UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D´ENGINYERIA AGRONÒMICA I DEL MEDI NATURAL

ANÁLISIS DE LA EXPRESIÓN ALTERNATIVA DE ISOFORMAS EN EL TIEMPO MEDIANTE DATOS DE RNA-SEQ

TRABAJO FIN DE GRADO EN BIOTECNOLOGÍA

ALUMNO: JORDI MARTORELL MARUGÁN

TUTORA: SONIA TARAZONA CAMPOS

COTUTORA EXTERNA: ANA CONESA CEGARRA

Curso Académico: 2014-2015

VALENCIA, JULIO DE 2015

Análisis de la expresión alternativa de isoformas en el tiempo mediante datos de RNA-Seq
Autor: D. Jordi Martorell Marugán
Tutor académico: Prof. Dña. Sonia Tarazona Campos
Cotutor: Dña. Ana Conesa Cegarra
Valencia, julio de 2015

Resumen
Los avances en tecnologías de secuenciación masiva han dado lugar al desarrollo de la transcriptómica por secuenciación, que permite analizar la expresión de genes e isoformas. El laboratorio de Genómica de la Expresión Génica participa en el proyecto STATegra, en el que se han generado datos de RNA-Seq de una serie de diferenciación de células B en ratón. Estos datos posibilitan el estudio de expresión de isoformas y de variantes de *splicing*, lo cual es muy interesante en organismos eucariotas superiores debido a que este es un elemento generador de regulación y complejidad transcripcional. Sin embargo, estos análisis siguen siendo difíciles, ya que los problemas de anotación y expresión diferencial de isoformas no están resueltos completamente para estos datos. El objetivo del trabajo fue comparar diferentes métodos de cuantificación de isoformas (eXpress y RSEM), analizar la expresión diferencial e interpretar los resultados con el fin de entender las diferencias funcionales asociadas a la expresión alternativa.

Abstract
The innovations in massive sequencing technologies have resulted in the development of sequencing transcriptomics that allows for the analysis of gene and isoform expression. The Genomics of Gene Expression laboratory participates in the STATegra project, in which RNA-Seq data have been generated for a B cell differentiation system. These data make it possible to study isoform expression and splicing variants, which is very interesting in superior eukaryotic organisms because it is the cause of regulation and transcriptional complexity. However, these analyses are still difficult because the annotation and isoform differential expression problems are not completely solved for these data yet. The objective of this work is to compare different methods for isoform quantification (eXpress and RSEM), analyze differential expression and interpret the results in order to understand the functional differences associated to alternative expression.

Palabras clave
RNA-Seq, expresión de isoformas, *splicing* alternativo, expresión diferencial, secuenciación masiva, bioinformática, diferenciación de células B.

Key words
RNA-Seq, isoform expression, alternative splicing, differential expression, massive sequencing, bioinformatics, B cells differentiation.

Para Carlos y Andrea, por hacer de mi carrera algo mucho más agradable;
Para mi abuela Mª Carmen, por conseguir aprenderse el nombre de mi carrera;
Para mis padres, por darme su apoyo;
Y para Eli, cuyo cariño me ha animado a seguir adelante durante estos 4 años.

AGRADECIMIENTOS

Me gustaría agradecer la colaboración y amabilidad de todos los miembros del laboratorio de Genómica de la Expresión Génica del CIPF, que no dudaron en sacrificar parte de su tiempo para explicarme aquello que necesitaba comprender. Merece una mención especial la Dra. Ana Conesa, directora del laboratorio y mi cotutora, por dirigir y supervisar mi trabajo.

Además, estoy especialmente agradecido a mi tutora, la Dra. Sonia Tarazona, que me ha soportado durante varios meses siempre con su amabilidad, paciencia y conocimiento. No solo ha hecho posible este trabajo preocupándose de que saliera bien como si fuera suyo, sino que me ha enseñado mucho y ha hecho que realmente haya valido la pena hacerlo.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ABREVIATURAS

bp: pares de bases

cDNA: DNA complementario

CIPF: Centro de Investigación Príncipe Felipe

CPM: Conteos por millón

DNA: Ácido desoxirribonucleico

ddNTP: dideoxinucleótido trifosfato

DE: *Differentially expressed*, diferencialmente expresado

dNTP: desoxinucleótido trifosfato

EE: E*qually expressed*, expresado de forma similar

EM: *Expectation-maximization*, esperanza-maximización

FPKM: *Fragments Per Kilobase of transcript per Million mapped reads*, fragmentos por kilobase de transcrito por millón de lecturas mapeadas

GB: *Gigabytes*

kb: kilobases

KEGG: *Kyoto Encyclopedia of Genes and Genomes*, enciclopedia de Kioto de genes y genomas

mRNA: RNA mensajero

NGS: *Next-Generation Sequencing*, secuenciación masiva

PCA: Análisis de componentes principales

PCR: *Polymerase chain reaction*, reacción en cadena de la polimerasa

pre-mRNA: mRNA precursor

RNA: Ácido ribonucleico

RPKM: *Reads Per Kilobase of transcript per Million mapped reads*, lecturas por kilobase de transcrito por millón de lecturas mapeadas

snRNP: Ribonucleoproteínas pequeñas nucleares

SR: Serina-arginina

TB: *Terabytes*

TMM: *Trimmed Mean of M-values*, Media recortada de los logaritmos de los ratios de expresión

TPM: Transcritos por millón

UTR: *Untranslated region*, región no traducida

1. INTRODUCCIÓN

1.1. *Splicing* alternativo

Según el dogma central de la biología molecular, propuesto por el famoso científico Francis Crick (1970), para que se produzca la expresión génica, las RNA polimerasas sintetizan RNA a partir del DNA mediante el fenómeno conocido como transcripción, mientras que el RNA contiene las instrucciones para que los ribosomas sinteticen proteínas en el proceso de traducción. Este dogma se ha ido adaptando a los nuevos descubrimientos realizados (figura 1.1).

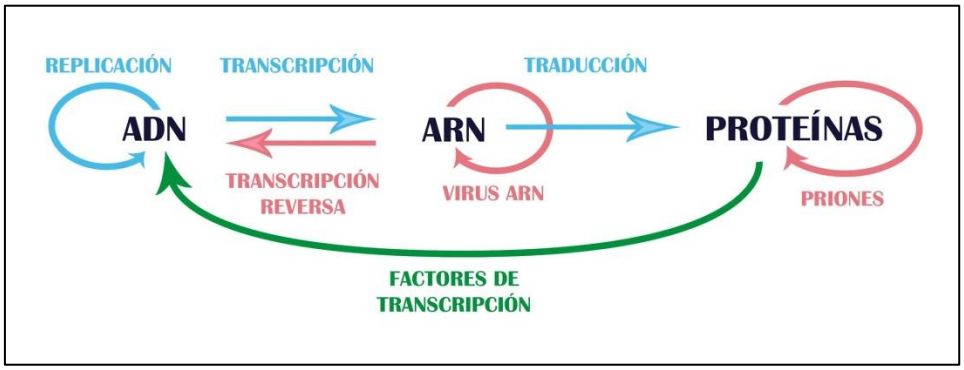


Figura 1.1. Dogma central de la biología molecular actualizado. Esquema que muestra como el DNA, que se multiplica mediante la replicación, pasa a RNA mediante la transcripción y este a proteínas mediante la traducción. Además, el RNA puede pasar a DNA mediante la transcripción reversa, algunos virus pueden replicar su RNA y los priones son proteínas que inducen su conformación tridimensional a otras proteínas, por lo que se da una especie de replicación proteica. Finalmente, las proteínas influyen en la expresión génica actuando sobre el DNA mediante factores de transcripción (Figura adaptada de Martini, 2010).

En el primer paso de la transcripción a partir del DNA se obtiene pre-mRNA, es decir, RNA mensajero que debe sufrir ciertas modificaciones para llegar a ser un mRNA maduro preparado para ser traducido a proteína. Algunas de estas modificaciones son el ***capping*** (adición de una capucha de guanosina en el extremo 5' del pre-mRNA), la **poliadeninación** (adición de la cola poli-A en el extremo 3' del pre-mRNA) y el ***splicing*** (eliminación de los intrones y unión de los exones). La figura 1.2 esquematiza este proceso.

Aunque existen intrones de algunos organismos y orgánulos (mitocondrias y cloroplastos) que son autocatalíticos, los intrones de los organismos eucariotas superiores necesitan de la actividad de una serie de enzimas para ser escindidos. Estas enzimas forman el **espliceosoma**, un conjunto de distintas snRNPs, proteínas con una parte ribonucleica que reconocen secuencias intrónicas y que se encuentran en el núcleo. El mecanismo consiste, básicamente, en la unión de las snRNPs en distintos puntos del intrón, curvándolo y formando un lazo que es cortado (Konarska *et al*., 1985). Este proceso está esquematizado en la figura 1.3 (A).

Durante el procesamiento del pre-mRNA en muchas ocasiones se da el fenómeno de ***splicing* alternativo**. Consiste en un método de regulación en el que a partir de un mismo transcrito primario de RNA se puede dar lugar a moléculas diferentes de mRNA, y con ello a distintas **isoformas** proteicas. Este proceso se da principalmente en organismos eucariotas, aunque también en algunos virus y en células procariotas de forma bastante diferente (Reinhold-Hurek y Shub, 1992).

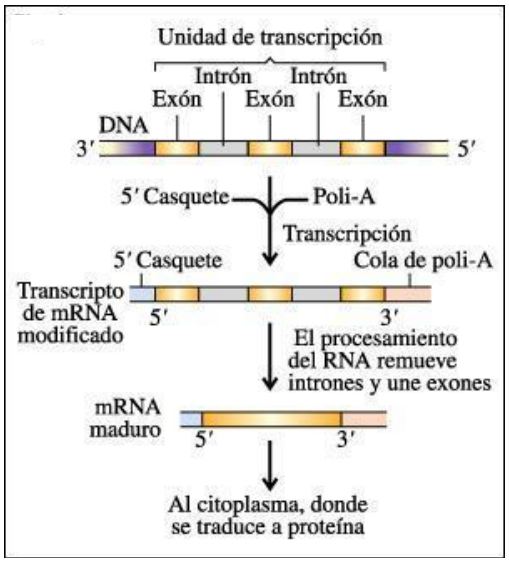


Figura 1.2. Procesamiento del pre-mRNA a mRNA maduro. En primer lugar, se añaden el 5' CAP (o 5' casquete) y la cola poli-A. A continuación se cortan los intrones y se unen los exones (González, 2009).

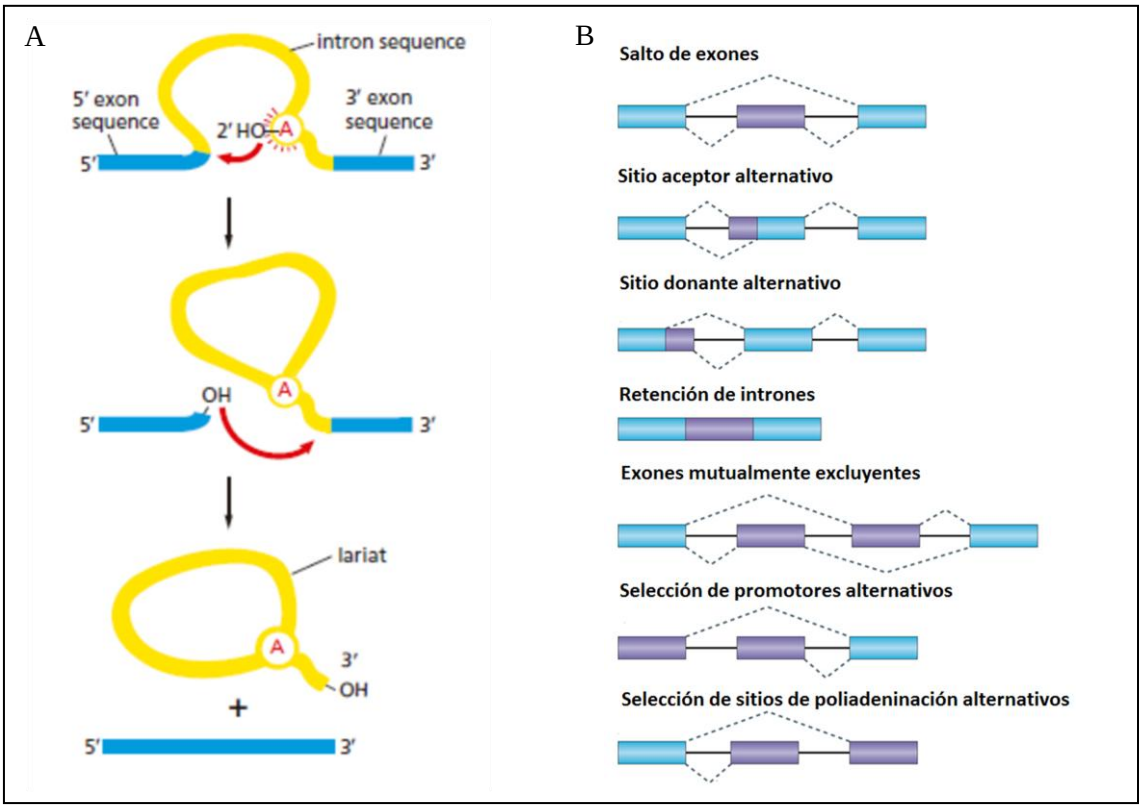


Figura 1.3. (A) Mecanismo de *splicing*. En primer lugar, un nucleótido adenina de la secuencia del intrón (marcado en rojo) ataca el sitio de splicing 5', que se suelta y se une covalentemente a la adenina formando un lazo. El grupo 3'-OH libre del exón reacciona con el inicio del siguiente exón, juntándose los dos exones y liberándose el intrón. Todo este proceso está asistido por el espliceosoma (Alberts *et al.*, 2015). **(B) Mecanismos de *splicing* alternativo.** Las cajas azules representan exones constitutivos y las moradas regiones que se pueden considerar como intrones o como exones (figura adaptada de Keren *et al.*, 2010).

Así, las células eucariotas son capaces de obtener distintas isoformas a partir de un solo gen en función de lo que se considere como exón y como intrón. Aunque en algunos casos el *splicing* alternativo se produce debido a la ambigüedad de la secuencia de los intrones, en muchos otros casos este proceso está regulado. La determinación de los sitios de corte para cada

alternativa de *splicing* es llevada a cabo por las proteínas nucleares SR, que cortan por los sitios adecuados en función de la isoforma proteica que es obtenida. Se conocen cinco métodos por lo que las células son capaces de realizar el *splicing* alternativo (Sammeth *et al*., 2008), esquematizados en la figura 1.3 (B):

- **Salto de exones**. Hay exones constitutivos que siempre se comportan como tales y exones *cassete* que a veces son cortados del pre-mRNA (son "saltados") y otras veces no.
- **Sitio aceptor alternativo**: El sitio de *splicing* en 3' (sitio aceptor) puede cambiar.
- **Sitio donante alternativo**: También puede variar el sitio de *splicing* en 5' (sitio donante).
- **Retención de intrones**: Se trata de un fallo a la hora de reconocer y eliminar ciertos intrones.
- **Exones mutuamente excluyentes**: Hay parejas de exones excluyentes, de manera que si uno se añade al mRNA el otro se elimina como si fuera un intrón.

Además, otros dos mecanismos que normalmente se asocian a la regulación de la expresión más que al propio *splicing* alternativo, pero que también producen cambios en la molécula de mRNA maduro (Black, 2003), son:

- **Selección de promotores alternativos**: dentro de la secuencia de un gen puede que se reconozcan distintas secuencias como promotores, produciéndose distintos mRNAs de diferente longitud.
- **Selección de sitios de poliadeninación alternativos**: El extremo 3' del mRNA se corta antes de añadirse la cola poli-A. Este corte puede producirse a diferentes distancias del extremo, produciéndose también mRNAs de distinta longitud.

Este proceso ha sido seleccionado evolutivamente porque presenta una serie de ventajas sobre las células, ya que permite almacenar la información de forma más económica (a partir de una sola secuencia de DNA se pueden obtener varias proteínas), ahorrando así espacio y energía (Kelemen *et al*, 2012). Se ha visto que el *splicing* alternativo regula la unión entre proteínas, entre proteínas y ácidos nucleicos y entre proteínas y membranas. Además, también regula la localización celular, la interacción con ligandos y las propiedades enzimáticas. Aunque en la mayoría de los casos los cambios de cada isoforma son pequeños, las células coordinan muchos cambios de isoformas que producen efectos en la supervivencia o en la proliferación celular, por ejemplo. Además, se ha sugerido que este sistema permite obtener nuevas proteínas cambiando los sistemas de regulación, permitiendo una evolución más rápida (Keren *et al*., 2010) y también contribuyendo a las diferencias entre las especies (Pan *et al*., 2005).

Distintos individuos de la misma especie pueden presentar diferencias en cuanto al *splicing* alternativo, contribuyendo a los distintos fenotipos de la población (Hull *et al*., 2007), por lo que es muy importante estudiarlo para entender mejor la expresión génica.

Este fenómeno tiene una gran importancia en los procesos de diferenciación celular. De hecho, se estima que en más del 90% de los genes humanos se da *splicing* alternativo en diferentes tejidos y tipos celulares (Chen *et al*., 2015). Se ha demostrado que las células madre, de las cuales se desarrollan los distintos tipos celulares, están muy influenciadas por el control de la expresión génica mediante *splicing* alternativo, que permite la síntesis de mRNAs e isoformas proteicas específicos de tejidos (Fu *et al*., 2009).

Es importante destacar el papel del *splicing* alternativo en distintas patologías. Se han asociado alteraciones en este proceso a enfermedades neurodegenerativas (Mills y Janitz, 2012), musculares (Poulos *et al*., 2011), cardíacas (Xu *et al*., 2005), etc. Tienen especial importancia

los descubrimientos que implican el *splicing* alternativo en mecanismos relacionados con el cáncer, como la oncogénesis, la supresión tumoral o la metástasis (Hagen y Ladomery, 2012).

El estudio del *splicing* alternativo ha permitido, por una parte, comprender mejor la regulación de la expresión génica en organismos eucariotas y, por otra, desarrollar fármacos efectivos para combatir distintas enfermedades (Tang *et al*., 2013) y biomarcadores para el diagnóstico y la prognosis del cáncer (Yi y Tang, 2011).

1.2. RNA-Seq

Los recientes avances en el campo de las *ómicas* han resultado en la disponibilidad de un amplio abanico de tecnologías de alto rendimiento (o *high throughput*) que permiten el estudio de la biología celular a diferentes niveles de organización molecular. Entre estas tecnologías destacan las tecnologías de secuenciación masiva (NGS), que en los últimos años han permitido rebajar el precio de la secuenciación de DNA hasta el punto de que una gran cantidad de laboratorios de biología ya se puede permitir enfocar sus investigaciones biomédicas a nivel de genomas enteros y sistemas.

Se denomina secuenciación de DNA a la determinación del orden de las bases nitrogenadas (adenina, timina, guanina y citosina) de una molécula de DNA. Las tecnologías de secuenciación han avanzado mucho desde sus inicios en la década de 1970. Durante muchos años la técnica de secuenciación más usada fue el método de Sanger (Sanger y Coulson, 1975) hasta que, a principios del siglo XXI, se desarrollaron las técnicas de secuenciación masiva que, por su eficiencia, permitieron bajar considerablemente el precio de secuenciación y con ello se abrieron muchas posibilidades de investigación en campos como la genómica o la biología de sistemas.

Aunque actualmente coexisten distintas tecnologías NGS (Shendure y Ji, 2008), una de las más utilizadas es la de la empresa **Illumina**, basada en la secuenciación por síntesis (figura 1.4). El primer paso para utilizar esta tecnología es preparar librerías de DNA a partir de las muestras. Resumidamente, este paso consta de la purificación del DNA a partir de las células estudiadas, su fragmentación en moléculas más pequeñas y la unión de adaptadores en los extremos de cada molécula. El siguiente paso es la realización de PCR en puente, en la que se añaden las moléculas resultantes del paso anterior desnaturalizadas a un panel que contiene los oligonucleótidos complementarios a los adaptadores. En cada región del panel se une uno o ningún fragmento y, después de realizar la PCR, se obtienen muchas copias de cada fragmento sobre los que se detecta la fluorescencia emitida en los distintos ciclos de secuenciación. Se trata de una polimerización con ddNTPs. En cada ciclo se añaden los 4 ddNTPs y, como están bloqueados, se detiene la elongación cuando se incorpora uno a cada molécula. Se mide entonces la fluorescencia, ya que los ddNTP añadidos están marcados cada uno de un color. Después se trata químicamente para convertir los ddNTPs añadidos en dNTPs y se elimina la etiqueta fluorescente. Este ciclo se repite muchas veces, obteniendo las secuencias de todos los fragmentos.

RNA-Seq es una metodología que se basa en aplicar alguna de las técnicas NGS disponibles para secuenciar moléculas de RNA. Por ejemplo, se puede utilizar la secuenciación con Illumina explicada anteriormente con la particularidad de que es necesario capturar los RNA de las células y retrotranscribirlas para obtener cDNA, que es lo que se secuencia en lugar del propio genoma. Para separar los mRNA de otras moléculas de RNA se utilizan bolas con colas poli-T unidas covalentemente que se hibridan a las colas poli-A de los transcritos (Morin *et al*., 2008). Dependiendo de lo que se quiera estudiar se utilizan los mRNA unidos a estas bolas (como es el caso) o el resto, los RNA no codificantes, que también aportan información muy útil para analizar la expresión génica.

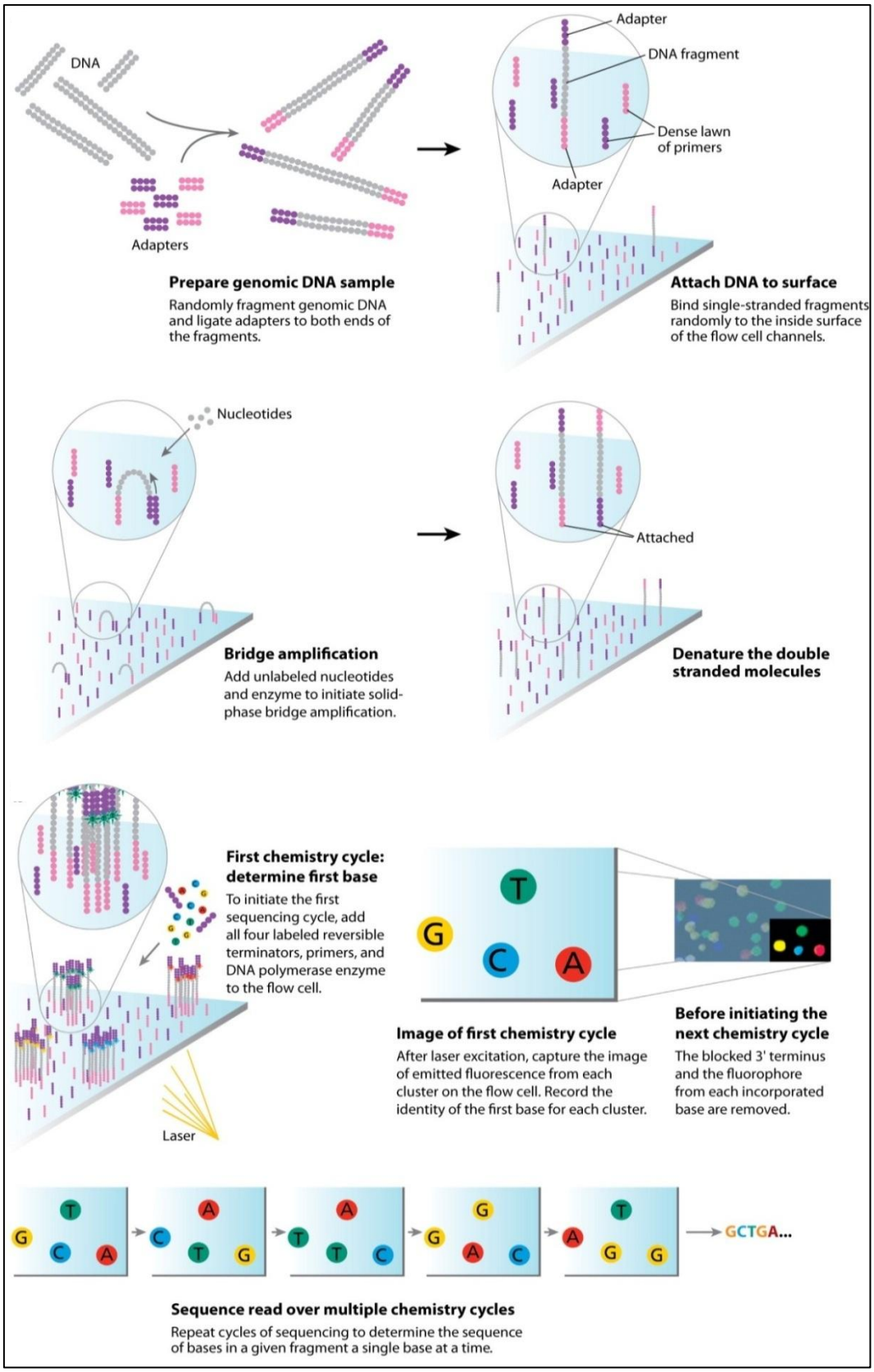


Figura 1.4. Metodología de secuenciación con Illumina. Se añaden *primers* a los fragmentos previamente amplificados y se añaden los 4 nucleótidos bloqueados con DNA polimerasa. Después de incorporarse cada nucleótido, se limpia la polimerasa y los nucleótidos sobrantes, se añade un tampón de escaneado y el sistema óptico detecta la fluorescencia del panel. Una vez escaneado, se añaden productos para desbloquear los nucleótidos y las etiquetas fluorescentes (Mardis, 2008).

Una vez obtenidas las secuencias, se debe ensamblar el transcriptoma para cuantificar los transcritos. Hay dos posibles aproximaciones para ello: el ensamblaje *de novo* y el ensamblaje con un genoma de referencia. En la primera se intenta obtener el transcriptoma sin apoyarse en ninguna información de secuencia previa, algo computacionalmente muy complicado y que requiere el apoyo de secuenciaciones Sanger para obtener lecturas más largas que las que se obtienen con las tecnologías NGS (Zerbino y Birney, 2008). En la segunda aproximación se alinean las lecturas sobre un genoma de referencia previamente generado, por lo que se requieren menos recursos informáticos, pero tiene el requisito de que se debe conocer el genoma de referencia del organismo con el que se está trabajando. Con estas técnicas de ensamblaje se han preparado distintos transcriptomas de referencia (Adamidi *et al*., 2011, Zeing *et al*., 2011, Garg *et al*., 2011) que facilitan el estudio de la expresión génica, ya que permiten mapear las lecturas directamente sobre ellos ahorrando mucho tiempo, aunque no se pueden descubrir nuevos transcritos de esta forma. En este proyecto se optó por mapear las lecturas sobre un transcriptoma de referencia, ya que el descubrimiento de nuevos transcritos no fue uno de los objetivos.

Bien se use un genoma de referencia o un transcriptoma de referencia sobre el que mapear las secuencias de RNA-Seq, cada lectura puede alinearse de forma única, es decir, solo se puede asociar a una posición en la referencia, o de forma ambigua. Los transcritos mapeados de forma ambigua tienen diferente significado dependiendo de si se está empleando un genoma o un transcriptoma de referencia: en el primer caso, se pueden corresponder a genes duplicados o a secuencias repetitivas, por lo que se puede descartar parte de este mapeado; en el segundo, lo más probable es que estas secuencias ambiguas pertenezcan a distintas isoformas del mismo gen, por lo que es importante no descartarlas.

La identificación de los distintos transcritos del mismo gen es un problema debido a la semejanza de secuencia entre ellos. Se puede encontrar un ejemplo en el gen humano WT1, que está formado por 9 variantes de *splicing*, algunas muy similares, y es muy difícil identificar de cuál de las variantes proviene cada lectura (Antón *et al*., 2008). Este problema se agrava con la cantidad de isoformas de los genes y también con lo parecidas que sean estas isoformas entre sí. Como ya se ha explicado, en RNA-Seq se utilizan lecturas cortas, dificultando aún más la tarea de identificar el transcrito del que proviene cada lectura. Por ello en los últimos años han surgido distintos algoritmos de cuantificación de lecturas (ver apartado 1.3.3).

1.3. Herramientas informáticas para el análisis de datos de RNA-Seq

Dada la explosión de datos *high throughput* (de alto rendimiento) acontecida en los últimos tiempos se han tenido que diseñar y adaptar herramientas informáticas para analizar este tipo de datos. Muchas de estas herramientas son software libre, es decir, programas creados por la comunidad científica que se ponen a disposición de cualquiera que pueda necesitarlos.

1.3.1. El sistema operativo Ubuntu

En el ámbito de la bioinformática es muy habitual el uso de sistemas operativos basados en GNU/Linux debido a que son libres, por lo que se pueden usar sin pagar caras licencias como en el caso de Windows, por ejemplo. Además, muchas de las herramientas bioinformáticas están diseñadas para ser usadas en este tipo de sistemas operativos.

Uno de los sistemas operativos basados en GNU/Linux más utilizado es Ubuntu. Ello es debido a que presenta una interfaz que facilita el manejo a usuarios poco familiarizados con Linux y, al mismo tiempo, conserva la versatilidad y la seguridad que caracterizan a estos sistemas operativos, por lo que fue el sistema operativo elegido para realizar el proyecto.

Es importante destacar la línea de comandos, presente en todos los sistemas operativos, pero muy utilizada en el caso concreto de los basados en GNU/Linux. No es más que un método que permite al usuario dar órdenes a los programas. Se trata de una ventana en la que se

escriben los comandos que son interpretados y ejecutados. En esta misma ventana se muestra la respuesta del sistema operativo al comando introducido. La única desventaja de la línea de comandos respecto a los entornos de escritorio es su complejidad. No obstante, una vez dominada se obtiene mayor control y velocidad, el acceso remoto es muy sencillo, consume muy pocos recursos informáticos y es posible automatizar tareas mediante la preparación de *scripts* en lenguaje de programación Bash. Además, los programas ejecutados en la línea de comandos permiten explorar ficheros de gran tamaño, como los generados por las tecnologías NGS, que ni siquiera es posible abrirlos mediante programas tradicionales.

1.3.2. El lenguaje de programación R

R (Ihaka y Gentleman, 1996) es un entorno y lenguaje de programación para análisis estadístico y gráfico. Es muy útil para realizar este tipo de análisis, y por ello es muy utilizado en campos como las matemáticas, la bioinformática o la biomedicina. La mayoría de las tareas relacionadas con este proyecto se realizaron con este lenguaje (aunque también se usaron puntualmente otros como Python o Bash). Existen diferentes interfaces gráficas para R, aunque la que se utilizó fue **RStudio** (Racine, 2012), que es potente, productiva y se puede usar en cualquier sistema operativo.

La comunidad científica desarrolla constantemente herramientas escritas en R (paquetes o librerías) que son organizadas en distintos repositorios dependiendo de su aplicación. El proyecto **Bioconductor** (Huber *et al.*, 2015) recopila las herramientas útiles para el análisis y la comprensión de grandes cantidades de datos genómicos. Algunos de los paquetes utilizados en el trabajo provienen de Bioconductor.

1.3.3. Cuantificación de la expresión de isoformas

Una vez obtenidos los archivos con las secuencias leídas por el secuenciador (lecturas de secuenciación o *reads*), es necesario identificar de qué isoforma proviene cada una, es decir, calcular cuantas moléculas de DNA secuenciadas pertenecen a cada isoforma del transcriptoma de referencia. Se le llama conteo a cada lectura de secuenciación que se mapea a un transcrito. A este paso de asignar un número de conteos a cada isoforma del genoma se le denomina **cuantificación**, y existen distintos paquetes de software que realizan este proceso (Garber *et al.*, 2011).

Algunos programas, como SOLAS (Richard *et al*., 2010), RUM (Grant *et al*., 2011), SpliceTrap (Wu et al. 2011) o SpliceSeq (Ryan *et al*. 2012) usan un mapeo sobre el genoma de referencia para cuantificar los transcritos. Otros, como RSEM (Li *et al*., 2011), IsoEM (Nicolae *et al*., 2011), NEUMA (Lee *et al*., 2011), BitSeq (Glaus *et al*., 2012), o eXpress (Roberts *et al*., 2013) se basan en mapear las lecturas sobre un transcriptoma de referencia. Esto aumenta la velocidad de cuantificación, aunque impide el descubrimiento de nuevas isoformas.

Dado que el descubrimiento de nuevas isoformas no fue uno de los objetivos del trabajo, se optó por utilizar métodos basados en el transcriptoma de referencia. La cuantificación con eXpress ya había sido realizada previamente al inicio del trabajo y se decidió compararla con RSEM porque es uno de los métodos más utilizados por la comunidad científica, incluso en proyectos internacionales de prestigio como el proyecto ENCODE (ENCODE Project Consortium, 2004).

1.3.4. Expresión diferencial

Una vez cuantificada la expresión de todos los transcritos de las diferentes muestras, se puede optar por realizar un análisis de expresión diferencial, que consiste en la identificación de aquellas isoformas que presentan un cambio medio de expresión estadísticamente significativo entre las distintas condiciones experimentales estudiadas. El programa elegido para ello fue **EBSeq** (Leng *et al*., 2013). Las ventajas de este programa respecto a otras herramientas de

análisis de expresión diferencial para el tipo de datos con el que se trató son las siguientes: permite detectar la expresión diferencial tanto a nivel de genes como a nivel de isoformas, mientras que otros programas no están optimizados para analizar isoformas, como DESeq (Anders y Huber, 2010) o edgeR (Robinson *et al*., 2010); es capaz de analizar datos de más de dos condiciones; es adecuado para la cuantificación tanto de eXpress como de RSEM, en contraste a otros programas que realizan su propia cuantificación, como DiffSplice (Hu *et al*., 2013) o DEXSeq (Anders *et al*., 2012) y, finalmente, algunos programas de análisis de expresión diferencial de isoformas, como MISO (Katz *et al.*, 2010) o FDM (Singh *et al*., 2010) solo tienen en cuenta las diferencias en el *splicing*, pero no el nivel de expresión.

1.4. El proyecto STATegra

Este trabajo se realizó en el laboratorio de Genómica de la Expresión Génica del Centro de Investigación Príncipe Felipe (CIPF), que coordina el proyecto europeo **STATegra**, financiado por el programa *FP7 Health*. El objetivo de este programa es llenar el hueco existente entre las herramientas de análisis estadístico, generalmente optimizadas para un solo tipo de datos, y la necesidad creciente de los investigadores biomédicos de analizar el comportamiento celular. En el proyecto colaboran expertos en estadística, biomedicina, genómica y desarrollo de software de manera que los estadísticos son conscientes de las necesidades de los científicos experimentales, entienden sus problemas y la generación de datos y crean soluciones analíticas.

