan analizarse de una manera más visual.

5. Agradecimientos

Agradecer al grupo de Cáncer Hereditario del Instituto Germans Trias i Pujol por darme esta oportunidad para poder formarme como bioinformática y aprender de grandes profesionales. Dar las gracias a todos mis compañeros sin los que esta experiencia no hubiera sido lo mismo.

6. Glosario

BH: corrección Benjamini-Hochberg

BP: Biological Process

CC: Cellular Component

DEG: Genes diferencialmente expresados/ Differentially expressed genes

EXP: experimento

FiPS: Fibroblastos procedentes de biopsia de piel, transformadas a células pluripotentes.

GO: *Gene Ontology*

GSEA: *Gene Set Enrichment Analysis*

iPSC: *induced Pluripotent Stem Cells*

LRT: *Likellyhood Ratio Test*

LFC: *log2 fold change*

MAP: máximo a posteriori

MLE: *maximum-likelihood estimates*

MM: células con genotipo *NF1*$^{-/-}$

MPNSTs: tumores malignos de la vaina del nervio periférico

NF1: neurofibromatosis

NF1*:* gen que codifica para la neurofibromina

PCA: Análisis de Componentes principales

PNF: tumor de neurofibroma plexiforme

PSC: célula pluripotente

p-adj: p valor ajustado por BH

SC: Célula de Schwann

TPM: *Transcripts per Million*

7. Bibliografía

1. Ratner, N. & Miller, S. J. A RASopathy gene commonly mutated in cancer: the neurofibromatosis type 1 tumour suppressor. *Nat.Rev.Cancer* **15,** 290–301 (2016).
2. Mirsky, R., Woodhoo, A., Parkinson, D. B. & Arthur-farraj, P. Novel signals controlling embryonic Schwann cell development , myelination and dedifferentiation. *J Peripher Nerv Syst* **135,** 122–135 (2008).
3. Gutmann, D. H. *et al.* Neurofibromatosis type 1. *Nat. Publ. Gr.* **3,** 1–18 (2017).
4. Jessen, K. R. & Mirsky, R. THE ORIGIN AND DEVELOPMENT OF GLIAL CELLS IN PERIPHERAL NERVES. *Nat. Rev. Neurosci.* **6,** 671–682 (2005).
5. Carrió, M. *et al.* Reprogramming NF1 plexiform neurofibroma captures the genetic status and tumorigenic properties. *Manuscript* 1–39 (2018).
6. Gonz, I. Statistical analysis of RNA-Seq data. *Tutorial* (2014).
7. Pereira, M. A., Imada, E. L., Eddie, M. & Guedes, L. M. RNA - seq : Applications and Best Practices RNA - seq : Applications and Best Practices. doi:10.5772/intechopen.69250
8. Conesa, A. *et al.* A survey of best practices for RNA-seq data analysis. *Genome Biol.* 1–19 (2016). doi:10.1186/s13059-016-0881-8
9. Trapnell, C. *et al.* Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat. Protoc.* (2014). doi:10.1038/nprot.2012.016
10. Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat : discovering splice junctions with RNA-Seq. *Bioinformatics* **25,** 1105–1111 (2009).
11. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).

12. Patro, R. *et al.* Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference. *Nat. Methods* **14,** 417–419 (2017).

13. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq. *Nat. Biotechnol.* **34,** 4–8 (2016).

14. Soneson, C., Love, M. I. & Robinson, M. D. Differential analyses for RNA-seq : transcript-level estimates improve gene-level inferences [version 2 ; referees : 2 approved] Referee Status : *Version 2. F1000Res* 1–19 (2016). doi:10.12688/f1000research.7563.1

15. Everaert, C. *et al.* Benchmarking of RNA-sequencing analysis workflows using whole- transcriptome RT-qPCR expression data. *Sci. Rep.* 1–11 (2017). doi:10.1038/s41598-017-01617-3

16. Robinson, M. D., Mccarthy, D. J. & Smyth, G. K. edgeR : a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139–140 (2010).

17. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 1–21 (2014). doi:10.1186/s13059-014-0550-8

18. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom : precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* 1–17 (2014).

19. Spies, D. & Ciaudo, C. Dynamics in Transcriptomics : Advancements in RNA-seq Time Course and Downstream Analysis. *CSBJ* **13,** 469–477 (2015).

20. Huber, W. *et al.* Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods* **12,** 115–121 (2015).

21. Thomas, P. D. *et al.* PANTHER : A Library of Protein Families and Subfamilies Indexed by Function. *Cold Spring Harb. Lab. Press* 2129–2141 (2003). doi:10.1101/gr.772403.2

22. Durinck, S., Spellman, P. T., Birney, E. & Huber, W. Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor package biomaRt. *Nat Protoc* **4,** 1184–1191 (2011).

23. R Core Team. R: A language and environment for statistical computing. *R Fundation for Statistical Computing, Vienna, Austria* (2017). at <https://www.r-project.org/>

24. RStudio Team. RStudio: Integrated Development for R. *RStudio, Inc., Boston, MA* (2016). at <http://www.rstudio.com/>

25. Patro, R., Duggal, G., Love, M., Irizarry, R. & Kings, C. Salmon Documentation. (2017).

26. Love, M. I., Anders, S., Kim, V. & Huber, W. RNA-Seq workflow : gene-level exploratory analysis and differential expression [version 1 ; referees : 2 approved] Referee Status : *Version 1. F1000Res* **1070,** 1–41 (2015).

27. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biol.* **11,** 1–12 (2010).

28. Young, M. D., Wakefield, M. J., Smyth, G. K. & Oshlack, A. Gene ontology analysis for RNA-seq : accounting for selection bias. *Genome Biol.* **11,** 1–12 (2010).

29. Gene, T. *et al.* Gene Ontology : tool for the unification of biology. *Nat Genet.* **25,** 25–29 (2011).

30. Luo, W., Friedman, M., Shedden, K., Hankenson, K. & Woolf, P. Generally Applicable Gene-set / Pathway Analysis. 1–21 (2018). doi:10.1186/1471-2105-10-161.2

31. Luo, W., Friedman, M. S., Shedden, K., Hankenson, K. D. & Woolf, P. J. GAGE : generally applicable gene set enrichment for pathway analysis. *BMC Bioinformatics* **17,** 1–17 (2009).

32. Luo, W. & Brouwer, C. Pathview : an R / Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* **29,** 1830–1831 (2018).

33. Ogata, H., Goto, S., Sato, K., Fujibuchi, W. & Bono, H. KEGG : Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* **27,** 29–34 (1999).

34. Monk, K. R., Feltri, M. L. & Taveggia, C. New Insights on Schwann Cell Development. *Glia* **63,** 1376–1393 (2016).

35. Zhu, A., Ibrahim, J. G. & Love, M. I. Heavy-tailed prior distributions for sequence count data : removing the noise and preserving large differences. *bioRxiv* (2018). doi:http://dx.doi.org/10.1101/303255

36. Stephens, M. False discovery rates : a new deal. *Biostatistics* 275–294 (2018). doi:10.1093/biostatistics/kxw041

37. Young, M. D., Wakefield, M. J. & Smyth, G. K. goseq : Gene Ontology testing for RNA-seq datasets Reading data. (2017).

8. Anexos

Anexo 1. Lista de los archivos adjuntados junto con la memoria final

Pipeline_analysis.html	**Script en *html* con el proceso de análisis que se ha llevado a cabo para conseguir los resultados mostrados en este proyecto.**
Parameters.html	Lista con los parámetros usados para el proceso de análisis
FiPS_parameteres.yaml	Lista con los parámetros de las FiPS
Funciones/RNAseqHelper.R	Funciones desarrolladas para analizar los datos.
Genes diferencialmente expresados	Archivos con los genes diferencialmente expresados de cada uno de los contrastes realizados.
Archivos *GSEA*	Archivos que contienen los resultados obtenidos con las funciones implementadas para conseguir los GO terms más representados, a partir de los genes diferencialmente expresados en el estadio de 30 días de diferenciación.
Anotación de los genes específicos	Anotación de los genes específicos obtenidos de cada estadio de diferenciación de este estudio.

