Universidad Europea

Máster en bioinformática

DESARROLLO DE UN MÓDULO DE ANÁLISIS TRANSCRIPTÓMICO, MECANÍSTICO E INTEGRATIVO

Autora: Elena Rodríguez De Tiedra

Tutora: María Peña Chilet

Curso 2022-23

AGRADECIMIENTOS

A mi familia, amigos y pareja por haberme apoyado y haber estado a mi lado durante el proceso.

A mi tutora, por mostrarme una aplicación tan interesante de todo lo aprendido en el máster y por ayudarme durante la realización de este proyecto.

Índice

1. Resumen

Los mapas mecanísticos de enfermedad juegan un papel cada vez más relevante en el diagnóstico y análisis de enfermedades, lo que permite abordar el estudio de enfermedades desde el punto de vista de la biología de sistemas. Cambios en el mecanismo celular y en las rutas de señalización son potencialmente responsables de un fenotipo, al alterar la función celular. Es razonable pensar que cambios en la expresión de genes se traducen en cambios funcionales, mediados por cambios en la ruta de señalización en la que están implicados esos genes, pero muchas veces mediante aproximaciones clásicas no podemos evaluar el escenario celular en su conjunto.

Existen métodos de análisis de estas redes mecanísticas, como son los métodos *MPA* (*Mechanistic Pathway Analysis*), como la herramienta Hipathia, que permiten modelizar situaciones en la célula, evaluando el impacto funcional sobre el mecanismo de un proceso, condición o enfermedad.

Los *miRNAs* son pequeñas moléculas de *RNA* no codificante, cuya función es la regulación epigenética de genes, mediante su silenciamiento al unirse a la región 3'UTR del *mRNA*. Este silenciamiento, sin embargo, no siempre se ve reflejado en cambios en la cuantificación del transcriptoma, por lo que al modelizar los mecanismos celulares empleando únicamente la expresión de los genes, nos estamos perdiendo información relevante que puede estar afectando al proceso global.

En este trabajo se ha desarrollado un pipeline de análisis, mediante scripts en R automatizados, que permite integrar la información de expresión de genes y el impacto de los *miRNAs* en una condición determinada, posibilitando la evaluación del efecto global sobre un mapa mecanístico.

Palabras clave: mecanístico, transcriptómico, miRNA, rutas, activación, pipeline

2. Introducción

En los últimos 20 años, gracias al auge de las ómicas, los esfuerzos por conocer la base genética tras procesos biológicos se han incrementado considerablemente, traduciéndose en mejoras en secuenciación y análisis bioinformático. Gracias a ello, se ha podido comprender mejor el funcionamiento de las enfermedades, entre otros avances. Sin embargo, el reto reside en que no todas son debidas a la mutación de un único gen, ni todas muestran una alta penetrancia (Peña-Chilet *et al.*, 2019). Enfermedades complejas como el cáncer o la diabetes, quedan lejos de poder comprenderse con detalle, lo que aleja la posibilidad de disponer de diagnósticos y tratamientos que se ajusten con mayor eficacia a las necesidades de los pacientes que los que existen en la actualidad (Loh *et al.*, 2019).

Aunque nuestro conocimiento de enfermedades como el cáncer de mama, segundo cáncer más diagnosticado y quinto con el mayor número de muertes relacionadas con el cáncer en mujeres (según GLOBOCAN 2018. Loh *et al.*, 2019), ha permitido aumentar la supervivencia global, todavía se requiere de esfuerzos para poder mejorar su diagnóstico y ofrecer un tratamiento personalizado, cuyo impacto y garantía de éxito son mayores en los estadíos más tempranos de la enfermedad. Se sabe que el cáncer de mama es causado por el crecimiento anómalo de células colindantes de conductos y glándulas mamarias.

Un ejemplo de la complejidad de esta patología es que, atendiendo a su perfil molecular, puede ser clasificado en 6 tipos intrínsecos: luminal A, luminal B, HER+, normal-like, basal (o triple negativo) y claudin-low, cada uno con su fenotipo único, grado de tumor y caracterización molecular (Loh *et al.*, 2019). Estos 6 tipos pueden clasificarse según sus patrones de expresión de genes, si las células luminales expresan ER (ER+) o no (ER-). En el primer grupo (ER+) podrían catalogarse luminal A, luminal B y normal-like, y en el segundo (ER-) basal, HER+ y claudin-low (Perou *et. al.*, 2000). Los subtipos luminales o ER+ presentan una mejor prognosis, con una respuesta intermedia a quimioterapia, frente a los ER-, con un mayor carácter invasivo y peor prognosis. (Prat y Perou, 2010).

2.1. Análisis mecanístico

Esta complejidad requiere de herramientas de análisis que puedan entender los mecanismos de la enfermedad de una manera más dinámica. Por ejemplo, es bien conocido que el fenotipo de proliferación, común en las células de cáncer, requiere de un apoyo intensivo para la biosíntesis de componentes celulares y generación de energía, lo cual se consigue, en general, mediante metabolismo reprogramado (Cubuk *et al.*, 2018).

Existen modelos que emplean conocimiento biológico en señalización celular para codificar valores de expresión de genes de manera individual en medidas que responden a funcionalidades celulares causadas por la actividad de la ruta (Hidalgo *et al.*, 2016).

Teniendo en cuenta esto, no todos los modelos que analizan rutas son capaces de capturar ese funcionamiento alterado en enfermedades como el cáncer de mama. Los análisis basados en topología ignoran que muchas rutas son multifuncionales y que pueden desencadenar funciones opuestas dependiendo de la progresión dinámica de la propagación de la señal entre el receptor y el efector (Amadoz *et al.*, 2019). Además, aunque se pueda detectar el nivel de activación de genes, el que ello desencadene funcionalidades o no se debe determinar a posteriori, mediante métodos heurísticos (Hidalgo *et al.*, 2016). Todas estas consideraciones son tomadas en cuenta por los MPA (*mechanistic pathway analysis*).

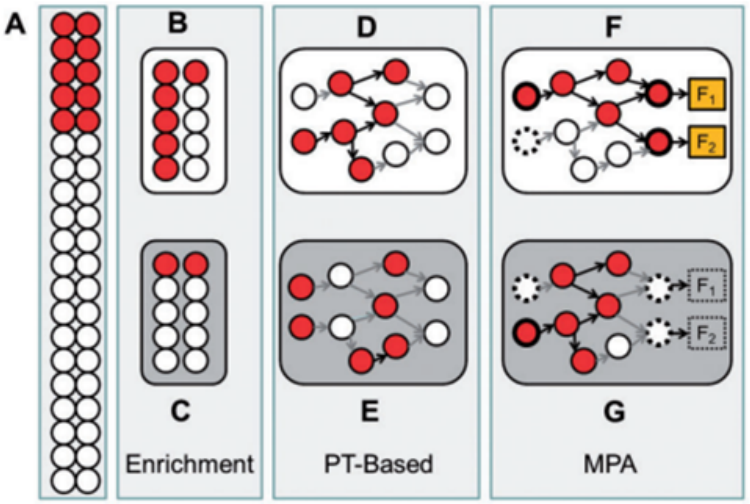


Figura 1. Familias de métodos de análisis funcionales: Enrichment (enriquecimiento): ORA (over-representation analysis), GSEA (gene set enrichment analysis)), PT-based (pathway-topology based, basados en topología): SPIA, CePa, Pathifier entre otros; y MPA: TAPPA, DEAP o Hipathia entre otros. Tomada de Amadoz *et al.*, 2019.

Los modelos basados en análisis mecanísticos (MPA) asumen que un sistema complejo puede comprenderse examinando el funcionamiento de sus partes y la manera en que se juntan. Un ejemplo de ello es el modelo Hipathia (acrónimo de *High-throughput pathway interpretation and analysis*). En Hipathia, los circuitos que conectan las proteínas receptoras, responsables de desencadenar actividades celulares, están basados en rutas KEGG. El método asume que, para transducir la señal, todas las proteínas que conectan el receptor con el efector deben estar presentes y que, cuanto más alta sea la cantidad de estas proteínas, mayor será la señal. Se toman medidas de de mRNA como proxis de la cantidad correspondiente de proteína (Peña-Chilet *et al.*, 2019).

2.2. Papel de los miRNA

Los micro RNA (miRNA) son RNA pequeños, de 19 a 24 nucleótidos según algunos papers (Borgmästars *et al.*, 2019) 21 a 23 nucleótidos según otros (Huberdeau y Simard, 2018). Son una familia evolutivamente conservada, endógena y de moléculas monocatenarias. El genoma humano contiene unos 2654 secuencias de miRNA maduros hasta la fecha. Los miRNA funcionan como elementos clave en la regulación post-transcripcional de expresión de genes en diferentes tejidos y su desarrollo mediante interacciones altamente específicas y redes regulatorias complejas (Loh *et al.*, 2019).

Los miRNA son generados en el núcleo celular por la RNA polimerasa II, formándose un bucle pre-miRNA. Este pre-miRNA continúa con su maduración en el citoplasma, donde acaba siendo dividido en dos: un miRNA maduro que realizará su función reguladora y una “hebra pasajera”, que acabará siendo degradada. El proceso un poco más detallado, puede verse ilustrado en la siguiente imagen.

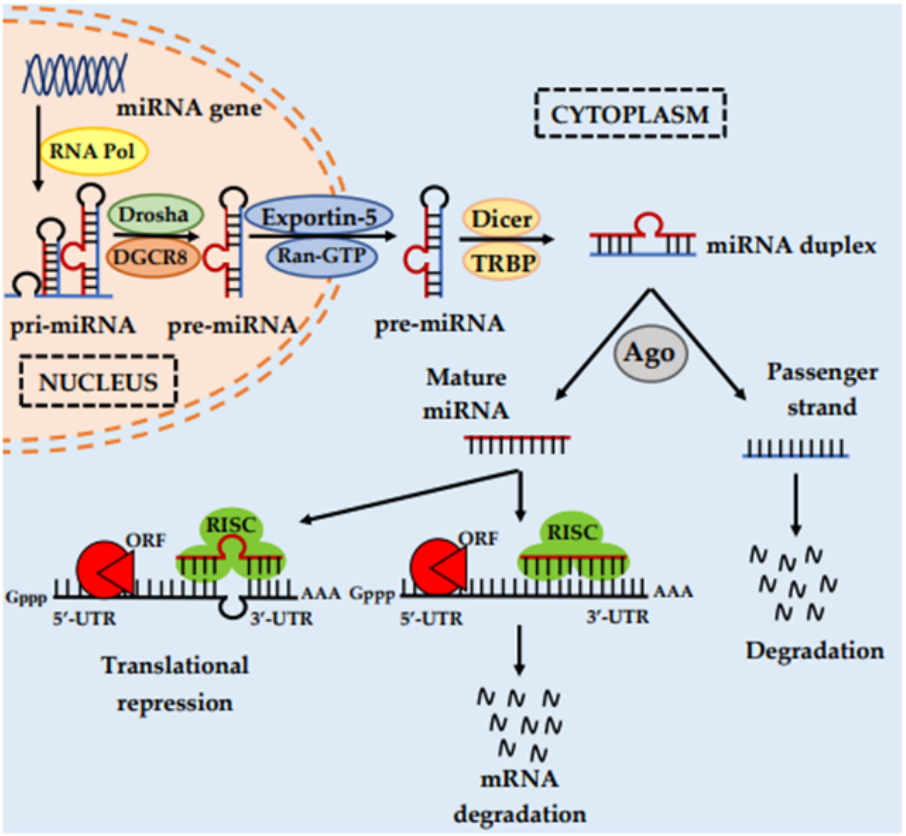


Figura 2. Creación de miRNA y modulación de su actividad. Tomada de Loh *et al.*, 2019

Este miRNA maduro regula al mRNA, pero dependiendo de su complementariedad, esta acción puede desembocar en represión traslacional o en degradación del mRNA. Es aquí donde radica uno de los desafíos a la hora de estudiar con los miRNA. Una complementariedad del 100% conlleva a que el mRNA se degrade, mientras que una complementariedad parcial provoca que el mRNA no se degrade (por lo que puede ser secuenciado) pero no realiza su función.

Numerosos estudios muestran la expresión de miRNA alterada en cáncer de mama. Estos miRNA alterados pueden subdividirse entre miRNA oncogénicos (oncomiRs) y miRNA supresores de tumores (tsmiRs). Los oncomiRs se encuentran habitualmente regulados de manera positiva en el cáncer de mama, suprimiendo la expresión de genes supresores de tumores y favoreciendo la activación de factores de transcripción oncogénicos abocando por tanto, a la promoción de tumorgénesis en la mama (Loh *et. al*, 2019) (van Schooneveld *et al.*, 2015). A su vez, los tsmiRs pueden inhibir la expresión de oncogenes que promueven la

tumorgénesis. Es por ello que habitualmente se encuentran desregulados en células cancerosas (van Schooneveld *et al.*, 2015).

Tanto oncomiRs como tsmiRs regulan de manera crítica el desarrollo del tumor de cáncer de mama y su progresión participando en complejas redes reguladoras. Estas redes incluyen varias capacidades distintivas del cáncer (también conocidas como *hallmarks* del cáncer) tales como mantenimiento del crecimiento y señales proliferativas, inmortalidad replicativa, iniciación de metástasis e invasión, resistencia a apoptosis y respuesta a muerte celular, inducción a angiogénesis, activación de metabolismo y apoyo al escape del sistema inmune celular.

2.3. Enfoque integrativo

Muchos de los métodos para realizar modelos mecanísticos de las variaciones en la actividad de las rutas celulares parten de secuenciaciones de mRNA y miRNA, no obstante, las herramientas empleadas y los pipeline diseñados no pueden ser únicos ni pueden ser usados en todos los casos. Estos pipelines suelen integrar datos de diferentes repositorios, como GDC, miRTarBase o GTEX, emplear algoritmos que se basen en repositorios de rutas (Hipathia emplea el repositorio de rutas KEGG).