Para diseñar los métodos estadísticos más apropiados para el procesamiento y el análisis de los datos en un contexto real, dentro del proyecto STATegra se generó un conjunto completo de datos experimentales de distintas *ómicas*, entre los que se encuentran los datos de RNA-Seq utilizados en este trabajo.

1.4.1. Diferenciación de células B

Los datos experimentales usados en el proyecto STATegra tienen como objetivo estudiar el rol del factor de transcripción Ikaros durante la diferenciación de células B. Las células B (o linfocitos B) son un tipo de linfocitos (células del sistema inmunitario adaptativo) que reconocen antígenos y segregan anticuerpos para ese antígeno. Son claves, por tanto, en la respuesta inmunitaria humoral, muy importante para la supervivencia de los organismos vertebrados (Delves y Roitt, 2003).

A grandes rasgos, las células B provienen de células pre-B, que a su vez son generadas a partir de células madre en la médula ósea (Rossi *et al*., 2006). Concretamente, el sistema estudiado fue el sistema B3, que modeliza la transición desde el estadío de diferenciación pre-BI a pre-BII. En el primero, los progenitores de las células B tienen una alta capacidad de proliferación, siendo capaces de auto-renovarse, mientras que en el segundo se detiene la proliferación y se estimula la diferenciación celular.

En respuesta a distintas señales biológicas, las células pre-BI se diferencian a células pre-BII, y una de las proteínas clave en este proceso es Ikaros. Ikaros es un factor de transcripción que regula la expresión de diversos genes gracias a su estructura de dedo de zinc, que le permite unirse a secuencias de DNA específicas (Merkenschlager, 2010). La línea celular estudiada expresa una versión inducible de Ikaros por 4-hidroxi tamoxifeno (Ikaros-ERt2).

En el experimento de STATegra se analizaron dos poblaciones de células pre-BI: unas en las que se activó el factor de transcripción Ikaros y por tanto se diferenciaron a células pre-BII, y otras a las que no se les activó Ikaros, por lo que no se diferenciaron, constituyendo estas últimas el control del experimento. Las dos poblaciones de células se analizaron en diferentes tiempos después de activar Ikaros en los cultivos: 0, 2, 6, 12, 18 y 24 horas. Para cada condición y tiempo se hicieron tres réplicas biológicas. Por tanto, finalmente se analizaron 36 muestras (células con y sin Ikaros activado, a 6 tiempos y con 3 réplicas biológicas).

2. OBJETIVOS

El objetivo principal de este trabajo es el estudio de la expresión alternativa de isoformas a partir de datos de secuenciación masiva. En concreto, se disponía de datos de RNA-Seq procedentes de un sistema de de diferenciación de células B en ratón. Por una parte, se compararán distintos métodos de cuantificación de isoformas (eXpress y RSEM) con el fin de entender cuáles son las diferencias en su funcionamiento, en qué circunstancias producen resultados similares y en cuáles existen discrepancias entre los dos métodos y, en definitiva, poder elegir justificadamente la cuantificación de un método u otro para la realización de análisis posteriores. Por otra parte, se analizará la expresión diferencial de isoformas a lo largo del tiempo durante la diferenciación de células B para compararla con la expresión diferencial de sus genes correspondientes. Para ello, se elegirá un método adecuado entre todos los disponibles, de acuerdo con las características de los datos del proyecto, y se intentará otorgar sentido biológico al comportamiento de distintos genes y de sus isoformas.

Para lograr estos objetivos se propusieron una serie de objetivos parciales:

1. Cuantificación de las isoformas génicas a partir de datos de RNA-Seq mediante el programa RSEM.

2. Comparación de distintos aspectos de la cuantificación realizada por eXpress y por RSEM.

3. Dilucidación de los motivos de las discrepancias entre los dos métodos de cuantificación.

4. Selección de los métodos de pre-procesamiento más adecuados para los datos tratados, desde la normalización hasta el filtrado de los resultados de cuantificación.

5. Aplicación de un método de análisis de expresión diferencial de isoformas.

6. Análisis de los resultados de la expresión diferencial y contextualización con el entorno biológico en cuestión.

3. MATERIAL Y MÉTODOS

3.1. Datos de expresión de transcritos

Los datos de RNA-Seq con los que se trabajó se obtuvieron mediante una serie de experimentos de secuenciación masiva con la tecnología Illumina. Para preparar la librería, se capturaron los transcritos con cola poli-A, enriqueciendo las muestras con mRNAs maduros. El tipo de protocolo para preparar la librería de cDNA fue específico de hebra. Esto significa que es posible saber si cada fragmento proviene de la hebra positiva del DNA (la que va en sentido 5' a 3') o de la hebra negativa (la que va en sentido 3' a 5'). El protocolo específico de hebra facilita el análisis y cuantificación de transcritos antisentido o solapantes, además del ensamblaje *de novo*, por lo que suele ser preferible usarlo.

El tipo de librería fue *paired-end*, que se diferencia de la secuenciación *single-end* en que se secuencian los dos extremos de los cDNAs en lugar de tan solo uno de ellos. Las librerías *paired-end* son más útiles para realizar análisis de expresión de isoformas (Katz *et al.*, 2011), aunque las librerías *single-end* son más baratas y suficientes para estudiar la expresión a nivel de genes en organismos muy estudiados.

La profundidad de secuenciación o tamaño de librería, es decir, el número de lecturas secuenciadas de cada muestra, fue de aproximadamente 100 millones. Durante el proceso de secuenciación, cada una de las muestras se analizó en varias tandas de secuenciación con el fin de llegar a la profundidad de secuenciación deseada. Se trata, por tanto, de una secuenciación profunda que permitió cuantificar de forma precisa transcritos con baja expresión. Se sabe que a mayor profundidad de secuenciación, más transcritos se detectan y la cuantificación es más precisa (Mortazavi *et al*., 2008).

El tamaño medio de cada lectura de secuenciación fue de unos 100 pares de bases. Por tanto se obtuvieron lecturas cortas, lo cual es un problema para diferenciar de qué isoforma génica proviene cada transcrito. Las lecturas cortas son muy características de las tecnologías de secuenciación masiva, siendo una de las desventajas principales y el motivo por el que aún se necesiten técnicas de secuenciación más tradicionales, con lecturas mucho más largas, para ciertos objetivos.

3.2. Recursos informáticos del CIPF

Durante la realización de este trabajo fue necesario aprovechar el clúster del CIPF. Un clúster es un conjunto de ordenadores conectados entre sí por una red de alta velocidad, por lo que todos ellos se comportan como si fueran una sola computadora. Estos conglomerados informáticos tienen un alto rendimiento de computación y pueden almacenar mucha memoria, por lo que tienen diversas aplicaciones. En la actualidad, el CIPF cuenta con un nodo de computación de 362 procesadores y 2.6 TB de memoria, además de un espacio de almacenamiento de 262 TB.

Para lanzar varios procesos al clúster con el fin de que sean computados de forma paralela es necesario preparar un tipo especial de *scripts* en el lenguaje de programación Bash llamado *arrayjob*. Un *arrayjob* permite gestionar el sistema de colas, que es la forma de gestionar los recursos del clúster, asignando trabajos a las distintas máquinas, dando diferente prioridad a los procesos en ejecución, etc.

Como ya se ha explicado, cada una de las muestras del estudio se secuenció en dos tandas de secuenciación para llegar a la profundidad de secuenciación deseada. Ello implica que se disponía de un total de 72 archivos con las secuencias, ocupando unos 5 GB cada uno. Esta

cantidad de datos no se puede almacenar ni manejar con un ordenador común, y por ello fue necesario el uso del clúster.

3.3. Cuantificación de las isoformas génicas

El primer paso para analizar la expresión alternativa de isoformas en el tiempo fue cuantificar cada uno de estos transcritos a partir de los datos de secuenciación masiva de RNA-Seq. Debido a la semejanza que existe entre las isoformas, esta cuantificación es una tarea complicada. Por ello existen varios programas especializados en ello. Se usaron dos de estos programas, eXpress y RSEM, para realizar dicha cuantificación y comparar los resultados, usando los del programa que se consideró que obtuvo mejores resultados en los análisis posteriores.

Ambos métodos usan el mapeado de Bowtie (Langmead *et al.*, 2009) para calcular los conteos y otras medidas explicadas en las tablas 3.1 y 3.2.

Tanto eXpress como RSEM utilizan el algoritmo EM (Dempster *et al.*, 1977), una aproximación general para el cálculo iterativo de estimaciones de máxima verosimilitud de los parámetros de un modelo estadístico. Cada iteración del algoritmo consiste en un paso de esperanza (paso E) en el que se genera una función para el valor esperado de la verosimilitud evaluada utilizando la estimación disponible de los parámetros desconocidos, seguido de un paso de maximización (paso M) en el que se calculan estimaciones de máxima verosimilitud de los parámetros maximizando la verosimilitud esperada del paso anterior. Los parámetros calculados en el paso M son usados en el siguiente paso E, repitiéndose el proceso de forma iterativa.

RSEM y eXpress aplican el algoritmo EM de la siguiente manera: en el paso E el algoritmo asigna lecturas a secuencias de referencia con una probabilidad acorde a los parámetros de abundancia y en el paso M actualiza las abundancias a la solución de máxima verosimilitud basándose en las asignaciones realizadas en el paso E.

3.3.1. Cuantificación con eXpress

La cuantificación de isoformas mediante el uso del programa eXpress ya había sido realizada con anterioridad por otros integrantes del proyecto STATegra, por lo que solo se tuvieron que descargar los archivos *output* desde el servidor del CIPF a través de internet. El eXpress da como resultados para cada muestra una tabla con varias columnas. El significado de cada columna se explica en la tabla 3.1.

Tabla 3.1. Nombre, ejemplos y significado de cada columna del archivo de resultados que genera el programa eXpress.

Número de columna	Nombre de columna	Ejemplo	Descripción
1	bundle_id	ENSMUSG00000021774	Identificación del gen al que pertenece el transcrito.
2	target_id	ENSMUST00000160880	Identificación del transcrito.
3	length	2182	Número de pares de bases del transcrito.
4	eff_length	783.136288	Longitud del transcrito ajustado por su tamaño, especificidad de la secuencia y posición relativa.
5	tot_counts	99	Número de fragmentos que se mapean (única o ambiguamente) a este transcrito.
6	uniq_counts	7	Número de fragmentos que se mapean únicamente a este transcrito.

7	est_counts	26.702456	Número estimado de fragmentos generados desde este transcrito en el experimento de secuenciación.
8	eff_counts	74.399258	Número estimado de fragmentos generados desde este transcrito en el experimento de secuenciación ajustado por longitud.
9	ambig_distr_alpha	3.154652	Parámetro α de la distribución β-binomial de las lecturas de secuenciación ambiguas.
10	ambig_distr_beta	2.293653	Parámetro β de la distribución β-binomial de las lecturas de secuenciación ambiguas.
11	fpkm	3.514176	Abundancia relativa estimada del transcrito en la muestra en unidades de fragmentos por kilobase por millón mapeados.
12	fpkm_conf_low	2.119151	Límite inferior del intervalo de confianza del 95 % para los FPKM.
13	fpkm_conf_high	4.909200	Límite superior del intervalo de confianza del 95 % para los FPKM.
14	solvable	T	Valor binario (T/F) que indica si la función de probabilidad tiene un solo máximo.
15	tpm	2.347222e+05	Transcritos por millón.

3.3.2. Cuantificación con RSEM

El paquete de software RSEM está compuesto principalmente por dos programas. El primero de ellos, *rsem-prepare-reference*, prepara el transcriptoma de referencia a partir de archivos FASTA[1] o GTF[2], mientras que el segundo, *rsem-calculate-expression*, calcula la expresión de los diferentes transcritos.

Para ejecutar el primer programa se preparó un script en lenguaje de programación Bash (Anexo I) que permitió lanzarlo en el clúster del CIPF, dentro del entorno de línea de comandos de Linux. Los parámetros introducidos al programa fueron *--no-polyA*, que evita que se añadan colas poli-A a las isoformas de referencia, algo que en el tipo de datos tratados empeora el alineamiento, y *--bowtie2*, que indica a RSEM que use Bowtie2 (Langmead y Salzberg, 2012) para mapear las secuencias. Cabe destacar que como transcriptoma de referencia se usó el producido a partir del genoma de ratón mm10, el mismo que se usó para ejecutar el eXpress, para tener un resultado comparable entre los dos programas.

Al ejecutar dicho *script* se generaron distintos archivos necesarios para realizar el siguiente paso del RSEM. Este segundo proceso fue llevado a cabo por el programa *rsem-calculate-expression*, que debe cuantificar juntas todas las secuencias de la misma muestra y tanda del secuenciador, y por ello fue necesario preparar un *arrayjob* (Anexo II) que pone en cola una lista de procesos en el clúster y se van realizando en función de la prioridad de dichos procesos. Como parámetros del programa se usaron: *--bowtie2*, que indica al programa que use el Bowtie2 para mapear las secuencias; *--no-bam-output*, que evita que el programa genere archivos BAM que no fueron necesarios en este caso; *--p 8*, que indica que se usen 8 hilos del ordenador para ejecutar el programa; *--samtools-sort-mem 3G*, que indica que se utilicen 3 GB de memoria RAM para ejecutar el paquete SAMtools (Li *et al*., 2009) y *--paired-end*, que indica que el método de secuenciación es *paired-end* y no *single-end*.

Al igual que eXpress, RSEM genera un archivo para cada muestra que contiene diversas columnas, cuyo significado se detalla en la tabla 3.2.

[1] Información sobre los archivos FASTA en la web http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml.
[2] Información sobre los archivos GTF en la web http://www.ensembl.org/info/website/upload/gff.html.

Tabla 3.2. Nombre, ejemplos y significado de cada columna del archivo de resultados que genera el programa RSEM.

Número de columna	Nombre de columna	Ejemplo	Descripción
1	transcript_id	ENSMUST00000000003	Nombre del transcrito.
2	gene_id	ENSMUSG00000060550	Nombre del gen al que pertenece el transcrito.
3	length	4209	Número de pares de bases del transcrito.
4	effective_length	4065.53	Longitud del transcrito teniendo en cuenta solo las posiciones que pueden generar un fragmento válido.
5	expected_counts	860.18	Suma de la probabilidad de que cada lectura de secuenciación provenga de este transcrito de entre todas las lecturas. Es decir, se corrigen los conteos por el ruido de fondo.
6	TPM	7.50	Transcritos por millón.
7	FPKM	7.23	Abundancia relativa estimada del transcrito en la muestra en unidades de fragmentos mapeados por kilobase y por millón.
8	IsoPct	62,54	Porcentaje de isoformas del gen que representa este transcrito.

3.3.3. Comparación de la cuantificación de eXpress y RSEM

Una vez obtenida la cuantificación de los transcritos con los programas eXpress y RSEM se compararon los resultados con el fin de decidir cuáles se usarían en análisis posteriores. Este análisis comparativo se llevó a cabo utilizando el lenguaje de programación R (Anexo III).

Fue necesario elegir qué medidas de cuantificación usar de entre todas las calculadas por los dos métodos. Un requisito indispensable para esta elección fue que ambas medidas debían ser comparables. Hay tres medidas que cumplen este requisito: conteos estimados (*effective counts* en eXpress y *expected counts* en RSEM), TPM y FPKM. En estudios anteriores del laboratorio se llegó a la conclusión de que la medida que mejor resultados dio fue la de los conteos estimados.

Para obtener la cuantificación a nivel de gen se sumaron las medidas de las isoformas pertenecientes a cada gen, ya que después de todo la expresión de un gen es la suma de la expresión sus posibles isoformas alternativas.

3.3.4. Pre-procesado de los datos de expresión

El primer paso en la preparación de los datos de expresión fue sumar los conteos de las distintas tandas de secuenciación para la misma muestra. De esta forma, se consiguió la profundidad de secuenciación adecuada para cada una de las 36 muestras del experimento.

Los datos deben representar con la máxima fidelidad posible el comportamiento biológico que los ha generado. Por ello, es necesario reducir potenciales sesgos sistemáticos que pueda estar introduciendo la propia tecnología y que distorsionan la señal. A este proceso se le llama **normalización** de los datos, y es un paso imprescindible durante el pre-procesado.

Uno de los métodos de normalización más básicos consiste en corregir por la profundidad de secuenciación, es decir, por el número total de conteos en cada muestra. Esto es necesario porque dos isoformas con el mismo número de conteos en dos muestras diferentes con distinta profundidad de secuenciación no deben considerarse isoformas con la misma expresión,

ya que su expresión relativa es distinta. Así pues, el método de normalización CPM (*counts per million*) transforma los conteos a valores CPM, que se calculan dividiendo los conteos de cada isoforma entre los conteos totales de la muestra y multiplicando por 10^6. Esto se hizo usando la función *rpkm()* del paquete de R NOISeq (Tarazona *et al*., 2011).

Otra normalización muy utilizada para este tipo de datos es el método RPKM (*Reads Per Kilobase and per Million Reads*), ideado por Mortazavi *et al*. (2008). Este método corrige los datos tanto por la profundidad de secuenciación (como CPM) como por la longitud de los transcritos. La corrección por la longitud del transcrito en RNA-Seq es bastante común, ya que un posible sesgo de esta tecnología es que los transcritos más largos tienen, en teoría, mayor probabilidad de que sus fragmentos sean secuenciados. Los RPKM se calculan dividiendo los conteos de cada isoforma entre los conteos totales de la muestra y la longitud del transcrito y multiplicando por 10^9. La función *rpkm()* del paquete NOISeq también se puede emplear para realizar estos cálculos, siempre y cuando se le proporcione la información de la longitud de cada transcrito.

La normalización TMM (*Trimmed Mean of M-values*), propuesta por Robinson y Oshlack (2010), ha alcanzado gran popularidad por sus buenos resultados (Dillies *et al*., 2013). Este método trata de corregir el sesgo producido cuando la distribución de conteos por transcrito a lo largo de una muestra es diferente para distintas muestras. Cuando se observan estas diferencias entre muestras puede ser debido a que algunos de los transcritos en una de las muestras están muy expresados y por tanto acumulan gran parte de las lecturas. Por tanto, el resto de transcritos estarían "infra-cuantificados" y esto dificultaría la comparación de su expresión con otras muestras donde esto no sucede y en las que, por consiguiente, tendrían una cuantificación mayor. El método TMM consiste en utilizar la media recortada de los logaritmos de los ratios de expresión entre muestras para corregir los datos con distribuciones diferentes. Esta normalización se hizo con la función *tmm()* del mismo paquete NOISeq.

Estos tres métodos de normalización son algunos de los más utilizados. En el apartado 4.1.2 se discute cuál de los métodos es más adecuado para los datos analizados en este caso.

Otro paso importante durante el pre-procesado de datos es el filtrado de las isoformas de baja expresión. La estimación de las isoformas poco expresadas suele ser poco fiable, presenta una elevada variabilidad y, por tanto, suele introducir ruido en los datos. Así pues, en los análisis estadísticos de este tipo de datos, el filtrado es una práctica muy extendida, ya que se aumenta la potencia estadística de los métodos. El filtrado se hizo mediante el método CPM de la función *filtered.data()* del paquete NOISeq, que elimina aquellos transcritos que tiene un valor medio de CPM inferior a un determinado valor de corte en todas las condiciones experimentales. El valor de corte elegido para el filtrado fue de 0,5 CPM.

3.3.5. Herramientas estadísticas para el análisis de resultados

La principal herramienta estadística usada fue el **coeficiente de correlación de Pearson**. Se trata de una medida estadística de la relación lineal entre dos variables aleatorias cuantitativas. Este coeficiente puede tomar un valor desde -1 a 1, siendo la correlación mayor cuanto más cerca esté de 1 (correlación positiva) o de -1 (correlación negativa). R incluye la función *cor()* para calcular el coeficiente de correlación, que aplica la expresión 1:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y} \quad (1)$$

Siendo ρ_{XY} el coeficiente de correlación de Pearson, σ_{XY} la covarianza de (X,Y), σ_X la desviación típica de la variable X y σ_Y la desviación típica de la variable Y.

El coeficiente de correlación se usó para medir la semejanza entre los resultados de RSEM y eXpress. Para ello, se calculó la correlación de los valores de cada transcrito en todas las muestras entre los *expected counts* de RSEM y los *effective counts* de RSEM. También se usó el coeficiente de correlación para medir la robustez de cada método. En este caso, se calculó la correlación entre las réplicas de la misma muestra para cada transcrito.

3.3.6. Herramientas para la visualización de resultados

Además de métodos gráficos sencillos como gráficos de barras, *boxplots* (diagrama de caja y bigotes), etc. se utilizaron herramientas de visualización más avanzadas.

Una de estas herramientas es el análisis de componentes principales (PCA), una técnica estadística que permite reducir la dimensión de los datos y ayuda a averiguar las causas de la variabilidad en los mismos. Para ello se realiza una transformación lineal que elige un nuevo sistema de coordenadas para el conjunto de datos originales en el cual la mayor varianza del conjunto de datos es registrado en el primer eje (primer componente principal), la segunda varianza más grande es el segundo eje, etc. El sentido de realizar un PCA sobre los resultados de eXpress y RSEM fue estudiar si uno de los dos métodos ofrece datos más consistentes con el contexto biológico tratado y si la agrupación de las muestras es acorde al diseño experimental. A *priori*, las dos primeras fuentes de variabilidad deberían ser la activación o no activación de Ikaros y el tiempo. Si los datos son adecuados, al representar la primera y la segunda componente principal las muestras se deben agrupar en función de las fuentes de variabilidad mencionadas. Los datos utilizados para realizar el PCA fueron los conteos estimados por eXpress y RSEM filtrados por un valor mínimo de 1 CPM en alguna de las condiciones, normalizados por el método TMM y con el logaritmo en base 2 aplicado (ver *script* de R en Anexo V).

También se utilizaron gráficos *heat map* o mapa de calor. Un mapa de calor permite agrupar las muestras mediante un clúster jerárquico, según la correlación entre ellas, y adjudicar el mismo color a las muestras agrupadas para ver la similitud entre ellas. En este caso, la utilidad de este tipo de gráficos reside en que permite visualizar si las muestras se agrupan por condiciones (indicando que los datos son coherentes con el diseño experimental) o si, por el contrario, las muestras no se agrupan por condiciones (indicando baja calidad de los datos). Por tanto, en este caso es una forma alternativa de evaluar la consistencia entre réplicas dentro de la misma condición. Los mapas de calor se realizaron aplicando la función *heatmap()* de R a la matriz de correlación entre los conteos filtrados por un valor mínimo de 1 CPM en alguna de las condiciones y normalizados a CPM.

Otra herramienta útil a la hora de tratar con lecturas de secuenciación mapeadas es el visor de genomas **IGV** (Robinson *et al*., 2011). Se trata de una herramienta de visualización para la exploración interactiva de datos genómicos que permite ver gráficamente la región de DNA deseada y las secuencias alineadas a esta. El mapeo visualizado fue realizado por otro integrante del proyecto STATegra con el programa TopHat (Trapnell *et al*., 2009). Se realizó un mapeo por cada muestra y tanda de secuenciación, de manera que se generó un archivo de tipo BAM por cada mapeo realizado. Dado que cada muestra fue secuenciada en dos tandas, fue necesario combinar los dos archivos BAM que pertenecen a la muestra estudiada en cada caso. Para ello se usó el comando *merge* del paquete SAMtools, ejecutado en la línea de comandos de Linux. También se usó SAMtools para ordenar (con el comando *sort*) e indexar (con el comando *index*) los archivos BAM combinados, algo necesario para poder visualizarlos en IGV.

Dado el elevado número de transcritos, se creó un tipo de gráfico al que se llamó "gráfico por intervalos" que nos permitió visualizar mejor la relación entre dos variables X e Y. Este gráfico consiste en agrupar los datos de la variable X en intervalos que contienen el mismo número de transcritos, normalmente 200. Para ello, se ordenan los valores de X de menor a mayor y se agrupan los 200 primeros, los 200 segundos, etc. En cada uno de estos intervalos, se

calcula la media de la variable Y para los transcritos que caen en dicho intervalo (ver script en Anexo III). De esta forma, se evita obtener una nube muy densa de puntos con todos los transcritos, de la que es difícil extraer alguna conclusión, y se representa únicamente un punto por cada intervalo creado.

La última herramienta utilizada fue Paintomics (García-Alcalde *et al*., 2011), una aplicación web que permite integrar y visualizar los datos de distintas *ómicas* en las rutas de la base de datos KEGG. Esta herramienta acepta datos de más de 100 especies distintas. Los datos subidos a Paintomics pueden tener distintos tipos de identificadores, ya que los relaciona con los identificadores de KEGG. En el caso de que existan ambigüedades (algo común en los metabolitos, por ejemplo) se le comunica al usuario, que específica exactamente de qué elementos dispone los datos. Una vez ejecutado, el usuario elige las rutas que desea visualizar. Dentro de los nódulos de cada ruta se pintan con distintos colores los datos en función de su valor a lo largo del tiempo. En este caso, los valores introducidos fueron el ratio de expresión de los genes entre las dos condiciones (Ikaros activado y control) a lo largo del tiempo. Colores rojos indican sobreexpresión en Ikaros activado y azules lo contrario. La intensidad del color indica la magnitud de la diferencia (a mayor intensidad, mayor cambio). Paintomics admite también una lista de elementos significativos para que los remarque en las rutas. En este caso se introdujo una lista de los genes expresados diferencialmente.

La base de datos KEGG (Kanehisa y Goto, 2000) es en realidad una colección de distintas bases de datos de genomas, biomoléculas y rutas enzimáticas muy útiles para investigaciones en el campo de la biología de sistemas, por ejemplo. En este caso la base de datos utilizada fue KEGG Pathway, que contiene una colección de mapas de rutas celulares en distintos organismos, incluyendo rutas metabólicas, de señalización, etc.

3.4. Análisis de expresión diferencial de isoformas

Para analizar la expresión diferencial de isoformas se utilizó el paquete de R EBSeq (Leng *et al*., 2013), que se encuentra disponible en el repositorio Bioconductor (ver script de R en anexo VI). El funcionamiento de EBSeq se basa en el uso de una aproximación empírica bayesiana para clasificar las isoformas en distintos patrones de expresión. Utiliza un modelo de distribución de probabilidad con distintos hiperparámetros α y β que describen las fluctuaciones técnicas y biológicas de los datos. Estos hiperparámetros son estimados mediante el algoritmo EM, siendo α común para todas las isoformas, mientras que β depende del número de isoformas del gen.

El paquete EBSeq requiere una serie de datos para realizar su tarea. En primer lugar necesita una lista de todos los transcritos y otra de los genes a los que pertenece cada uno de estos transcritos. Este requerimiento es debido a que EBSeq tiene en cuenta que la cuantificación de los genes con una sola isoforma presenta menos variabilidad que la de los genes con 2 o más isoformas, por lo que trata los datos de forma distinta dependiendo del número de isoformas de cada gen. EBSeq también requiere factores de normalización para cada muestra para normalizar los datos. Como factores de normalización se introdujeron los TMM, calculados con la función *calcNormFactors()* del paquete edgeR (Robinson *et al*., 2010). No se introdujeron los datos de todas las muestras, sino solamente los de las condiciones control a 0 y 24 horas e Ikaros activado a 0 y 24 horas para facilitar el análisis y la interpretación de los resultados. Se prepararon los posibles patrones de expresión para 4 condiciones con la función *GetPatterns* (tabla 3.3). Por ejemplo, un posible patrón de expresión para las 4 condiciones estudiadas sería 1-1-1-1, que indicaría que la expresión de la isoforma es aproximadamente igual para las 4 condiciones comparadas. Otro patrón podría ser 1-1-2-2, que indica que la expresión no cambia entre 0 y 24 horas, pero sí cambia entre las condiciones Ikaros activado y control. En este estudio interesaba identificar las isoformas con distinto perfil de expresión temporal en Ikaros activado y en control. Dado que en este último ejemplo el perfil es plano en ambos casos, no se consideró diferencialmente expresada una isoforma con dicho perfil.

Siguiendo este mismo razonamiento, se consideró que una isoforma no estaba diferencialmente expresada (EE) si presentaba un perfil con los patrones 1, 4 o 6 de la tabla 3.3, y diferencialmente expresada (DE) en cualquier otro caso. Reducir el estudio a dos instantes de tiempo en lugar de considerar toda la serie temporal supone ciertas limitaciones. Sin embargo, uno de los inconvenientes de EBseq es el incremento de complejidad al generar e interpretar los patrones cuando se aumenta el número de condiciones. Para estudiar la serie temporal completa (12 condiciones experimentales), se hubieran tenido que evaluar más de 4 millones de patrones diferentes, por lo que se decidió simplificar el análisis al caso de 2 tiempos (4 condiciones experimentales).

Dado que EBSeq utiliza el algoritmo EM, y este es un algoritmo iterativo, se le debió introducir el número de iteraciones a realizar como parámetro de la función *EBMultiTest()*. Se introdujeron 7 iteraciones, ya que en pruebas anteriores se vio que a partir de 5-6 iteraciones la diferencia entre los parámetros calculados fue menor de 10^{-3} (convergencia).

Ya que se analizaron 4 condiciones distintas, EBSeq clasificó cada gen e isoforma en uno de los 15 patrones posibles recogidos en la tabla 3.3. Los resultados del análisis de expresión diferencial, obtenidos con la función *GetMultiPP()*, incluyen una matriz con el patrón asignado a cada gen o transcrito y otra matriz con la probabilidad de cada gen o transcrito a permanecer a cada uno de los patrones.

Tabla 3.3. Posibles patrones de expresión para 4 condiciones. Las siglas DE corresponden a "diferencialmente expresado" (d*ifferentially expressed*) y las EE a "igualmente expresado" (*equally expressed*).

	Control 0 horas	Control 24 horas	Ikaros 0 horas	Ikaros 24 horas	Tipo
Patrón 1	1	1	1	1	EE
Patrón 2	1	1	1	2	DE
Patrón 3	1	1	2	1	DE
Patrón 4	1	1	2	2	EE
Patrón 5	1	2	1	1	DE
Patrón 6	1	2	1	2	EE
Patrón 7	1	2	2	1	DE
Patrón 8	1	2	2	2	DE
Patrón 9	1	1	2	3	DE
Patrón 10	1	2	1	3	DE
Patrón 11	1	2	2	3	DE
Patrón 12	1	2	3	1	DE
Patrón 13	1	2	3	2	DE
Patrón 14	1	2	3	3	DE
Patrón 15	1	2	3	4	DE

.

4. RESULTADOS Y DISCUSIÓN

4.1. Comparación de la cuantificación de eXpress y RSEM

El análisis comparativo entre eXpress y RSEM realizado en este trabajo tenía dos objetivos. El primero fue entender el funcionamiento de los métodos y en qué situaciones producen resultados distintos. El otro objetivo fue elegir el método de cuantificación más adecuado para los datos tratados de acuerdo con una serie de parámetros evaluados, como la eficiencia, la robustez, etc.

4.1.1. Eficiencia

La eficiencia de un método de cuantificación se define como la proporción del total de lecturas asignadas a algún transcrito (conteos) respecto al total de lecturas mapeadas. Se hizo la suposición de que el número de lecturas mapeadas por cada método fue el mismo, ya que ambos usaron Bowtie2 para mapear. Esta suposición se realizó para evitar dividir los conteos entre el número de lecturas mapeadas en cada muestra y método, ya que no se disponía de dicha información. Por tanto en este caso la eficiencia es simplemente el número total de conteos en cada muestra. Cabe destacar que, en consecuencia de esta suposición, es posible comparar la eficiencia de la misma muestra entre los dos métodos, pero no es correcto comparar distintas muestras entre sí, ya que cada muestra tiene un número diferente de lecturas mapeadas. Dado que el objetivo era comparar la eficiencia entre métodos y no entre muestras, esto no supuso ningún problema.

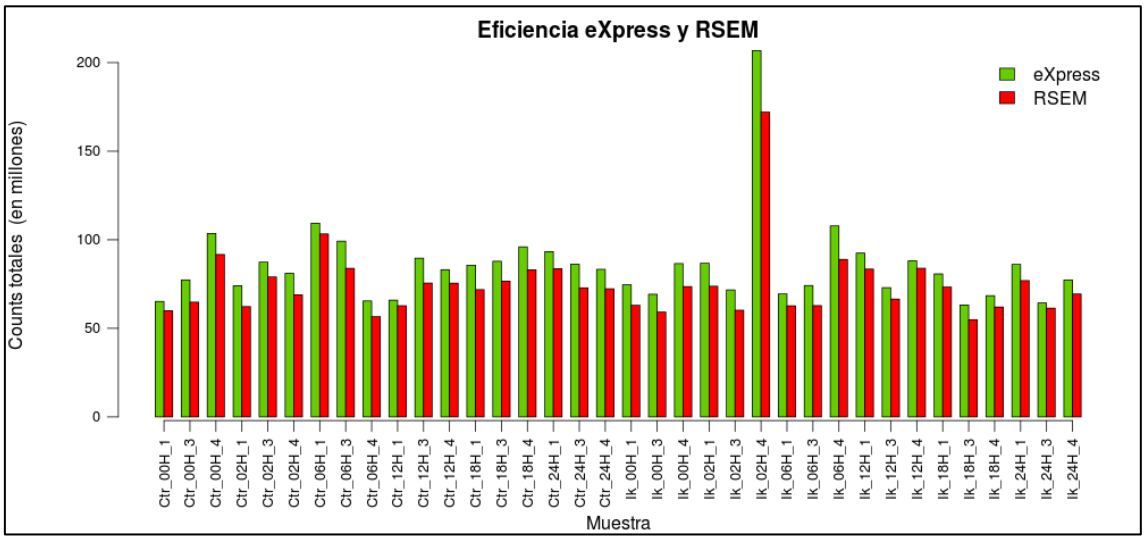


Figura 4.1. Eficiencia de cada método para cada muestra. En el eje Y se encuentran los conteos en millones, mientras que en el eje X están las distintas muestras analizadas en formato "condición_tiempo_número de réplica". La condición "Ctr" es el control, mientras que la "Ik" se corresponde a las muestras con el factor de transcripción Ikaros activado. Las barras verdes representan los datos de eXpress y las rojas los de RSEM.

En todas las muestras, eXpress presenta mayor eficiencia que RSEM (figura 4.1), ya que asigna un mayor número de lecturas a isoformas en todas y cada una de las 36 muestras analizadas. Además, una de las muestras (Ikaros activado después de 2 horas, réplica 4) contiene un número mucho mayor de conteos que el resto de muestras. Ello es debido a que en el proceso de secuenciación se analizó esta muestra con una profundidad de secuenciación mayor a la del resto.