Anexo 2. Lista de los 10 primeros genes más correlacionados con los marcadores analizaos por *RT-qPCR*

CDH19	**genes**	**EGR2**	**genes**
1,000	CDH19	1,000	EGR2
0,999	CADM4	0,997	SNORA15B-1
0,999	TCF4	0,997	SNORA15B-2
0,999	FAM47E,STBD1	0,995	B4GALT1-AS1
0,997	COL18A1-AS2	0,994	TP53TG5
0,995	SECISBP2L	0,991	PHF24
0,995	TAS2R42	0,990	NEURL1
0,994	FLRT3	0,989	FAM46A
0,993	RIPK3	0,989	LOC101928381
0,991	ALDH1A3	0,988	KAT2B

ITGA4	genes	**MPZ**	**genes**
1,000	ITGA4	1,000	MPZ
0,999	STK38	0,999	MIR4534
0,999	LINC01433	0,999	LINC00342
0,999	RC3H2	0,998	GFI1
0,999	MIR22HG	0,998	BVES-AS1
0,999	SLC35E1	0,996	SRGAP2D
0,999	ACTR10	0,996	TFAP2A-AS2
0,999	ABR	0,996	TMX4
0,998	RPL23AP82	0,996	ANKRD36
0,998	TNR	0,996	ARID5A

PLP1	genes	**PMP22**	**genes**
1,000	PLP1	1,000	PMP22
0,997	ABHD2	0,998	KCNT1
0,996	ZFHX4-AS1	0,998	DST
0,996	LGI4	0,997	MAMLD1
0,994	AFAP1L2	0,997	OR2C1
0,993	SEC14L2	0,997	INSC
0,992	LINC02003	0,997	PLAT
0,992	VEGFB	0,997	C1orf53
0,992	SHROOM4	0,996	SYT11
0,992	JHY	0,996	POP3

POU5F1	**genes**	**POU3F1**	**genes**
1,0000	POU5F1	1,000	POU3F1
1,0000	LNCPRESS1	1,000	RNU12
1,0000	OTX2	0,999	FGF5
1,0000	BEND4	0,999	XPNPEP3
1,0000	HLA-DPB2	0,999	LOC100130587
1,0000	OR52A1	0,999	TNFAIP8L1
1,0000	C2orf80	0,999	PPIL6

1,0000	CD7	0,999	LOC105377849
1,0000	CLC	0,999	NTSR2
1,0000	FAM24B-CUZD1	0,999	SEPSECS

GAP43	**genes**	**NGFR**	**genes**
1,000	GAP43	1,000	NGFR
0,999	ARHGAP42	0,999	RNF180
0,999	SNORD3P3	0,999	ZNF404
0,999	LINC00092	0,998	LOC101927815
0,998	SMURF1	0,998	PTX3
0,998	LY6G6C	0,997	MEF2C
0,997	PTPRJ	0,996	ZNF76
0,997	ADH1B	0,996	MACROD2
0,997	NELFCD	0,996	UG0898H09
0,996	IDH3G	0,996	CDADC1

S100B	**genes**	**SOX10**	**genes**
1,000	S100B	1,000	SOX10
0,998	RASGEF1C	1,000	BHLHE40-AS1
0,998	NKX6,1	0,999	OSGIN1
0,997	FAHD2B	0,998	IFT172
0,997	VTRNA1-1	0,998	GATAD2B
0,997	CIITA	0,998	HOXB7
0,997	RGMB	0,997	CXorf40A
0,996	PLXNB3	0,996	KLF11
0,995	ITGB8	0,996	SPRNP1
0,995	LOC101928160	0,996	INPP1