Un esquema muy general de un pipeline para el análisis computacional de secuenciación de RNA incluye etapas de pre análisis (diseño experimental, diseño de secuenciación y control de calidad), perfil del transcriptoma, análisis (expresión diferencial e interpretación) y etapas de estudio avanzado (visualización, secuenciación de otros tipos de RNA e integración de resultados) (Conesa *et al.*, 2016). Estos pipelines o sus etapas se pueden realizar gracias servicios y aplicaciones (DEAapp, RNAseqViewer, Integrative Genomics Viewer, miEAA) o mediante trabajo computacional con lenguajes y entornos de programación, como R y RStudio.

Un ejemplo más puede ser Hipathia, que puede ser empleado en página web (http://hipathia.babelomics.org/) o mediante el uso de la librería Hipahtia

(https://www.bioconductor.org/packages/release/bioc/html/hipathia.html), en R, un sistema de computación estadística y generador de gráficos. Consiste en un lenguaje y un entorno con gráficos, un depurador, acceso a ciertas funciones de sistema y la habilidad para correr programas almacenados en archivos tipo *script* (https://cran.r-project.org/). El entorno RStudio permite una visualización y edición del código, script, herramientas empleadas, archivos generados y gráficos creados con sencillez y eficacia.

Emplear un único script en R facilita integrar todos las fuentes de datos y herramientas (librerías) empleadas en un único lugar. Aunque requiere una inversión de tiempo y esfuerzo, compensa a su vez en el manejo detallado y minucioso de todo el proceso. Esto a su vez, favorece el control y la cantidad de información que se puede extraer.

3. Hipótesis

La activación diferencial de rutas de señalización tiene un papel relevante en la progresión y pronóstico del cáncer de mama, reflejado en sus subtipos intrínsecos. Esta activación diferencial puede estimarse a partir de los datos de expresión de genes. Además, existe un perfil de expresión de miRNAs diferencial en los subtipos de cáncer de mamá que tiene un impacto sobre la activación de rutas de señalización.

4. Objetivos

Desarrollar los scripts de R necesarios para optimizar un pipeline de análisis de datos procedentes de tecnologías transcriptómicas (principalmente expresión de genes y elementos no codificantes, como miRNAs) enfocándose en el análisis mecanístico con Hipathia y visualización intuitiva de datos. Para ello empleamos la entidad cáncer de mama luminal (de mejor pronóstico) frente al resto de subtipos.

- Seleccionar datos para evaluar la utilidad y validez del pipeline desarrollado como caso de uso:
  - Descarga: De repositorios como GDC (datos clínicos, expresión de genes y expresión de miRNA, todo de TCGA) miRTarBase y GTEx
  - Preparación: Limpieza y normalización por TMM
- Evaluar la desregulación funcional de la entidad seleccionada en el caso de uso empleando datos de expresión de genes.
- Obtener el perfil de miRNAs desregulados en la entidad seleccionada en el caso de uso.
- Estimar el impacto mecanístico sobre la activación de rutas de una selección de miRNAs desregulados.
- Comparar los resultados obtenidos con un método de análisis funcional tradicional como el enriquecimiento funcional.
- Integrar y automatizar los pasos en el entorno de análisis de datos R.

5. Materiales y metodología

5.1. Materiales

Para cumplir con los objetivos planteados se emplean datos de fuentes diversas, tales como Genomic Data Commons (GDC) (https://portal.gdc.cancer.gov/), miRTarBase (https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/search.php) y GTEX (https://gtexportal.org/home/). Estos datos proceden de muestras de mama de pacientes sanos (GTEx) y enfermos (GDC).

Repositorio	Tipo de dato empleado	Descarga y uso en R
GDC	RNA-seq, miRNA-seq, datos clínicos	Descarga desde RStudio
miRTarBase	relación de miRNA-gen diana	Descarga de archivo .xlsx
GTEx	RNA-seq	Descarga de archivo .gct.gz

Tabla 1. Descripción de repositorios, tipo de datos empleados, descarga y uso en R. Realización propia.

5.1.1. Genomic Data Commons (GDC)

GDC es un programa de investigación del Instituto Nacional del Cáncer (NCI). Su misión es la de proveer a la comunidad investigadora del cáncer un repositorio unificado y una base de datos sobre el cáncer que permita compartir datos entre estudios genómicos. Todo con el objetivo de apoyar la medicina de precisión. Dispone de datos de acceso público y datos con acceso restringido.

Los datos descargados, datos clínicos, expresión de genes y miRNA se corresponden con la versión v37.0, del 29 de marzo de 2023, de acceso público. Pertenecen al proyecto *The Cancer Genome Atlas Program* (TCGA) de cáncer de mama *(Breast Cancer gene*, BRCA), uno de los 82 proyectos del repositorio, que cuenta también con 68 sitios primarios, 88.991 casos, 1.003.747 archivos, 22.588 genes y 2.903.037 mutaciones.

Trabajar con expresión de genes y su información clínica nos proporciona la oportunidad de emplear Hipathia para disponer de una imagen global de las rutas de señalización alteradas debido al cáncer. La matriz de expresión de miRNA será empleada para estudiar el impacto de su expresión diferencial sobre la activación de rutas de señalización.

Tipos de datos empleados	Características
Expresión genes	60.660 muestras de RNA-seq
	1.231 pacientes
Información clínica	1.881 muestras de miRNA-seq
	3.622 pacientes
Expresión miRNA	114 columnas de información clínica
	1.020 pacientes

Tabla 2. Descripción de datos empleados y sus características. Realización propia

5.1.2. miRTarBase

Es una base de datos de interacciones entre miRNA y dianas, siendo estas interacciones validadas experimentalmente. Ha llegado a acumular más de 360.000 interacciones miRNA-diana (MTIs), que son recolectadas manualmente, revisando manualmente la literatura pertinente mediante uso de NLP (natural language processing) de forma sistemática para filtrar artículos de investigación relacionados con estudios funcionales de miRNA. De manera general, los MTIs recogidos son validados experimentalmente mediante técnicas como *reporter assay, western blot, microarray* y secuenciación *next-generation*. Aparte de almacenar MTIs validados, miRTarBase compara esta colección continuamente actualizada con otras bases de datos previamente desarrolladas. El acceso y descarga de datos es público.

Contiene alrededor de 470.000 MTIs, de 32 especies. Almacena también alrededor de 4.312 miRNA, con 23.426 genes dianas aproximadamente. La versión activa en el momento de la descarga de datos es la versión miRTarBase 8.0, a quien se pueden atribuir los datos mostrados. La matriz que hemos descargado, perteneciente a la especie humana, contiene 957.000 muestras y presenta información tal y como muestra la siguiente tabla.

miRTarBase ID	miRNA	Species (miRNA)	Target Gene	Target Gene (Entrez ID)	Species (Target Gene)	Experiments	Support Type	References (PMID)
MIRT000002	hsa-mir-20a	Homo sapiens	HIF1A	3091	Homo sapiens	Luciferase reporter assay//Western blot//Northern blot//qRT-PCR	Functional MTI	18632605
MIRT000002	hsa-mir-20a	Homo sapiens	HIF1A	3091	Homo sapiens	HITS-CLIP	Functional MTI (Weak)	22473208
MIRT000002	hsa-mir-20a	Homo sapiens	HIF1A	3091	Homo sapiens	Luciferase reporter assay//qRT-PCR//Western blot	Functional MTI	23911400

MIRT000006	hsa-mir-146a	Homo sapiens	CXCR4	7852	Homo sapiens	qRT-PCR//Luciferase reporter assay//Western blot	Functional MTI	18568019
MIRT000006	hsa-mir-146a	Homo sapiens	CXCR4	7852	Homo sapiens	Microarray	Functional MTI (Weak)	20375304
MIRT000006	hsa-mir-146a	Homo sapiens	CXCR4	7852	Homo sapiens	Luciferase reporter assay//qRT-PCR//Western blot	Functional MTI	26082201
MIRT000012	hsa-mir-122	Homo sapiens	CYP7A1	1581	Homo sapiens	qRT-PCR//Luciferase reporter assay	Functional MTI	20351063
MIRT000018	hsa-mir-222	Homo sapiens	STAT5A	6776	Homo sapiens	qRT-PCR//Luciferase reporter assay//Western blot	Functional MTI	20489169
MIRT000018	hsa-mir-222	Homo sapiens	STAT5A	6776	Homo sapiens	Luciferase reporter assay	Functional MTI	24736554
MIRT000019	hsa-mir-21	Homo sapiens	RASGRP1	10125	Homo sapiens	qRT-PCR//Luciferase reporter assay//Western blot	Functional MTI	20483747

Tabla 3. Extracto de tabla de MTIs humanos procedente de miRTarBase. Creación propia

Empleamos estos datos como medio para conocer qué potenciales dianas pueden tener los miRNA de TGCA-BRCA descritos anteriormente. Nos servirá para crear una tabla que relacione los miRNA diferencialmente expresados con un factor de silenciamiento diseñado y las dianas de estos miRNA.

5.1.3. GTEx

El proyecto GTEx (*Genotype-Tissue Expression*) es un esfuerzo continuo para construir una fuente de información comprensible sobre expresión de genes específica de tejidos, así

como su regulación. Las muestras son recolectadas de 54 tipos de tejidos sanos de un número próximo a los 1.000 individuos, principalmente mediante ensayos moleculares como WGS, WES y RNA-seq. Las muestras restantes se encuentran disponibles en el biobanco GTEx. El portal GTEx proporciona un acceso abierto a los datos de expresión de genes, QLTs e imágenes histológicas.

La versión activa en el momento de la descarga es la versión V8, que contiene un total de 54 tipos de tejidos, con 948 donantes y 17.382 muestras. Hemos descargado datos de acceso abierto correspondientes al tejido de mama sano. La matriz que empleamos en este trabajo recoge conteos de 462 muestras como mostramos a continuación:

	GTEX.1117F.2826.SM.5GZXL	**GTEX.111YS.1926.SM.5GICC**	**GTEX.1122O.1226.SM.5H113**	**GTEX.117XS.1926.SM.5GICO**	**GTEX.117YX.1426.SM.5H12H**
ENSG00000223972.5	0	1	0	1	0
ENSG00000227232.5	286	135	110	246	44
ENSG00000278267.1	0	0	0	0	0
ENSG00000243485.5	0	0	0	0	0
ENSG00000237613.2	0	0	0	0	0
ENSG00000268020.3	1	3	0	6	0
ENSG00000240361.1	2	2	0	7	0
ENSG00000186092.4	0	3	2	2	1
ENSG00000238009.6	0	0	0	3	0
ENSG00000233750.3	0	14	0	8	0

Tabla 4. Extracto de tabla de conteos de secuenciación RNA-seq procedentes de GTEx. Creación propia

Empleamos datos procedentes de tejido sano ya que nos sirven para estudiar cómo su expresión puede verse afectada por miRNA de los cuales son dianas. Estos miRNA proceden de TGCA-BRCA descritos anteriormente.

5.2. Metodología

5.2.1. Estudio bioinformático en RStudio

El estudio bioinformático se realiza empleando el lenguaje de programación R (https://cran.r-project.org/, versión 3.4.1) mediante el entorno RStudio. Para ello, se hace uso de una serie de paquetes o librerías:

Librería	Fuente	Versión	Uso
TCGA-biolinks (Colaprico et al., 2015)	Bioconductor	3.18	Análisis integrativo con datos procedentes de GDC
edgeR (Robinson et al., 2010)	Bioconductor	3.18	Análisi empírico de expresión de genes digital en R
annotationDBI (Pagès et al., 2023)	Bioconductor	3.18	Manipulación de anotaciones basadas en SQLite
org.Hs.eg.db (Carlson, 2019)	Bioconductor	3.18	Anotación del genoma completo para humanos
GO (Carlson, 2019)	Bioconductor	3.18	Set de mapas de anotación referentes a Gene Ontology
BiocGenerics (Huber et al., 2015)	Bioconductor	3.18	Funciones genéricas S4
dplyr (Morgan, Cheng, 2023)	Tidyverse	1.1.3	Gramática de manipulación de datos
hipathia (Hidalgo et al., 2017)	Bioconductor	3.18	Análisis de alto rendimiento de rutas
readx (Wickham, Bryan, 2023)	R-Cran	4.3.1	Lectura de archivos tipo Excel
Clusterprofiler (Yu et al., 2012)	Bioconductor	3.18	Herramienta de enriquecimiento universal para la interpretación de datos ómicos
ggplot2 (Wickham 2016)	Tidyverse	3.4.4	Sistema para crear gráficos
EnhancedVolcano (Blighe et al., 2023)	Bioconductor	3.18	Creación de volcano plots

Tabla 5. Descripción de librerías empleadas en este trabajo. Creación propia.

Empleando las librerías base (https://cran.r-project.org/, versión 3.4.1) y BiocGenerics (Huber *et al.*, 2015), facilitamos la automatización de procesos.