4.1.2. Distribución de la expresión

Se comparó la distribución de la expresión de los transcritos tanto entre muestras como entre ambos métodos de cuantificación (figura 4.2) mediante un gráfico *boxplot* múltiple con los CPM de cada muestra y método. Se puede observar que la distribución es similar para los dos métodos, aunque cambia ligeramente entre las distintas muestras, habiendo algunas con menor cuantificación, como la muestra con Ikaros activado 12 horas, réplica 4. Las diferencias existentes entre distintas muestras al normalizar por CPM llevaron a la decisión de normalizar los datos con el método TMM con el fin de corregir estas diferencias. Así pues, se descartó la normalización RPKM, ya que el sesgo producido por la diferente distribución de la expresión entre muestras es más importante que el sesgo por la longitud de los transcritos que se corregiría con la normalización RPKM. Además, estudios recientes demuestran que la normalización TMM es uno de los métodos que ofrece mejores resultados para el análisis de expresión diferencial (Dillies *et al*., 2013).

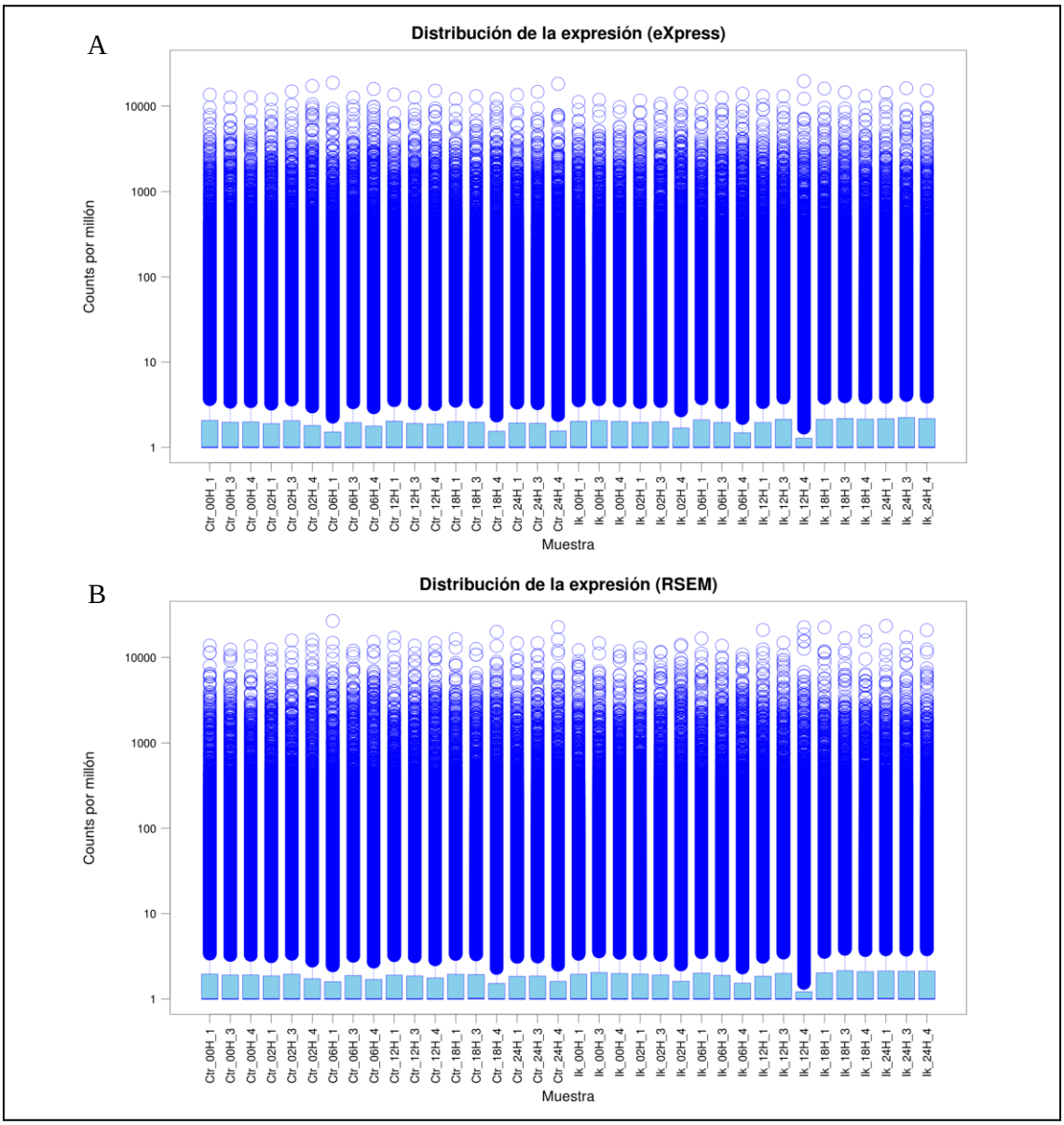


Figura 4.2. Distribución de la expresión de transcritos cuantificados por eXpress (A) y RSEM (B). En el eje Y se encuentran los CPM (en escala logarítmica), mientras que en el eje X están las distintas muestras analizadas en formato "condición_tiempo_número de réplica". La condición "Ctr" es el control, mientras que la "Ik" se corresponde a las muestras con el factor de transcripción Ikaros activado.

4.1.3. Número de genes e isoformas detectados

Se define un transcrito detectado como aquel para el que el software de cuantificación ha asignado al menos un conteo en alguna de las muestras, mientras que un gen detectado es aquel con al menos un conteo en alguna de sus isoformas en cualquier muestra. Comparando los transcritos y genes detectados con los anotados para el genoma analizado se puede obtener una estimación del porcentaje de transcritos e isoformas expresados en las células durante el experimento. Otra medida útil es el número medio de isoformas por gen que detecta cada método, ya que permite descubrir qué programa diferencia mejor entre los distintos transcritos de un mismo gen. De todos estos datos (tabla 4.1) se concluye que RSEM detecta algunos transcritos y genes más que eXpress y que, sobre todo en el caso de los genes, los dos métodos coinciden bastante. Por otro lado, el número de isoformas detectadas por gen es mayor en los resultados de eXpress.

Tabla 4.1. Resultados de la cuantificación de isoformas.

	Transcritos en el genoma	Transcritos detectados	Genes en el genoma	Genes detectados	Isoformas por gen
eXpress	94 647	65 889	38 803	21 769	3,0267
RSEM		66 384		23 914	2,7759
En común		61 975		21 278	2,9126

Para investigar con más detalle el número de isoformas detectadas por gen se representó el número de genes con distintos números de isoformas detectados por cada método, y se compararon con los datos reales del genoma para los genes detectados por alguno de los dos programas (figura 4.3).

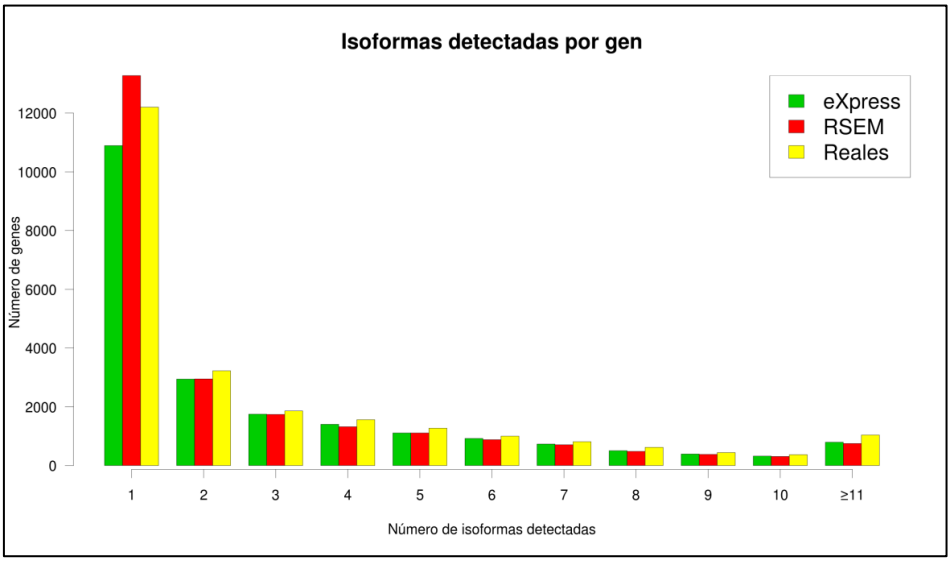


Figura 4.3. Número de genes con distintos números de isoformas. Las barras verdes corresponden al número de genes detectados por eXpress en cada caso, las rojas al de RSEM y las amarillas al número real de genes en cada caso según la anotación del genoma.

En la gráfica anterior se observa que, en general, eXpress acierta más el número de isoformas de cada gen que RSEM, ya que su número de genes con cada número de isoformas detectadas se acerca más a los datos reales. Por otro lado, los dos métodos detectan menos isoformas que las que hay para cada gen. Esto es coherente, ya que no todos los transcritos de un mismo gen se suelen expresar simultáneamente. Existe una clara excepción a esto: RSEM sobreestima el número de genes con una sola isoforma. En consecuencia, subestima en el resto de los casos el número de isoformas con respecto a eXpress. En los resultados de eXpress para los genes para los que RSEM detecta una sola isoforma (figura 4.4 (A)) se aprecia que muchos

de ellos no son detectados por eXpress (0 isoformas detectadas), explicando esto gran parte de la diferencia. También hay algunos genes para los que eXpress detecta más isoformas que RSEM, explicando el resto de las desemejanzas. Lo mismo se puede decir en el caso contrario: muchos de los genes con una sola isoforma detectada por eXpress no son detectados por RSEM, mientras que otros son detectados con más isoformas (figura 4.4 (B)).

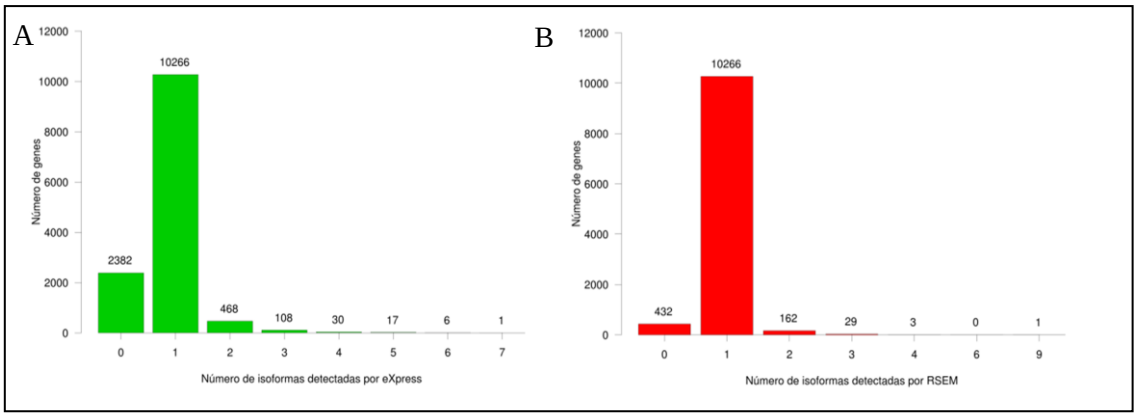


Figura 4.4. (A) Número de isoformas detectadas por eXpress para los genes con una sola isoforma detectada por RSEM. (B) Número de isoformas detectadas por RSEM para los genes con una sola isoforma detectada por eXpress.

4.1.4. Correlación entre las cuantificaciones de RSEM y eXpress

Para evaluar la semejanza de los resultados se calculó el coeficiente de correlación de cada transcrito entre los *expected counts* de RSEM y los *effective counts* de eXpress de las 36 muestras. Como se puede observar (figura 4.5 (A)), la correlación está cercana a 1 para la mayoría de los transcritos, lo cual indica que en esos casos la cuantificación es similar en ambos métodos. Sin embargo, es interesante estudiar aquellos transcritos que son cuantificados de forma diferente para tratar de averiguar qué método cuantifica mejor en cada caso.

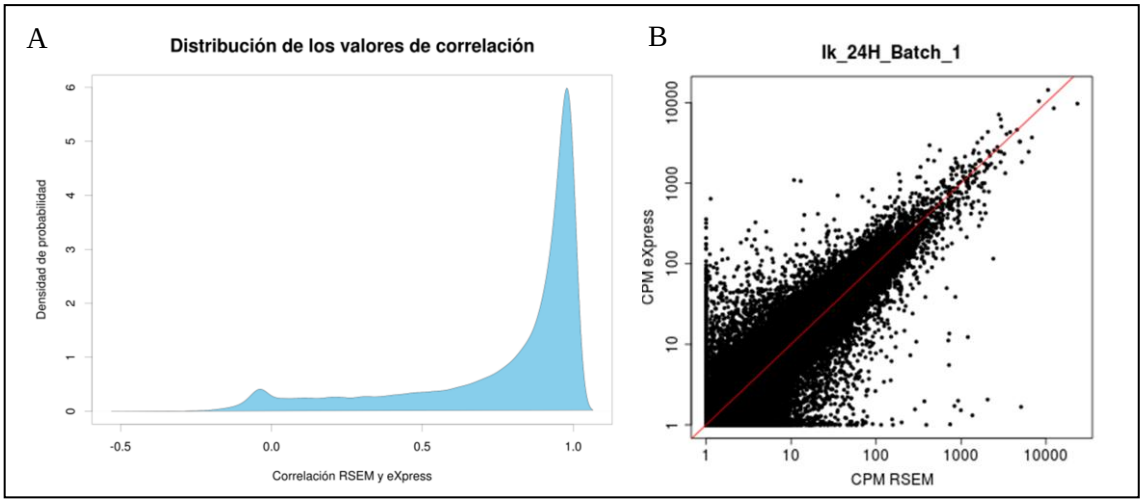


Figura 4.5. (A) Distribución de los valores de correlación. El eje X muestra el coeficiente de correlación entre los resultados de RSEM y eXpress. El eje Y muestra la densidad de probabilidad. **(B) Expresión de los transcritos en eXpress y RSEM.** La expresión está normalizada por CPM (conteos por millón). Los ejes X e Y están en escala logarítmica. La línea roja marca la diagonal del gráfico. Los datos se corresponden a la muestra con Ikaros activado después de 24 horas, réplica 1.

Para ello, se representaron los *effective counts* de eXpress se cada transcrito contra los *expected counts* de RSEM, ambos normalizados a CPM, de cada muestra (Anexo IV, ejemplo

en figura 4.5 (B)). En estas gráficas, además de las pequeñas diferencias típicas del uso de distintos algoritmos de cuantificación, destacan casos muy llamativos de transcritos que están muy cuantificados por un método y muy poco por el otro, que son los casos que interesó estudiar. Con este fin se eligieron algunos ejemplos para visualizar su mapeo en el visor de genomas IGV e investigar el motivo de estas discrepancias. Se estudiaron 10 genes con transcritos destacados, eligiendo como ejemplos representativos tres: Rps17, Actb y H3f3a.

En el caso de Rps17, un gen con 5 isoformas diferentes, su transcrito Rps17-04 está cuantificado con 1,68 CPM por eXpress y 2314,94 CPM por RSEM en la muestra control a 0 horas, réplica 1. Este transcrito no es codificante, sino que es un caso de *splicing* alternativo debido a la retención de un intrón que se debería haber eliminado. La isoforma Rps17-04 se encuentra muy poco representada en el mapeo, mientras que la Rps17-01, más cuantificada por eXpress que por RSEM, se encuentra muchas veces en el mapeo, tal y como se puede comprobar al comparar las estructuras de estas isoformas (figura 4.6) con el mapeo del gen (figura 4.7). Por tanto, parece que eXpress fue el método que hizo la cuantificación correcta en este caso.

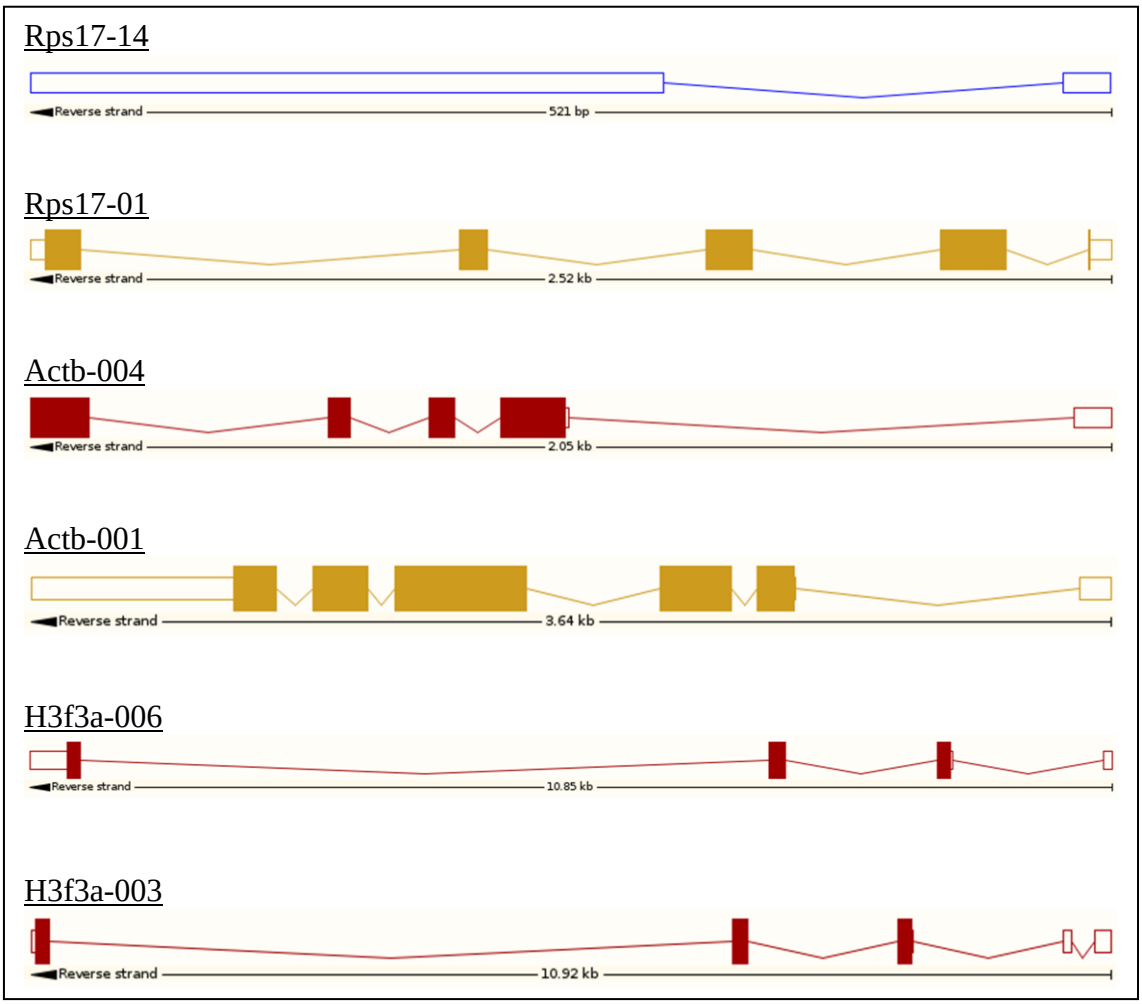


Figura 4.6. Estructuras de distintas isoformas. Se representan dos de las distintas isoformas de los genes Rps17, Actb y H3f3a. Las líneas representan los intrones de cada isoforma, las cajas rellenas los exones codificantes y las cajas vacías los exones no codificantes. Los transcritos rojos y dorados son codificantes, mientras que los azules son no codificantes. Las unidades de longitud de transcritos se expresan en pares de bases (bp) o en kilobases (kb). Información obtenida de ENSEMBL (2015).

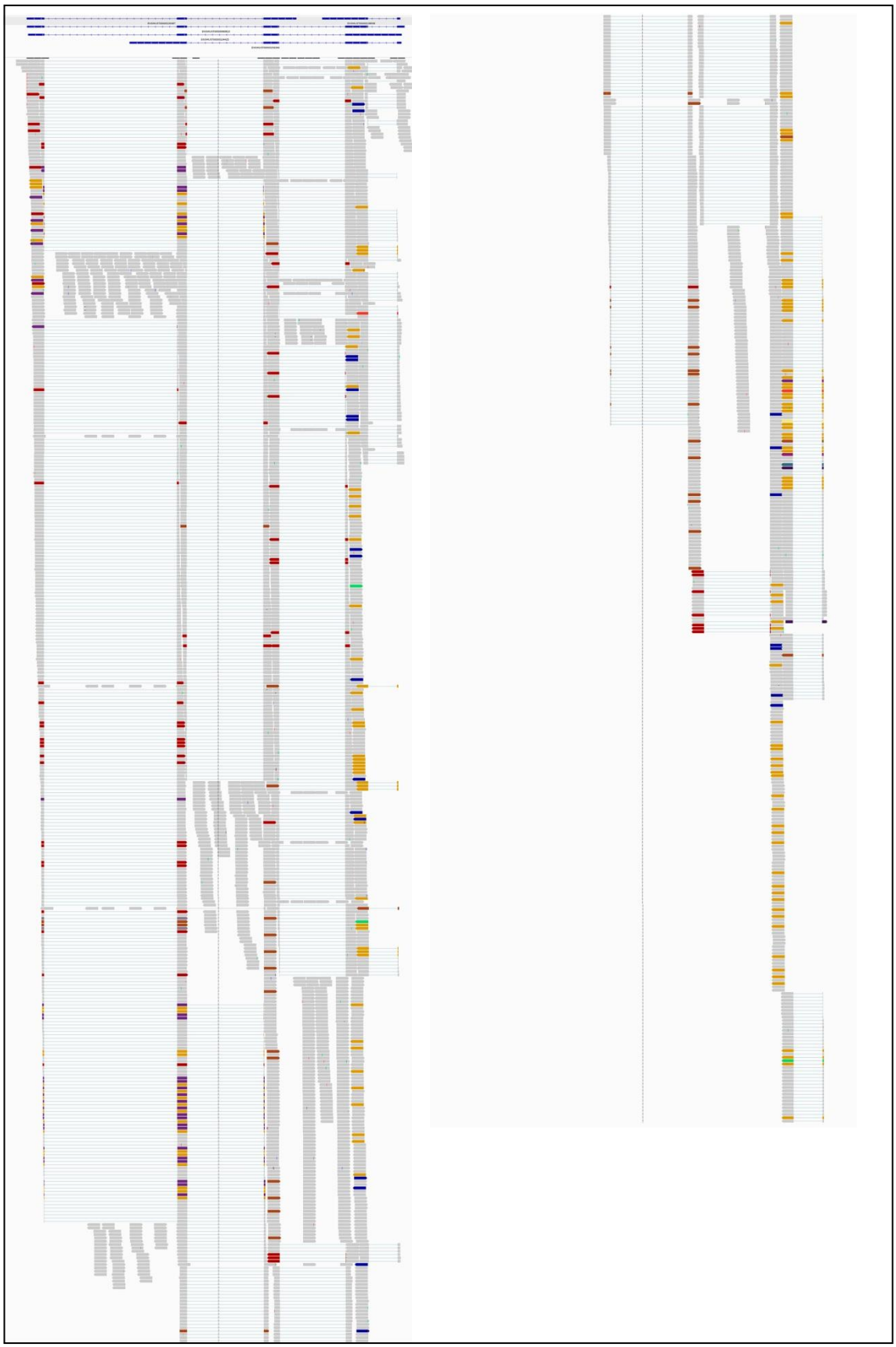


Figura 4.7. Mapeado de lecturas sobre el gen Rps17. Lecturas visualizadas con IGV. La columna de la derecha es la continuación de la de la izquierda. En la parte de arriba se muestran los transcritos de referencia. Cada caja es una lectura, siendo las cajas rojas y amarillas secuencias truncadas, las azules transcritos no codificantes y las verdes inserciones o deleciones.

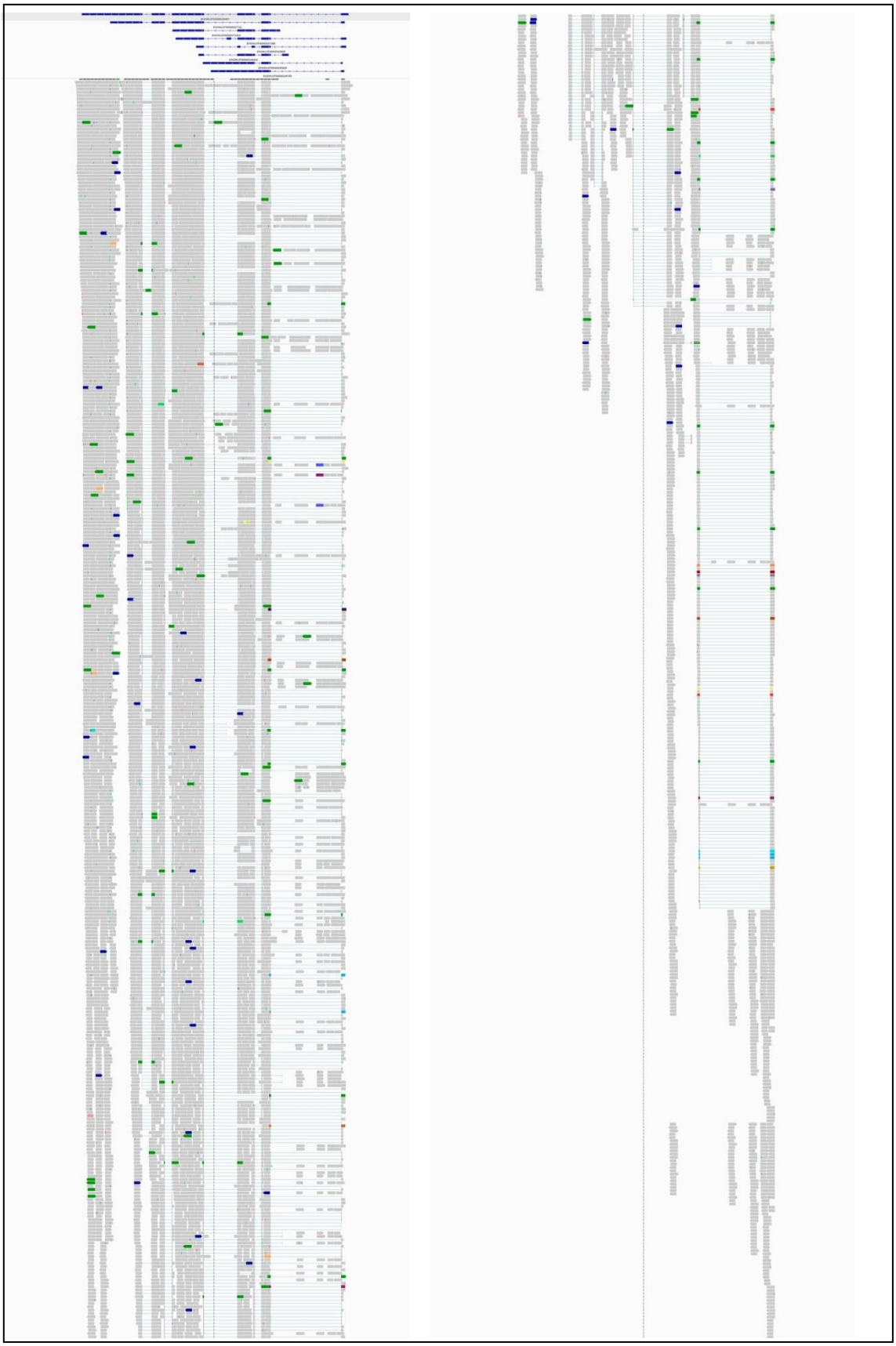


Figura 4.8. Mapeado de lecturas sobre el gen Actb. Lecturas visualizadas con IGV. La columna de la derecha es la continuación de la de la izquierda. En la parte de arriba se muestran los transcritos de referencia. Cada caja es una lectura, siendo las cajas rojas y amarillas secuencias truncadas, las azules transcritos no codificantes y las verdes inserciones o deleciones.

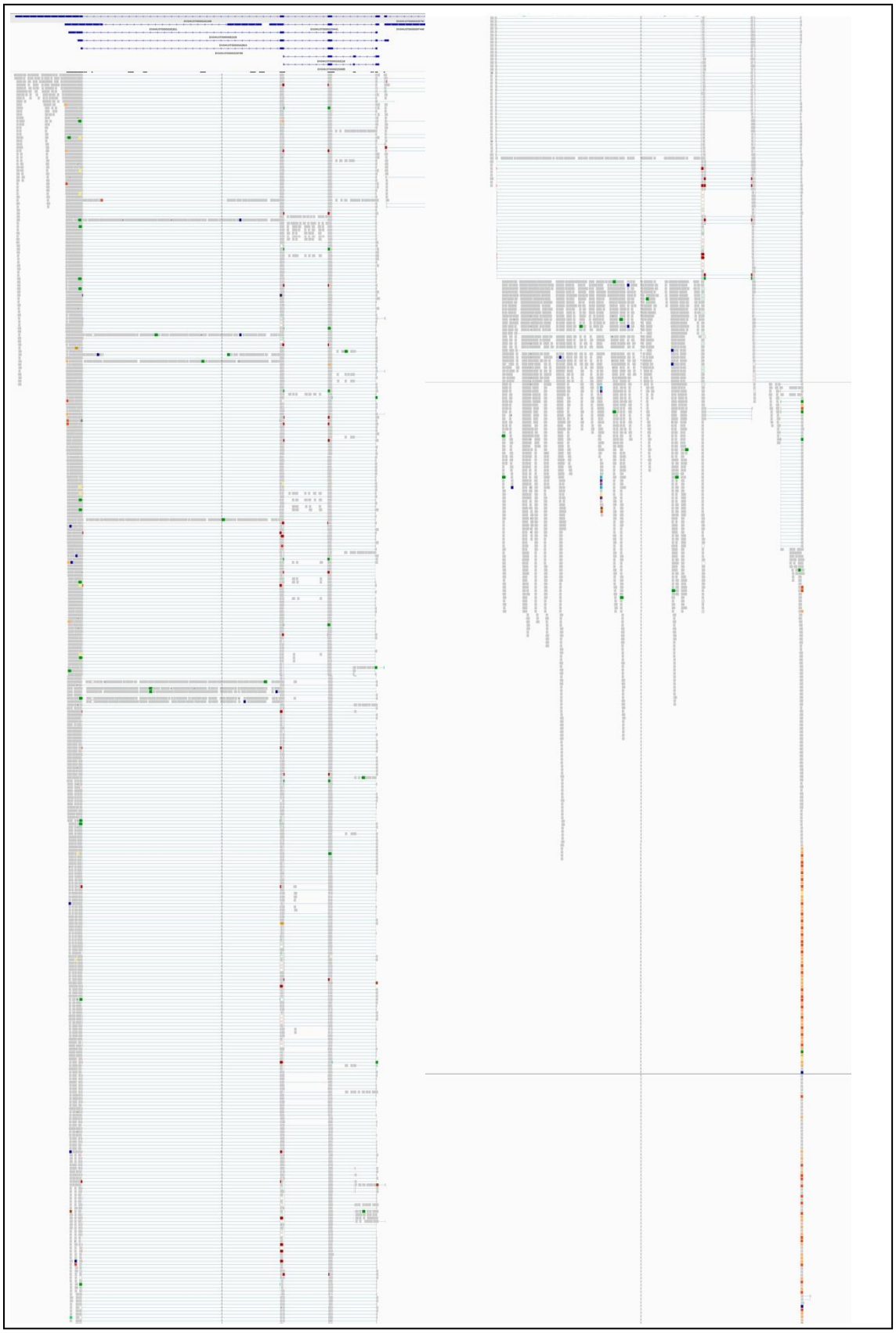


Figura 4.9. Mapeado de lecturas sobre el gen H3f3a. Lecturas visualizadas con IGV. La columna de la derecha es la continuación de la de la izquierda. En la parte de arriba se muestran los transcritos de referencia. Cada caja es una lectura, siendo las cajas rojas y amarillas secuencias truncadas, las azules transcritos no codificantes y las verdes inserciones o deleciones.

Otro gen en el que ocurre algo similar es Actb, que tiene 9 isoformas: mientras que en la muestra control a 6 horas, réplica 3, eXpress le asigna a la isoforma Actb-004 tan solo $1{,}2 \cdot 10^{-6}$ CPM, RSEM le asigna 7500,05 CPM en promedio. Al observar el mapeo de este gen (figura 4.8) apenas se puede encontrar dicha isoforma, formada por 5 exones, sino que la isoforma que se puede apreciar mayoritariamente es Actb-001, que contiene 6 exones (figura 4.6). Los valores asignados a la isoforma Actb-001 por eXpress y RSEM en esta muestra son 5712,5 CPM y 2548,2 CPM en promedio respectivamente. Por tanto, en el caso del gen Actb de nuevo es eXpress el que parece que hizo la cuantificación más adecuada.

El último ejemplo de este tipo de casos es el gen H3f3a, que tiene 7 isoformas conocidas. A la isoforma H3f3a-006 en la muestra con Ikaros activado a 24 horas, réplica 1, eXpress le asigna 1096,9 CPM en promedio y RSEM tan solo 9,8 CPM. En otra isoforma, H3f3a-003, ocurre lo contrario. Estas dos isoformas, muy parecidas, se diferencian principalmente en que H3f3a-006 contiene tan solo un exón no codificante en el extremo 3', mientras que H3f3a-003 contiene dos de estos exones (figura 4.6). En el mapeo (figura 4.9) se puede ver que, aunque están presentes las dos isoformas, H3f3a-006 es la que más veces se repite y, por tanto, vuelve a ser eXpress el que parece que ha cuantificado mejor las isoformas del gen.

El siguiente paso fue analizar los posibles motivos de la baja correlación de algunos transcritos entre los dos métodos de cuantificación. Para ello, la primera hipótesis que se planteó fue si los transcritos con baja correlación se correspondían con los de baja expresión debido a que los transcritos poco expresados presentan más variabilidad, son más difíciles de estimar y, por tanto, difieren más los resultados obtenidos por ambos métodos. Para verificar esta hipótesis se realizaron gráficas por intervalos (ver apartado 3.3.6) que se muestran en la figura 4.10 y que representan la correlación en función del nivel de expresión obtenido mediante eXpress (A) o RSEM (B). En ellas se puede observar la tendencia a aumentar el coeficiente de correlación a medida que aumenta la expresión. Como método alternativo para visualizar conjuntamente para ambos métodos esta tendencia, se realizaron gráficos por intervalos que enfrentaban la cuantificación de ambos métodos y se coloreaba cada punto en base al coeficiente de correlación medio entre eXpress y de RSEM para los transcritos incluidos en cada intervalo (figura 4.11). Con estas gráficas se confirmó que los valores más altos de correlación (colores más intensos) se corresponden a mayor expresión de los transcritos, y se observó que basta con que la cuantificación sea baja en solo unos de los métodos para que el valor de la correlación se reduzca.