5.2.2. Descarga de datos para desarrollar el pipeline:

Accedemos a la página de GDC (https://portal.gdc.cancer.gov/) y en concreto al proyecto de cáncer de mama dentro del programa TCGA, mencionado anteriormente, como mostramos a continuación:

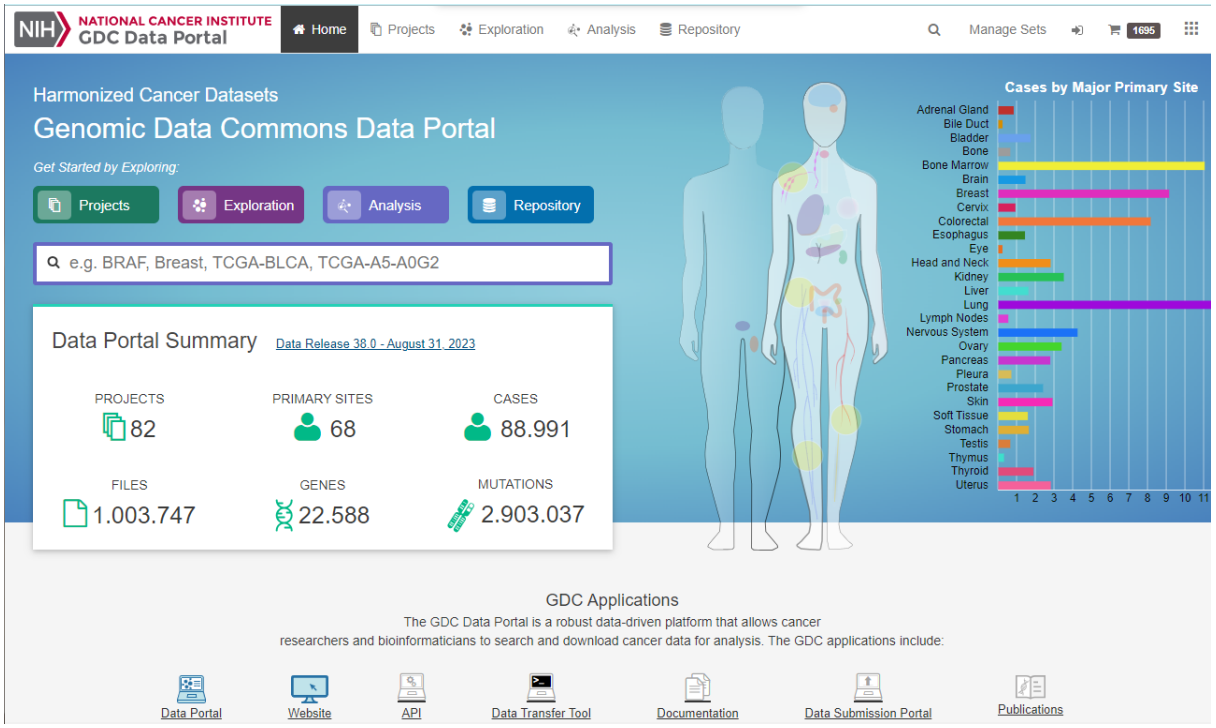


FIgura 3: Página inicial GDC. Extraído de https://portal.gdc.cancer.gov/

Los datos se pueden descargar de 4 maneras posibles:

1. Directamente desde la herramienta web de GDC, tras aplicar los filtros deseados
2. Mediante una herramienta gráfica llamada Transfer Tool, facilitada por GDC
3. Mediante una utilidad de línea de comandos
4. Mediante el paquete TCGAbiolinks en R, que nos permite no sólo la descarga sino el tratamiento directo de los datos en R.

En el caso de este desarrollo, la diferencia de tamaño y cantidad de los datos a tratar nos influye en el camino a seguir: mientras que los datos clínicos agregados de todos los sujetos de estudio vienen disponibles en un sólo archivo agregado, la información correspondiente a sus expresiones de genes y miRNA han de descargarse individualmente para cada uno de los pacientes.

Es por esto que utilizamos, para la matriz clínica, la descarga de los datos desde la web de GDC: :

- Descarga de datos clínicos:

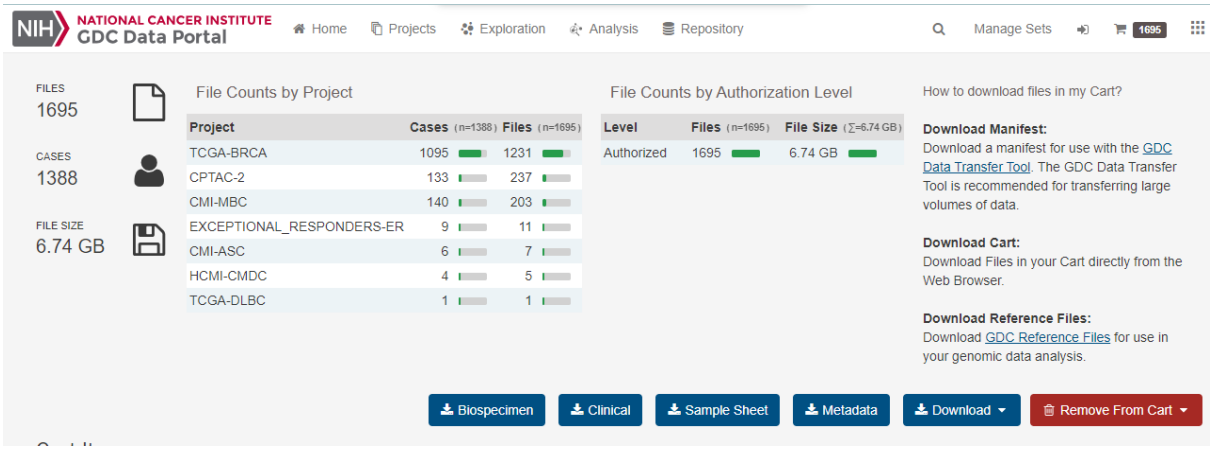


FIgura 4: Extracto de la pantalla de descarga de datos del portal GDC. Extraído de https://portal.gdc.cancer.gov/

La descarga es rápida y el archivo final es un conjunto de archivos .tsv contenidos en una carpeta comprimida:

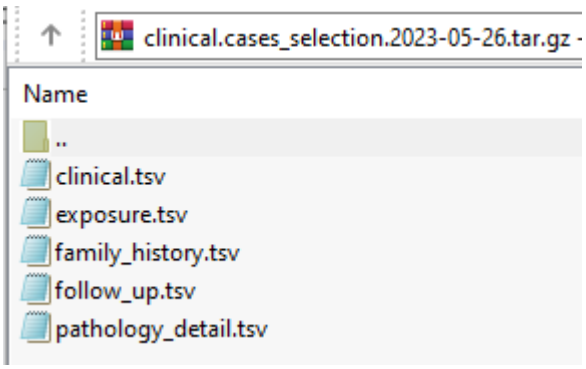


FIgura 5: Extracto del contenido de la carpeta de datos clínicos descargada del portal GDC. Creación propia

- Descarga de matrices de conteos:

Como indicamos anteriormente, la cantidad de archivos a descargar nos hacen complicado el uso de métodos manuales. Es por esto que en un principio se evalúa utilizar la herramienta *Transfer Tool*, facilitada y recomendada por GDC, pero resulta un proceso poco práctico debido a la lentitud de la herramienta, así como su poca fiabilidad (fue necesario reiniciarla en varias ocasiones) y el hecho de que hubiera que igualmente importar los datos en R manualmente una vez descargados. .

Tras evaluar el método 3, con resultados similares, nos acabamos decantando por el método 4: el uso del paquete TCGA-biolinks (Colaprico *et al.*, 2015).

Este paquete nos facilita, no sólo la descarga de los datos de manera integrada en el entorno R, sino también la importación directa de los mismos para ser tratados, facilitando la reproducibilidad del experimento.

```
    #Creamos query para datos de CONTEOS DE GENES

query_TCGA_gene <- GDCquery(project = 'TCGA-BRCA',
                            data.category = 'Transcriptome Profiling',
                            data.type = 'Gene Expression Quantification',
                            access = 'open',
                            experimental.strategy = 'RNA-Seq'
                            )

getResults(query_TCGA_gene)


      #descargamos los datos - GDC download - conteo genes
GDCdownload(query_TCGA_gene)


      #preparamos los datos
tcga_sample_data <- GDCprepare(query_TCGA_gene, summarizedExperiment = TRUE)
tcga_gene_data <- tcga_sample_data
raw_gene_matrix <- assay(tcga_gene_data,'unstranded')

save(raw_gene_matrix, file = 'raw_gene_matrix')
```

FIgura 6: Conjunto de comandos empleados para la descarga de datos de conteos de mRNA a través de R. Creación propia

Los datos de conteos que descargamos son de dos tipos, variando el campo *data.type* de la query que realizamos a GDC:

- *Gene Expression Quantification*: conteos de expresión de mRNA procedentes de RNAseq.
- *miRNA Expression Quantification:* conteos de expresión de microRNA.

Los datos son guardados sin modificación alguna con la etiqueta de "*raw*".

<u>Datos procedentes de miRTarBase y GTEx</u>

miRTarBase: descarga de datos

(https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/search.php)

Accedemos a la página web del repositorio. En el apartado "*Download*" descargamos el archivo .xlsx correspondiente a la especie humana, en el catálogo por especies, tal y como se puede apreciar en la siguiente figura.

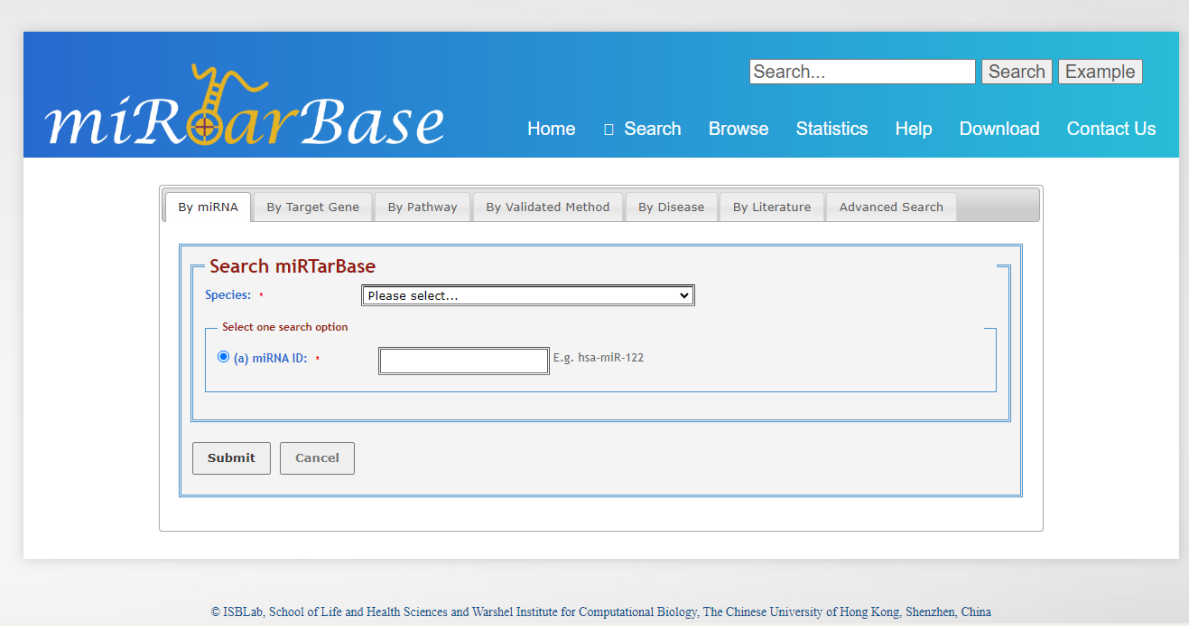


Figura 7: Página inicial miRTarBase. Extraida de https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/search.php

Release 9.0 - Download

miRTarBase Download	
All published miRNA target interaction data in EXCEL format	MTI.xlsx

MTI PubMed Abstract Manually Curation Corpus	
Manually curation of miRNA and target genes, and their relations in PubMed abstracts README	MTI-PubMed_corpus.txt

MicroRNA Target Sites Provided by Original Literatures	
Manually curation of miRNA and target sites, and the target sites were provided by original literatures	MicroRNA_Target_Sites.xlsx

Catalog by Species	
Arabidopsis thaliana	ath_MTI.xlsx
Bombyx mori	bmo_MTI.xlsx
Bos taurus	bta_MTI.xlsx
Caenorhabditis elegans	cel_MTI.xlsx
Canis familiaris	cfa_MTI.xlsx

Figura 8: Apartado de descargas del portal miRTarBase. Extraida de https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/search.php

GTEx (https://gtexportal.org/home/)

Accedemos a la página web. En el apartado "*Download*" (esquina superior izquierda de la figura __) accedemos a "*Open Access Database*". Escogemos la pestaña "*Bulk tissue expression*" y descargamos el archivo .gct.gz correspondiente a *gene_reads_2017-06-05_v8_breast_mammary_tissue.gct.gz*.