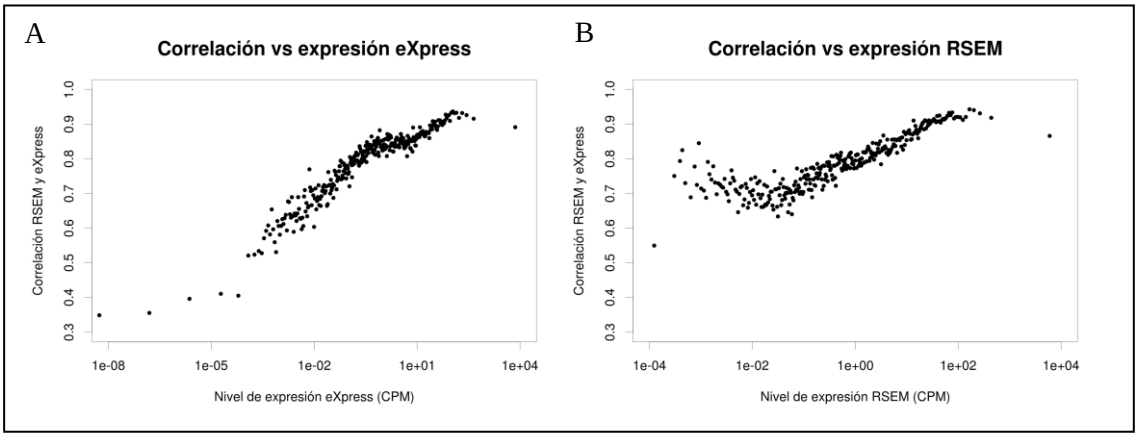


Figura 4.10. Relación entre correlación entre métodos y expresión. El eje X (en escala logarítmica) muestra los valores de expresión normalizados por CPM (conteos por millón) agrupados en intervalos a partir de los resultados de eXpress **(A)** y RSEM **(B)**. El eje Y muestra el valor de correlación entre la cuantificación de RSEM y eXpress.

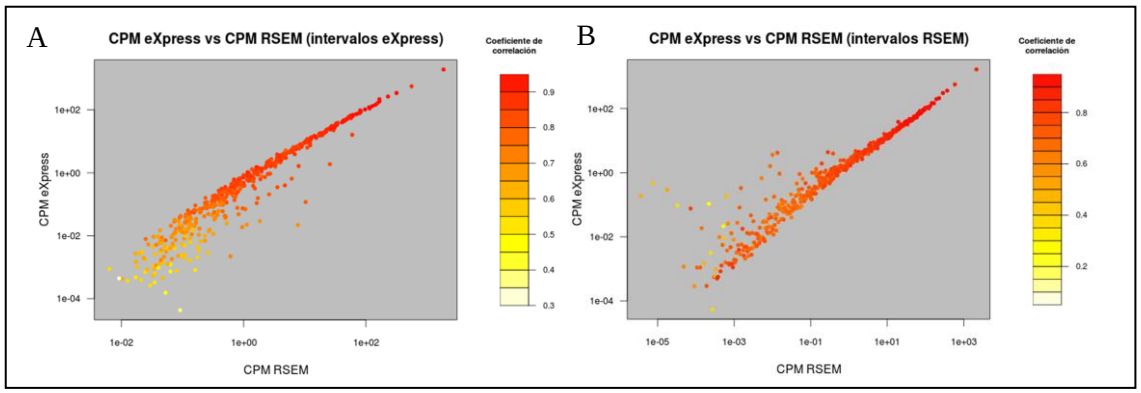


Figura 4.11. Expresión medida por eXpress y RSEM coloreada por correlación. Los datos están agrupados en intervalos a partir de los datos de eXpress **(A)** y de RSEM **(B)**. Los ejes X e Y están en escala logarítmica. La expresión se mide en CPM (conteos por millón). Los colores de los puntos indican la correlación entre los resultados de RSEM y eXpress.

La siguiente hipótesis planteada fue si la longitud de los transcritos podría estar afectando de alguna manera al coeficiente de correlación entre los resultados de los dos programas. En este caso no se observó una tendencia clara a que la correlación suba al aumentar la longitud de los transcritos (figura 4.12). Este hecho, sumado a que el coeficiente de correlación entre la correlación y la longitud es de tan solo 0,12, llevó a la conclusión de que la longitud de los transcritos no es uno de los factores que causan discrepancia entre los dos métodos.

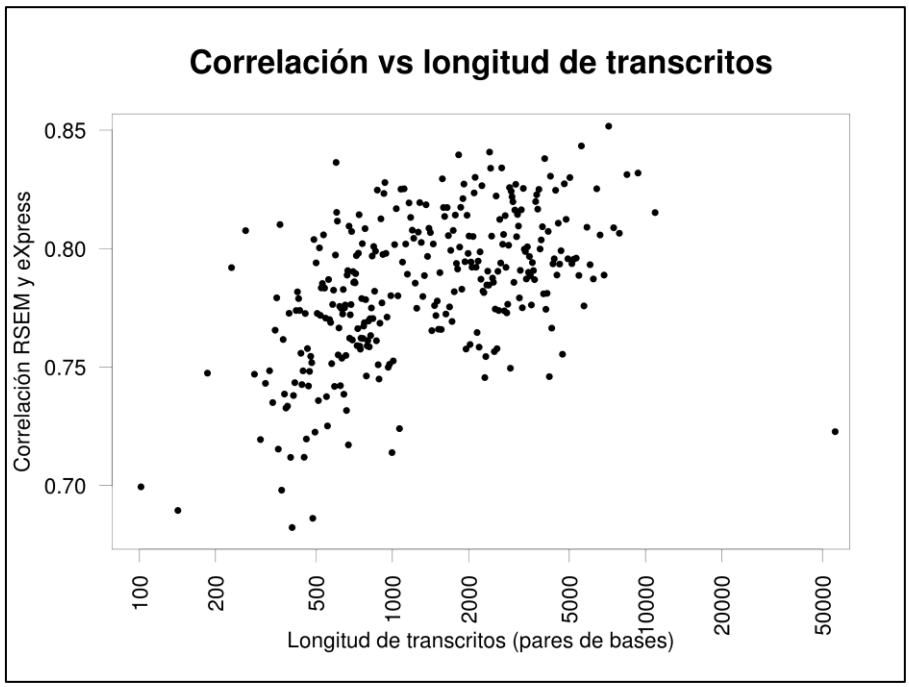


Figura 4.12. Relación entre la correlación entre métodos y la longitud de los transcritos. Los datos están agrupados por intervalos. El eje X se encuentra en escala logarítmica.

También se planteó la hipótesis de si cuantas más isoformas posibles tiene un gen, más difícil es la diferenciación de cada una de ellas y por tanto los dos métodos discrepan más entre sí. Se puede apreciar que sí que hay cierta tendencia a que la correlación disminuya a medida que aumenta el número de isoformas de los genes (figura 4.13).

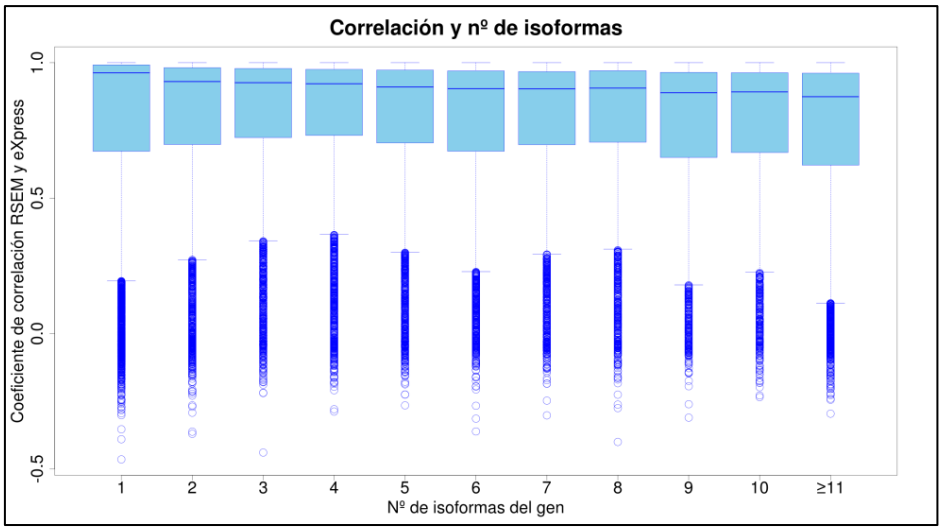


Figura 4.13. Correlaciones para los distintos números de isoformas de los genes.

4.1.5. Coherencia de los datos con el diseño experimental

Con el fin de descubrir qué cuantificación era más consistente con el diseño experimental, se realizó un análisis PCA para la cuantificación producida por cada método normalizada por TMM (figura 4.14). Se puede observar que en el caso de eXpress se separan perfectamente las muestras control de las Ikaros activado y, además, también se agrupan bien las réplicas de cada tiempo con Ikaros activado. En control, todas las muestras y tiempos están mezcladas como era de esperar, ya que las células de esta condición no se diferencian. Además las muestras de Ikaros activado se van separando progresivamente de las de control en orden cronológico (0 horas, 2 horas, 6 horas, etc.), como es lógico, a medida que las células de esta condición se van diferenciando. En cambio, en RSEM no se separan correctamente las muestras, indicando una menor correspondencia entre estos datos y el diseño experimental.

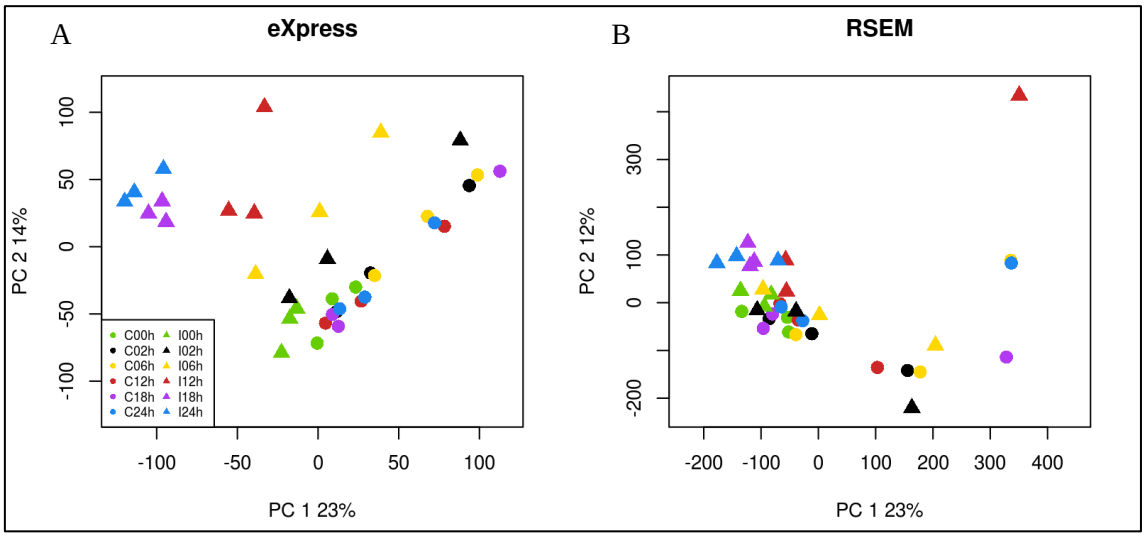


Figura 4.14. Análisis PCA de los resultados. El eje X se corresponde a la primera componente principal, que explica el 23% de la variabilidad de los datos en ambos métodos. El eje Y se corresponde a la segunda componente principal, que explica el 14% y el 12% de la variabilidad de los datos para eXpress **(A)** y RSEM **(B)** respectivamente.

También se realizó un mapa de calor para cada método (figura 4.15). Se puede observar que, para los datos de los dos métodos, generalmente se separan las muestras con Ikaros activado de las control, aunque no siempre se agrupan correctamente las réplicas de cada condición.

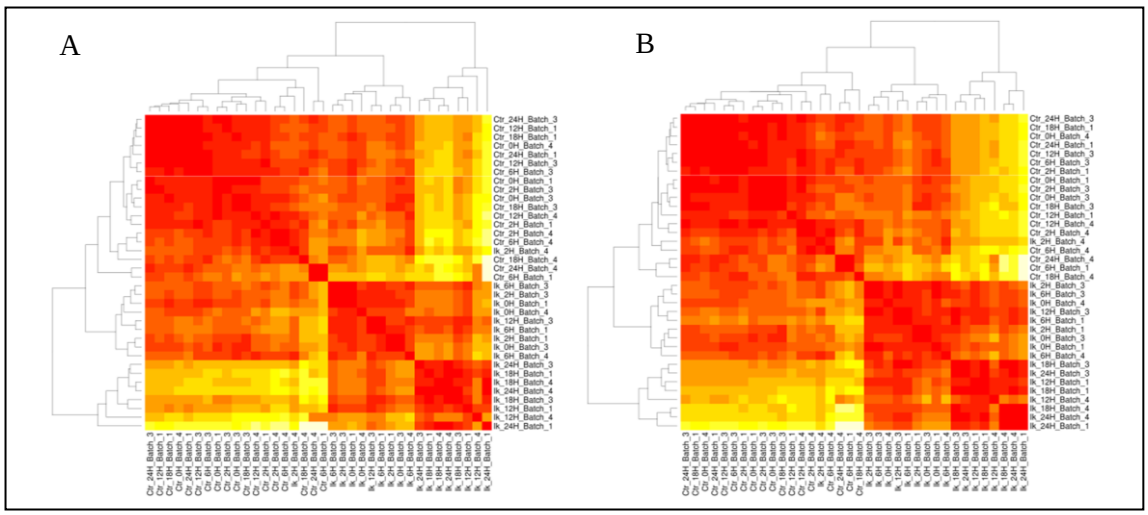


Figura 4.15. Mapas de calor de las cuantificaciones de eXpress (A) y de RSEM (B).

Mientras que los PCA se realizaron con los datos normalizados por TMM, los mapas de calor se hicieron con los normalizados por CPM. El hecho de que se separen mucho mejor las muestras en el PCA de los datos de eXpress que en cualquiera de los mapas de calor reforzó la decisión de usar los datos normalizados por TMM en los análisis posteriores.

4.1.6. Robustez

El siguiente paso consistió en analizar la robustez de cada método, es decir, la similitud en la cuantificación para las distintas réplicas de cada condición experimental. La robustez de un método es una característica muy importante, ya que si las réplicas dentro de cada condición presentan una mayor variabilidad que las propias condiciones experimentales comparadas es muy difícil detectar cambios de expresión entre dichas condiciones. Para evaluar la robustez, se extrajeron las correlaciones entre réplicas de la misma condición. Para cada pareja de réplicas se representó la correlación obtenida entre la cuantificación de RSEM (eje X) y de eXpress (eje Y), así como la diagonal para compararlas mejor (figura 4.16). Se puede observar que los resultados de eXpress presentan mayor correlación entre réplicas de la misma condición, y por tanto son más robustos que los de RSEM, ya que la mayoría de los puntos están por encima de la diagonal.

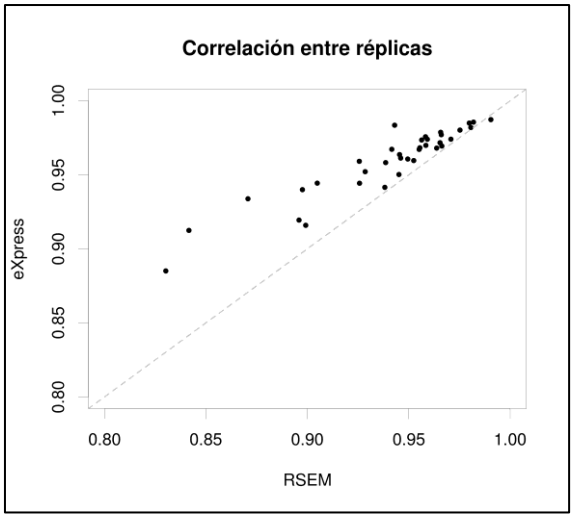


Figura 4.16. Correlación entre réplicas de cada método. Cada punto representa la correlación entre una pareja de réplicas pertenecientes a la misma condición experimental para los resultados de eXpress (eje Y) y RSEM (eje X). La línea discontinua marca la diagonal del gráfico.

4.1.7. Resumen

En función de los aspectos evaluados se eligió el método de cuantificación eXpress, ya que presenta mayor eficiencia, las isoformas detectadas de cada gen se acercan más a las reales, el análisis visual de los casos muy discrepantes sugieren que es este el que cuantifica adecuadamente, los resultados son más coherentes con el diseño experimental y también son más robustos.

Por otra parte, el método de normalización elegido fue TMM, ya que la distribución de los datos normalizados por CPM presenta diferencias entre muestras y, además, las muestras se separan mejor en el PCA de eXpress (preparado con datos normalizados con TMM) que en los mapas de calor (preparados con datos normalizados con CPM). La normalización RPKM se descartó porque no podría corregir la distribución diferente entre muestras. Además, estudios recientes afirman que la normalización TMM reduce el número de falsos positivos en los análisis de expresión diferencial (Dillies *et al.*, 2013).

Por tanto, los datos usados en análisis posteriores fueron los *effective counts* de eXpress normalizados por TMM.

4.2. Análisis de expresión diferencial de isoformas

Aunque los resultados del apartado anterior llevaron a la decisión de utilizar los datos producidos por eXpress para analizar la expresión diferencial de isoformas, se decidió descartar las isoformas de baja expresión, que no solo son motivo de discrepancia entre los métodos, sino que además su cuantificación es menos fiable y por tanto es una práctica común no tenerlas en cuenta en los análisis de expresión diferencial, reduciendo así el ruido y aumentando la potencia de los métodos estadísticos.

Para filtrar las isoformas se utilizó el método CPM del paquete NOISeq (descrito en la sección 3.3.4). Dado que este método requiere que se establezca un valor de CPM de corte, se representaron para distintos valores de corte el número de transcritos que se mantuvieron tras el filtro y la correlación resultante entre RSEM y eXpress (figura 4.17).

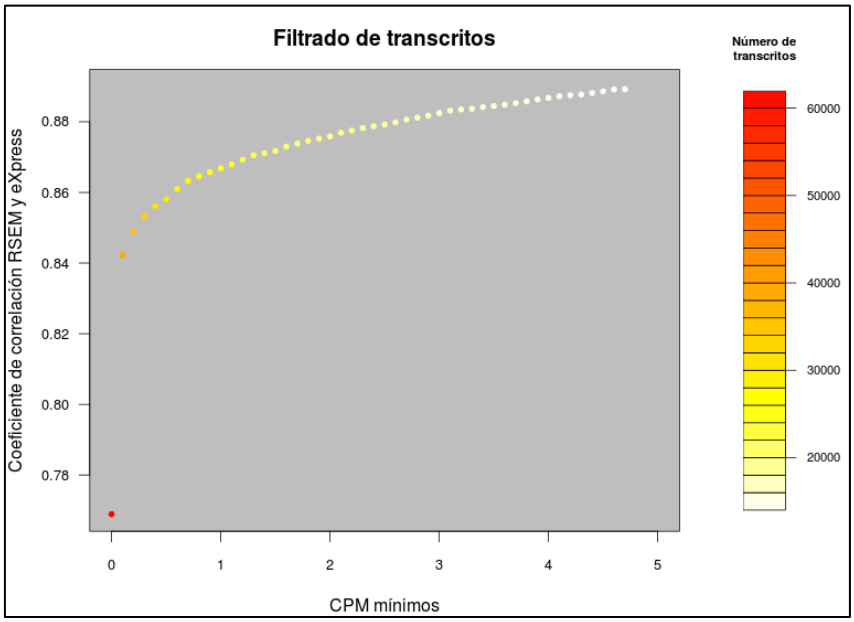


Figura 4.17. Filtrado de transcritos por nivel de expresión. El eje X muestra el valor de corte del filtrado en CPM (conteos por millón)**.** El eje Y muestra la correlación entre los resultados de eXpress y RSEM tras filtrar los datos con los diferentes valores de corte. Los colores representan el número de transcritos restantes tras el filtrado.

Se puede apreciar que la correlación entre métodos aumentó considerablemente al filtrar los transcritos. Evidentemente, conforme se aumentó el valor del filtro también disminuyó el número de transcritos que quedaron sin filtrar, por lo que se eligió un valor de filtro en el que se mantiene un número aceptable de transcritos al mismo tiempo que aumenta suficientemente la correlación (y por tanto la fiabilidad de los datos). El valor de filtro elegido fue de 0,5 CPM. En la tabla 4.2 se recopilan los resultados tras aplicar dicho filtro, comparándolos con los datos sin aplicárselo.

Tabla 4.2. Resultados antes y después de filtrar los datos.

		Datos sin filtrar	Datos filtrados
Número de isoformas detectadas	eXpress	65 889	34 160
	RSEM	66 384	34 395
	En común	61 975	30 545
Número de genes detectados	eXpress	21 769	12 574
	RSEM	23 914	13 206
	En común	21 278	12 208
Número de isoformas por gen	eXpress	3,0267	2,7167
	RSEM	2,7759	2,6045
Correlación media entre eXpress y RSEM		0,769	0,858

Una vez filtrados los datos, se aplicó EBSeq a nivel de gen y de transcrito para un nivel de significación de 0,05. La tabla 4.3 recoge las diferencias entre el análisis de los genes y de los transcritos. Se puede comprobar que en general ocurre lo esperado: los genes diferencialmente expresados contienen alguna isoforma diferencialmente expresada, mientras que en la mayoría de los genes no diferencialmente expresados las isoformas correspondientes tampoco lo están.

Tabla 4.3. Diferencias en la expresión diferencial de los genes y de sus transcritos. Las siglas DE corresponden a "diferencialmente expresado" (d*ifferentially expressed*) y las EE a "igualmente expresado" (*equally expressed*).

	Totales	Con alguna isoforma DE	Con ninguna isoforma DE	Con todas las isoformas filtradas
Nº de genes DE	8 238	7 711	251	276
Nº de genes EE	4 390	2 004	2 234	152

En la tabla 4.4 se recopilan los resultados del análisis de expresión diferencial para una serie de genes implicados en la diferenciación de las células B. Los resultados concuerdan con lo esperado, ya que la mayoría de estos genes están diferencialmente expresados. Se puede observar que normalmente un gen se clasifica como diferencialmente expresado cuando la mayoría de sus isoformas lo están.

Tabla 4.4. Comportamiento de genes y sus isoformas implicados en la diferenciación de células B. Las siglas DE corresponden a "diferencialmente expresado" (d*ifferentially expressed*) y las EE a "igualmente expresado" (*equally expressed*).

Gen	DE/EE	Isoformas DE	Isoformas EE	Isoformas filtradas
Igll1	DE	3	0	0
Myc	DE	4	1	1
Slc7a5	DE	2	0	1
Ldha	DE	3	1	4
Foxo1	DE	1	0	0
Lig4	DE	2	0	0
Ikzf1	EE	2	1	3
Ptgs1	DE	5	0	0
Slc22a8	DE	2	0	0
Pnp	DE	2	0	0

Gfra2	DE	1	0	0
Rhoq	EE	0	1	1
Vpreb1	DE	2	1	0
Gtf3c5	DE	5	2	0
Trp53inp1	DE	3	0	1
St3gal5	DE	2	0	0
Cd93	EE	0	1	0
Tifab	DE	1	0	0
Tns3	DE	2	0	2
Kdelc1	DE	4	0	3
Polr1b	DE	5	0	0
Cyb561a3	DE	1	0	0
Pax7	DE	2	0	0
Alpl	DE	2	0	6
Srfbp1	DE	1	0	0
Plekho2	DE	1	0	0
Pik3ap1	DE	1	0	0
Plch1	DE	7	1	5
Anks1	DE	4	0	0
Ranbp9	DE	1	0	0
Cdk6	DE	2	0	0
Rel	DE	1	0	0
Bcl11a	DE	4	1	1
1810032O08Rik	DE	6	1	0
Trim59	EE	1	1	3
Hip1	DE	1	0	0
Lef1	DE	6	0	0
Ankrd28	EE	0	1	0
Rmnd5b	DE	2	1	1
Plekha2	DE	3	0	1
Optn	DE	2	2	1
Slc37a3	DE	1	0	0
Hivep2	DE	1	0	0
Elovl6	DE	1	0	0
Nedd9	DE	1	1	0
Vpreb2	DE	1	0	0
Dusp12	DE	6	0	1
Ddc	DE	3	1	6
Cdkn1a	DE	3	0	0
St6galnac4	DE	3	0	1
Blnk	DE	3	0	0
Rapgef5	DE	1	0	0

En el siguiente apartado se estudia con más detalle el comportamiento de algunos genes e isoformas involucrados en procesos relevantes para la diferenciación celular.

4.3. Discusión biológica de los resultados

En primer lugar se investigó el comportamiento, tanto a nivel de gen como a nivel de isoformas, de los genes involucrados en 3 rutas que se activan durante la diferenciación de las células B: la ruta de la glicólisis, la ruta de la regulación de la autofagia y la ruta de señalización FOXO. Para ello, primero se cargaron los datos de los genes en la herramienta Paintomics con el fin de obtener una visión general de cada ruta (Anexo VIII, ejemplo en figura 4.18). Los datos representados son los ratios de la expresión para cada tiempo entre Ikaros activado y el control.

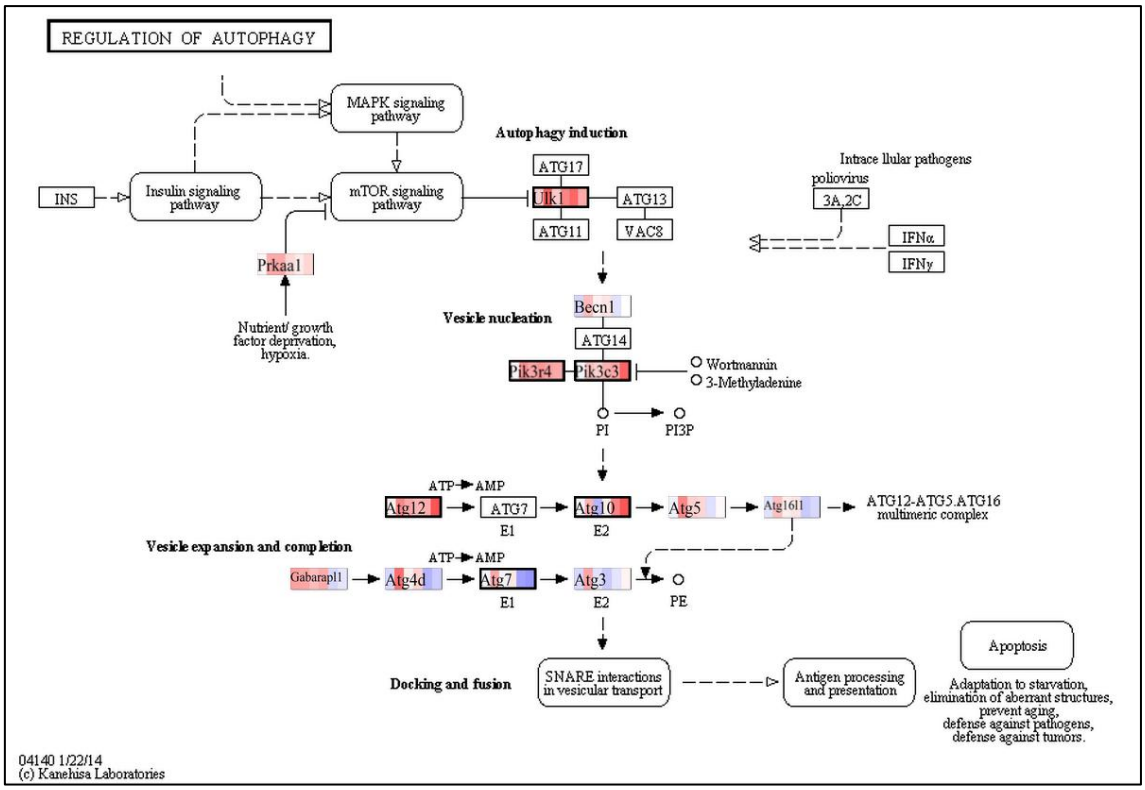


Figura 4.18. Ruta de regulación de la autofagia en Paintomics. El color azul representa una expresión mayor en control que en Ikaros activado para ese tiempo, el rojo una mayor expresión en Ikaros activado que en control y el blanco una expresión similar. Los genes con el recuadro más remarcado están expresados diferencialmente según EBSeq.

También se representó la expresión de los genes e isoformas implicados en estas rutas (figura 4.19 y Anexo IX). En cada gráfica está representada la expresión media para cada muestra de un gen diferente (círculo y líneas rojas) y de cada una de sus isoformas (triángulos y líneas de otros colores). Además, se diferencia entre los genes e isoformas con expresión diferencial según EBSeq (líneas continuas y círculos o triángulos rellenos) y sin expresión diferencial (líneas discontinuas y círculos o triángulos vacíos).

Se puede observar que muchos genes implicados en la glicólisis aumentan su expresión en las primeras horas después de activar Ikaros (Anexo VIII), como por ejemplo los genes Bpgm y Aldh3a2 (figura 4.19). Esto tiene sentido biológico, ya que las células necesitan aumentar su cantidad de energía disponible en forma de ATP, y para ello deben transformar la glucosa en piruvato. La gluconeogénesis es el proceso contrario (transformación de piruvato a glucosa), aunque la mayoría de las enzimas implicadas son las mismas que para la glicólisis, catalizando las reacciones en sentido contrario. No obstante, hay enzimas que catalizan reacciones específicas de la gluconeogénesis, y los genes que codifican estas enzimas (Aldoa, Aldob, Fbp1, Pck2 (figura 4.19) y Pcx) se expresaron menos al activar Ikaros (Anexo IX), lo que confirma la necesidad de las células de obtener energía.

Respecto a los genes implicados en la ruta de la regulación de la autofagia, también se puede ver que la mayoría de los genes implicados aumentan su expresión después de activar el factor de transcripción Ikaros (figura 4.18), como por ejemplo Atg12 y Gabarapl2 (figura 4.19). La autofagia es un mecanismo celular en el que macromoléculas y orgánulos son degradados por el lisosoma. Se ha demostrado que es un proceso muy importante para la diferenciación celular, ya que permite que las células cambien muy rápidamente su composición y otorga las biomoléculas necesarias (Vessoni *et al*., 2012). Además, se ha comprobado experimentalmente que es un proceso crucial en la diferenciación de las células B (Mizushima y Levine, 2010). Por

tanto, era de esperar que los genes implicados en la autofagia estuvieran más expresados en las células que se encuentran en un proceso de diferenciación.

Finalmente, los genes de la familia FoxO también tienen un rol central durante la diferenciación de las células B (Reth y Nielsen, 2014). La regulación de esta familia es compleja e intervienen muchos otros genes, por lo que no es tan sencillo describir un comportamiento general de los genes implicados como en las dos rutas anteriores (Anexo VIII). La conclusión más clara es que todos los genes de la familia FOXO, como Foxo1 (figura 4.16) aumentaron su expresión con Ikaros activado (Anexo IX). A su vez, muchos de los genes que activan la expresión de los genes de la familia FOXO también aumentan su expresión, como Cdnk1b (figura 4.16), aunque otros disminuyen. Lo mismo pasa con los genes que reducen la expresión de los genes de la familia FOXO, aunque el aumento de la expresión de estos inducen a pensar que prevalece la estimulación a la represión de estos genes.

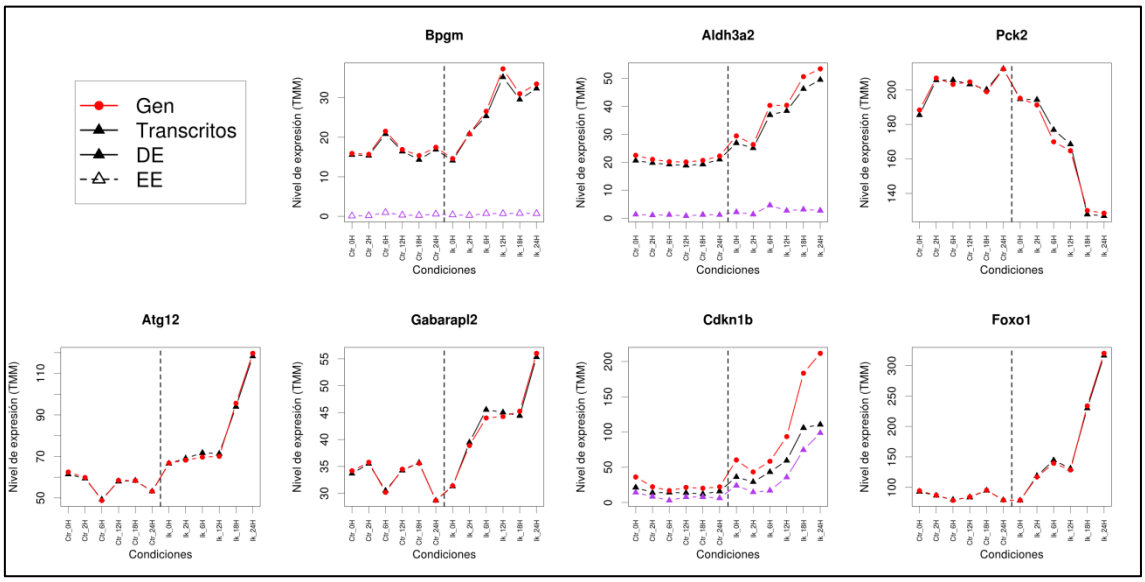


Figura 4.19. Perfiles de expresión de algunos genes y sus isoformas. El eje X muestra las distintas condiciones y tiempos. El eje Y muestra la expresión normalizada por TMM. Los genes están representados por círculos y líneas rojas y las isoformas por triángulos y líneas de otros colores. Los genes e isoformas con expresión diferencial se representan con líneas continuas y círculos o triángulos rellenos y el resto con líneas discontinuas y círculos o triángulos vacíos. Las siglas DE corresponden a "diferencialmente expresado" (d*ifferentially expressed*) y las EE a "igualmente expresado" (*equally expressed*).