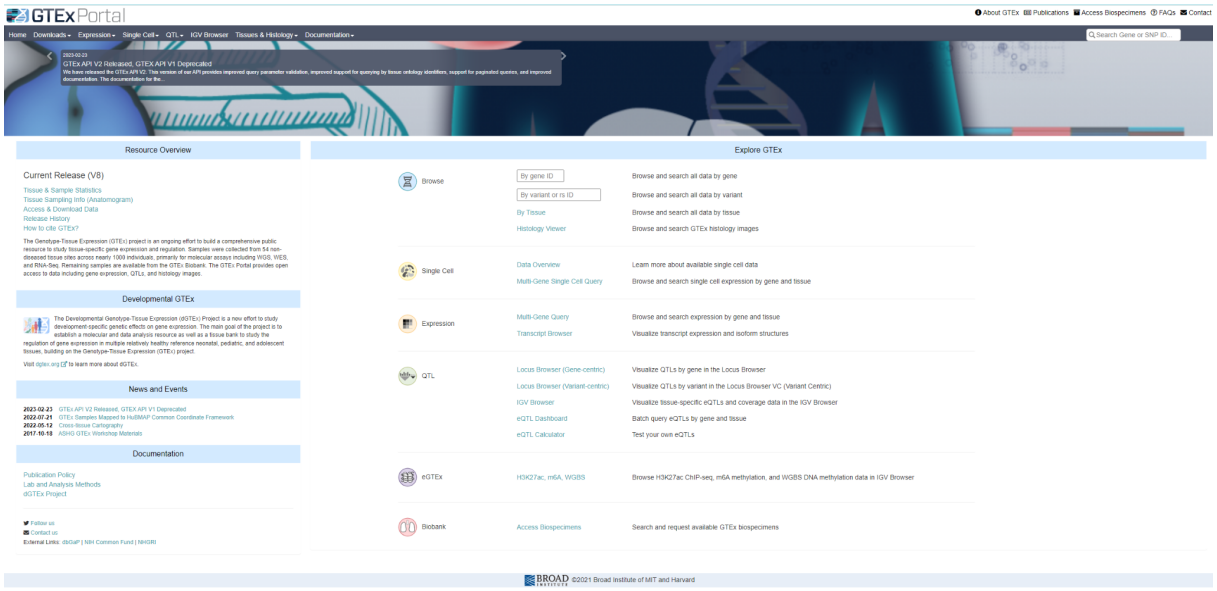


Figura 9: Página inicial GTEx. Extraida de https://gtexportal.org/home/

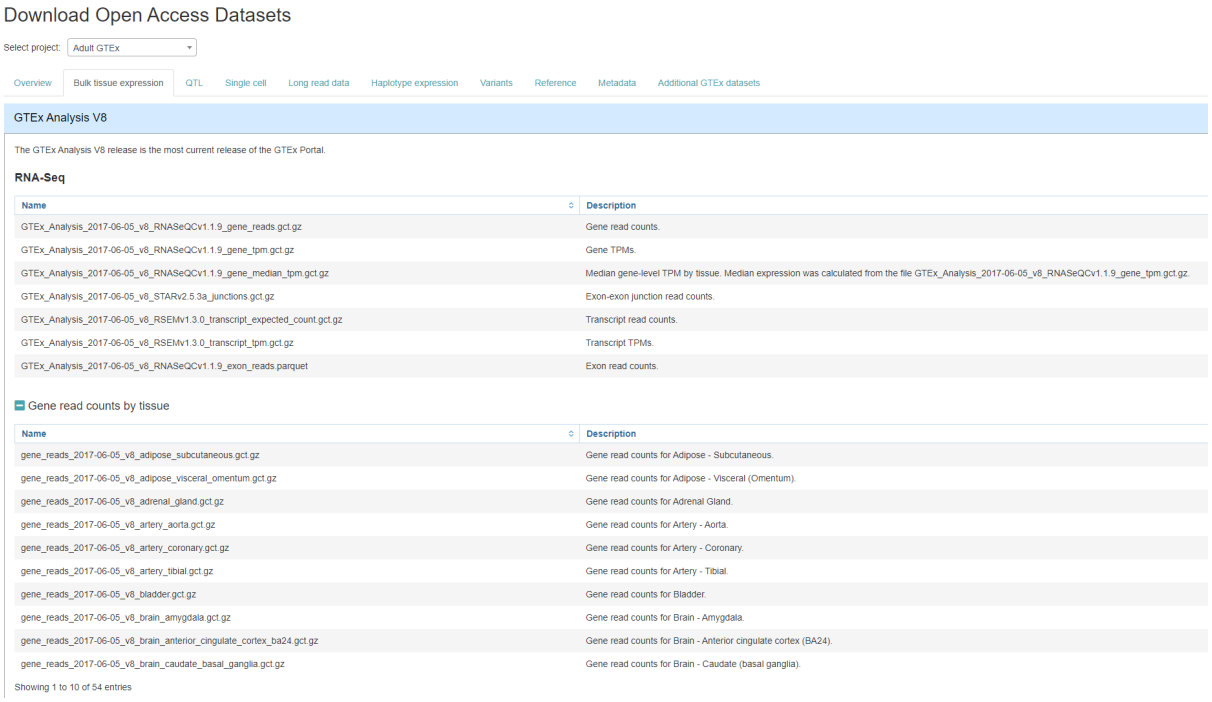


Figura 10: Apartado de descargas de datos de acceso público del GTEx. Extraida de https://gtexportal.org/home/

5.2.3. Limpieza y ajuste de las matrices de datos

En la limpieza y ajuste observamos falta de concordancia en el orden de las columnas, nombres de columnas y filas, así como duplicados entre las matrices de genes y miRNA.

Las columnas de ambos conjuntos de datos fueron ordenadas para aparecer en el mismo orden en cada matriz y filtradas para mantener sólo casos coincidentes en ambos datasets. Para posibilitar esta concordancia, hemos tenido que editar los nombres de las columnas quedándonos con sólo los tres primeros grupos del identificador de muestra ("TCGA -_ _ - _ _ _ _ "), información suficiente para identificar cada una de las muestras únicas una vez hemos comprobado el esquema en la web de TCGA y verificado que cada participante sólo cuenta con una muestra.

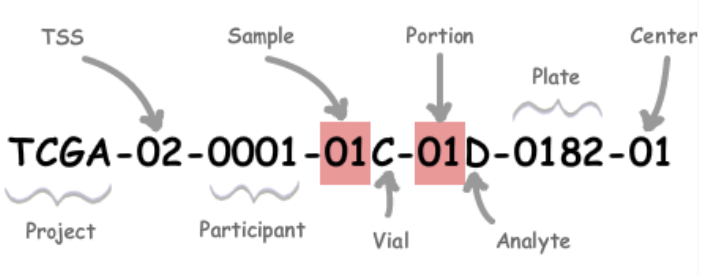


Figura 11: Significado de cada una de las partes que componen los nombres de muestras en el repositorio GDC. Extraído de https://docs.gdc.cancer.gov/Encyclopedia/pages/TCGA_Barcode/#reading-barcodes

A continuación mostramos los resultados de la limpieza y ajuste realizados:

- Datos de expresión de genes

Hemos editado los nombres de columna y ordenado las columnas para que se dispongan en el mismo orden que las columnas de miRNA, quedando así una muestra del resultado final:

	TCGA-3C-AAAU	TCGA-3C-AALI	TCGA-3C-AALJ	TCGA-3C-AALK	TCGA-4H-AAAK
ENSG00000000003.15	651	418	1303	2969	2572
ENSG00000000005.6	2	2	52	4	15
ENSG00000000419.13	1872	1818	1281	1386	1439
ENSG00000000457.14	951	2437	380	1315	1008
ENSG00000000460.17	339	490	239	460	366
ENSG00000000938.13	248	388	371	384	356
ENSG00000000971.16	911	1755	972	2672	3166
ENSG00000001036.14	1128	1458	767	1530	1652
ENSG00000001084.13	1587	1621	685	1452	1468
ENSG00000001167.14	3458	1252	731	2550	1632

Tabla 6: Extracto de la matriz final tras limpieza y recolocación de columnas de la matriz de conteos de genes raw. Creación propia

- Datos de expresión de miRNA

Para este set de datos, filtramos la información para quedarnos con la de interés para el resto del proceso::

- Columna **miRNA_ID:** identifica el miRNA. Mantenido sin cambios
- Columnas **read_count_TCGA-B6-A0IJ-01A-11R-A035-13** (una por cada muestra): contiene la información de lecturas para cada miRNA. La mantenemos, pero cambiando el nombre para quedarnos con los tres primeros grupos del identificador (**TCGA-B6-A0IJ**), para así poder concordar con las matrices clínicas y de mRNA.
- Columnas **reads_per_million_miRNA_mapped_TCGA-...** y **cross-mapped_TCGA-**...: no son de relevancia para nuestro análisis, así que las eliminamos.

Columna original raw_miRNA	**Columna nueva**
miRNA_ID	miRNA_ID
read_count_TCGA-B6-A0IJ-01A-11R-A035-13	TCGA-B6-A0IJ
reads_per_million_miRNA_mapped_TCGA-B6-A0IJ-01A-11R-A035-13	*descartada*
cross-mapped_TCGA-B6-A0IJ-01A-11R-A035-13	*descartada*
read_count_TCGA-D8-A1XY-01A-11R-A14L-13	TCGA-D8-A1XY
reads_per_million_miRNA_mapped_TCGA-D8-A1XY-01A-11R-A14L-13	*descartada*
cross-mapped_TCGA-D8-A1XY-01A-11R-A14L-13	*descartada*
read_count_TCGA-AQ-A54O-01A-11R-A25Z-13	TCGA-AQ-A54O

Tabla 7: Extracto de tabla comparativa de nombres de columnas de la matriz raw de miRNA y de la matriz de conteos de miRNA editada. Creación propia

5.2.4. Normalización por TMM

Existen varios métodos para normalizar los datos, como RPKM (*Reads Per Kilobase Million*) , FPKM (*Fragments Per Kilobase Million*), TPM (*Transcripts Per Kilobase Million*) y TMM (*Trimmed Mean of M-values*). También existen diferentes librerías que realizan estos métodos, como EdgeR, limma o DESeq. Para esta tarea, hemos escogido TMM con la librería EdgeR. Esto es debido a que RPKM, FPKM y TPM muestran un desempeño pobre cuando las muestras presentan una heterogeneidad en las distribuciones, es decir, cuando los transcritos presentan características altamente diferenciadas que pueden introducir un sesgo en la distribución de los conteos, que TMM no introduce (Conesa et al., 2016).

Realizamos la normalización por TMM (*trimed m-means*) usando el paquete edgeR (Robinson *et al.*, 2010) de los conteos de los genes procedentes de TCGA-BRCA.

```
  #GEN
y<-DGEList(gen_sub)
y<-calcNormFactors(y, method = "TMM")
gen_norm <- cpm(y,log=TRUE,prior.count = 3)

#guardamos archivos
save(gen_norm, file = 'gen_norm')
```

Figura 12: Lista de comandos empleada para realizar la normalización por TMM sobre la matriz de conteos de genes . Creación propia

Original	TCGA-3C-AAAU	TCGA-3C-AALI	TCGA-3C-AALJ	TCGA-3C-AALK	TCGA-4H-AAAK	Normalizado	TCGA-3C-AAAU	TCGA-3C-AALI	TCGA-3C-AALJ	TCGA-3C-AALK	TCGA-4H-AAAK
ENSG00000000003.15	651	418	1303	2969	2572	ENSG00000000003.15	3.378602	3.203245	5.6061443	5.790811	5.650944
ENSG00000000005.6	2	2	52	4	15	ENSG00000000005.6	-3.584943	-3.390446	0.9952275	-2.987857	-1.538357
ENSG00000000419.13	1872	1818	1281	1386	1439	ENSG00000000419.13	4.897782	5.317792	5.5816039	4.693294	4.814285
ENSG00000000457.14	951	2437	380	1315	1008	ENSG00000000457.14	3.923133	5.740072	3.8320816	4.617585	4.301843
ENSG00000000460.17	339	490	239	460	366	ENSG00000000460.17	2.443795	3.431337	3.1661697	3.107846	2.846861
ENSG00000000938.13	248	388	371	384	356	ENSG00000000938.13	1.997856	3.096423	3.7976281	2.849025	2.807185
ENSG00000000971.16	911	1755	972	2672	3166	ENSG00000000971.16	3.861354	5.266978	5.1838552	5.638903	5.950437
ENSG00000001036.14	1128	1458	767	1530	1652	ENSG00000001036.14	4.168613	4.999891	4.8426721	4.835628	5.013091
ENSG00000001084.13	1587	1621	685	1452	1468	ENSG00000001084.13	4.659952	5.152550	4.6798597	4.760277	4.843018
ENSG00000001167.14	3458	1252	731	2550	1632	ENSG00000001167.14	5.781995	4.780517	4.7734446	5.571551	4.995547

...

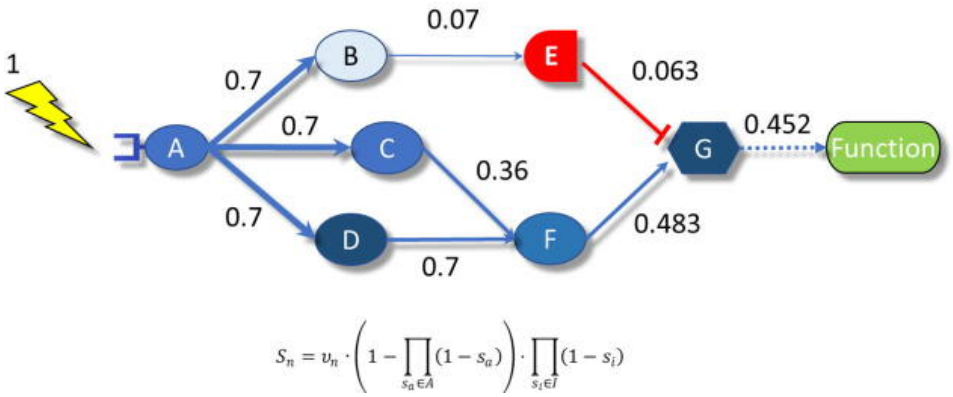
Tabla 8 : Extracto de tabla comparativa de matriz de conteos de genes procedentes de TCGA-BRCA, antes y después de la normalización por TMM. Creación propia

La matriz de datos procedentes del portal GTEX sigue un proceso similar al mostrado. Para el set de datos de miRNA, del proyecto TCGA, el proceso se encuentra integrado en el análisis diferencial, que mostraremos más adelante.

5.2.5. Análisis funcional

- ORA (*Over-representation analysis*): Una aproximación común a la hora de realizar la anotación funcional de datos de RNA-seq. En pocas palabras, consiste en comparar una lista de genes con el resto del genoma para observar funciones sobrerrepresentadas (Conesa et al., 2016). Se aplica el método propuesto por Benjamini y Hochberg, que controla el ratio de falso descubrimiento (FDR). El FDR es la proporción esperada de falsos positivos entre todos los positivos que rechazan la hipótesis nula y no entre los test realizados. (Jafari y Ansari-Pour, 2019).

- Hipathia: Para cuantificar la señal de la transducción se normalizan los valores de expresión de genes (aportados por las medidas de mRNA), se re-escala a valores en el rango de 0 a 1. Esto es tomado como *proxy* de la actividad de las proteínas. Se le asigna un valor máximo de señal de 1 al receptor, la cual es propagada por los nodos de los circuitos de señalización de acuerdo a la fórmula recursiva de la figura que mostramos a continuación:



Gene	Exp.
A	0.7
B	0.1
C	0.5
D	1.0
E	0.9
F	0.6
G	1.0

Figura 13. Ejemplo de funcionamiento del modelo Hipathia. Ejemplo de ruta con receptor A, 5 proteínas que transducen la señal (B, C, D, E, F, siendo E un inhibidor) y G como proteína receptora. La tabla representa valores normalizados de expresión génica. Tomada de Rian *et al.*, 2021.