En la mayoría de los genes que forman parte de las rutas estudiadas se observó que las distintas isoformas tenían un perfil de expresión muy similar al del gen o que simplemente no estaban expresadas (figura 4.19). Sin embargo, son casos más interesantes para este estudio aquellos en los que alguna de las isoformas presenta un comportamiento diferente al del gen, ya que ello puede implicar alguna diferencia funcional entre las distintas isoformas. Se investigaron estos casos con el fin de descubrir a qué se debieron estas diferencias en la expresión de isoformas. Algunos de los genes en los que se observó esto fueron Flii, Mxi1 y Dnase1l1. Estos tres genes están diferencialmente expresados, así como sus isoformas que han pasado el filtro de 0,5 CPM.

El gen Flii (figura 4.20) tiene cuatro isoformas, una codificante (Flii-001) y las otras tres no codificantes (Flii-002, Flii-003 y Flii-004). Las isoformas no codificantes son casos de variantes de *splicing* debido a la retención de intrones que se deberían eliminar, y por tanto los transcritos son productos de fallos en el *splicing*, y no tienen ninguna función biológica conocida. No obstante, el hecho de que la isoforma Flii-004 tenga una expresión diferencial (y no al nivel mínimo como Flii-003) sugiere que es posible que esta isoforma tenga alguna

función biológica desconocida hasta ahora (podría tener un papel en la regulación génica, por ejemplo), pero habría que hacer otros tipos de experimentos para investigarlo.

Las tres isoformas del gen Mxi1 son codificantes (figura 4.20). Cada isoforma tiene una función diferente en sus homólogas humanas (Dugast-Darzacq *et al.*, 2004). Mxi1-201 tiene mucha capacidad de represión del crecimiento celular, mientras que Mxi1-202 presenta la misma actividad, aunque en menor medida, y además reprime la transformación celular (Dugast-Darzacq *et al.*, 2007). El hecho de que la isoforma más activa aumente su expresión y la menos activa (y represora de la transformación) la disminuya encaja muy bien en el contexto de la diferenciación de células pre-BI a pre-BII, ya que en este proceso las células detienen su crecimiento. Por otra parte, la isoforma Mxi1-203 carece de actividad represora, por lo que también es coherente que generalmente se exprese menos que las otras dos isoformas.

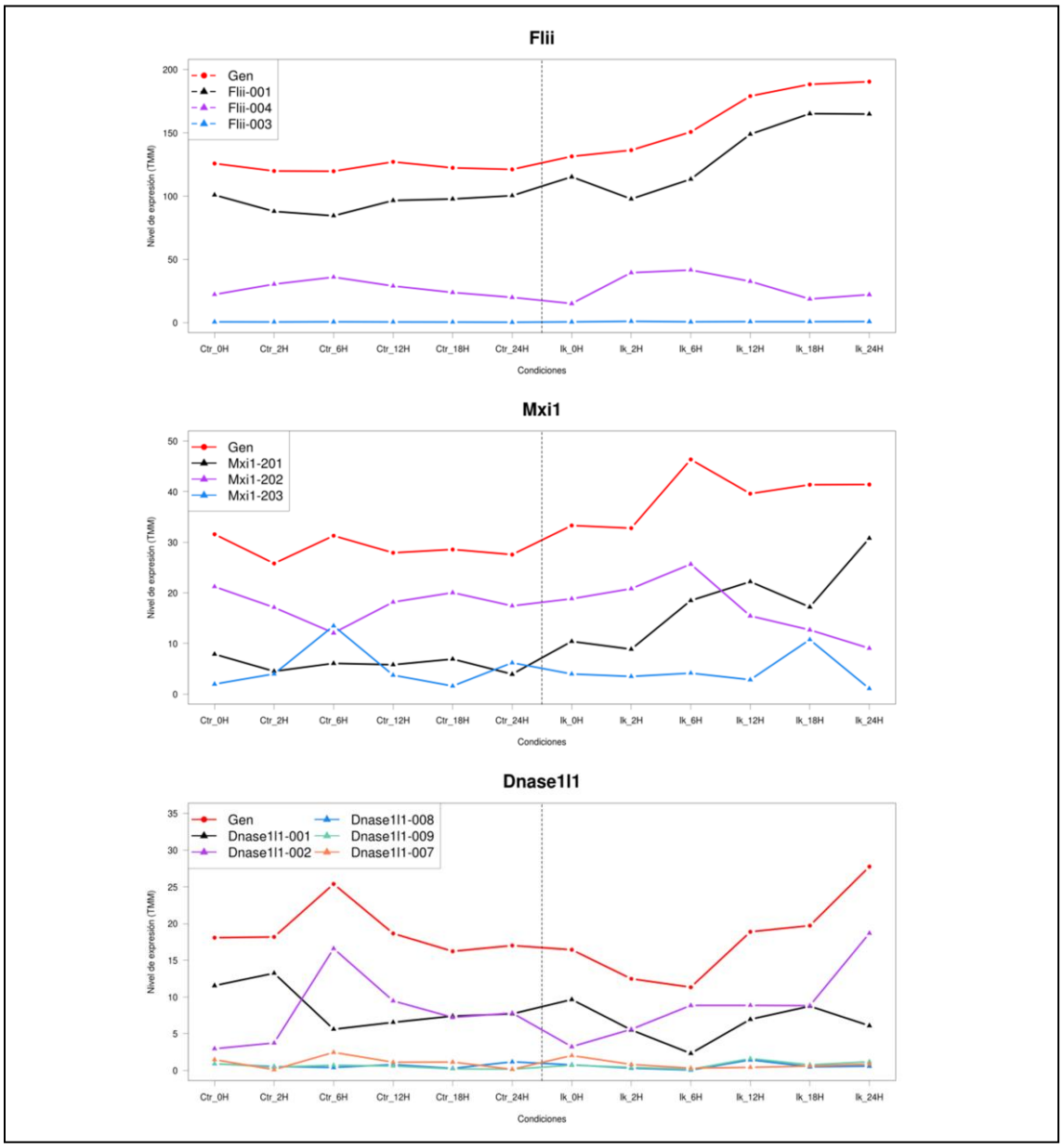


Figura 4.20. Perfiles de expresión de genes e isoformas con comportamientos diferentes. La línea discontinua vertical señala la separación entre las condiciones control e Ikaros. El nivel de expresión está normalizado por TMM. Cada gráfica contiene el perfil de expresión de un gen (representado con puntos y líneas rojas) y de sus isoformas (representadas con triángulos y líneas de otros colores).

En el caso de Dnase1l1 (figura 4.20) las isoformas que se comportan de forma diferente son Dnase1l1-001 y Dnase1l1-002, ambas codificantes. Hasta la fecha no se conoce ninguna diferencia funcional entre estas dos isoformas y, de hecho, codifican una proteína idéntica, ya que se diferencian en el extremo 3' UTR del mRNA, que no es traducido. Por tanto, es posible que coexistan las dos isoformas realizando la misma función, tal y como muestran los datos.

Así, estos hechos ponen en relevancia la importancia de estudiar la expresión alternativa de isoformas, ya que es una manera de identificar transcritos que se expresan de forma diferente en determinadas condiciones para ser estudiados experimentalmente y descubrir así nuevas funciones desconocidas actualmente. Por otra parte, es una manera más fina de estudiar la expresión diferencial, pudiendo obtener más conclusiones que haciendo estos estudios a nivel de gen.

5. CONCLUSIONES

Las nuevas tecnologías de secuenciación masiva permiten estudiar la expresión génica a nivel de isoformas, permitiendo investigar el papel del *splicing* alternativo en distintos procesos biológicos. En concreto, en este trabajo se han utilizado datos de RNA-Seq para llevar a cabo dicho estudio.

El pre-procesado de los datos de secuenciación es muy importante para eliminar posibles sesgos de la tecnología y reducir el ruido de los datos. En este trabajo se estudiaron distintos métodos para realizarlo, obteniendo los mejores resultados al aplicar la normalización TMM y filtrar las isoformas con expresión media menor de 0,5 CPM en todas las condiciones experimentales.

La cuantificación de isoformas es una tarea compleja, especialmente cuando se utilizan tecnologías de secuenciación con lecturas cortas (como RNA-Seq). Por ello, es importante conocer los distintos métodos de cuantificación y determinar sus limitaciones. En este caso, se cuantificaron los mismos datos con los programas eXpress y RSEM. Distintos análisis indicaron que, para estos datos, eXpress realizó en general una mejor cuantificación, ya que presentó mayor eficiencia y robustez, detectó más isoformas por gen en promedio y, en casos concretos en los que los dos métodos presentaron discrepancias, el análisis visual de los mapeos sugiere que la estimación de eXpress resultó ser más correcta. Además, los resultados de eXpress son más coherentes con el diseño experimental que los de RSEM según el análisis PCA realizado.

Se analizaron las posibles causas de las discrepancias entre los dos métodos. Se demostró que el nivel de expresión es un factor importante: los transcritos con baja expresión fueron cuantificados de forma más diferente por los dos métodos. Otro factor que disminuyó la correlación entre los resultados de eXpress y RSEM fue el número de isoformas de los genes, aunque su importancia es menor que el nivel de expresión. La longitud de los transcritos no parece que afectara significativamente el coeficiente de correlación.

Se eligió el método EBSeq para realizar el análisis de expresión diferencial, ya que permite realizar el análisis tanto a nivel de gen como a nivel de isoformas, así como comparar más de dos condiciones. Además, evalúa los cambios en el nivel de expresión, al contrario que otros métodos que estudian si cambia la proporción de lecturas entre isoformas.

Se comprobó que el perfil de expresión de muchos genes que forman parte de algunas rutas relevantes en la diferenciación (glicólisis, regulación de la autofagia y señalización de FOXO) sigue la tendencia que cabía esperar en el contexto biológico.

Se investigaron algunos genes cuyas isoformas se comportaron de forma diferente entre ellas. En el caso de Flii, es posible que una de sus isoformas tenga una función biológica desconocida hasta ahora, por lo que habría que estudiarlo con más detalle. Cada una de las isoformas del gen Mxi1, que tienen funciones diferentes, se comportan de manera lógica dentro del proceso de diferenciación estudiado. Por último, las isoformas que se comportan de forma diferente en el gen Dnase1l1 realizan, hasta donde se sabe, la misma función, por lo que no es extraño que se expresen simultáneamente.

El alcance de este trabajo no ha permitido profundizar en el estudio funcional de la expresión alternativa de isoformas, aunque ha servido como punto de partida para dicho análisis, que será llevado a cabo por el laboratorio de Genómica de la Expresión Génica del CIPF.

6. BIBLIOGRAFÍA

ADAMIDI, C.; WANG, Y.; GRUEN, D.; MASTROBUONI, G.; YOU, X.; TOLLE, D.; DODT, M.; MACKOWIAK, S.D.; GOGOL-DOERING, A.; OENAL, P.; RYBAK, A.; ROSS, E.; SANCHEZ ALVARADO, A.; KEMPA, S.; DIETERICH, C.; RAJEWSKY, N.; CHEN, W. (2011). De novo assembly and validation of planaria transcriptome by massive parallel sequencing and shotgun proteomics. *Genome Res.,* 21:1193-1200.

ALBERTS, B.; JOHNSON, A.; LEWIS, J.; MORGAN, D.; RAFF, M.; ROBERTS, K.; WALTER, P. (2015). Control of Gene Expression, en: *Molecular Biology of the Cell*, 6ª Ed. Garland Science, Nueva York, 369-438.

ANDERS, S.; HUBER, W. (2010). Differential expression analysis for sequence count data. *Genome Biol.,* 11:R106-2010-11-10-r106.

ANDERS, S.; REYES, A.; HUBER, W. (2012). Detecting differential usage of exons from RNA-seq data. *Genome Res.,* 22:2008-2017.

ANTON, M.A.; GOROSTIAGA, D.; GURUCEAGA, E.; SEGURA, V.; CARMONA-SAEZ, P.; PASCUAL-MONTANO, A.; PIO, R.; MONTUENGA, L.M.; RUBIO, A. (2008). SPACE: an algorithm to predict and quantify alternatively spliced isoforms using microarrays. *Genome Biol.,* 9:R46-2008-9-2-r46.

BLACK, D.L. (2003). Mechanisms of alternative pre-messenger RNA splicing. *Annu.Rev.Biochem.,* 72:291-336.

CHEN, K.; DAI, X.; WU, J. (2015). Alternative splicing: An important mechanism in stem cell biology. *World J.Stem Cells,* 7:1-10.

CRICK, F. (1970). Central dogma of molecular biology. *Nature,* 227:561-563.

DELVES, P.; ROITT, I. (2003). Activación de los linfocitos, en: *Inmunologia. Fundamentos*. 10ª Ed. Editorial Médica Panamericana S.A, Buenos Aires, 183-198.

DEMPSTER,A.; LAIRD, N.M.; RUBIN, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc*., 39:1–38.

DILLIES, M.A.; RAU, A.; AUBERT, J.; HENNEQUET-ANTIER, C.; JEANMOUGIN, M.; SERVANT, N.; KEIME, C.; MAROT, G.; CASTEL, D.; ESTELLE, J.; GUERNEC, G.; JAGLA, B.; JOUNEAU, L.; LALOE, D.; LE GALL, C.; SCHAEFFER, B.; LE CROM, S.; GUEDJ, M.; JAFFREZIC, F.; FRENCH STATOMIQUE CONSORTIUM. (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief Bioinform*, 14:671-683.

DUGAST-DARZACQ, C.; GRANGE, T.; SCHREIBER-AGUS, N.B. (2007). Differential effects of Mxi1-SRalpha and Mxi1-SRbeta in Myc antagonism. FEBS J., 274:4643-4653.

DUGAST-DARZACQ, C.; PIRITY, M.; BLANCK, J.K.; SCHERL, A.; SCHREIBER-AGUS, N. (2004). Mxi1-SRalpha: a novel Mxi1 isoform with enhanced transcriptional repression potential. Oncogene, 23:8887-8899.

ENSEMBL (2015). Visto el 27 de junio de 2015. http://www.ensembl.org/.

ENCODE PROJECT CONSORTIUM. (2004). The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science*, 306:636-640.

FU, R.H.; LIU, S.P.; OU, C.W.; YU, H.H.; LI, K.W.; TSAI, C.H.; SHYU, W.C.; LIN, S.Z. (2009). Alternative splicing modulates stem cell differentiation. *Cell Transplant.,* 18:1029-1038.

GARBER, M.; GRABHERR, M.G.; GUTTMAN, M.; TRAPNELL, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat.Methods,* 8:469-477.

GARCIA-ALCALDE, F.; GARCIA-LOPEZ, F.; DOPAZO, J.; CONESA, A. (2011). Paintomics: a web based tool for the joint visualization of transcriptomics and metabolomics data. *Bioinformatics*, 27:137-139.

GARG, R.; PATEL, R.K.; TYAGI, A.K.; JAIN, M. (2011). De novo assembly of chickpea transcriptome using short reads for gene discovery and marker identification. *DNA Res.,* 18:53-63.

GLAUS, P.; HONKELA, A.; RATTRAY, M. (2012). Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics*, 28:1721-1728.

GONZÁLEZ, C. (2009). Procesamiento de ARNm en la Transcripción ("splicing"). Visto el 7 de mayo de 2015. http://www.botanica.cnba.uba.ar/Pakete/Dibulgeneral/Splicing/Splicing.htm

GRANT, G.R.; FARKAS, M.H.; PIZARRO, A.D.; LAHENS, N.F.; SCHUG, J.; BRUNK, B.P.; STOECKERT, C.J.; HOGENESCH, J.B.; PIERCE, E.A. (2011). Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27:2518-2528.

HAGEN, R.M.; LADOMERY, M.R. (2012). Role of splice variants in the metastatic progression of prostate cancer. *Biochem.Soc.Trans.,* 40:870-874.

HU, Y.; HUANG, Y.; DU, Y.; ORELLANA, C.F.; SINGH, D.; JOHNSON, A.R.; MONROY, A.; KUAN, P.F.; HAMMOND, S.M.; MAKOWSKI, L.; RANDELL, S.H.; CHIANG, D.Y.; HAYES, D.N.; JONES, C.; LIU, Y.; PRINS, J.F.; LIU, J. (2013). DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Res.,* 41:e39.

HUBER, W.; CAREY, V.J.; GENTLEMAN, R.; ANDERS, S.; CARLSON, M.; CARVALHO, B.S.; BRAVO, H.C.; DAVIS, S.; GATTO, L.; GIRKE, T.; GOTTARDO, R.; HAHNE, F.; HANSEN, K.D.; IRIZARRY, R.A.; LAWRENCE, M.; LOVE, M.I.; MACDONALD, J.; OBENCHAIN, V.; OLES, A.K.; PAGES, H.; REYES, A.; SHANNON, P.; SMYTH, G.K.; TENENBAUM, D.; WALDRON, L.; MORGAN, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nat.Methods*, 12:115-121.

HULL, J.; CAMPINO, S.; ROWLANDS, K.; CHAN, M.S.; COPLEY, R.R.; TAYLOR, M.S.; ROCKETT, K.; ELVIDGE, G.; KEATING, B.; KNIGHT, J.; KWIATKOWSKI, D. (2007). Identification of common genetic variation that modulates alternative splicing. *PLoS Genet.,* 3:e99.

IHAKA, R.; GENTLEMAN, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5-3: 299-314

JIANG, T.; QIN, B.; HE, J.; LIN, S.; DING, S. (2013). Three isoforms of the Atg16L1 protein contribute different autophagic properties. *Mol.Cell.Biochem.,* 378:257-266.

KANEHISA, M.; GOTO, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res., 28:27-30.

KATZ, Y.; WANG, E.T.; AIROLDI, E.M.; BURGE, C.B. (2010). Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nat.Methods,* 7:1009-1015.

KELEMEN, O.; CONVERTINI, P.; ZHANG, Z.; WEN, Y.; SHEN, M.; FALALEEVA, M.; STAMM, S. (2013). Function of alternative splicing. *Gene,* 514:1-30.

KEREN, H.; LEV-MAOR, G.; AST, G. (2010). Alternative splicing and evolution: diversification, exon definition and function. *Nat.Rev.Genet.,* 11:345-355.

KONARSKA, M.M.; GRABOWSKI, P.J.; PADGETT, R.A.; SHARP, P.A. (1985). Characterization of the branch site in lariat RNAs produced by splicing of mRNA precursors. *Nature,* 313:552-557.

LANGMEAD, B.; TRAPNELL, C.; POP, M.; SALZBERG, S.L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.,* 10:R25-2009-10-3-r25.

LANGMEAD, B.; SALZBERG, S.L. (2012). Fast gapped-read alignment with Bowtie 2. Nat.Methods, 9:357-359.

LEE, S.; SEO, C.H.; LIM, B.; YANG, J.O.; OH, J.; KIM, M.; LEE, S.; LEE, B.; KANG, C.; LEE, S. (2011). Accurate quantification of transcriptome from RNA-Seq data by effective length normalization. *Nucleic Acids Res.*, 39:e9.

LENG, N.; DAWSON, J.A.; THOMSON, J.A.; RUOTTI, V.; RISSMAN, A.I.; SMITS, B.M.; HAAG, J.D.; GOULD, M.N.; STEWART, R.M.; KENDZIORSKI, C. (2013). EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics,* 29:1035-1043.

LI, B.; DEWEY, C.N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics,* 12:323-2105-12-323.

LI, H.; HANDSAKER, B.; WYSOKER, A.; FENNELL, T.; RUAN, J.; HOMER, N.; MARTH, G.; ABECASIS, G.; DURBIN, R.; 1000 GENOME PROJECT DATA PROCESSING SUBGROUP. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics,* 25:2078-2079.

MARDIS, E.R. (2008). Next-generation DNA sequencing methods. *Annu.Rev.Genomics Hum.Genet.,* 9:387-402.

MARTINI, M. (2010). Cuando el dogma no se cumple: el caso de virus y priones. Visto el 7 de mayo de 2015. http://www.aportes.educ.ar/sitios/aportes/recurso/index?rec_id=107708&nucleo=biologia_nucleo_arte#sthash.5begv4bV.dpufhttp://www.aportes.educ.ar/sitios/aportes/recurso/index?rec_id=107708&nucleo=biologia_nucleo_arte

MERKENSCHLAGER, M. (2010). Ikaros in immune receptor signaling, lymphocyte differentiation, and function. *FEBS Lett.,* 584:4910-4914.

MILLS, J.D.; JANITZ, M. (2012). Alternative splicing of mRNA in the molecular pathology of neurodegenerative diseases. *Neurobiol.Aging,* 33:1012.e11-1012.e24.

MIZUSHIMA, N.; LEVINE, B. (2010). Autophagy in mammalian development and differentiation. *Nat.Cell Biol.,* 12:823-830.

MORIN, R.; BAINBRIDGE, M.; FEJES, A.; HIRST, M.; KRZYWINSKI, M.; PUGH, T.; MCDONALD, H.; VARHOL, R.; JONES, S.; MARRA, M. (2008). Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *BioTechniques,* 45:81-94.

MORTAZAVI, A.; WILLIAMS, B.A.; MCCUE, K.; SCHAEFFER, L.; WOLD, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat.Methods,* 5:621-628.

NICOLAE, M.; MANGUL, S.; MANDOIU, I.I.; ZELIKOVSKY, A. (2011). Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms Mol.Biol*., 6:9-7188-6-9.

PAN, Q.; BAKOWSKI, M.A.; MORRIS, Q.; ZHANG, W.; FREY, B.J.; HUGHES, T.R.; BLENCOWE, B.J. (2005). Alternative splicing of conserved exons is frequently species-specific in human and mouse. *Trends Genet.,* 21:73-77.

POULOS, M.G.; BATRA, R.; CHARIZANIS, K.; SWANSON, M.S. (2011). Developments in RNA splicing and disease. *Cold Spring Harb Perspect.Biol.,* 3.

RACINE, J.S. (2012). RStudio: A Platform-Independent IDE for R and Sweave. *J.Appl.Econometrics*, 27:167-172.

REINHOLD-HUREK, B.; SHUB, D.A. (1992). Self-splicing introns in tRNA genes of widely divergent bacteria. *Nature,* 357:173-176.

RETH, M.; NIELSEN, P. (2014). Signaling circuits in early B-cell development. *Adv.Immunol.,* 122:129-175.

ROBERTS, A.; PACHTER, L. (2013). Streaming fragment assignment for real-time analysis of sequencing experiments. *Nat.Methods,* 10:71-73.

ROBINSON, J.T.; THORVALDSDOTTIR, H.; WINCKLER, W.; GUTTMAN, M.; LANDER, E.S.; GETZ, G.; MESIROV, J.P. (2011). Integrative genomics viewer. *Nat.Biotechnol.,* 29:24-26.

ROBINSON, M.D.; MCCARTHY, D.J.; SMYTH, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics,* 26:139-140.

ROBINSON, M.D.; OSHLACK, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.,* 11:R25-2010-11-3-r25.

ROSSI, B.; ESPELI, M.; SCHIFF, C.; GAUTHIER, L. (2006). Clustering of pre-B cell integrins induces galectin-1-dependent pre-B cell receptor relocalization and activation. *J.Immunol.,* 177:796-803.

RYAN, M.C.; CLELAND, J.; KIM, R.; WONG, W.C.; WEINSTEIN, J.N. (2012). SpliceSeq: a resource for analysis and visualization of RNA-Seq data on alternative splicing and its functional impacts. *Bioinformatics*, 28:2385-2387.

SAMMETH, M.; FOISSAC, S.; GUIGO, R. (2008). A general definition and nomenclature for alternative splicing events. *PLoS Comput.Biol.,* 4.

SANGER, F.; COULSON, A.R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J.Mol.Biol.,* 94:441-448.

SHENDURE, J.; JI, H. (2008). Next-generation DNA sequencing. *Nat.Biotechnol.,* 26:1135-1145.

SINGH, D.; ORELLANA, C.F.; HU, Y.; JONES, C.D.; LIU, Y.; CHIANG, D.Y.; LIU, J.; PRINS, J.F. (2011). FDM: a graph-based statistical method to detect differential transcription using RNA-seq data. *Bioinformatics,* 27:2633-2640.

TANG, J.Y.; LEE, J.C.; HOU, M.F.; WANG, C.L.; CHEN, C.C.; HUANG, H.W.; CHANG, H.W. (2013). Alternative splicing for diseases, cancers, drugs, and databases. *ScientificWorldJournal,* 2013:703568.

TARAZONA, S.; GARCIA-ALCALDE, F.; DOPAZO, J.; FERRER, A.; CONESA, A. (2011). Differential expression in RNA-seq: a matter of depth. *Genome Res.,* 21:2213-2223.

TRAPNELL, C.; PACHTER, L.; SALZBERG, S.L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics,* 25:1105-1111.

TRAPNELL, C.; WILLIAMS, B.A.; PERTEA, G.; MORTAZAVI, A.; KWAN, G.; VAN BAREN, M.J.; SALZBERG, S.L.; WOLD, B.J.; PACHTER, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat.Biotechnol.,* 28:511-515.

VESSONI, A.T.; MUOTRI, A.R.; OKAMOTO, O.K. (2012). Autophagy in stem cell maintenance and differentiation. *Stem Cells Dev.,* 21:513-520.

WU, J.; AKERMAN, M.; SUN, S.; MCCOMBIE, W.R.; KRAINER, A.R.; ZHANG, M.Q. (2011). SpliceTrap: a method to quantify alternative splicing under single cellular conditions. *Bioinformatics*, 27:3010-3016.

XU, X.; YANG, D.; DING, J.H.; WANG, W.; CHU, P.H.; DALTON, N.D.; WANG, H.Y.; BERMINGHAM, J.R.,JR; YE, Z.; LIU, F.; ROSENFELD, M.G.; MANLEY, J.L.; ROSS, J.,JR; CHEN, J.; XIAO, R.P.; CHENG, H.; FU, X.D. (2005). ASF/SF2-regulated CaMKIIdelta alternative splicing temporally reprograms excitation-contraction coupling in cardiac muscle. *Cell,* 120:59-72.

YI, Q.; TANG, L. (2011). Alternative spliced variants as biomarkers of colorectal cancer. *Curr.Drug Metab.,* 12:966-974.

ZENG, V.; VILLANUEVA, K.E.; EWEN-CAMPEN, B.S.; ALWES, F.; BROWNE, W.E.; EXTAVOUR, C.G. (2011). De novo assembly and characterization of a maternal and developmental transcriptome for the emerging model crustacean Parhyale hawaiensis. *BMC Genomics,* 12:581-2164-12-581.

ZERBINO, D.R.; BIRNEY, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.,* 18:821-829.

7. ANEXOS

7.1. Anexo I: *Script* de *rsem-prepare-reference*

```
#!/bin/sh

#$ -N rsem_reference
#$ -S /bin/bash
##$ -pe smp 1
#$ -l h_vmem=8G
#$ -o /clinicfs/projects/stategra/isoformas/queue/stdout
#$ -e /clinicfs/projects/stategra/isoformas/queue/stderr
#$ -m beas -M jordiferretes2@gmail.com
#$ -cwd
##$ -wd working_dir
##$ -V
##$ -v variable[=value]
##$ -l long=true
##$ -q long.q
##$ -l fast=true
##$ -q fast.q


# Se cargan los módulos necesarios (Bowtie 2 y RSEM)

module load bowtie/2.1.0
module load rsem/1.2.18

# Se asignan las rutas del archivo de referencia y el nombre del output

reference=/clinicfs/projects/stategra/upload/RNAseq_KI/eXpress_REF/mm10_EnsemblSeq_U
CSC.fa
output=rsem_reference

# Se ejecuta el RSEM para preparar el transcriptoma de referencia

rsem-prepare-reference --no-polyA --bowtie2 $reference $output
```

7.2. Anexo II: *Arrayjob*

```
#!/bin/sh
#$ -N rsem-expression
#$ -S /bin/bash
#$ -o /clinicfs/projects/stategra/isoformas/queue/stdout
#$ -e /clinicfs/projects/stategra/isoformas/queue/stderr
#$ -m beas
#$ -M jordiferretes2@gmail.com
#$ -t 1-5
##$ -tc 10
#$ -cwd
#$ -l h_vmem=4G
#$ -q long.q
#$ -l long=true
##$ -R y
#$ -pe smp 4

module load rsem/1.2.18
module load bowtie/2.1.0

SEEDFILE=/clinicfs/projects/stategra/isoformas/RSEM/scripts/rsem_commands_RUN1.cmd
SEED=$(cat $SEEDFILE | head -n $SGE_TASK_ID | tail -n 1)
$SEED
```