Donde la n_i es la actividad el nodo actual *n*, A es el número total de aristas que llegan al nodo que representan el flujo de metabolitos producidos en otros nodos con valores de actividad S_a, y S_i es el flujo del nodo de respuesta

(Cubuk *et al.*, 2018).

La aplicación de esta fórmula a todos los circuitos definidos en todas las rutas permite transformar el perfil de expresión de un gen a su correspondiente perfil de actividad de señalización de circuitos para cada muestra estudiada. Si dos condiciones son comparadas, se puede emplear un test Wilcoxon para evaluar las diferencias en actividad de señalización de circuitos entre los dos tipos de muestras (Peña-Chilet *et al.*, 2019).

5.2.6. Simulación de silenciamiento in silico

- **Diseño del experimento**

Un silenciamiento *in silico* consiste en, a partir de lecturas de conteos de genes en tejidos sanos, replicar estos datos multiplicando los conteos de un set de estos genes por un factor de silenciamiento. De este modo, obtenemos dos valores: la lectura original del tejido sano por un lado, y por otro, una simulación de cómo afectaría la desregulación de la expresión de los microRNAs a dicha lectura.

Estos dos datos en paralelo (el tejido sano, y el tejido afectado por la desregulación) nos permiten construir una matriz de diseño acorde que indique si el sujeto analizado corresponde a los datos de tejido sano o a los desregulados. Con estos conjuntos de datos, podemos simular el efecto de una desregulación de los miRNA significativos anteriormente descritos con Hipathia para obtener los valores de activación de las rutas de señalización afectadas.

Para realizar esto, generamos un factor de silenciamiento proporcional al *Fold Change* (FC) de cada miRNA. Separamos los FC por deciles y establecemos rangos escalados entre el mayor y el menor entre 0 y 1, siendo 0 nada de efecto y 1 silenciamiento total. Con este FC escalado multiplicamos el valor de expresión de la diana por el inverso de su factor de silenciamiento (1-score). De esta manera, tenemos dos tipos de datos, unos valores de expresión normales de tejido sano y otros con los valores alterados, simulando así un silenciamiento.

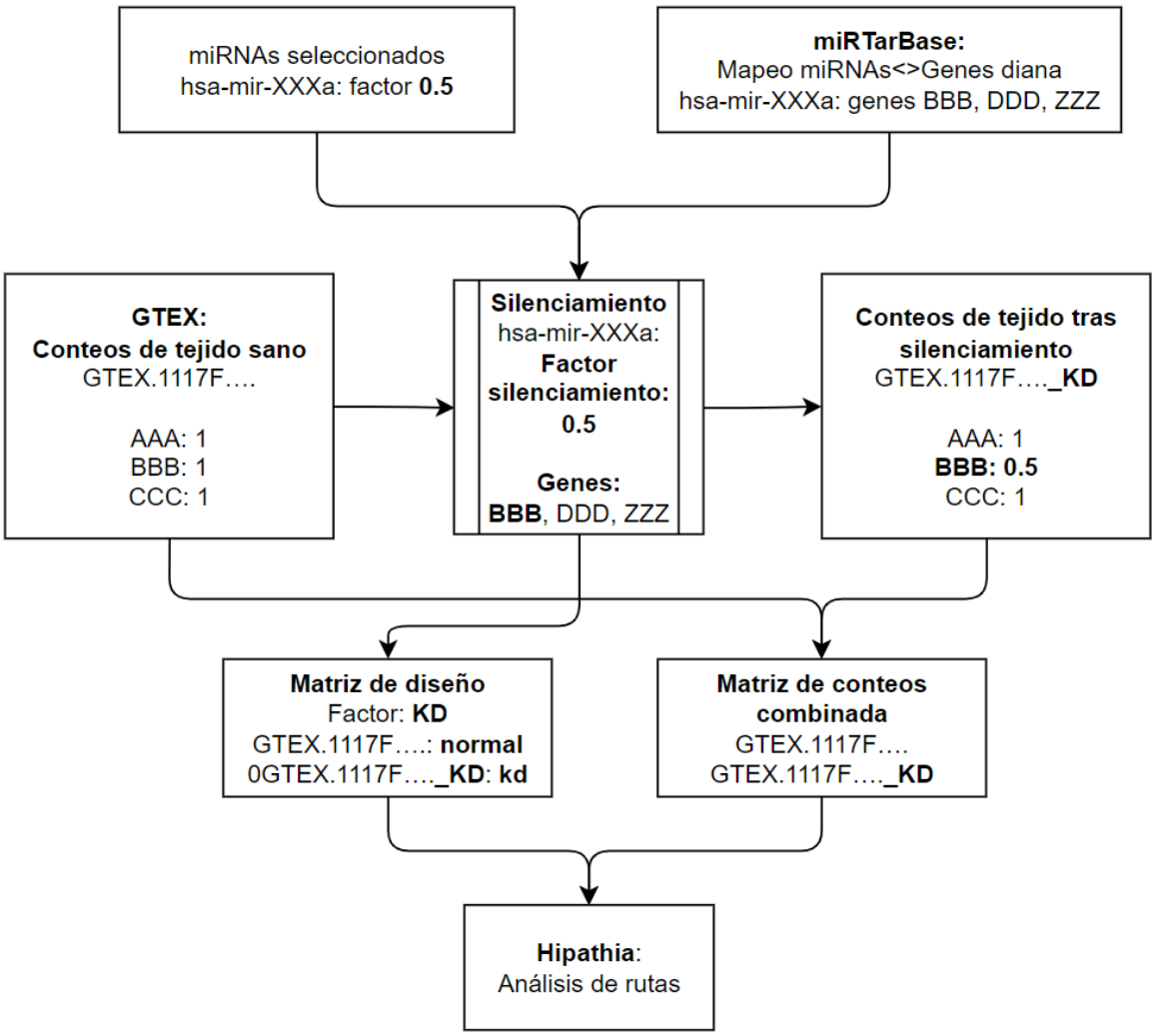


Figura 14: Diagrama de flujo sobre el diseño del silenciamiento in silico realizado en este trabajo. Creación propia

- **Tabla miRNA-KD-diana**

Cargamos la matriz descargada de miRTarBase como hemos mostrado en el objetivo 6.1.

De las 9 columnas que presenta (miRTarBase, miRNA, Species (miRNA), Target Gene, Target Gene (Entrez ID), Species (Target Gene), Experiments, Support Type y References (PMID)), nos quedamos con miRNA, Target Gene y Target Gene (Entrez ID), creando una nueva tabla que utilizamos para obtener, para cada uno de los miRNA, cuáles son sus genes diana para simular en el silenciamiento.

- **Selección de miRNA significativos y genes diana**

Una vez eliminados duplicados y filtrados con los miRNA significativos obtenidos del análisis diferencial, seleccionamos 6 miRNA, los 3 más sobreregulados y 3 más reprimidos. Una vez agrupados para obtener los genes diana asociados a cada uno, obtenemos una matriz como la siguiente:

miRNA	Número de genes diana	Genes diana	Factor silenciamiento
hsa-mir-934	26	TIPRL, ARL8B, IKBIP, SERPINC1, HSBP1a, ...	1
hsa-mir-18a	615	PSMB7, RNF138, SLC25A37, ETFA, ...	0.131963613584509
hsa-mir-577	134	ZNF648, NDUFS3, FKBP1A, NDUFAF3, NUP58, ...	0.612847491878828
hsa-mir-190b	66	DAB2, GPC5, TATDN2, MAP3K2, MTMR6, ...	0.00319001328822565
hsa-mir-342	453	SLC30A9, TMEM43, CLSTN3, KIF3A, NAA10, ...	0.011023906011702
hsa-mir-449a	167	FAM127B, PHF19, RRAS, MCFD2, MFSD8, ...	0

Tabla 9: Relación entre miRNA desregulados, genes diana y factor de silenciamiento. Creación propia

Estas técnicas se integran en una secuencia de procesos que quedan ilustrados en el siguiente flujo de trabajo.

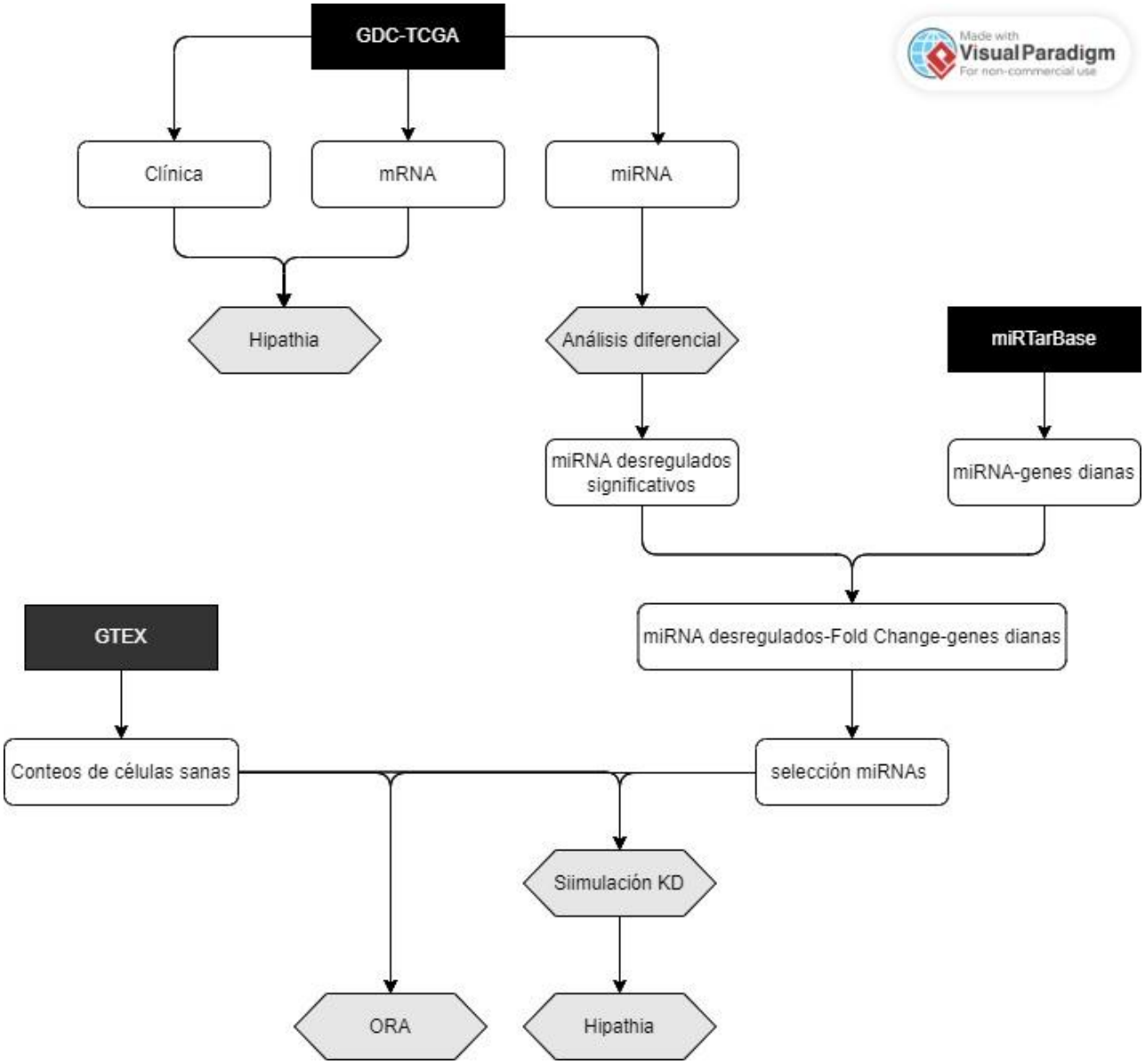


Figura 15: Diagrama de flujo de trabajo general de este proyecto. Incluye los repositorios empleados (recuadros en negro), datos empleados (recuadrosen blanco) y procedimientos resumidos por las herramientas empleadas (hexágono gris). Realizado en VisualParadigm https://www.visual-paradigm.com/. Creación propia

Ilustraremos el flujo del trabajo mediante tablas y gráficos generados en RStudio mediante la escritura de código en R.

6. Resultados

6.1. Hipathia con datos de TCGA-BRCA

6.1.1. Preparación Hipathia

Hipathia necesita dos sets de datos para realizar el análisis mecanístico de rutas, que se han de incluir en un objeto de tipo"Summarizedexperiment" (Hidalgo *et al.*, 2017):

- Matriz normalizada de conteos (objetivo 6.2), modificada para que los genes estén en formato ENTREZ ID y no en ENSEMBL y para adecuarse con la matriz de diseño (1020 columnas)

	TCGA-5T-A9QA	TCGA-A1-A0SB	TCGA-A1-A0SD	TCGA-A1-A0SE
ENSG00000000003	5.368208	6.291956	5.3554948	6.431044
ENSG00000000005	-3.683490	3.517718	-0.1894593	-1.125877
ENSG00000000419	5.861887	4.497689	5.5191936	4.729922
ENSG00000000457	4.930637	4.236758	4.5642452	4.908213
ENSG00000000460	3.611741	2.681052	2.6420078	4.473514
ENSG00000000938	1.379555	1.102279	2.7766914	1.831547
ENSG00000000971	1.583164	7.415987	6.3522563	5.598836
ENSG00000001036	5.720854	3.899848	5.7978845	4.455798
ENSG00000001084	4.777245	4.744208	5.9485547	4.743675
ENSG00000001167	4.480330	5.456895	6.0034149	6.378829

Tabla 10: Extracto de la matriz normalizada de conteos de genes procedentes de TCGA-BRCA por TMM. Creación propia

- Una matriz de diseño (1020 filas) que indique la condición a evaluar. En este caso, vamos a tomar la entidad cáncer de mama luminal (de mejor pronóstico) frente al resto de subtipos

	er_status_by_ihc
TCGA-5T-A9QA	Positive
TCGA-A1-A0SB	Positive
TCGA-A1-A0SD	Positive
TCGA-A1-A0SE	Positive
TCGA-A1-A0SF	Positive
TCGA-A1-A0SG	Positive
TCGA-A1-A0SH	Negative
TCGA-A1-A0SI	Positive
TCGA-A1-A0SJ	Positive
TCGA-A1-A0SK	Negative

Tabla 11: Extracto de la matriz de diseño para ejecutar Hipathia con la matriz de conteos normalizada de genes de TCGA-BRCA. Creación propia.