7.3. Anexo III: Script de comparación de los resultados de eXpress y RSEM

```
# Este script compara los resultados de la cuantificación de eXpress y RSEM.

#########################################################################
################### 0. PREPARACIÓN DE FUNCIONES Y DATOS #################
#########################################################################

# Se preparan algunas funciones necesarias en los análisis

## Función cpm, útil para representar gráficas con puntos de colores
cpm = function (correlaciones, main = "CPM eXpress vs CPM RSEM", log = "xy",
                xlab = "CPM RSEM", ylab = "CPM eXpress", title = "Coeficiente
                 de\ncorrelación") {
  miscolores = heat.colors(100)

  percentiles = as.data.frame(correlaciones[,3])[,1]
  maximo = max(percentiles)
  minimo = min(percentiles)
  for (i in 1:length(percentiles)) {
    percentiles[i] = (percentiles[i]-minimo) * 100 / (maximo-minimo)
  }
  percentiles = 100 - round(percentiles)

  z = correlaciones[,3]
  zlim = range(z, finite = TRUE)
  nlevels = 20
  levels = pretty(zlim, nlevels)

  par(mar = c(5,3,4,4))
  mar.orig <- (par.orig <- par(c("mar", "las", "mfrow")))$mar
  on.exit(par(par.orig))
  key.extend = FALSE
  w <- (3 + mar.orig[2L]) * par("csi") * 2.54
  w <- lcm(w * ifelse(key.extend, 0.9, 1.0))
  layout(matrix(c(2, 1), ncol = 2L), widths = c(1, w))
  par(las = 2)
  mar <- mar.orig
  mar[4L] <- mar[2L]
  mar[2L] <- 1
  par(mar = mar)
  plot.new()
  plot.window(xlim = c(0, 1), ylim = range(levels), xaxs = "i",
              yaxs = "i")
  rect(0, levels[-length(levels)], 1, levels[-1L], col =
       rev(heat.colors(length(levels))))
  axis(4, cex.axis = 0.7)
  title(title, cex.main = 0.7)

  par(bg="white", mar = c(4,4,3,2), cex.axis = 0.8)
  plot(0, 0, type="n", ann=FALSE, axes=FALSE)
  u <- par("usr") # The coordinates of the plot area
  rect(u[1], u[3], u[2], u[4], col="grey", border=NA)

  par(new=TRUE)
  plot(correlaciones[,1:2], col = miscolores[percentiles+1], pch = 20, main =
       main, xlab = xlab, ylab = ylab, log = log, las = 1)
  par (mypar)
}

## Funciones length.dat y length.plot, necesarias para representar datos
agrupados por intervalos.

length.dat <- function (input, datos, long, factor = NULL, norm = FALSE)  {

  ceros = which(rowSums(datos) == 0)
```

```
  if (length(ceros) > 0) {
    print(paste("Warning:", length(ceros),
                "features with 0 counts in all samples are to be removed for
                 this analysis."))
    datos = datos[-ceros,]
  }

  nsam <- NCOL(datos)
  if (nsam == 1) datos <- as.matrix(datos)

  if (is.null(factor)) {  # per sample
    print("Length bias detection information is to be computed for:")
    print("Media")

  }

  # Length
  if (length(ceros) > 0) long = long[-ceros]

  infobio = NULL; biotypes = NULL; bionum = NULL

  ## Calculations for plot

  longexpr = vector("list", length = 1 + length(biotypes))
  names(longexpr) = c("global", names(biotypes))

  numXbin = 200

  for (i in 1:length(longexpr))  {

    if (i == 1) {  # GLOBAL

      numdatos = length(long)
      numbins = floor(numdatos / numXbin)
      misbins = quantile(long, probs = seq(0,1,1/numbins), na.rm = TRUE)

      if (length(misbins) != length(unique(misbins))) {
        repes = names(table(misbins))[which(table(misbins) > 1)]
        for (rr in repes) {
          cuantos = length(which(misbins == rr))
          cuales = which(misbins == rr)
          sumo = (misbins[cuales[1]+cuantos] - misbins[cuales[1]])/cuantos
          for (j in cuales[-1]) misbins[j] = misbins[j-1] + sumo
        }
      }

      miclasi = cut(long, breaks = misbins, labels = FALSE)
      misbins = sapply(1:numbins, function (i) mean(misbins[i:(i+1)]))
      miclasi = misbins[miclasi]
      longexpr[[i]] = aggregate(datos, by = list("lengthbin" = miclasi), FUN =
                      mean, trim = 0.025)
    }
  }
  ## SPLINES REGRESSION MODEL
  library(splines)

  datos = longexpr[[1]]
  longi = datos[,1]
  knots =  c(rep(longi[1],3), seq(longi[1], longi[length(longi)-1],
length.out=round(length(longi)/10, 0)),
             rep(longi[length(longi)], 4))
  bx = splineDesign (knots, longi, outer.ok = TRUE)


  mismodelos = vector("list", length = ncol(datos)-1)
  names(mismodelos) = colnames(datos)[-1]

  for (i in 2:ncol(datos)) {
```

```
    mismodelos[[i-1]] = lm(datos[,i] ~ bx)
  }

  ## Results

  list("data2plot" = longexpr, "RegressionModels" = mismodelos)
}


length.plot <- function (dat, main, ylab, xlab, ylim, samples = NULL, toplot =
                         "global", toreport = FALSE,...)  {

  datos = dat[["data2plot"]]
  mismodelos = dat[["RegressionModels"]]

  if (is.null(samples)) samples <- 1:(ncol(datos[[1]])-1)
  if (is.numeric(samples)) { samples = colnames(datos[[1]])[samples+1] }
  if (is.numeric(toplot)) {
    if (toplot == 1) { toplot = "global"}
    else { toplot = names(toplot)[toplot + 1] }
  }

  for (i in 1:length(samples)) {
    matplot(datos[[1]][,1], cbind(datos[[1]][,samples[i]],
            mismodelos[[samples[i]]]$fit),
            type = "pl", col = c(1, rgb(1,1,1,0)), main=main, pch=20,
            ylab = ylab, xlab = xlab, ylim = ylim, ...)
  }
}



# Se cargan los datos de eXpress:
        # Se crea una lista con los archivos de entrada

express_files = list.files("/home/jordi/TFG/data/eXpress",full.names=T)

        # Se crea una lista de los transcritos a partir de un fichero
cualquiera
targets = sort(read.delim("/home/jordi/TFG/data/eXpress/
              RUN1_Batch_1_Ctr_6H_.xprs", as.is = TRUE)[,2])

        # Se crea un data.frame con una columna con todos los transcritos, a
la que se le irán añadiendo los datos
express_effcounts = data.frame(targets, row.names=1)

        # Cada archivo de eXpress tiene un orden de transcritos diferente, por
ello la importación de los datos es más compleja que los de RSEM

        #El siguiente bucle introduce todas las columnas "eff_counts" de cada
archivo en el data frame express_effcounts
for (k in 1:length(express_files)){
  file_data = data.frame(read.delim(express_files[k], header=TRUE, as.is =
                         TRUE)[,"target_id"], read.delim(express_files[k],
                         header=TRUE, as.is=TRUE)[,"eff_counts"],row.names=1)

  file_data = file_data[order(rownames(file_data)), ] # Ordena los valores
  column = gsub("_.xprs", "", gsub("/home/jordi/TFG/data/eXpress/", "",
                gsub("_2_", "_3_", express_files[k])))
  express_effcounts[column] = file_data[,1]
  print (paste("Columna", column, "añadida"))
}

# Se cargan los datos de RSEM:

rsem_files = list.files("/home/jordi/TFG/data/RSEM",full.names=T)
rsem_expcounts = data.frame(targets, row.names=1)
```

```
for (k in 1:length(rsem_files)){
  column = gsub(".isoforms.results", "", gsub("/home/jordi/TFG/data/RSEM/",
               "", gsub("_2_", "_3_", rsem_files[k])))
  rsem_expcounts[column] = read.delim(rsem_files[k], header=TRUE, as.is =
                                      TRUE)[,"expected_count"]
  print (paste("Columna", column, "añadida"))
}

# Se calcula la suma de los RUNs de cada muestra para cada matriz

sample_list = paste(rep(c("Ctr_", "Ik_"), each=18),
                    rep(c("0H_","2H_","6H_","12H_","18H_","24H_"),
                    times=rep(3,6)), c("Batch_1","Batch_3","Batch_4"),
                    sep = "")

# Se crean las nuevas matrices que serán rellenadas
express_effcounts_final = matrix(0, nrow = length(targets), ncol =
                                 length(sample_list), dimnames=list(targets,
                                 sample_list))
rsem_expcounts_final = express_effcounts_final


# Se rellenan las matrices
library(stringr) # Librería necesaria para usar la función str_sub

for (k in colnames(express_effcounts_final)) {
  express_effcounts_final[,k] = rowSums(express_effcounts[,grep(paste
                                        (str_sub(k,-7), str_sub(k, 1, -9),
                                        sep="_"),colnames(
                                        express_effcounts))])
}

for (k in colnames(rsem_expcounts_final)) {
  rsem_expcounts_final[,k] = rowSums(rsem_expcounts[
                                     ,grep(k,colnames(rsem_expcounts))])
}

# Normalización de los datos por CPM
library(NOISeq) # Necesaria para las funciones rpkm() y filtered.data()
cpm_express = rpkm(express_effcounts_final, long=1000, k=0)
cpm_rsem = rpkm(rsem_expcounts_final, long = 1000, k=0)

## Medias de los CPM por condiciones y tiempos
cpm_express_mean = matrix(0, nrow = length(targets), ncol =
length(sample_list_mean), dimnames=list(targets, sample_list_mean))
cpm_rsem_mean = cpm_express_mean

for (k in sample_list_mean){
  i = grep(k, colnames(cpm_express))
  cpm_express_mean[,k] = rowMeans(cpm_express[,i])
  cpm_rsem_mean[,k] = rowMeans(cpm_rsem[,i])
  print(paste("Cálculo de ", k, " finalizado", sep=""))
}


#####################################
############ Eficiencia #############
#####################################

# Etiquetas del eje x del gráfico
barnames = paste(rep(c("Ctr_", "Ik_"), each=18), rep(c("00H_","02H_","06H_",
                "12H_","18H_","24H_"), times=rep(3,6)), c("1","3","4"),
                sep = "")

eficiencia = matrix(0, ncol = 36, nrow = 2, dimnames = list(c("eXpress",
                    "RSEM"), sample_list))

eficiencia["eXpress", ] = colSums(express_effcounts_final)
```

```
eficiencia[ "RSEM",] = colSums(rsem_expcounts_final)

mypar = par() # Parámetros gráficos predeterminados
par(cex.axis=0.8, mgp = c(4.2, 0.8, 0), mar = c(5.2, 5.2, 2, 1))

barplot(eficiencia/1e+6, names.arg=barnames, ylab = "Counts totales  (en
        millones)", xlab = "Muestra", axis.lty = 1, las = 2, beside=TRUE, main
        = "Eficiencia eXpress y RSEM", col=c("chartreuse3",2),
        legend.text=FALSE)
legend("topright", legend = c("eXpress", "RSEM"), fill = c("chartreuse3",2),
       xjust=1, bty = "n")
par(mypar)

################################################
############# Conteos por millón #############
################################################

# Se representan en boxplots los datos transformados a CPM

par(mgp = c(4.2, 0.8, 0), mar = c(5.2, 5.2, 2, 1), cex=2)
boxplot(cpm_express+1, names = barnames, log = "y", las = 2, main =
        "Distribución de la expresión (eXpress)", xlab = "Muestra", ylab =
        "Counts por millón",  col = "Sky blue", border = "Blue", ylim =
        c(1, 30000))
boxplot(cpm_rsem+1, col = "Sky blue", border = "Blue", names = barnames, log =
        "y", ylim = c(1, 30000), las = 2, main = "Distribución de la expresión
        (RSEM)", xlab = "Muestra", ylab = "Counts por millón")
par(mypar)

################################################
############# N° de isoformas por gen #############
################################################

# Se calcula el número medio de isoformas por gen a los que asigna reads cada
programa

        # Tabla que relaciona los nombres de los transcritos con los genes a
los que pertenecen

transcritos.genes = read.delim("/home/jordi/TFG/ensemblToGeneName.txt", as.is
                               = TRUE, row.names = 1)
rownames(transcritos.genes) = paste("mm10_ensGene_",
                                    rownames(transcritos.genes), sep = "")

express_effcounts_final_genes = cbind(express_effcounts_final,
transcritos.genes[rownames(express_effcounts_final),])
rsem_expcounts_final_genes = cbind(rsem_expcounts_final,
transcritos.genes[rownames(rsem_expcounts_final),])


suma_express = rowSums(express_effcounts_final) # Se suman los counts de cada
transcrito para saber si se ha detectado

isoformas_detectadas_express = data.frame(Gen = express_effcounts_final_genes[
                                          ,37], "N° isoformas" = 0)

isoformas_detectadas_express[which(suma_express > 0),2] = 1

isonumbers_express = matrix(0, nrow = length(unique(
                            isoformas_detectadas_express[,1])), ncol=1,
                            dimnames = list(unique(
                            isoformas_detectadas_express[,1]), "N° isoformas"))

for (i in rownames(isonumbers_express)) {
  isonumbers_express[i,1] = sum(isoformas_detectadas_express
                                [isoformas_detectadas_express[,1] == i,2])
}
```

```
suma_rsem = rowSums(rsem_expcounts_final)
isoformas_detectadas_rsem = data.frame(Gen = rsem_expcounts_final_genes[,37],
                                       "N° isoformas" = 0)
isoformas_detectadas_rsem[which(suma_rsem > 0),2] = 1
isonumbers_rsem = matrix(0, nrow = length(unique(
                         isoformas_detectadas_rsem[,1])), ncol=1, dimnames =
                         list(unique(isoformas_detectadas_rsem[,1]),
                         "N° isoformas"))

for (i in rownames(isonumbers_rsem)) {
  isonumbers_rsem[i,1] = sum(isoformas_detectadas_rsem
                             [isoformas_detectadas_rsem[,1] == i,2])
}

# Genes que tienen alguna isoforma con al menos un count
express_genes = unique(isoformas_detectadas_express[which(
                       isoformas_detectadas_express[,2] == 1),1])
rsem_genes = unique(isoformas_detectadas_rsem[which(
                    isoformas_detectadas_rsem[,2] == 1),1])

isoformas_por_gen_express = sum(rowSums(express_effcounts_final) != 0) /
                                length(express_genes)
isoformas_por_gen_rsem = sum(rowSums(rsem_expcounts_final) != 0) /
                             length(rsem_genes)

isoformas_comunes = length(which(isoformas_detectadas_rsem[,2]
                           [isoformas_detectadas_express[,2] == 1] == 1))
genes_comunes = length(intersect(express_genes, rsem_genes))

# Se calculan el nª de isodormas reales de los genes detectados
genes_detectados = union(express_genes, rsem_genes)
isonumbers_reales = matrix(0, ncol = 1, nrow = length(genes_detectados),
                           dimnames = list(genes_detectados, "N° isoformas"))

for (i in genes_detectados) {
  isonumbers_reales[i,1] = sum(i == transcritos.genes[,1])
}

# Se representan todos los resultados
isototal = matrix(0, ncol = 3, nrow = 42, dimnames=list(c(1:42), c("eXpress",
                  "RSEM", "Reales")))
for (i in 1:42) {
  isototal[i,1] = length(isonumbers_express
                         [which(isonumbers_express[,1]==i)])
  isototal[i,2] = length(isonumbers_rsem[which (isonumbers_rsem[,1] == i)])
  isototal[i,3] = length(isonumbers_reales[which(isonumbers_reales[,1] == i)])

}

#Se agrupan los genes con más de 10 isoformas
isototal2 = matrix(c(isototal[1:10,1], 0,isototal[1:10,2], 0,
                   isototal[1:10,3], 0), ncol = 3, nrow = 11,
                  dimnames=list(c(1:10, ">10"),c("eXpress", "RSEM",
                  "Reales")))
isototal2[11,1] = sum(isototal[11:42,1])
isototal2[11,2] = sum(isototal[11:42,2])
isototal2[11,3] = sum(isototal[11:42,3])


png("isoformas.png", 2400, 1400)
par(cex=3, cex.main=1.5)
barplot(t(isototal2), names.arg = c(1:10, "≥11"), ylab = "Número de genes",
        xlab = "Número de isoformas detectadas", axis.lty = 1, las = 1,
        beside=TRUE, main = "Isoformas detectadas por gen", col=c(3,2,7),
        legend.text=FALSE)
```

```
legend("topright", legend = c("eXpress", "RSEM", "Reales"), fill = c(3,2,7),
xjust=1, cex=1.5)
dev.off()


# Isoformas detectadas por eXpress para los genes detectados con una sola
isoforma por RSEM

isoformas_1_rsem = rownames(isonumbers_rsem)[which (isonumbers_rsem[,1] == 1)]

isoformas_1_rsem_express = matrix(isonumbers_express[isoformas_1_rsem,],
                                  dimnames = list(isoformas_1_rsem, "Isoformas
                                  eXpress"))

isototal_1_rsem[,1]+1 = matrix(0, ncol = 1, nrow = 8, dimnames=list(c(0:7),
                                "eXpress"))
for (i in 0:7) {
  isototal_1_rsem[i+1,1] = length(isoformas_1_rsem_express[which
                                  (isoformas_1_rsem_express[,1] == i)])
}

png("isoformas_1.png", 2400, 1400)
par(cex=4)
barplot(t(isototal_1_rsem), names.arg = c(0:7), ylim = c(0,12000),ylab =
        "Número de genes", xlab = "Número de isoformas detectadas por
        eXpress", axis.lty = 1, las = 1, col=3, legend.text=FALSE)
text(x = seq(0.68, 9.08, by = 1.2), y = isototal_1_rsem[,1]+500,
labels=as.character(isototal_1_rsem[,1]))
dev.off()

##############################################
############# Gráficos de puntos #############
##############################################

# Se realiza un gráfico de puntos para los CPM de cada muestra

for (k in sample_list_mean) {
  png(paste(k, ".png", sep = ""), 1119, 371)
  par(mfrow = c(1,3), cex.main = 1.8, cex.lab = 1.5, cex.axis = 1.5)
  j = grep(k, sample_list)
  for (i in j) {
    main = sample_list[i]
    plot(cpm_rsem[,i] + 1, cpm_express[,i] + 1, main = main, xlab = "CPM
         RSEM", ylab = "CPM eXpress", pch = 20, log = "xy")
    abline(a=0, b=1, col = 2)
  }
  dev.off()
  par(mypar)
}



#######################################
##### Coeficientes de correlación #####
#######################################

cor.a_counts = data.frame(targets, row.names=1)

for (k in rownames(cor.a_counts)){
  i = i + 1
  cor.a_counts[k,1] = cor(rsem_expcounts_final[k,],
                          express_effcounts_final[k,])
}
```

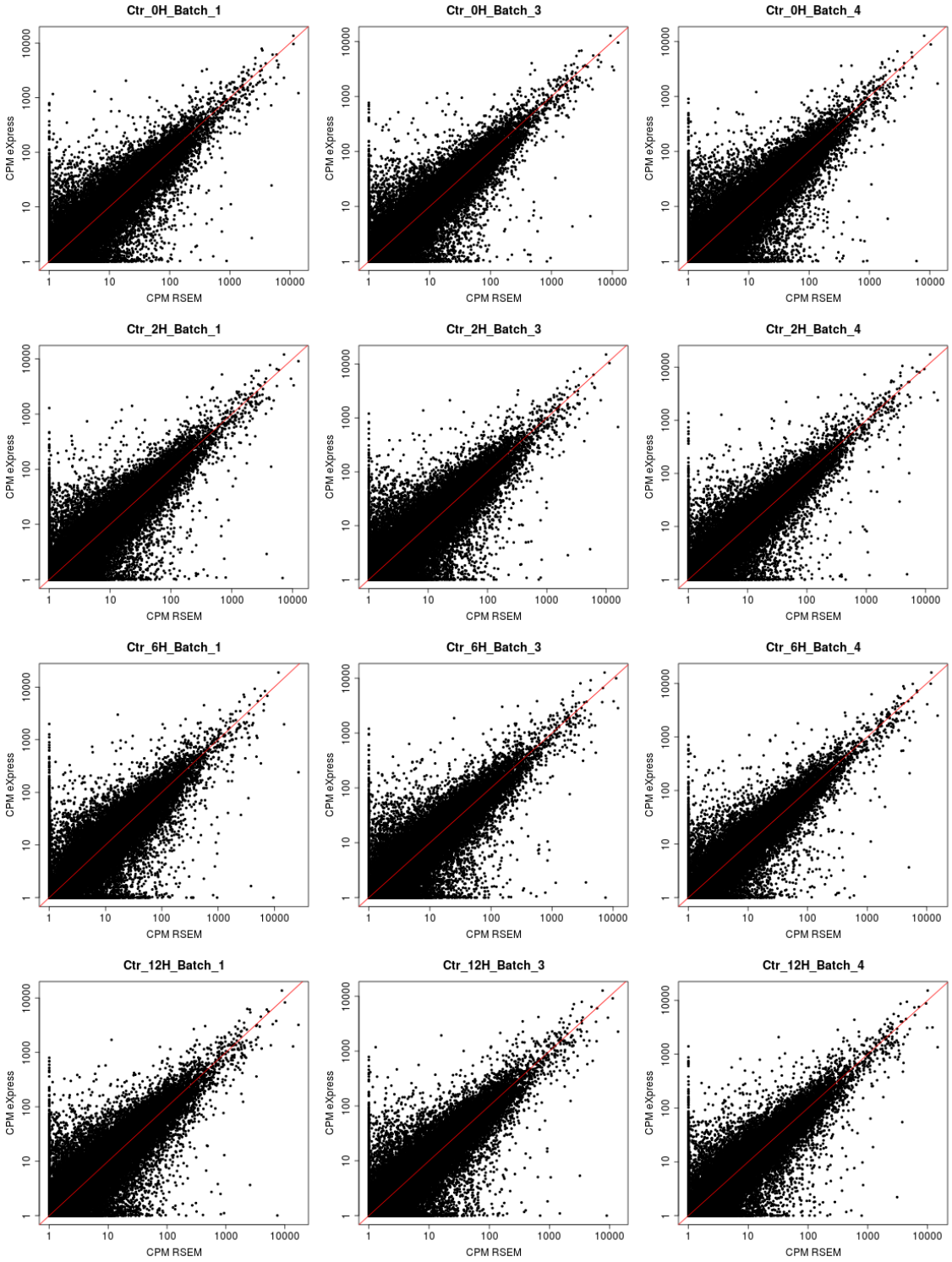
```
##############################################
#### Representación de las correlaciones #####
##############################################

# Se obtiene un gráfico de densidad

png("distribucion.png", 2400,1900)
par(cex=4, cex.main=1.5)
plot(density(cor.a_counts[,1], na.rm = TRUE), main = "Distribución de los
     valores de correlación", xlab = "Correlación RSEM y eXpress", ylab =
     "Densidad")
polygon(density(cor.a_counts[,1], na.rm=TRUE), col = "skyblue")
dev.off()


############################################################
##### Motivos de las discrepancias entre eXpress y RSEM #####
############################################################

# Matriz con la longitud de cada transcrito
isolength = matrix(read.delim("/home/jordi/TFG/data/RSEM/
                  Ctr_0H_Batch_1_RUN3.isoforms.results")[,3],
                  dimnames = list(targets, "Length"))

m_data_1a = as.matrix(cor.a_counts)
m_data_1a = matrix(m_data_1a, ncol = 1, dimnames = list(targets))
m_data_1a = na.omit(m_data_1a)

m_data_2a = m_data_1a


m_length_1a = rowMeans(cpm_rsem[rownames(m_data_1a),])
m_length_1a_length = isolength[rownames(m_data_1a),]

isonumber_1a = data.frame(Gen = express_effcounts_final_genes[
                          rownames(m_data_1a),37], "N° isoformas" = 0)
for (i in 1:nrow(isonumber_1a)) {
  isonumber_1a[i,2] = sum(isonumber_1a[i,1] == isonumber_1a[,1])
}

m_length_1a_iso = isonumber_1a[,2] #N° de isoformas de ese gen


m_length_2a = rowMeans(cpm_express[rownames(m_data_2a),])

mydata_1a = readData(data = m_data_1a, length = m_length_1a, factors =
                     "Media")
mydata_1a_length = readData(data = m_data_1a, length = m_length_1a_length,
                            factors = "Longitud")
mydata_1a_iso = readData(data = m_data_1a, length = m_length_1a_iso,
                         factors = "N° isoformas")


mydata_2a = readData(data = m_data_2a, length = m_length_2a,
                     factors = "Media")

# Se ejecutan las funciones
png("a cpm rsem.png", 2000, 1500)
par(cex=4, cex.main=1.5)
length.plot(length.dat(input = my_data_1a, datos = m_data_1a, long =
             m_length_1a), xlim = c(0.0001, 10000), ylim=c(0.3, 1), log = "x",
             main = "Correlación vs expresión RSEM", ylab = "Correlación RSEM y
             eXpress", xlab = "Nivel de expresión RSEM (CPM)")
dev.off()

png("a cpm express.png", 2000, 1500)
par(cex=4, cex.main=1.5)
length.plot(length.dat(input = my_data_2a, datos = m_data_2a, long =
```

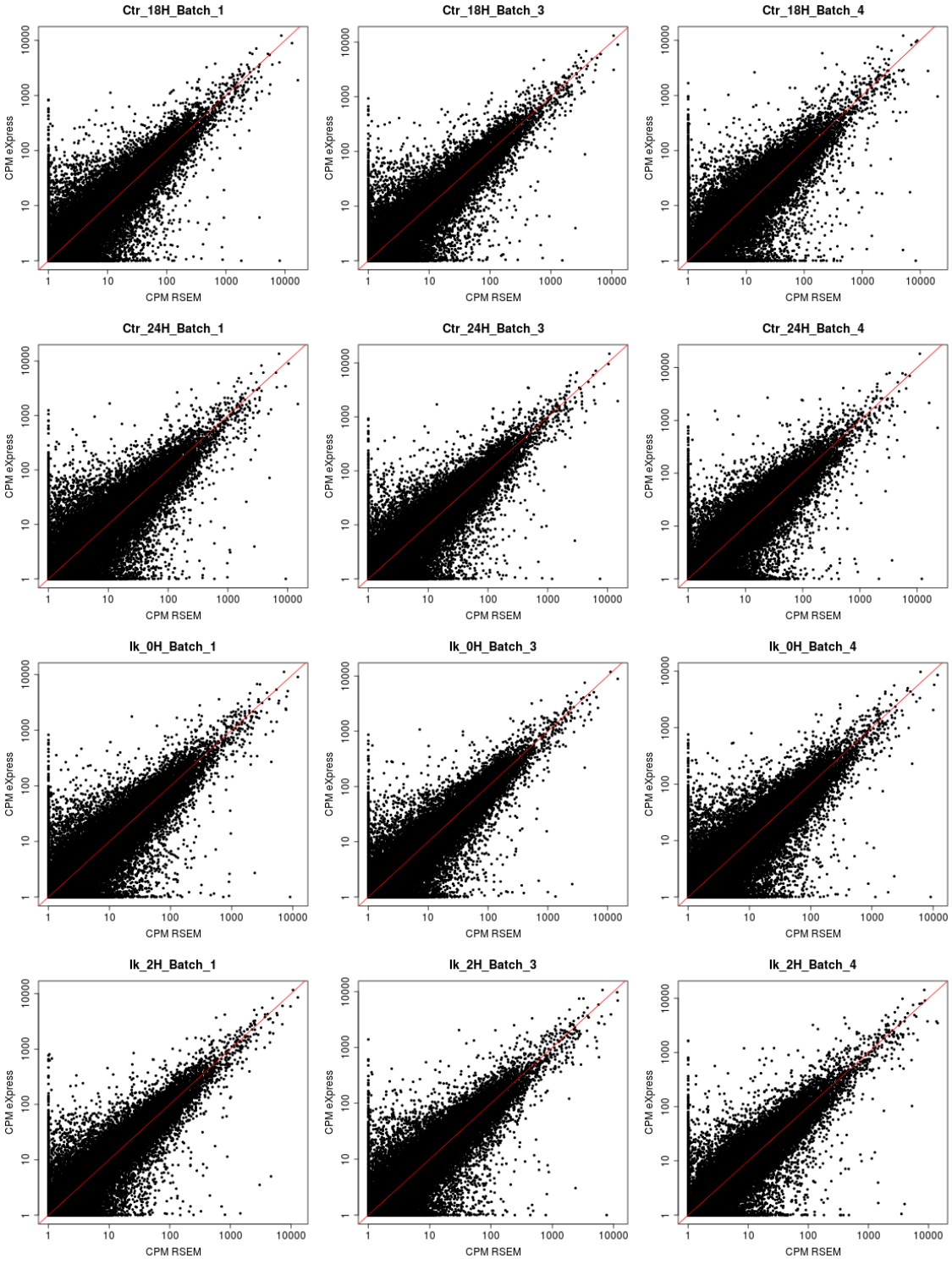
```
            m_length_2a), xlim = c(0.00000001,10000), ylim = c(0.3,1), log =
            "x", main = "Correlación vs expresión eXpress", ylab =
            "Correlación RSEM y eXpress", xlab = "Nivel de expresión eXpress
            (CPM)")
dev.off()

png("longitud.png", 2000, 1500)
par(cex=4, cex.main=1.5)
length.plot(length.dat(input = my_data_1a_length, datos = m_data_1a, long =
            m_length_1a_length), log = "x", xlim = c(100, 5e+04), ylim=c(0.68,
            0.85), main = "Correlación vs longitud de transcritos", ylab =
            "Correlación RSEM y eXpress", xlab = "Longitud de transcritos
            (pares de bases)", las = 2)
dev.off()


# Se hace un gráfico de puntos coloreados

correlaciones = data.frame(RSEM = rowMeans(cpm_rsem), eXpress =
                           rowMeans(cpm_express), Correlación = cor.a_counts,
                           row.names = targets)

# Se dividen los cpms en intervalos
numXbin = 200
numbins = floor(length(targets) / numXbin)

misbins_express = quantile(correlaciones[,1][which(correlaciones[,1] != 0)],
probs = seq(0,1,1/numbins), na.rm = TRUE)
misbins_rsem = quantile(correlaciones[,2][which(correlaciones[,2] != 0)],
probs = seq(0,1,1/numbins), na.rm = TRUE)

targets_bins = data.frame(RSEM = cut(correlaciones[,1], breaks =
                          misbins_rsem), eXpress = cut(correlaciones[,2],
                          breaks = misbins_express), row.names = targets)

correlaciones_bins_rsem = matrix(0, nrow = numbins, ncol = 3, dimnames =
                                 list(levels(cut(correlaciones[,1],
                                 misbins_rsem)), c("CPM RSEM", "CPM eXpress",
                                 "Correlación")))

for (i in rownames(correlaciones_bins_rsem)) {
  k = grep(i, targets_bins[,1], fixed = TRUE)
  correlaciones_bins_rsem[i,1] = mean(correlaciones[k,1], na.rm = TRUE)
  correlaciones_bins_rsem[i,2] = mean(correlaciones[k,2], na.rm = TRUE)
  correlaciones_bins_rsem[i,3] = mean(correlaciones[k,3], na.rm = TRUE)
}



correlaciones_bins_express = matrix(0, nrow = numbins_express, ncol = 3,
                                    dimnames = list(levels(
                                    cut(correlaciones[,2], misbins_express)),
                                    c("CPM RSEM", "CPM eXpress",
                                    "Correlación")))

for (i in rownames(correlaciones_bins_express)) {
  k = grep(i, targets_bins[,2], fixed = TRUE)
  correlaciones_bins_express[i,1] = mean(correlaciones[k,1], na.rm = TRUE)
  correlaciones_bins_express[i,2] = mean(correlaciones[k,2], na.rm = TRUE)
  correlaciones_bins_express[i,3] = mean(correlaciones[k,3], na.rm = TRUE)
}


cpm(correlaciones_bins_express, "CPM eXpress vs CPM RSEM (bins eXpress)")
cpm(correlaciones_bins_rsem[7:473,], "CPM eXpress vs CPM RSEM (bins RSEM)")
```

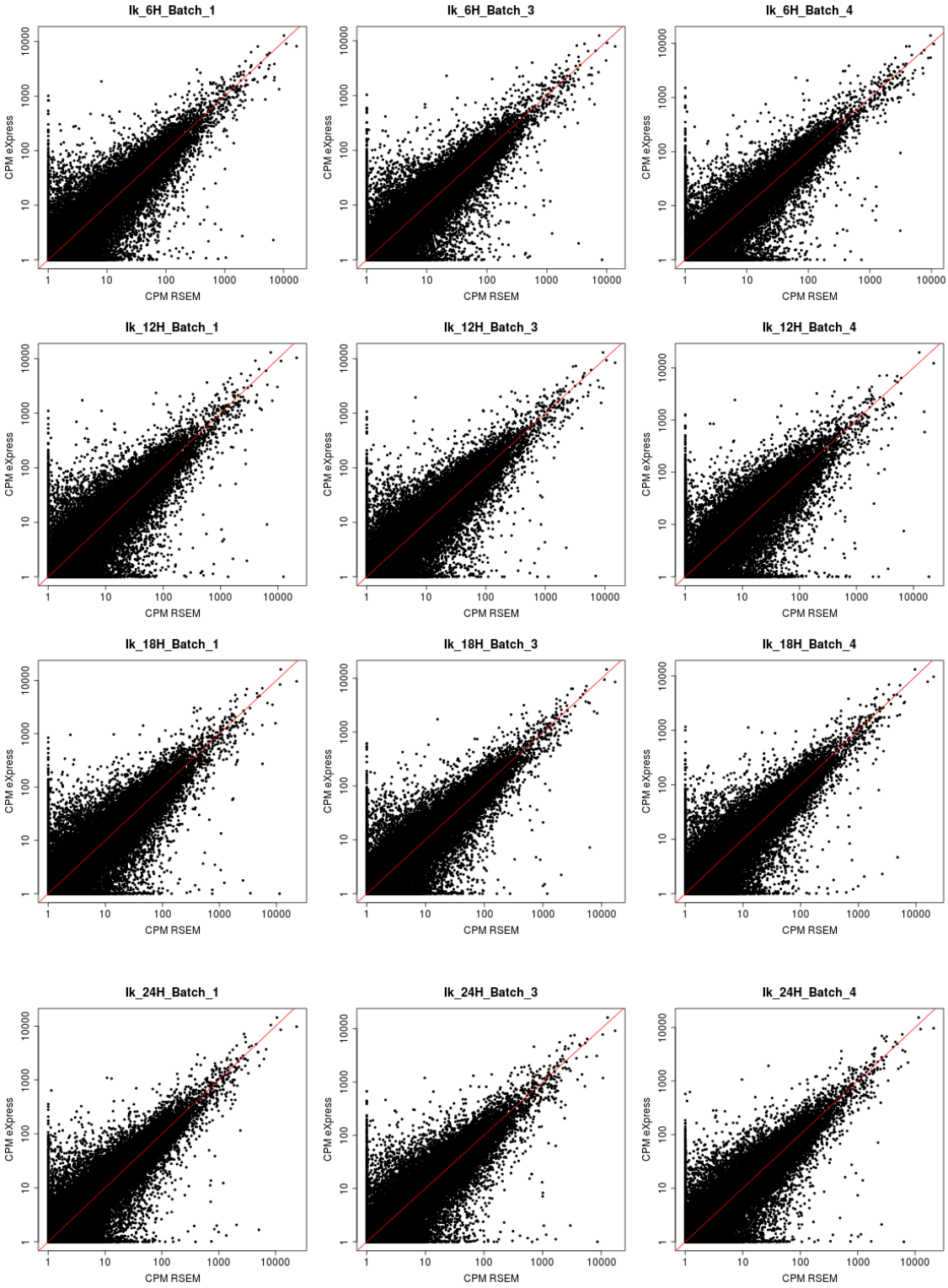
```
# Relación entre correlación y número de isoformas del gen

isonumber_cor = data.frame(Correlación = cor.a_counts[,1], "N° isoformas" = 0,
                           row.names = targets)
for (i in targets) {
  isonumber_cor[i,2] = sum(express_effcounts_final_genes[i,37] ==
transcritos.genes[,1])
}

box = list()
list11 = c()
for (i in 1:42) {
  if (isototal[i,3] > 0 & i < 11) {
    box[as.character(i)] = list(isonumber_cor[which(
                                  isonumber_cor[,2] == i),1])
  }

  if (isototal[i,3] > 0 & i > 11) {
    list11 = append(list11, isonumber_cor[which(isonumber_cor[,2] == i),1])
  }
}

box["≥11"] = list(list11)

png("nisoformas.png", 2700, 1500)
par(cex=2, cex.main=2.5, cex.lab=2, mar=c(5,5,4,4), cex.axis=2)
boxplot(box, col = "Sky blue", border = "Blue", main = "Correlación y n° de
        isoformas", xlab = "N° de isoformas del gen", ylab = "Coeficiente de
        correlación RSEM y eXpress" )
dev.off()


# Se calcula la correlación entre las muestras para cada método

library(NOISeq)
express_heatmap = filtered.data(express_effcounts_final, norm = FALSE,
                                factor = rep(sample_list_mean, each = 3),
                                method = 1, cpm = 1)
express_heatmap = rpkm(express_heatmap, long = 1000, k = 0)

rsem_heatmap = filtered.data(rsem_expcounts_final, norm = FALSE,
                             factor = rep(sample_list_mean, each = 3),
                                          method = 1, cpm = 1)
rsem_heatmap = rpkm(rsem_heatmap, long = 1000, k = 0)

png("heatmap express.png",2000,1500)
heatmap(-cor(express_heatmap), symm = TRUE, cexRow=2.5, cexCol=2.5,
        margins=c(18,10))
dev.off()

png("heatmap rsem.png",2000,1500)
heatmap(-cor(rsem_heatmap), symm = TRUE, cexRow=2.5, cexCol=2.5,
        margins=c(18,10))
dev.off()

express_cor = cor(express_effcounts_final)
rsem_cor = cor(rsem_expcounts_final)

sample_list_mean = paste(rep(c("Ctr_", "Ik_"), each=6),
                         rep(c("0H","2H","6H","12H","18H","24H"),
                         times=rep(1,6)), sep = "")

express_cor_reps = matrix (0, nrow = 12, ncol = 3, dimnames =
                             list(sample_list_mean, c("1-3", "1-4", "3-4")))
rsem_cor_reps = express_cor_reps

for (i in sample_list_mean) {
```

```
  k = grep(i, colnames(express_cor))
  express_cor_reps[i,1] = express_cor[k,k][1,2]
  express_cor_reps[i,2] = express_cor[k,k][1,3]
  express_cor_reps[i,3] = express_cor[k,k][2,3]
  rsem_cor_reps[i,1] = rsem_cor[k,k][1,2]
  rsem_cor_reps[i,2] = rsem_cor[k,k][1,3]
  rsem_cor_reps[i,3] = rsem_cor[k,k][2,3]
}


png("correlaciones.png", 2000,1800)
par(cex=5)
plot(rsem_cor_reps, express_cor_reps, xlim = range(0.8,1), ylim =
     range(0.8,1), pch = 20, main = "Correlación entre réplicas", xlab =
     "RSEM", ylab = "eXpress")
abline(a=0,b=1,col="grey",lty=2,lwd=6)
dev.off()
```

7.4. Anexo IV: Gráficas de expresión de transcritos




Ctr_18H_Batch_1
Ctr_18H_Batch_3
Ctr_18H_Batch_4
Ctr_24H_Batch_1
Ctr_24H_Batch_3
Ctr_24H_Batch_4
Ik_0H_Batch_1
Ik_0H_Batch_3
Ik_0H_Batch_4
Ik_2H_Batch_1
Ik_2H_Batch_3
Ik_2H_Batch_4
CPM eXpress
CPM RSEM
1
10
100
1000
10000


Ik_6H_Batch_1
Ik_6H_Batch_3
Ik_6H_Batch_4
Ik_12H_Batch_1
Ik_12H_Batch_3
Ik_12H_Batch_4
Ik_18H_Batch_1
Ik_18H_Batch_3
Ik_18H_Batch_4
Ik_24H_Batch_1
Ik_24H_Batch_3
Ik_24H_Batch_4
CPM eXpress
CPM RSEM
1
10
100
1000
10000