Asimismo, son necesarias las rutas de referencia (KEGG) a ser evaluadas, cuya carga se realiza con la función load_pathways del paquete de Hipathia.

6.1.2. Ejecución del modelo mecanístico Hipathia

Una vez que tenemos todo lo necesario, realizamos el análisis mecanístico con Hipathia. Este modelo nos permite representar de varias maneras los datos obtenidos, generar un report y una matriz con los valores de activación de los circuitos para cada muestra, como ilustran las dos imágenes que mostramos a continuación:

```r
# HIPATHIA

#nos aseguramos de que Hipathia pueda leer los conteos normalizados como cifras
num_gen_norm3 <- matrix(as.numeric(gen_norm3), ncol=ncol(gen_norm3))
rownames(num_gen_norm3) <- rownames(gen_norm3)
colnames(num_gen_norm3) <- colnames(gen_norm3)

design2[design2=="[Not Evaluated]"] <- "NotEvaluated"

#generamos el objeto summarizedExperiment que necesita hipathia para hacer su función
ER_sumexp <- SummarizedExperiment(assays=SimpleList(raw=num_gen_norm3),
                                  colData=design2)
save(ER_sumexp, file = 'ER_sumexp')

#cargamos rutas de humanos
pathways <- load_pathways(species = "hsa")

#Corremos Hipathia
hidataER <- hipathia(ER_sumexp, pathways, uni.terms = TRUE, GO.terms = TRUE,
                     decompose = FALSE, verbose=TRUE)
save(hidataER, file = 'hidataER')

#DAcomp
DadataER <- DAcomp(hidataER, design2$er_status_by_ihc, "Positive", "Negative")
save(DadataER, file = 'dadataER')

#Vemos los datos
DAoverview(DadataER)
DAsummary(DadataER)
DAtop(DadataER)
DApathway("hsa04010", pathways, DadataER)
#Obtenemos un report
ERreport <- DAreport(DadataER,pathways)
visualize_report(ERreport)
#Guardamos los valores de activación de los circuitos para cada muestra
path_vals <- get_paths_data(DadataER, matrix = TRUE)
save(path_vals, file = 'path_vals')
```

Figura 16: Lista de pasos y comandos para la ejecución de Hipathia. Creación propia

	ID	name	UP/DOWN	statistic	p.value	FDRp.value	N.nodes	N.gene.nodes	N.measured.nodes
1	P-hsa03320-37	PPAR signaling pathway: HMGCS2	DOWN	-4.45461337	8.409792e-06	1.910020e-05	6	2	2
2	P-hsa03320-61	PPAR signaling pathway: APOA1	UP	16.49405001	4.057920e-61	3.045063e-59	6	2	2
3	P-hsa03320-46	PPAR signaling pathway: APOA2	UP	10.52121805	6.907591e-26	9.061986e-25	6	2	2
4	P-hsa03320-57	PPAR signaling pathway: APOC3	DOWN	-2.63831199	8.335338e-03	1.327427e-02	6	2	2
5	P-hsa03320-64	PPAR signaling pathway: APOA5	DOWN	-0.86426287	3.875184e-01	4.552188e-01	6	2	2
6	P-hsa03320-47	PPAR signaling pathway: PLTP	DOWN	-2.44531464	1.447805e-02	2.219022e-02	6	2	2
7	P-hsa03320-65	PPAR signaling pathway: ME1	DOWN	-0.14025020	8.885700e-01	9.169182e-01	6	2	2
8	P-hsa03320-55	PPAR signaling pathway: SCD	UP	1.70958038	8.736872e-02	1.187708e-01	11	3	3
9	P-hsa03320-56	PPAR signaling pathway: FADS2	UP	5.33182450	9.730390e-08	2.585582e-07	6	2	2
10	P-hsa03320-33	PPAR signaling pathway: CYP7A1	UP	5.77179305	7.849589e-09	2.378971e-08	6	2	2
11	P-hsa03320-58	PPAR signaling pathway: CYP8B1	UP	1.23935377	2.152619e-01	2.710278e-01	6	2	2
12	P-hsa03320-59	PPAR signaling pathway: NR1H3	UP	1.75755712	7.884611e-02	1.081252e-01	11	3	3
13	P-hsa03320-63	PPAR signaling pathway: DBI	UP	11.59060876	4.605849e-31	9.290938e-30	6	2	2
14	P-hsa03320-44	PPAR signaling pathway: FABP1	UP	2.83185454	4.629863e-03	7.618967e-03	6	2	2
15	P-hsa03320-36	PPAR signaling pathway: SLC27A4	UP	1.63379893	1.023297e-01	1.379099e-01	11	3	3
16	P-hsa03320-30	PPAR signaling pathway: LPL	UP	3.34378820	8.268159e-04	1.492884e-03	13	4	4
17	P-hsa03320-28	PPAR signaling pathway: ACSL1	UP	5.38061903	7.428636e-08	2.010984e-07	11	3	3

Showing 1 to 18 of 1,876 entries, 13 total columns

Figura 17: Extracto de resultado creado por Hipathia. Describe las rutas activadas, nombre, tipo de activación, estadísticos empleados y número de nodos, genes por nodos y nodos medidos. Creación propia

En la tabla podemos observar también a qué circuito pertenece cada ruta, cuyo nombre se debe al ser el efector/el último nodo del circuito.

Al correr cada uno de los tres comandos de la anterior figura, se generan gráficos que ofrecen información interesante, como se puede comprobar a continuación. En *DAoverview* se puede contemplar fácilmente cuántos nodos y rutas se encuentran activados o desactivados, así como los uni.terms o los GO.terms.

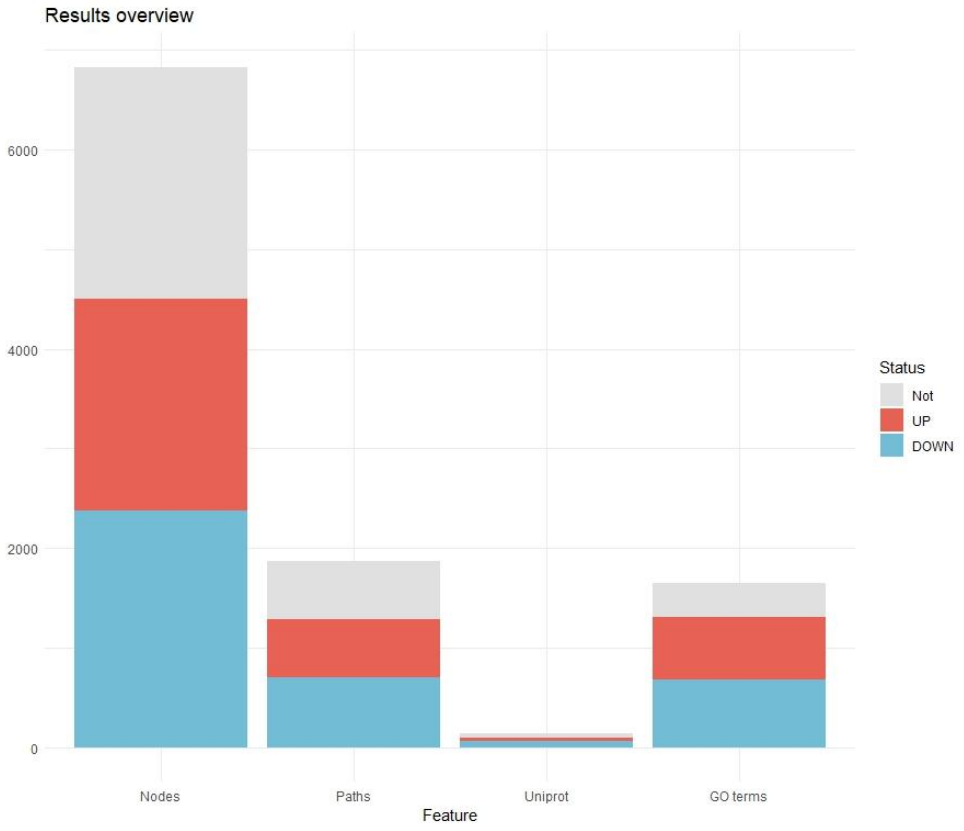


Figura 18: Resumen de resultados del análisis de activación de rutas realizado por Hipathia. Se puede apreciar tipo de activación, así como el número de nodos, rutas, rutas en Uniprot y rutas en términos Go. Creación propia.

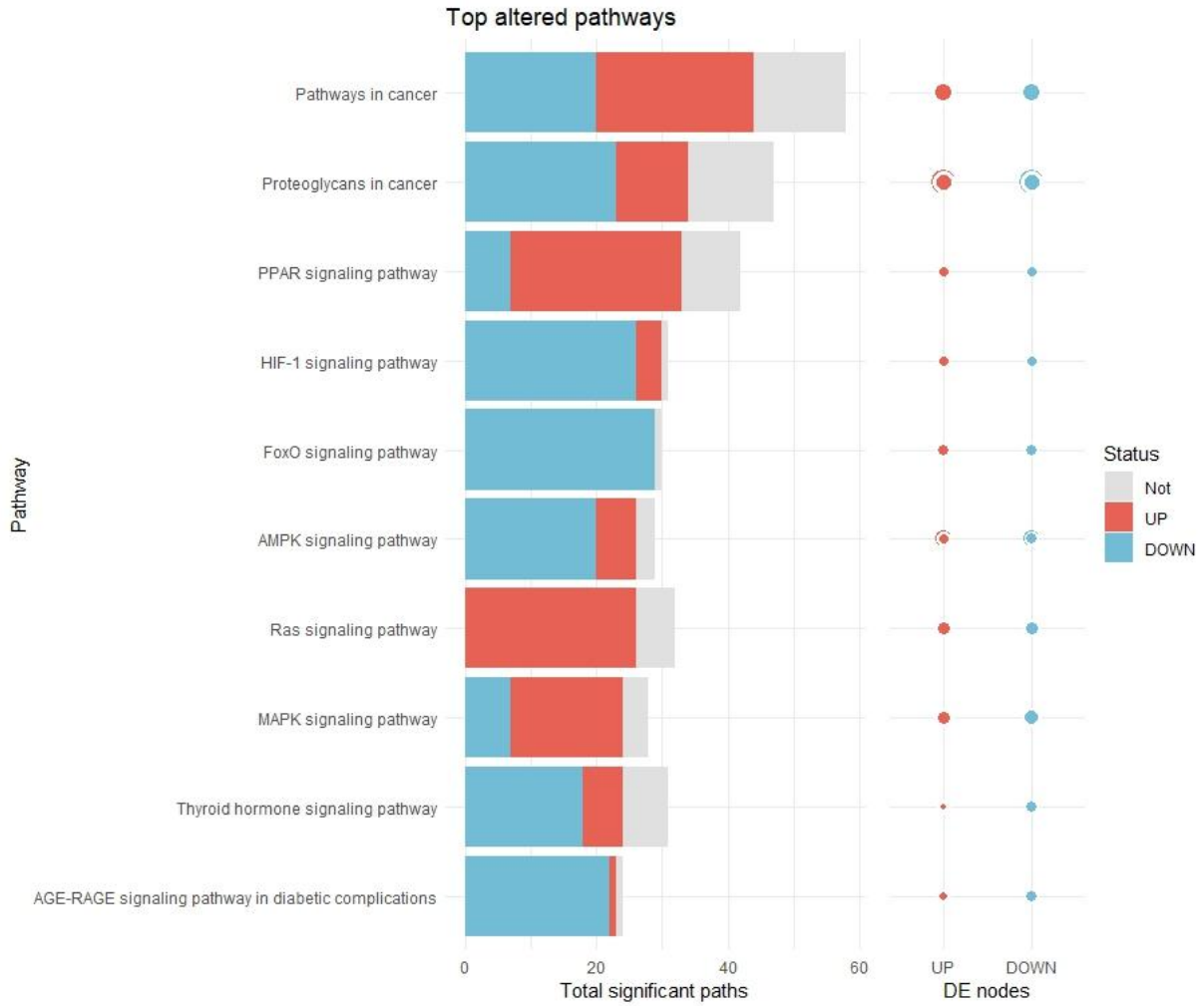


Figura 19: Resumen de rutas más alteradas tras el análisis de activación de rutas realizado por Hipathia. Se puede observar el tipo de activación, el nombre de las rutas alteradas, el número de rutas significativas y la expresión diferencial de los nodos de las rutas alteradas. Creación propia.

Cada uno de los tres gráficos que ofrece Hipathia muestra información concisa y clara sobre los valores de activación de rutas de señalización de nuestras muestras. En *top altered pathways*, por ejemplo, podemos ver qué rutas son las más alteradas, así como el número total de rutas significativas y nodos diferencialmente expresados. Podemos ver que cada gráfico obtenido amplía la información que aporta el anterior. En t*op altered pathways podemos* ver cómo, entre las rutas alteradas hay cierta variedad, como la ruta de proteoglicanos, o la de HIF-1 y la ruta PPAR además de la ruta de FoxO o la ruta de señalización de ciclo celular. Sin embargo se puede ver cómo en "*Top 10 altered features*", estas rutas se ven más agrupadas en dos, ciclo celular y FoxO.

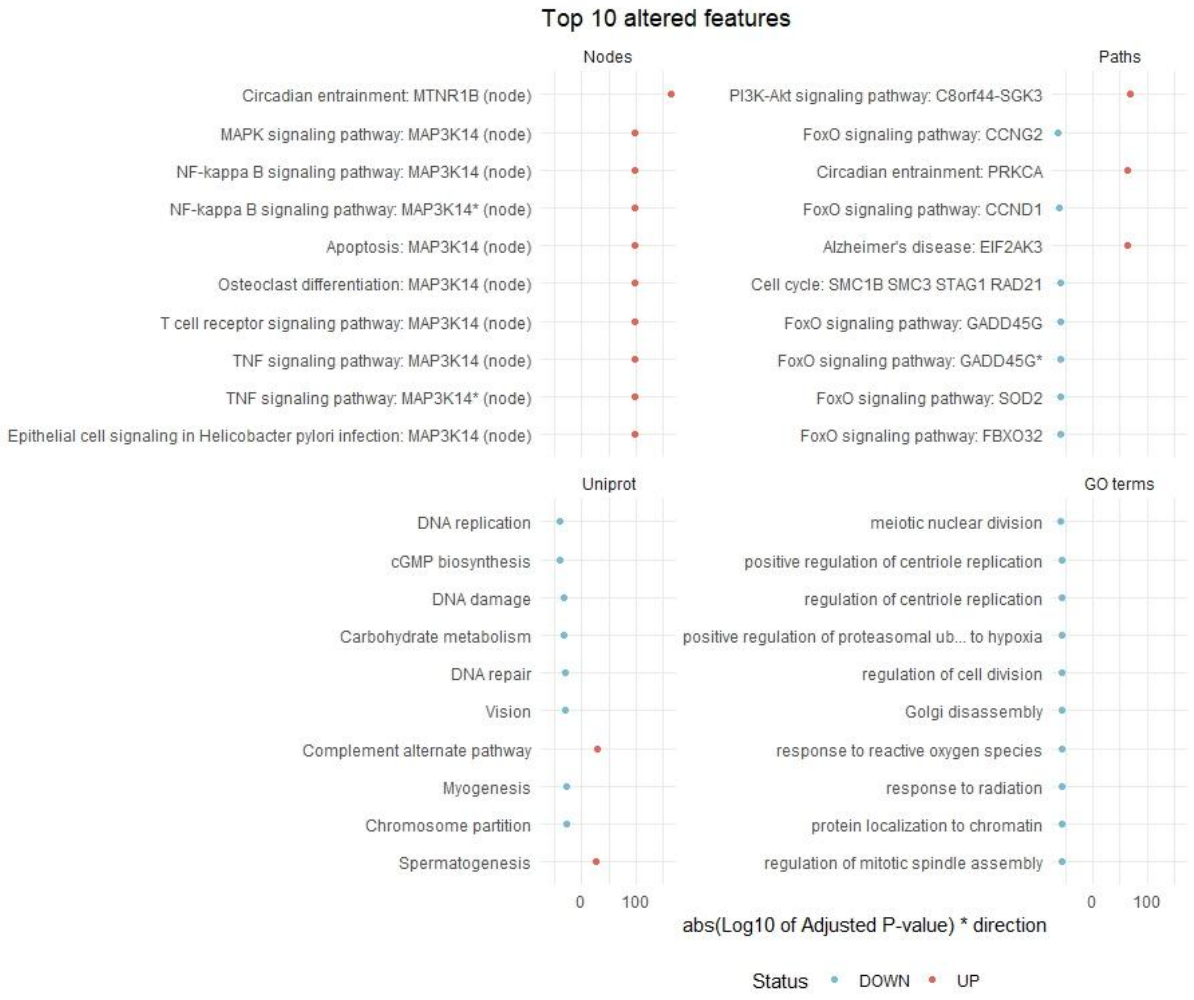


Figura 20: 10 rutas más alteradas tras el análisis de activación de rutas realizado por Hipathia.Se puede observar el tipo de activación en los nodos, rutas, rutas uniprot y rutas en términos GO representados. Creación propia.

6.2. Análisis diferencial de miRNA de TCGA-BRCA

Comenzamos realizando un análisis diferencial para conocer cuáles de los miRNA que hemos obtenido de GDC son los más relevantes. Tras el análisis diferencial, realizado con el paquete EdgeR (Robinson *et al.*, 2010), nos quedamos con una lista de 228 miRNA significativos.

	logFC	logCPM	F	PValue	FDR
hsa-mir-934	4.94	2.97	1010	1.97e-152	6.18e-150
hsa-mir-18a	2.04	3.76	669	9.45e-113	1.48e-110
hsa-mir-577	4.24	3.82	661	1.02e-111	1.07e-109
hsa-mir-505	1.32	5.55	493	8.88e-89	6.97e-87
hsa-mir-584	1.74	5.57	439	8.18e-81	5.14e-79
hsa-mir-135b	3	4.62	408	4.19e-76	2.19e-74
hsa-mir-17	1.29	9.39	372	1.5e-70	6.73e-69
hsa-mir-452	1.97	5.75	333	3.01e-64	1.18e-62
hsa-mir-190b	-2.69	4.61	316	1.79e-61	6.24e-60
hsa-mir-19a	1.34	5.02	282	7.54e-56	2.37e-54
hsa-mir-224	2.25	5.55	281	1.38e-55	3.94e-54
hsa-mir-590	0.911	4.38	261	3.31e-52	8.67e-51
hsa-mir-576	0.921	3.88	245	2.22e-49	5.35e-48
hsa-mir-455	1.34	8.07	241	8.14e-49	1.83e-47
hsa-mir-942	1.09	2.56	217	1.99e-44	4.16e-43

...

Tabla 12: Extracto del resultado del análisis diferencial de los miRNA procedentes de TCGA-BRCA. Creación propia.

Sabiendo que, según se encuentren sobreregulados o reprimidos desarrollan diferentes acciones reguladoras sobre el mRNA, separamos esa lista en dos, obteniendo 121 miRNA sobreregulados y 107 reprimidos. Esta división entre sobreregulados y reprimidos se puede observar con facilidad en el siguiente volcano plot, donde a la izquierda del Log2 del Fold Change se encuentran los miRNA reprimidos y a la derecha del Log2 del Fold Change se

encuentran los sobreregulados.

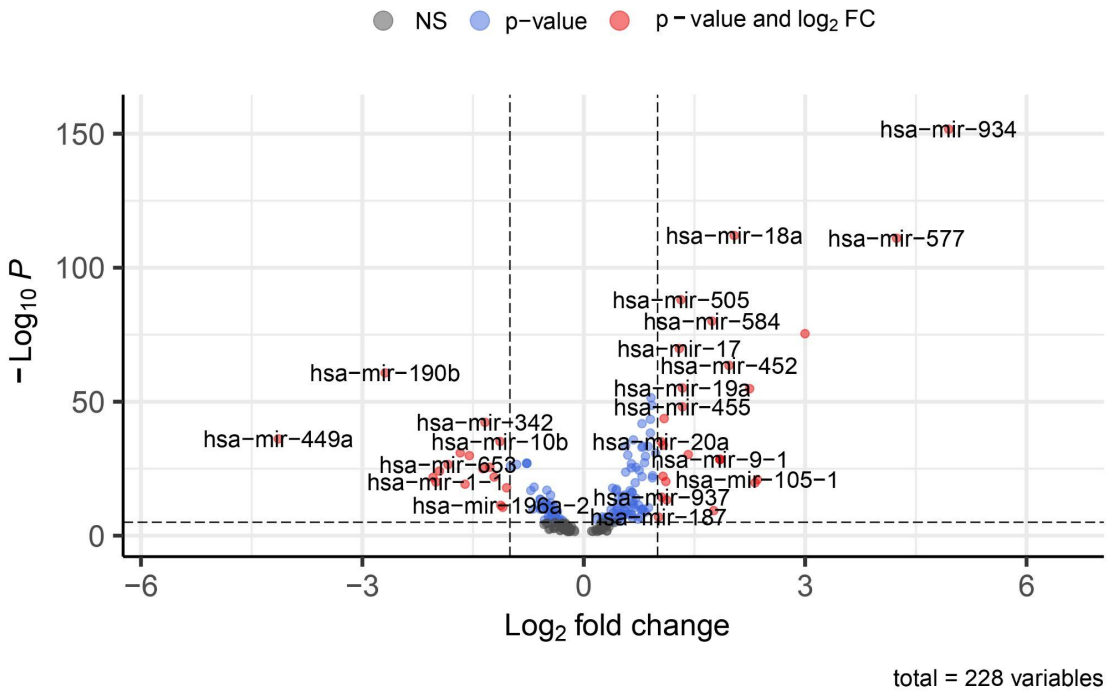


Figura 21: Representación mediante Volcano plot de los miRNA significativos tras la realización del análisis diferencial de los miRNA procedentes de TCGA-BRCA. Creación propia.

6.3. Simulación del efecto de silenciamiento in silico sobre rutas de señalización

Dada la matriz anterior con los miRNA a evaluar y sus genes diana, creamos una serie de funciones que vamos ejecutando para cada fila de la matriz:

- **autoKD**: dada una matriz de conteos de genes, una lista de genes a silenciar, y un factor de silenciamiento, devuelve un objeto SummarizedExperiment para poder ejecutar Hipathia que contiene:
    - los conteos originales con su identificador original
    - una copia de los conteos con los **genes** multiplicados por el factor de silenciamiento, con **_FC** añadido a cada identificador
    - una matriz de diseño con los identificadores, y el factor **normal** para los identificadores normales y **kd** a los que hemos realizado knockdown.

- **gen_kd_data:** para la matriz arriba descrita de miRNAs, convierte los identificadores de los genes al formato EntrezID usado por Hipathia, y llama a **autoKD,** devolviendo una lista de miRNAs con su correspondiente matriz de diseño.
- **run_hipathia:** ejecuta Hipathia sobre las salidas de gen_kd_data, devolviendo para cada miRNA su nombre, los resultados de la ejecución de Hipathia, y la matriz de diseño.
- **gen_dacomp:** para cada una de los elementos de **run_hipathia**, ejecuta un análisis comparativo teniendo en cuenta el factor de la matriz de diseño (**normal** o **kd**)
- **gen_reports:** para cada uno de los análisis comparativos de **gen_dacomp**, genera los reportes de las rutas afectadas que nos facilita hacer Hipathia (DAsummary, DAoverview), que adjuntamos en la sección de Resultados
- **gen_pathvals:** para cada uno de los análisis comparativos, devuelve una matriz de **valores de activación** para cada una de las rutas activadas.

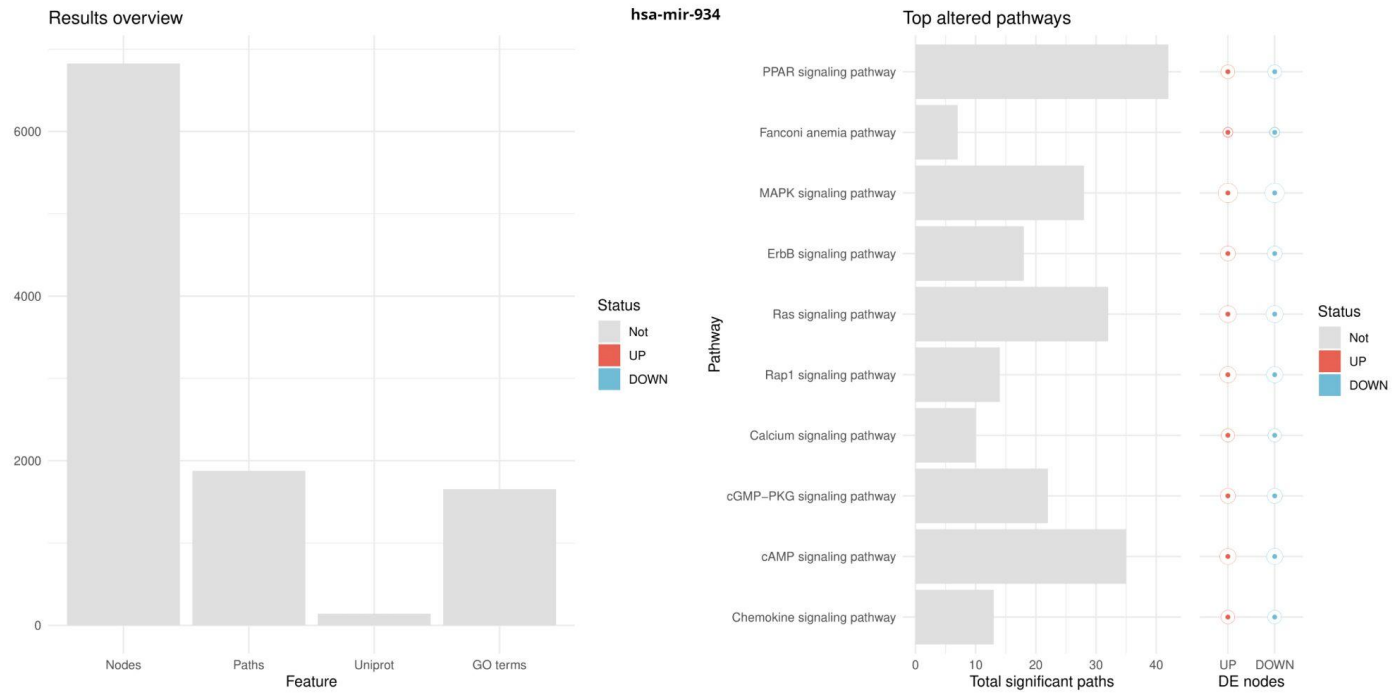


Figura 22: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-934. Creación propia.