7.5. Anexo V: *Script* de PCA

```
PCA.GENES<-function(X)
{
  n<-ncol(X)
  p<-nrow(X)
  offset<-apply(X,2,mean)
  Xoff<-X-(cbind(matrix(1,p,1))%*%rbind(offset))

  eigen<-eigen(Xoff%*%t(Xoff)/(p-1))
  var<-cbind(eigen$values/sum(eigen$values),
       cumsum(eigen$values/sum(eigen$values)))

  loadings2<-eigen$vectors
  scores2<-t(Xoff)%*%loadings2

  normas2<-sqrt(apply(scores2^2,2,sum))

  scores1<-loadings2%*%diag(normas2)
  loadings1<-scores2%*%diag(1/normas2)

  output<-list(eigen,var,scores1,loadings1)
  names(output)<-c("eigen","var.exp","scores","loadings")
  output
}

# Para hacer el PCA se usan los datos filtrados por 1 CPM, normalizados por
TMM y transformados a log2

library(NOISeq)

express_pca = filtered.data(express_effcounts_final, norm = FALSE, factor =
rep(sample_list_mean, each = 3), method = 1, cpm = 1)
express_pca = tmm(express_pca, long = 1000, k = 0)
express_pca = log2(express_pca+1)

rsem_pca = filtered.data(rsem_expcounts_final, norm = FALSE, factor =
rep(sample_list_mean, each = 3), method = 1, cpm = 1)
rsem_pca = tmm(express_pca, long = 1000, k = 0)
rsem_pca = log2(rsem_pca+1)

# Lista pasada a PCA.GENES
datos_todos=list("eXpress"=express_pca, "RSEM"=rsem_pca)

# Función para dividir vectores en "n" partes
n_split = function(vec, n) { split(vec, factor(sort(rank(vec)%%n))) }

# Carácteres para controles e ikaros
mychr = list("control" = 16, "ikaros" = 17)

# Colores para distintos tiempos (en orden: 0, 2, 6, 12, 18, 24 h)
mycolours = c("chartreuse3", "black", "gold1", "firebrick3", "darkorchid2",
              "dodgerblue2")

# Índices de cada tiempo (3 réplicas por clase y tiempo) en los datos
tiempos_control = n_split(1:18, 6)
tiempos_ikaros  = n_split(19:36, 6)


# Leyenda
datos_leyenda = c(paste(rep("C", 6), sprintf("%02d", c(0,2,6,12,18,24)), "h",
                  sep=""), paste(rep("I", 6), sprintf("%02d",
                  c(0,2,6,12,18,24)), "h", sep=""))

# Cálculo de PCA
pca.results <- lapply(datos_todos, function (x) PCA.GENES(t(x)))

# Scores plot
```

```
png("PCA_scores_tecrep.png", width = 8, height = 4*2, units = "in",
    res = 200)

par(mfcol = c(2,1))

for (i in 1:length(pca.results)) {

  # PC1 & PC2
  rango = diff(range(pca.results[[i]]$scores[,1:2]))

  plot(pca.results[[i]]$scores[,1:2], col = "white",
       xlab = paste("PC 1 ", round(pca.results[[i]]$var.exp[1,1]*100,0),
                    "%", sep = ""),
       ylab = paste("PC 2 ", round(pca.results[[i]]$var.exp[2,1]*100,0),
                    "%", sep = ""),
       main = names(pca.results)[i],
       xlim = range(pca.results[[i]]$scores[,1:2]) + 0.02*rango*c(-1,1),
       ylim = range(pca.results[[i]]$scores[,1:2]) + 0.02*rango*c(-1,1))
  if (i == 1) {
    legend("bottomleft", datos_leyenda, lty=0, pch = c(rep(16,6), rep(17,6)),
           col = rep(mycolours, 2), ncol = 2, cex = 0.65)
  }
  # 18 primeros: control
  for (j in 1:length(tiempos_control)) {
    points(pca.results[[i]]$scores[tiempos_control[[j]],1],
           pca.results[[i]]$scores[tiempos_control[[j]],2],
           pch = mychr[["control"]], col = mycolours[j], cex = 1.5)
  }

  # 18 últimos: ikaros
  for (j in 1:length(tiempos_ikaros)) {
    points(pca.results[[i]]$scores[tiempos_ikaros[[j]],1],
           pca.results[[i]]$scores[tiempos_ikaros[[j]],2],
           pch = mychr[["ikaros"]], col = mycolours[j], cex = 1.5)
  }
}

graphics.off()
```

7.6. Anexo VI: *Script* de EBSeq

```
# Primero se elige el tipo de filtrado

cors = matrix(0, nrow = 51, ncol = 3, dimnames = list(seq(0,5, by = 0.1),
             c("Counts mínimos", "Correlación", "N° de transcritos")))
cors[,1] = seq(0,5, by = 0.1)
cors[1,2] = mean(cor.a_counts[,1], na.rm = TRUE)
cors[1,3] = length(intersect(which(rowSums(express_effcounts_final) > 0),
which(rowSums(rsem_expcounts_final) > 0)))

library(NOISeq)

contador = 1
for (k in seq(0,5, by = 0.1)){
  contador = contador + 1
  express = filtered.data(express_effcounts_final, norm = FALSE, factor =
                          rep(sample_list_mean, each = 3), method = 1, cpm = k)
  rsem = filtered.data(rsem_expcounts_final, norm = FALSE, factor =
                       rep(sample_list_mean, each = 3), method = 1, cpm = k)
  common = intersect(rownames(express), rownames(rsem))
  cors[contador,3] = length(common)
  rsem_express = matrix(0, nrow = length(common), ncol = 1,
                        dimnames = list(common, "Correlación"))
  for (i in common){
    rsem_express[i,1] = cor(rsem[i,], express[i,])
  }
  cors[contador,2] = mean(rsem_express)
  print(k)
}

cpm(cors, main = "Filtrado de transcritos", log = "", ylab = "Coeficiente de
    correlación RSEM y eXpress", xlab = "CPM mínimos", title = "Número
    de\ntranscritos")


# Filtrado de los datos por 0,5 CPM

express_filtered = filtered.data(express_effcounts_final, norm = FALSE,
                                 factor = rep(sample_list_mean, each = 3),
                                 method = 1,cpm = 0.5)
rsem_filtered = filtered.data(rsem_expcounts_final, norm = FALSE,
                              factor = rep(sample_list_mean, each = 3),
                              method = 1, cpm = 0.5)

# Se calculan distintos datos para los datos de eXpress después de filtrar por
0,5 CPM
        # Se relacionan los transcritos con sus genes

express_filtered_genes = cbind(express_filtered,
                               transcritos.genes[rownames(express_filtered),])
rsem_filtered_genes = cbind(rsem_filtered,
                            transcritos.genes[rownames(rsem_filtered),])

isoformas_detectadas_express_filtered = data.frame(Gen=express_filtered_genes
                                                   [,37], "N° isoformas" = 1)
isoformas_detectadas_rsem_filtered = data.frame(Gen=rsem_filtered_genes[,37],
                                                "N° isoformas" = 1)


isonumbers_express_filtered = matrix(0, nrow=
                                     length(unique(express_filtered_genes
                                     [,37])), ncol=1, dimnames = list
                                     (unique(express_filtered_genes[,37]),
                                      "N° isoformas"))

isonumbers_rsem_filtered = matrix(0, nrow = length(unique(rsem_filtered_genes
```

```
                                    [,37])), ncol=1, dimnames = list
                                     (unique(rsem_filtered_genes[,37]),
                                     "N° isoformas"))

for (i in rownames(isonumbers_express_filtered)) {
  isonumbers_express_filtered[i,1] = sum(isoformas_detectadas_express_filtered
                                         [isoformas_detectadas_express_filtered
                                         [,1] == i,2])
}

for (i in rownames(isonumbers_rsem_filtered)) {
  isonumbers_rsem_filtered[i,1] = sum(isoformas_detectadas_rsem_filtered
                                      [isoformas_detectadas_rsem_filtered[,1]
                                      == i,2])
}

      # Genes que tienen alguna isoforma con al menos un count
express_filtered_ngenes = unique(isoformas_detectadas_express_filtered[,1])
rsem_filtered_ngenes = unique(isoformas_detectadas_rsem_filtered[,1])

isoformas_por_gen_express_filtered = sum(rowSums(express_filtered) != 0) /
                                         length(express_filtered_ngenes)
isoformas_por_gen_rsem_filtered = sum(rowSums(rsem_filtered) != 0) /
                                      length(rsem_filtered_ngenes)

        # Genes detectados por los dos métodos
genes_detectados_filtered = intersect(express_filtered_ngenes,
                                      rsem_filtered_ngenes)

# Se cargan los datos necesarios
library(EBSeq)

Conditions = as.factor(rep(c("Ctr_0H", "Ctr_24H", "Ik_0H", "Ik_24H"), each=3))

IsoNames = targets # Nombre de los transcritos

IsosGeneNames = transcritos.genes[rownames(express_effcounts_final),] # Nombre
de los genes a los que corresponde cada transcrito

IsoMat = express_filtered[,c(1:3, 16:21, 34:36)] # Se cargan Ctr e Ik a 0 y 24
horas

        # Se carga la cuantificación de las isoformas por eXpress

express_gen = matrix(0, nrow = length(unique(IsosGeneNames)), ncol=36,
                     dimnames = list(unique(IsosGeneNames), sample_list))

for (i in colnames(express_gen)) {
  for (j in rownames(express_gen)) {
    k = paste("^", j, "$", sep="")
    x = grep(k, IsosGeneNames)
    express_gen[j,i] = sum(express_effcounts_final[x,i])
  }
  print(i)
}

library(NOISeq)

express_gen_filtered = filtered.data(express_gen, norm = FALSE, factor =
                                     rep(sample_list_mean, each = 3),
                                     method = 1, cpm = 0.5)

IsoMat_genes = express_gen_filtered[,c(1:3, 16:21, 34:36)]

library(edgeR) # Se carga el paquete edgeR para calculas los factores de
normalización por TMM

IsoSizes = calcNormFactors(IsoMat, method = "TMM")
```

```
names(IsoSizes) = colnames(IsoMat)

IsoSizes_genes = calcNormFactors(IsoMat_genes, method = "TMM")
names(IsoSizes_genes) = colnames(IsoMat_genes)

NgList = GetNg(IsoNames, IsosGeneNames)
IsoNgTrun = NgList$IsoformNgTrun

patterns = GetPatterns(Conditions)


IsoMultiOut = EBMultiTest(Data=IsoMat, NgVector=IsoNgTrun, AllParti=patterns,
                          Conditions=Conditions,sizeFactors=IsoSizes,
                           maxround=7)

MultiOut=EBMultiTest(IsoMat_genes,NgVector=NULL,Conditions=Conditions,
                     AllParti=patterns, sizeFactors=IsoSizes_genes,
                     maxround=7)

IsoMultiPP=GetMultiPP(IsoMultiOut)

IsoMultiPP_genes=GetMultiPP(MultiOut)

# Se crea un data frame juntando los resultados

comp = data.frame(Transcrito = rownames(IsoMat), Patrón_Transcrito =
                  IsoMultiPP$MAP, Gen = transcritos.genes[rownames(IsoMat),1],
                  Patrón_Gen =
                  IsoMultiPP_genes$MAP[transcritos.genes[rownames(IsoMat),1]])

# Se agrupan los patrones no deseados en "EE" y el resto en "DE"

comp_DE = comp
genes_DE = IsoMultiPP_genes$MAP
patterns_EE = c("Pattern1", "Pattern4", "Pattern6")

for (i in patterns_EE) {
  k = paste("^", i, "$", sep="")
  comp_DE[,2] = gsub(k, "EE", comp_DE[,2])
  comp_DE[,4] = gsub(k, "EE", comp_DE[,4])
  genes_DE = gsub(k, "EE", genes_DE)
}
comp_DE[,2][which(comp_DE[,2] != "EE")] = "DE"
comp_DE[,4][which(comp_DE[,4] != "EE")] = "DE"
genes_DE[which(genes_DE != "EE")] = "DE"


# Se calculan otros resultados

ngenes_EE = which(genes_DE == "EE")

ngenes_DE = which(genes_DE == "DE")

ngenes_DE_DE = list() # Genes DE con alguna isoforma DE
ngenes_DE_EE = list() # Genes DE sin ninguna isoforma DE
ngenes_DE_nois = list() # Genes DE sin ninguna isoforma que haya pasado el
filtro

for (i in ngenes_DE) {
  k = rownames(IsoMat_genes)[i]
  x = grep(paste("^", k, "$", sep=""),comp_DE[,3])
  if (length(x) == 0) {
    ngenes_DE_nois = append(ngenes_DE_nois, k)
  }
  else{
    j = length(which(comp_DE[x,2] == "DE"))
    if (j > 0){
      ngenes_DE_DE = append(ngenes_DE_DE,k)
```

```
    }
    else {
      ngenes_DE_EE = append(ngenes_DE_EE,k)
    }
  }
}

ngenes_EE_DE = list() # Genes EE con alguna isoforma DE
ngenes_EE_EE = list() # Genes EE sin ninguna isoforma DE
ngenes_EE_nois = list() # Genes EE sin ninguna isoforma que haya pasado el
filtro

for (i in ngenes_EE) {
  k = rownames(IsoMat_genes)[i]
  x = grep(paste("^", k, "$", sep=""),comp_DE[,3])
  if (length(x) == 0) {
    ngenes_EE_nois = append(ngenes_EE_nois, k)
  }
  else{
    j = length(which(comp_DE[x,2] == "DE"))
    if (j > 0){
      ngenes_EE_DE = append(ngenes_EE_DE,k)
    }
    else {
      ngenes_EE_EE = append(ngenes_EE_EE,k)
    }
  }
}

# Se estudian los resultados de genes implicados en la diferenciación de
células B

genes_B = scan("../../genes_diferenciacion.txt", what="list", sep="\n") # Se
guarda la lista de genes importantes en la diferenciación de células B

genes_B_matrix = matrix(0, ncol=4, nrow=length(genes_B),
                        dimnames=list(genes_B, c("DE/EE", "Isoformas DE",
                        "Isoformas EE", "Isoformas filtradas")))

for (i in genes_B) {
  x = grep(paste("^", i, "$", sep = ""),comp_DE[,3])
  genes_B_matrix[i,1] = genes_DE[i]
  genes_B_matrix[i,2] = length(which(comp_DE[x,2] == "DE"))
  genes_B_matrix[i,3] = length(which(comp_DE[x,2] == "EE"))
  genes_B_matrix[i,4] = length(grep(paste("^", i, "$", sep = ""),
IsosGeneNames)) - as.integer(genes_B_matrix[i,2]) -
as.integer(genes_B_matrix[i,3])
}

write.csv(genes_B_matrix, "Tabla genes B.csv", quote=FALSE)
```

7.7. Anexo VII: *Script* de contextualización de datos

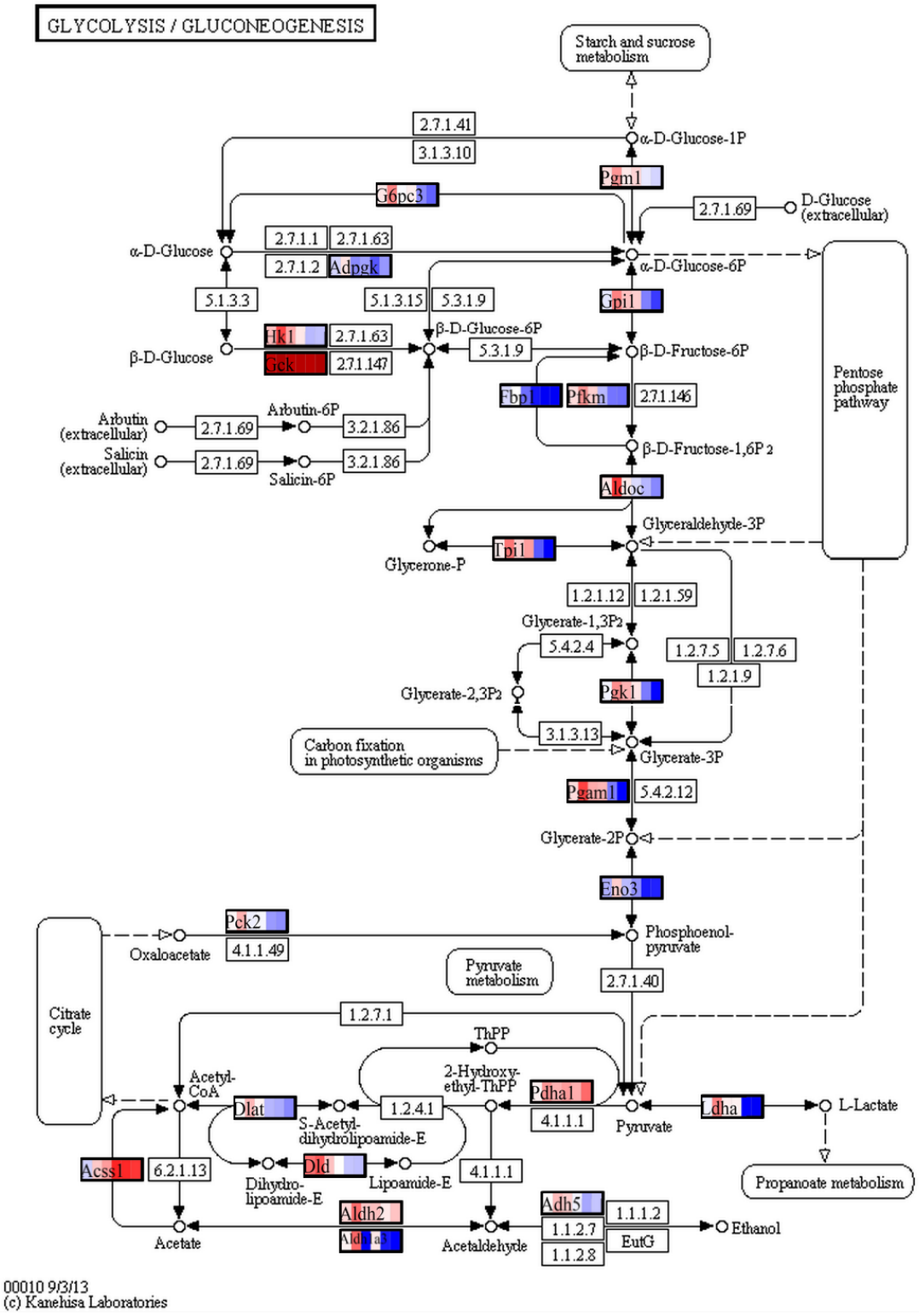
```
mycolors = c("chartreuse3", "black", "gold2", "firebrick3", "darkorchid2",
             "dodgerblue2", "aquamarine3", "coral", "bisque4", "deeppink",
             "grey60", "lemonchiffon3", "lightblue2", "slategrey")

 # Se normalizan los datos de los genes y transcritos filtrados a TMM

library(NOISeq)
tmm_express_gen = tmm(express_gen_filtered, long=1000, k=0)
tmm_express_filtered = tmm(express_filtered, long=1000, k=0)

# Se calculan las medias de las réplicas

tmm_express_gen_mean = matrix(0, nrow = length(rownames(tmm_express_gen)),
                              ncol = length(sample_list_mean),
                              dimnames=list(rownames(tmm_express_gen),
                                            sample_list_mean))

tmm_express_filtered_mean = matrix(0, nrow = length(
                                    rownames(tmm_express_filtered))
                                    ncol = length(sample_list_mean),

dimnames=list(rownames(tmm_express_filtered), sample_list_mean))

for (k in sample_list_mean){
  i = grep(k, colnames(cpm_express_gen))
  tmm_express_gen_mean[,k] = rowMeans(tmm_express_gen[,i])
  tmm_express_filtered_mean[,k] = rowMeans(tmm_express_filtered[,i])
}


# Se importan las listas de genes implicados en las rutas de la glicólisis, la
autofagia y FOXO

genes_glicolisis = read.delim("../../genes_glicolisis.txt", sep="\t",
                              header=FALSE)[,2])
genes_autofagia = read.delim("../../genes_autofagia.txt", sep="\t",
                             header=FALSE)[,2])
genes_foxo = read.delim("../../genes_foxo.txt", sep="\t", header=FALSE)[,2])

# Se representa el perfil de expresión de cada gen y de sus transcritos

contador3 = 1
for (j in list(genes_glicolisis, genes_autofagia, genes_foxo)){
  if (contador3==1){
    ruta="glicolisis"
  }
  if (contador3==2) {
    ruta = "autofagia"
  }
  if (contador3==3) {
    ruta = "foxo"
  }
  png(paste(ruta,"_1", ".png", sep=""), 3000, 1500)
  contador2 = 1
  par(mfrow = c(2,4), cex=2.5)
  plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')
  legend("center", cex=1.8, legend=c("Gen", "Transcritos", "DE", "EE"),
         pch=list(16,17,17,2), lty=c(1,1,1,2), col=c(2,1,1,1), lwd=5)
  par(new=T)
  plot(0,xaxt='n',yaxt='n',bty='n',pch='',ylab='',xlab='')

  for (i in j) {
    x= paste("^", i, "$", sep="")
    if (length(grep(x,rownames(tmm_express_gen_mean))) > 0) {
      transcritos = intersect(IsoNames[grep(x, IsosGeneNames, fixed=FALSE)],
```

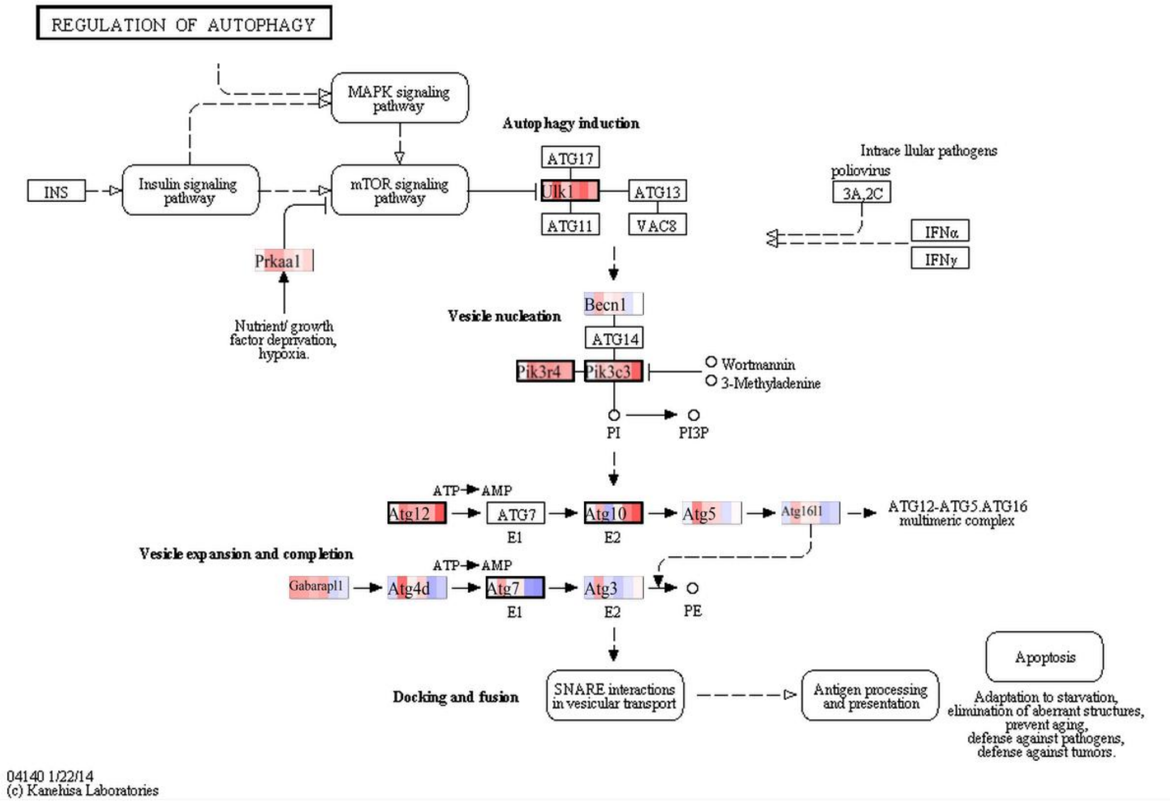
```
                                rownames(tmm_express_filtered))
      tmms = tmm_express_filtered_mean[transcritos,]
      tmms_gen = tmm_express_gen_mean[i,]
      minimo = min(c(tmms, tmms_gen))
      maximo = max(c(tmms, tmms_gen))
      contador=1

      for (k in transcritos){
        if (comp_DE[k,2] == "DE" & length(transcritos) > 1) {
          matplot(tmms[k,], type=c("b"), pch=17, ylim = c(minimo, maximo), lty
                  = 1, col = mycolors[c(2,5:14)][contador],
                  xlab = "", ylab = "", xaxt = "n", yaxt="n", lwd=3)
        }
        if (comp_DE[k,2] == "DE" & length(transcritos) == 1) {
          matplot(tmms, type=c("b"), pch=17, ylim = c(minimo, maximo), lty =
                  1, col = mycolors[c(2,5:14)][contador],
                  xlab = "", ylab = "", xaxt = "n", yaxt="n", lwd=3)
        }
        if (comp_DE[k,2] == "EE" & length(transcritos) > 1) {
          matplot(tmms[k,], type=c("b"), pch=2, ylim = c(minimo, maximo), lty
                  = 2, col = mycolors[c(2,5:14)][contador],
                  xlab = "", ylab = "", xaxt = "n",yaxt="n", lwd=3)
        }
        if (comp_DE[k,2] == "EE" & length(transcritos) == 1) {
          matplot(tmms, type=c("b"), pch=2, ylim = c(minimo, maximo), lty = 2,
                  col = mycolors[c(2,5:14)][contador],
                  xlab = "", ylab = "", xaxt = "n", yaxt="n", lwd=3)
        }
        par(new=T)
        contador = contador+1
      }

      if (genes_DE[i] == "DE")
        matplot(tmms_gen, type=c("b"), pch=16, ylim = c(minimo, maximo), lty =
                1, col = 2, main = i, xlab = "Condiciones",
                ylab = "Nivel de expresión (TMM)", xaxt = "n", lwd=3)

      if (genes_DE[i] == "EE")
        matplot(tmms_gen, type=c("b"), pch=1, ylim = c(minimo, maximo), lty =
                 2, col = 2, main = i, xlab = "Condiciones",
                 ylab = "Nivel de expresión (TMM)", xaxt = "n", lwd=3)

      axis(1, at = 1:12, labels = sample_list_mean, cex.axis = 0.6, las=2)
      abline(v=6.5, lty=2, lwd=3.5)
      print(i)
      contador2 = contador2+1
      if (contador2 %% 8 == 0){
        dev.off()
        png(paste(ruta, "_", contador2/8 + 1, ".png", sep=""), 3000, 1500)
        par(mfrow = c(2,4), cex=2.5)
      }
    }
  }
  dev.off()
  contador3 = contador3 + 1
}


# Se representan los perfiles de genes con isoformas cambiantes
# Flii

isoflii = intersect(IsoNames[grep("Flii", IsosGeneNames, fixed=TRUE)],
rownames(tmm_express_filtered))

tmm_flii = tmm_express_filtered_mean[isoflii,]
tmm_flii_gen = tmm_express_gen_mean["Flii",]
```

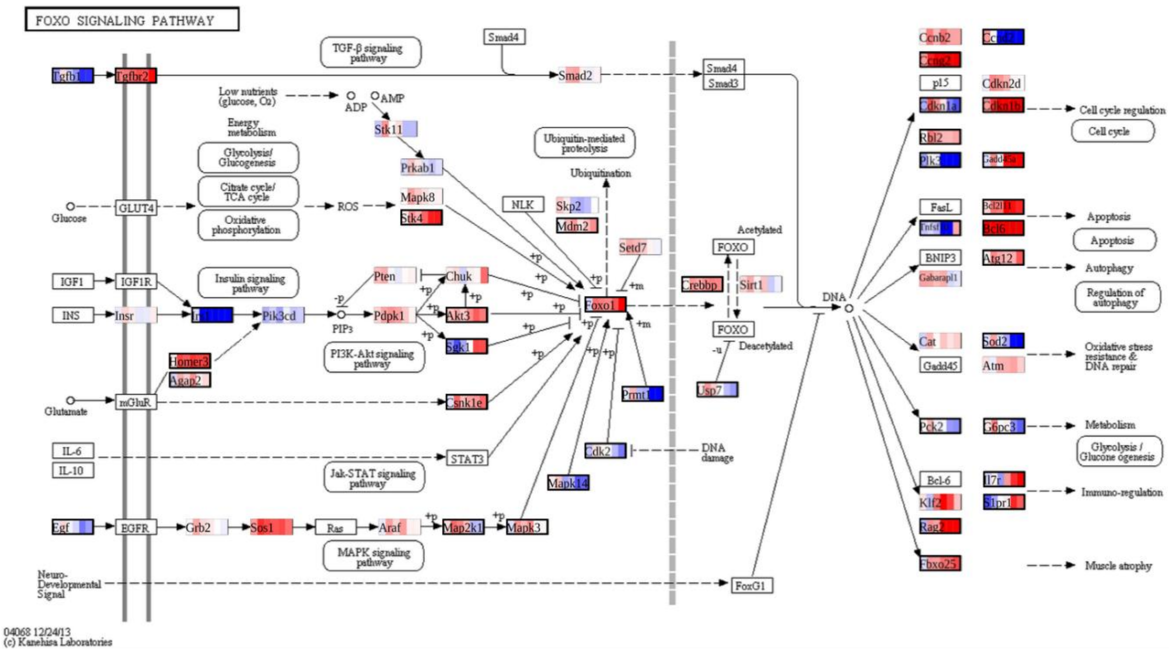
```
png("flii.png", 2500, 1200)
par(cex=2.4, cex.main=2)
matplot(t(tmm_flii), type=c("b"), pch=17, lty = 1, lwd = 7, col =
        mycolors[c(2,5:6)], main = "Flii", xlab = "Condiciones", ylab = "Nivel
        de expresión (TMM)", xaxt = "n", las=2, ylim=c(0,200))
par(new = T)
matplot(tmm_flii_gen, type=c("b"), pch=16, lty = 1, lwd = 7, col = 2, ylab =
        "", xaxt = "n", yaxt="n", ylim=c(0,200))
axis(1, at = 1:12, labels = sample_list_mean)
abline(v=6.5, lty=2, lwd=2.5)
legend("topleft", legend = c("Gen", "Flii-001", "Flii-004", "Flii-003"),
        cex=1.5, lty=2, col=c(2, mycolors[c(2,5:6)]), pch=c(16, 17, 17, 17),
        lwd=5)
dev.off()

# Mxi1

isomx = intersect(IsoNames[grep("Mxi1", IsosGeneNames, fixed=TRUE)],
rownames(tmm_express_filtered))

tmm_mx = tmm_express_filtered_mean[isomx,]
tmm_mx_gen = tmm_express_gen_mean["Mxi1",]

png("Mx1i.png", 2500, 1200)
par(cex=2.4, cex.main=2)
matplot(t(tmm_mx), type=c("b"), pch=17, lty = 1, lwd=7, col =
        mycolors[c(2,5:6)], main = "Mxi1", xlab = "Condiciones", ylab = "Nivel
        de expresión (TMM)", xaxt = "n", las=2, ylim=c(0,50))
par(new = T)
matplot(tmm_mx_gen, type=c("b"), pch=16, lty = 1, lwd=7, col = 2, ylab = "",
        xaxt = "n", yaxt="n", ylim=c(0,50))
axis(1, at = 1:12, labels = sample_list_mean)
abline(v=6.5, lty=2, lwd=2.5)
legend("topleft", legend = c("Gen", "Mxi1-201", "Mxi1-202", "Mxi1-203"),
        cex=1.5, lty=1, col=c(2, mycolors[c(2,5:6)]), pch=c(16, 17, 17, 17),
        ncol=1, lwd=5)
dev.off()

# Dnase1l1
isodn = intersect(IsoNames[grep("Dnase1l1", IsosGeneNames, fixed=TRUE)],
rownames(tmm_express_filtered))

tmm_dn = tmm_express_filtered_mean[isodn,]
tmm_dn_gen = tmm_express_gen_mean["Dnase1l1",]

png("Dnase1l1.png", 2500, 1200)
par(cex=2.4, cex.main=2)
matplot(t(tmm_dn), type=c("b"), pch=17, lty = 1, lwd=7, col =
        mycolors[c(2,5:14)], main = "Dnase1l1", xlab = "Condiciones", ylab =
        "Nivel de expresión (TMM)", xaxt = "n", las=2, ylim=c(0,35))
par(new = T)
matplot(tmm_dn_gen, type=c("b"), pch=16, lty = 1, lwd=7, col = 2, ylab = "",
        xaxt = "n", yaxt="n", ylim=c(0,35))
axis(1, at = 1:12, labels = sample_list_mean)
abline(v=6.5, lty=2, lwd=2.5)
legend("topleft", legend = c("Gen", "Dnase1l1-001", "Dnase1l1-002", "Dnase1l1-
       008", "Dnase1l1-009", "Dnase1l1-007"), cex=1.5,
       lty=1, col=c(2, mycolors[c(2,5:14)]), pch=c(16, 17, 17, 17, 17, 17),
       ncol=2, lwd=5)
```

```
dev.off()

# Se prepara la tabla para introducir en Paintomics

        # Se calcula el ratio para cada muestra

express_effcounts_ratio = (express_filtered[,19:36] + 0.0001) /
                          (express_filtered[,1:18] + 0.0001)
express_gen_ratio = (express_gen_filtered[,19:36] + 0.0001) /
                    (express_gen_filtered[,1:18] + 0.0001)


sample_list_ratio_mean = paste(rep("Ik/Ctr_", each=6),
                               rep(c("0H","2H","6H","12H","18H","24H"),
                               times=rep(1,6)), sep = "")

express_effcounts_ratio2 = express_effcounts_ratio
colnames(express_effcounts_ratio2) = paste(rep("Ik/Ctr_", 18),
                           rep(c("0H_","2H_","6H_","12H_","18H_","24H_"),
                           times=rep(3,6)), c("Batch_1","Batch_3","Batch_4"),
                           sep = "")
express_gen_ratio2 = express_gen_ratio
colnames(express_gen_ratio2) = paste(rep("Ik/Ctr_", 18),
                                     rep(c("0H_","2H_","6H_",
                                     "12H_","18H_","24H_"), times=rep(3,6)),
                                     c("Batch_1","Batch_3","Batch_4"),
                                     sep = "")

express_effcounts_ratio_mean = matrix(0, nrow = length(
                                      rownames(express_effcounts_ratio))
                                      ncol = length(sample_list_ratio_mean),
                                      dimnames=list(
                                      rownames(express_effcounts_ratio),
                                      sample_list_ratio_mean))

express_gen_ratio_mean = matrix(0, nrow = length(rownames(express_gen_ratio)),
                                   ncol = length(sample_list_ratio_mean),
                                   dimnames=list(rownames(express_gen_ratio),
                                   sample_list_ratio_mean))

for (k in sample_list_ratio_mean){
  i = grep(k, colnames(express_effcounts_ratio2))
  express_effcounts_ratio_mean[,k] = rowMeans(express_effcounts_ratio2[,i])
  express_gen_ratio_mean[,k] = rowMeans(express_gen_ratio2[,i])
}

express_effcounts_ratio_mean = log10(express_effcounts_ratio_mean)
express_gen_ratio_mean = log10(express_gen_ratio_mean)


paintomics = express_gen_ratio_mean
rownames(paintomics) = gsub("mm10_ensGene_", "", rownames(paintomics))


write.table(paintomics, "GenesP.txt", quote=FALSE, sep="\t")
write.table(names(ngenes_DE), "GenesDE.txt", quote=FALSE, sep="\n",
            col.names=FALSE, row.names=FALSE)
```

7.8. Anexo VIII: Rutas visualizadas con Paintomics

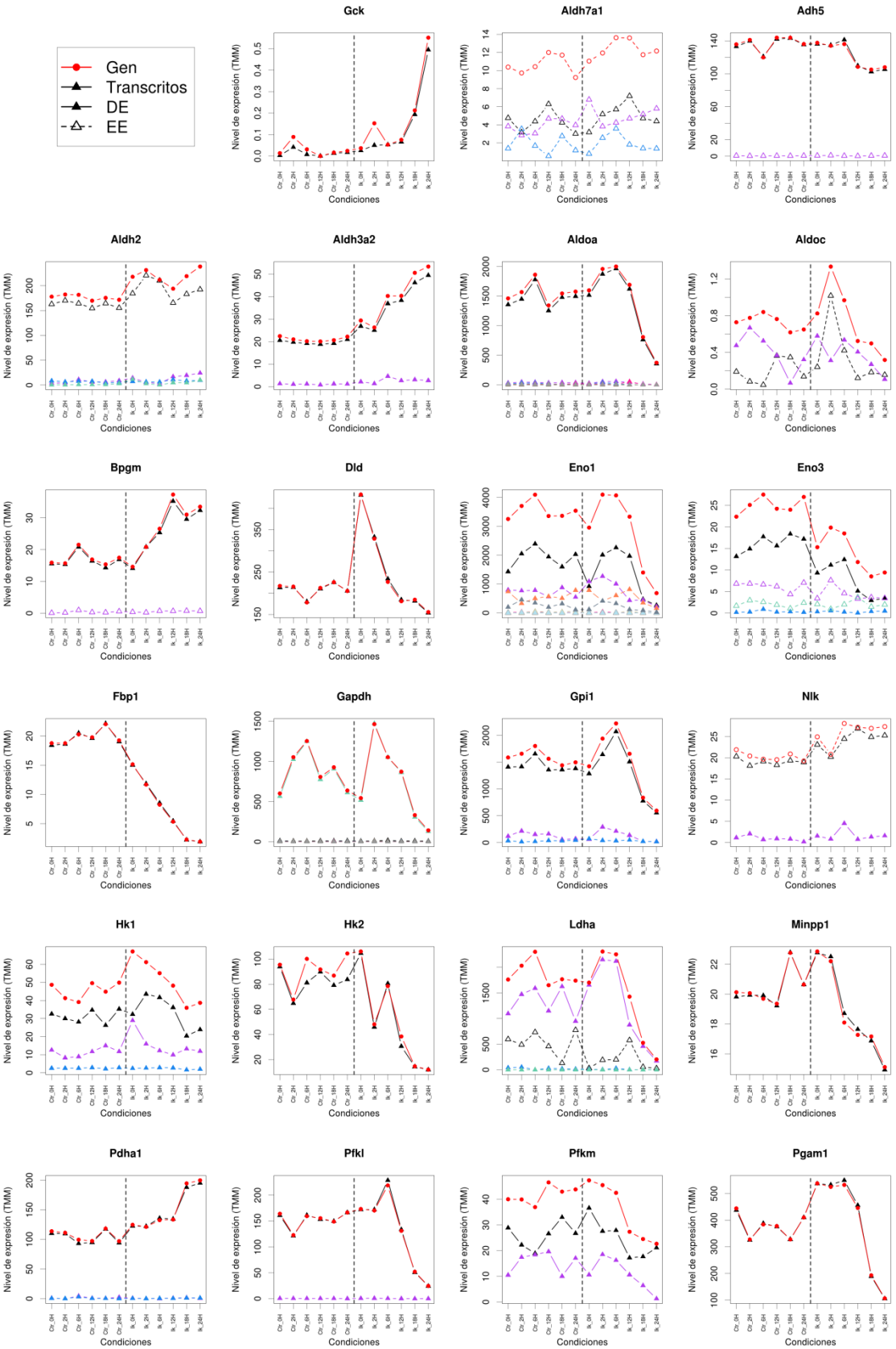


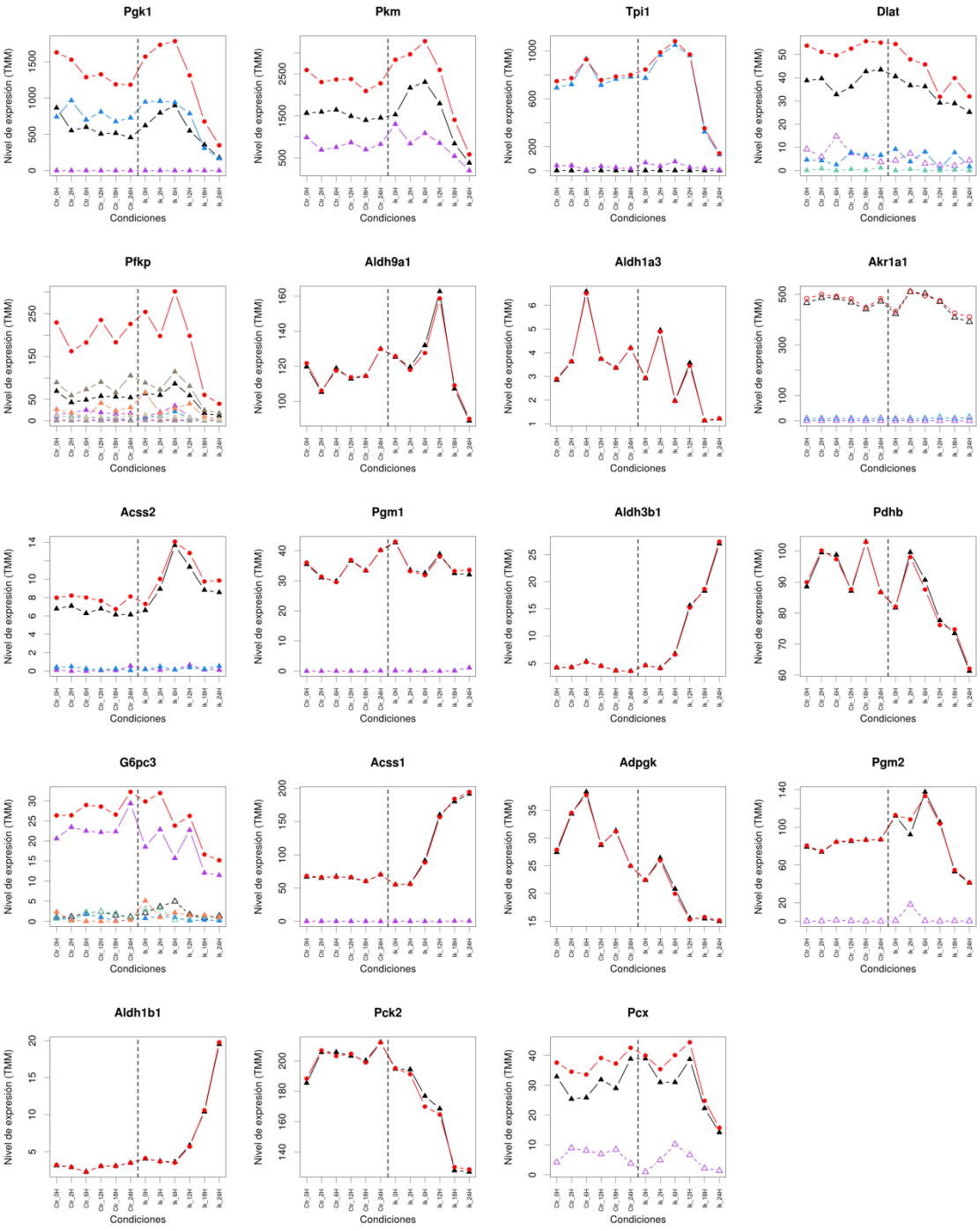

REGULATION OF AUTOPHAGY
MAPK signaling pathway
Autophagy induction
INS
Insulin signaling pathway
mTOR signaling pathway
ATG17
Ulk1
ATG13
ATG11
VAC8
Intracellular pathogens
poliovirus
3A,2C
IFNα
IFNγ
Prkaa1
Nutrient/ growth factor deprivation, hypoxia.
Vesicle nucleation
Becn1
ATG14
Pik3r4
Pik3c3
Wortmannin
3-Methyladenine
PI
PI3P
ATP
AMP
Atg12
ATG7
Atg10
Atg5
Atg16l1
ATG12-ATG5.ATG16 multimeric complex
E1
E2
Vesicle expansion and completion
Gabarapl1
Atg4d
Atg7
Atg3
PE
Docking and fusion
SNARE interactions in vesicular transport
Antigen processing and presentation
Apoptosis
Adaptation to starvation, elimination of aberrant structures, prevent aging, defense against pathogens, defense against tumors.
04140 1/22/14
(c) Kanehisa Laboratories


FOXO SIGNALING PATHWAY
TGF-β signaling pathway
Smad4
Smad2
Smad3
Tgfb1
Tgfbr2
Low nutrients (glucose, O2)
ADP
AMP
Energy metabolism
Stk11
Glycolysis/ Gluconeogenesis
Prkab1
Citrate cycle/ TCA cycle
Oxidative phosphorylation
ROS
Mapk8
Stk4
Glucose
GLUT4
Ubiquitin-mediated proteolysis
Ubiquitination
NLK
Skp2
Mdm2
Setd7
IGF1
IGF1R
Insulin signaling pathway
Pten
Chuk
INS
Insr
Pik3cd
PIP3
Pdpk1
Akt3
Foxo1
Crebbp
Sirt1
FOXO
Acetylated
Deacetylated
Homer3
Agap2
Sgk1
PI3K-Akt signaling pathway
Glutamate
mGluR
Csnk1e
Prmt1
Usp7
IL-6
IL-10
STAT3
Jak-STAT signaling pathway
Cdk2
DNA damage
Mapk14
Egf
EGFR
Grb2
Sos1
Ras
Araf
Map2k1
Mapk3
MAPK signaling pathway
Neuro-Developmental Signal
FoxG1
DNA
Ccnb2
Ccng2
p15
Cdkn2d
Cdkn1a
Cdkn1b
Cell cycle regulation
Cell cycle
Rbl2
Pik3
Gadd45
FasL
Bcl2l11
Apoptosis
Bcl6
BNIP3
Atg12
Autophagy
Gabarapl1
Regulation of autophagy
Cat
Sod2
Oxidative stress resistance & DNA repair
Gadd45
Atm
Pck2
G6pc3
Metabolism
Glycolysis / Gluconeogenesis
Bcl-6
Il7r
Immuno-regulation
Klf2
S1pr1
Rag1
Fbxo25
Muscle atrophy
04068 12/24/13
(c) Kanehisa Laboratories

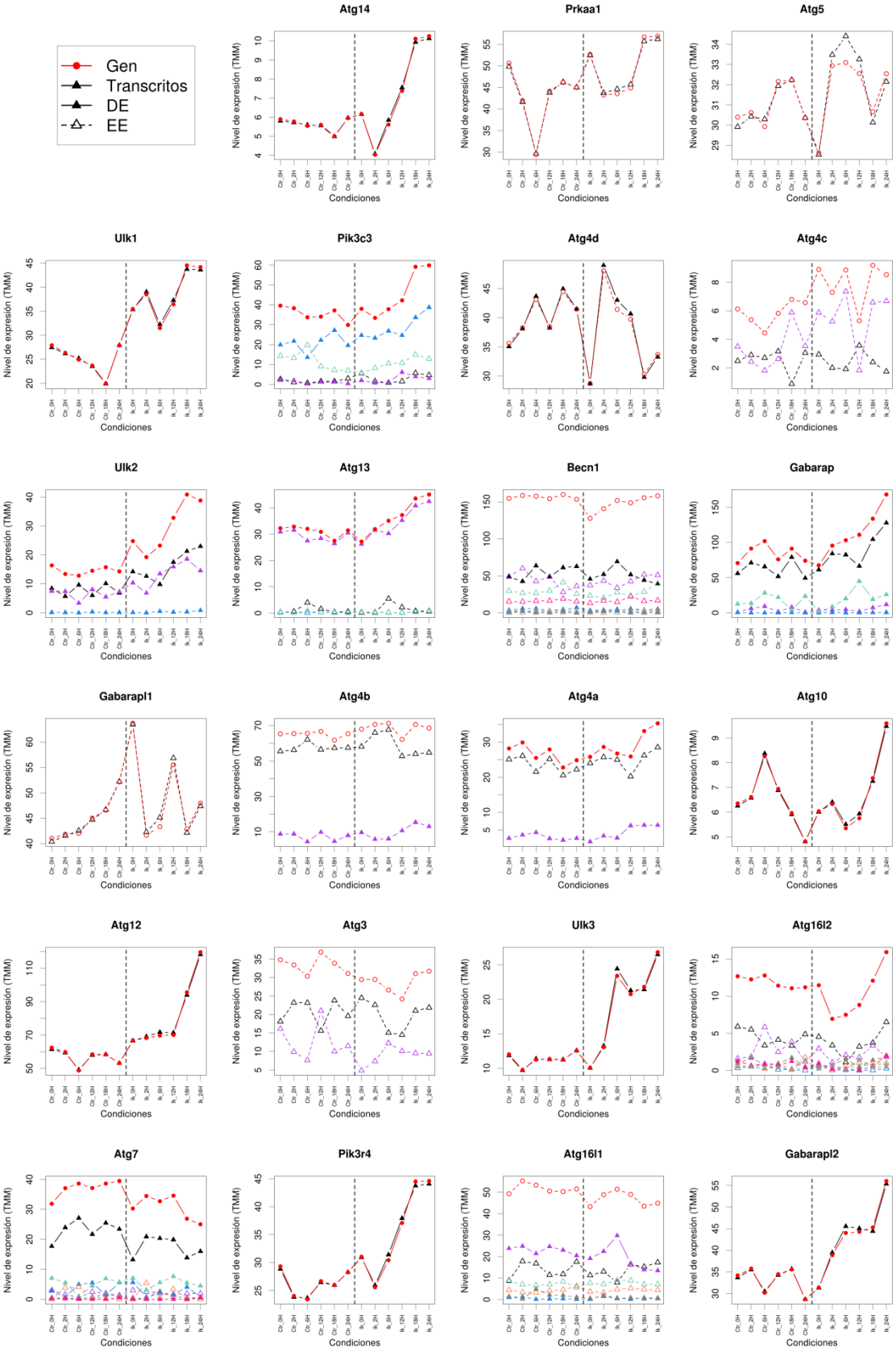
7.9. Anexo IX: Expresión de genes y sus isoformas de distintas rutas

Glicólisis/Gluconeogénesis

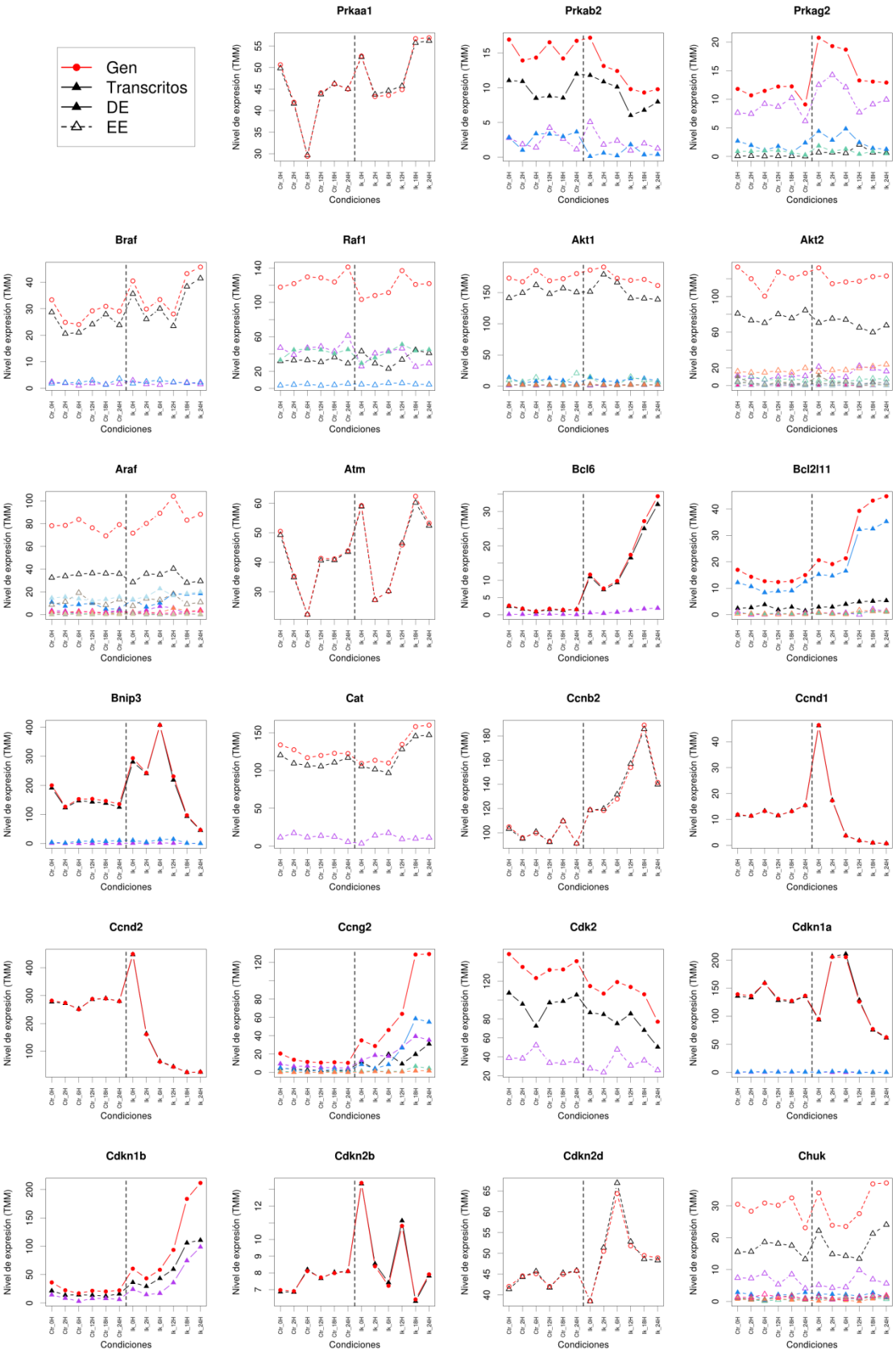


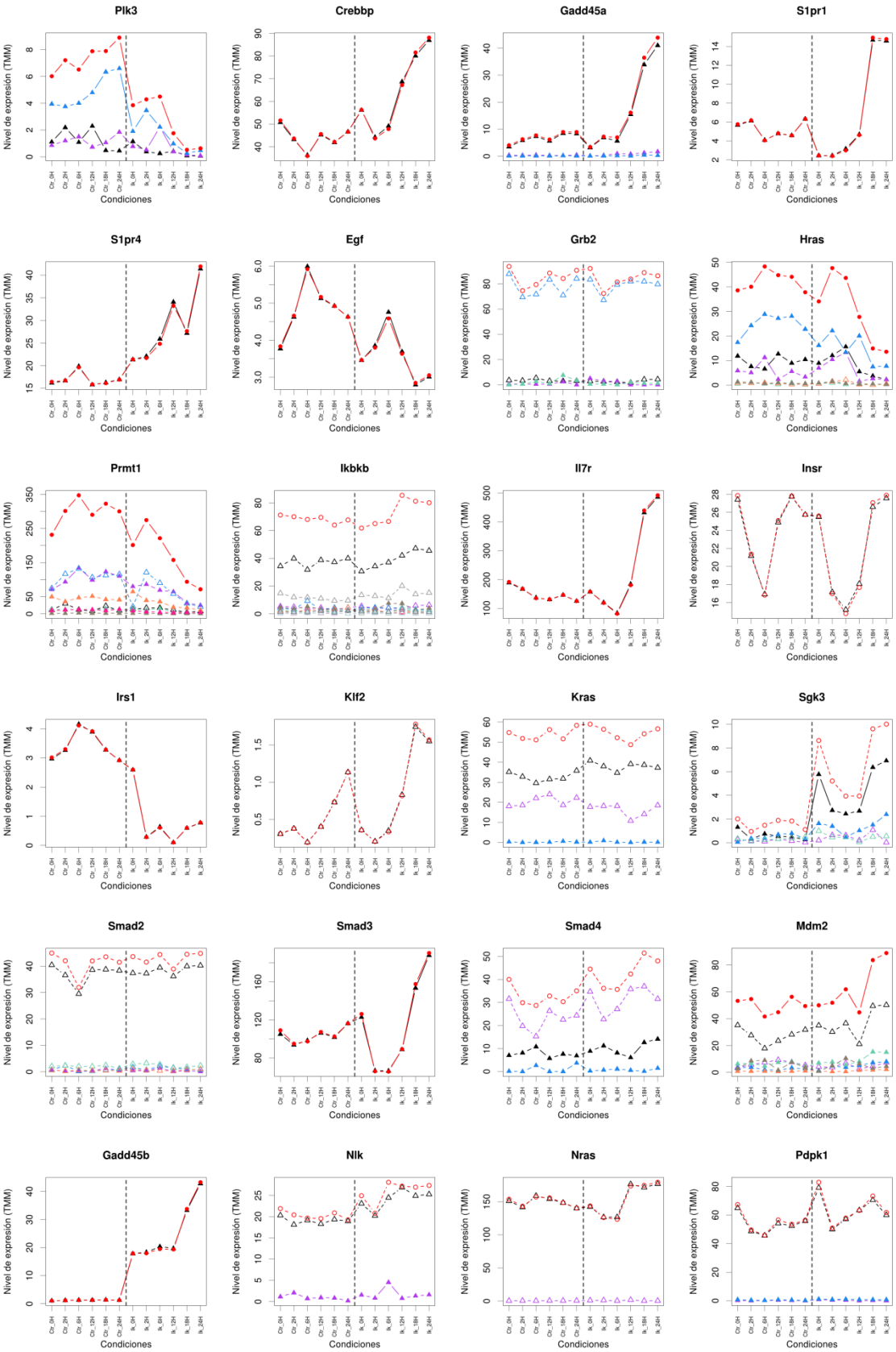

Pgk1
Pkm
Tpi1
Dlat
Pfkp
Aldh9a1
Aldh1a3
Akr1a1
Acss2
Pgm1
Aldh3b1
Pdhb
G6pc3
Acss1
Adpgk
Pgm2
Aldh1b1
Pck2
Pcx
Nivel de expresión (TMM)
Condiciones

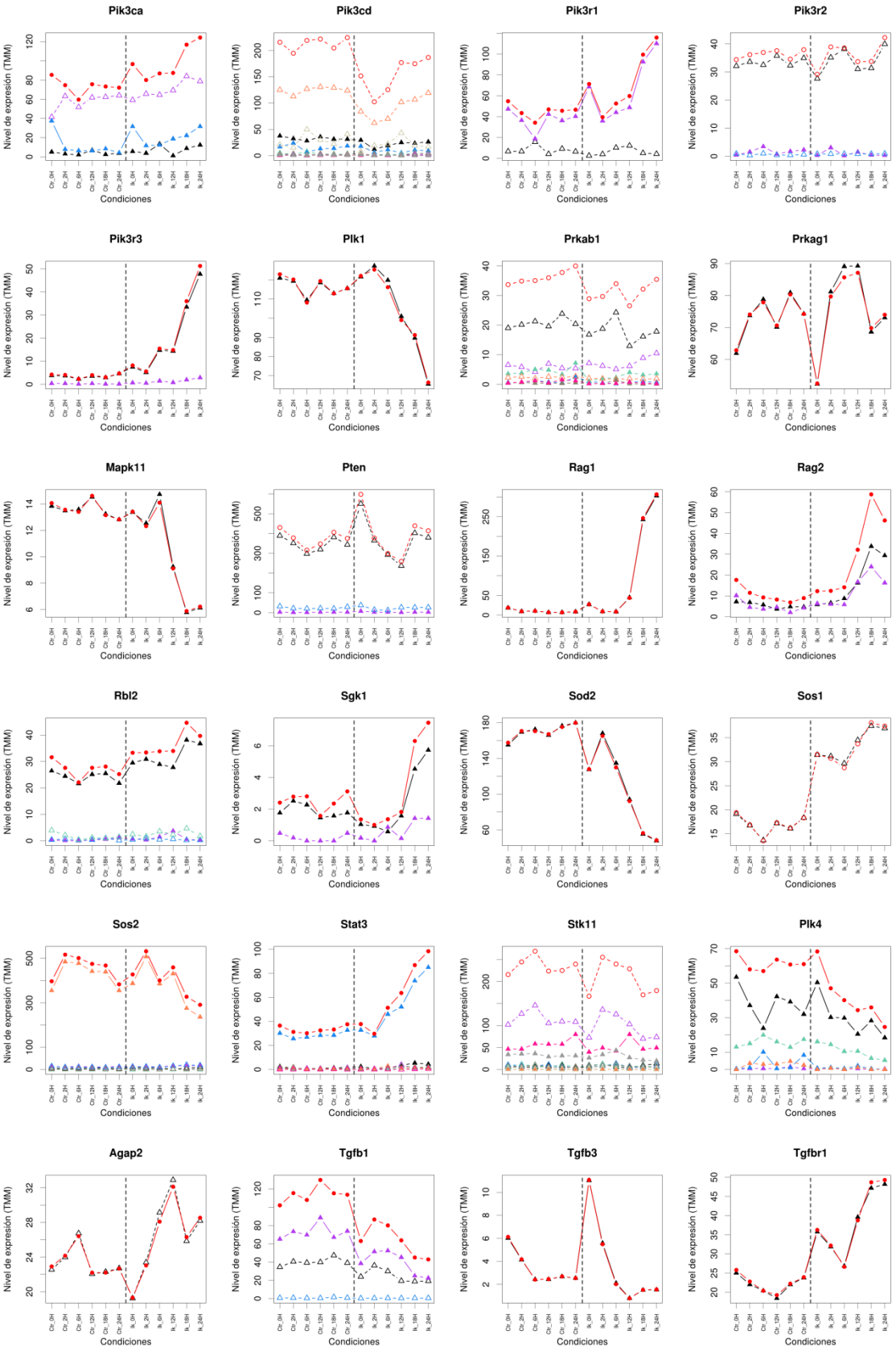
Regulación de la autofagia:

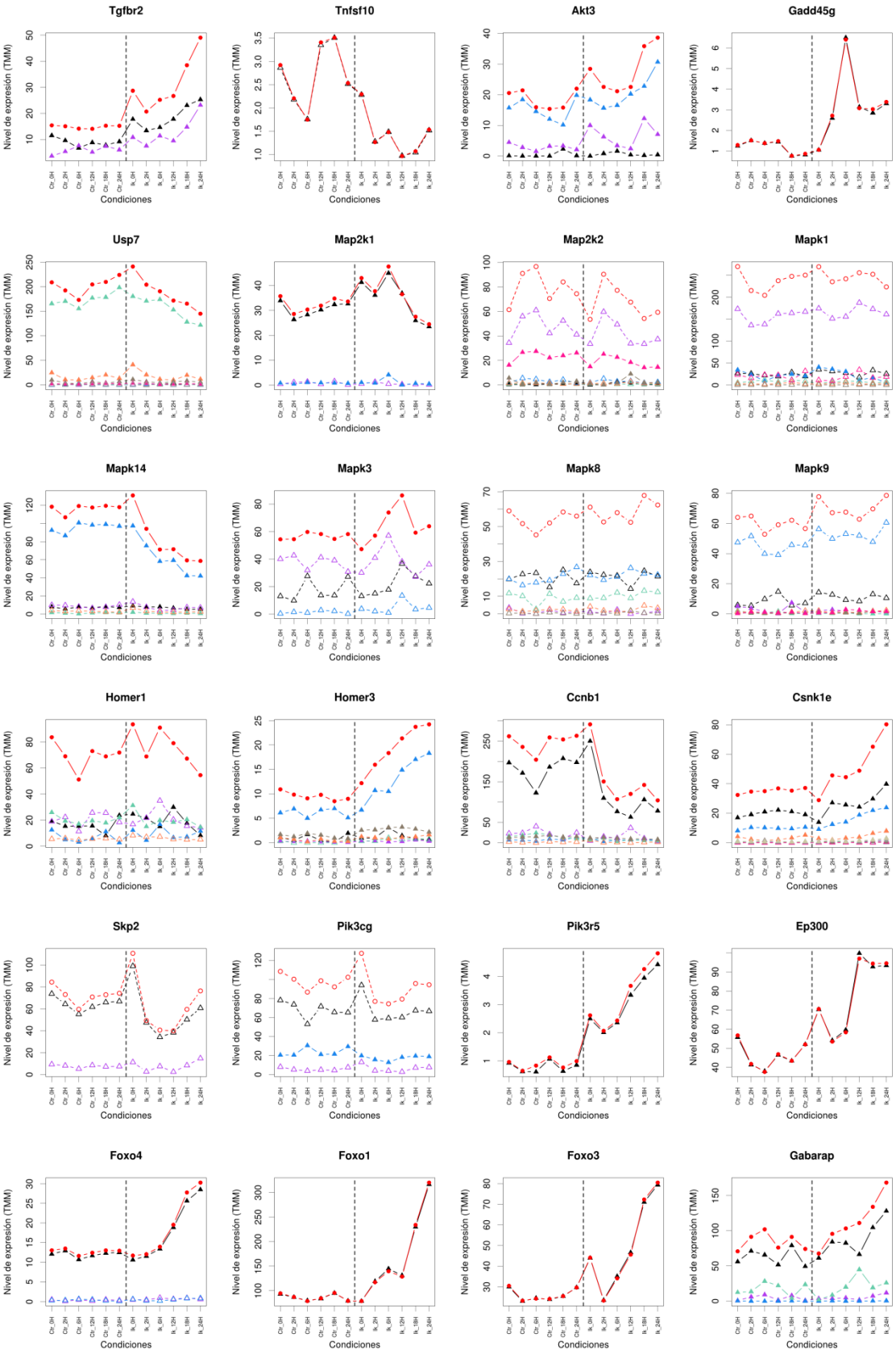


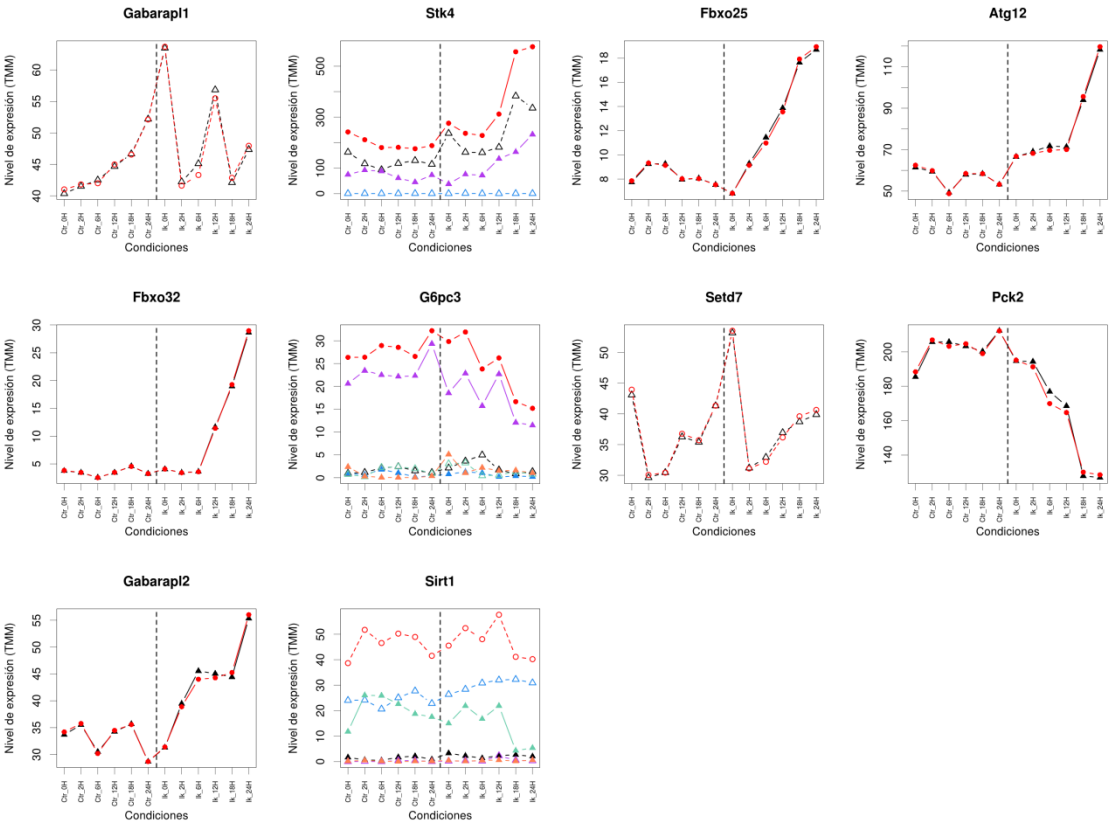
Señalización de FOXO




Plk3
Crebbp
Gadd45a
S1pr1
S1pr4
Egf
Grb2
Hras
Prmt1
Ikbkb
Il7r
Insr
Irs1
Klf2
Kras
Sgk3
Smad2
Smad3
Smad4
Mdm2
Gadd45b
Nlk
Nras
Pdpk1
Nivel de expresión (TMM)
Condiciones


Pik3ca
Pik3cd
Pik3r1
Pik3r2
Pik3r3
Plk1
Prkab1
Prkag1
Mapk11
Pten
Rag1
Rag2
Rbl2
Sgk1
Sod2
Sos1
Sos2
Stat3
Stk11
Plk4
Agap2
Tgfb1
Tgfb3
Tgfbr1
Nivel de expresión (TMM)
Condiciones


Tgfbr2
Tnfsf10
Akt3
Gadd45g
Usp7
Map2k1
Map2k2
Mapk1
Mapk14
Mapk3
Mapk8
Mapk9
Homer1
Homer3
Ccnb1
Csnk1e
Skp2
Pik3cg
Pik3r5
Ep300
Foxo4
Foxo1
Foxo3
Gabarap
Nivel de expresión (TMM)
Condiciones


Gabarapl1
Stk4
Fbxo25
Atg12
Fbxo32
G6pc3
Setd7
Pck2
Gabarapl2
Sirt1
Nivel de expresión (TMM)
Condiciones