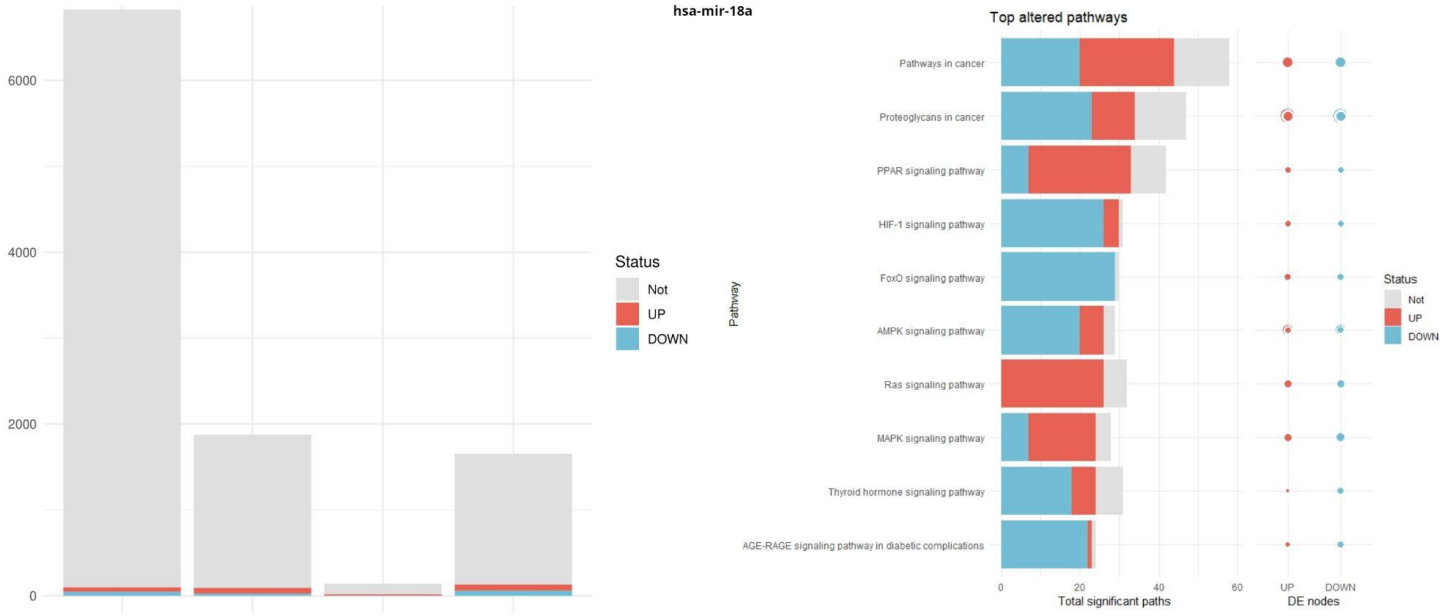


Figura 23: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-18a. Creación propia.

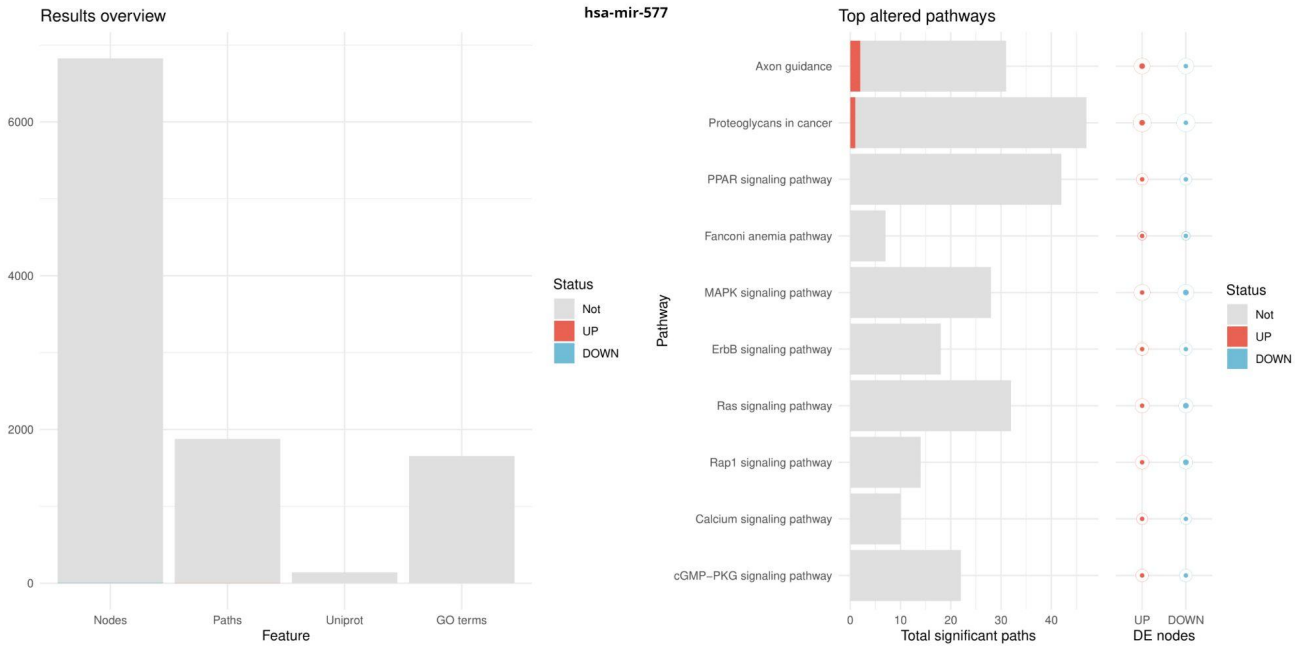


Figura 24: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-577. Creación propia.

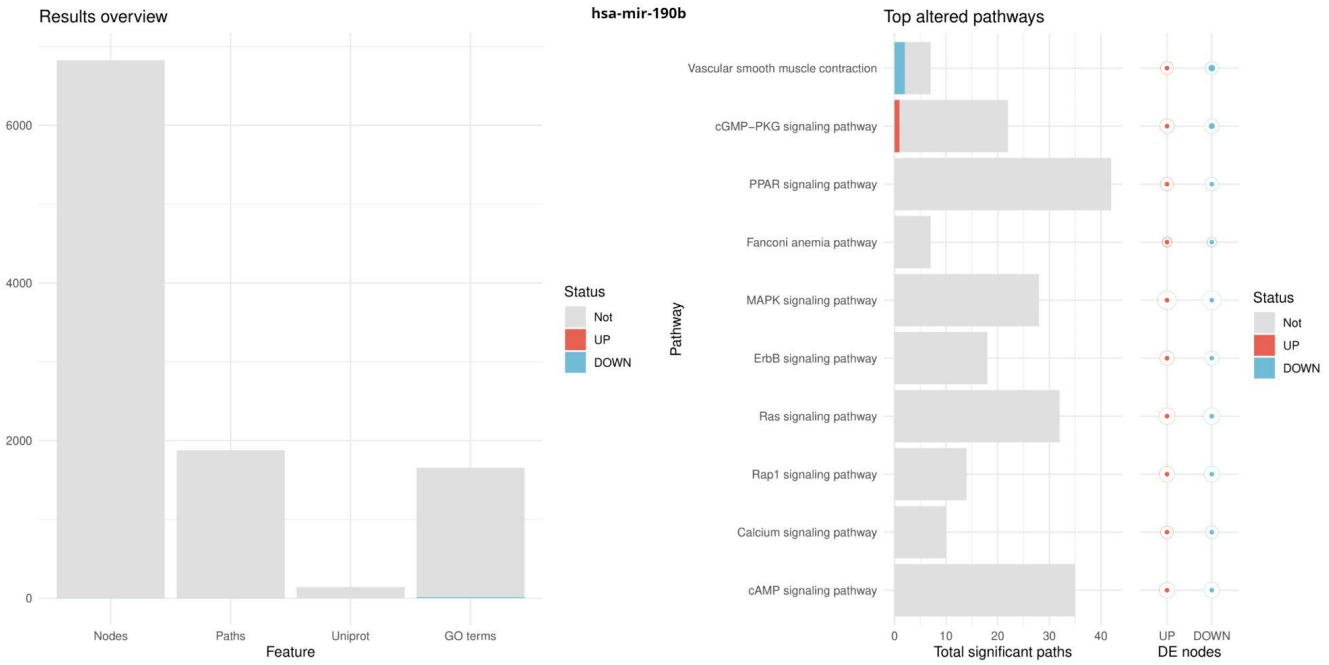


Figura 25: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-190b. Creación propia.

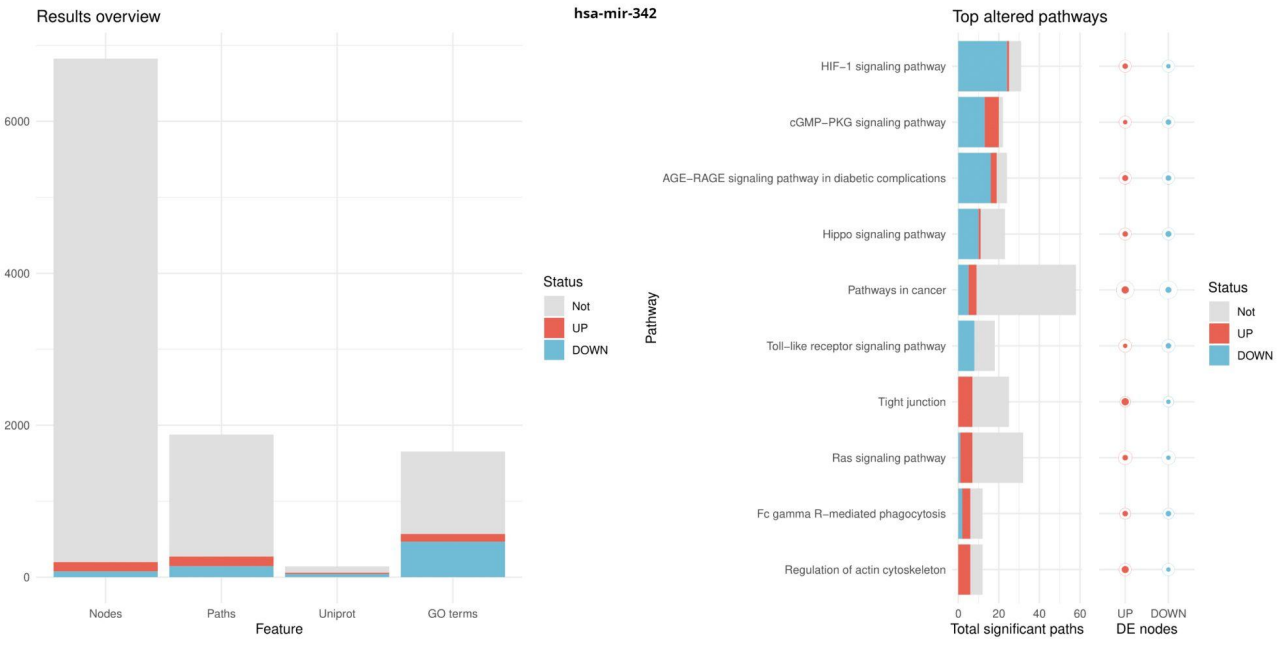


Figura 26: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-342. Creación propia.

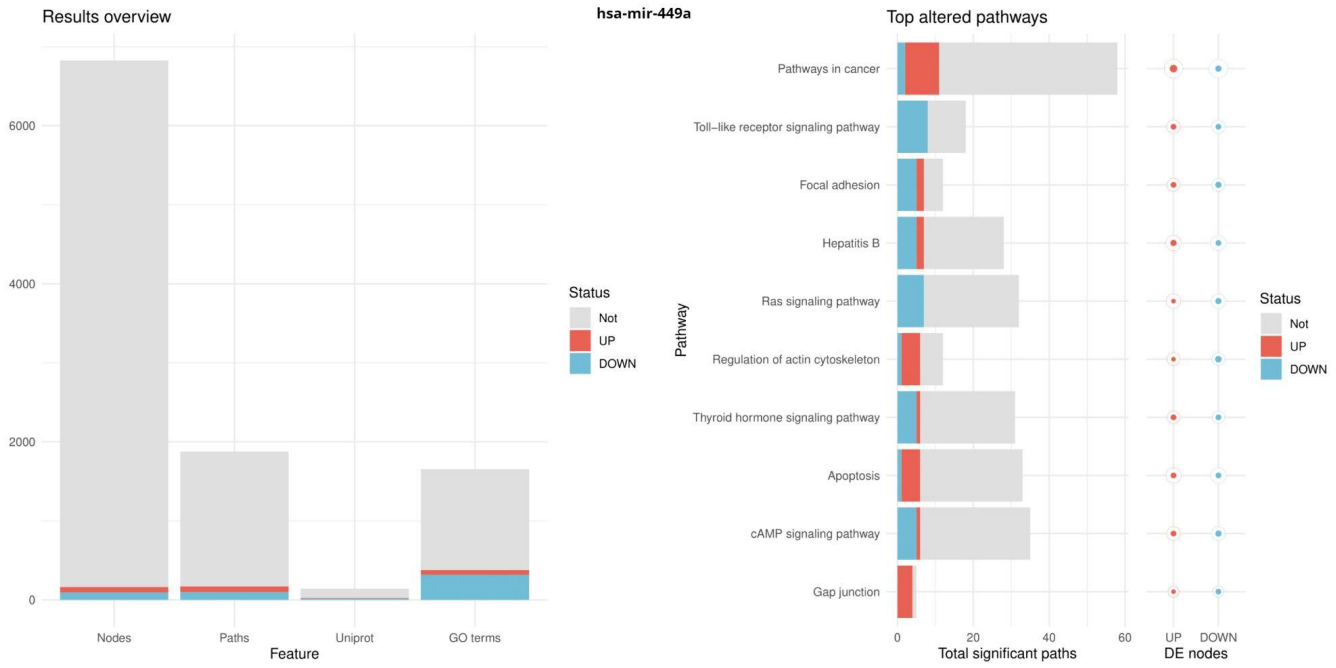


Figura 27: Resumen de resultados y rutas más alteradas para las rutas de los genes diana del miRNA hsa-mir-449a. Creación propia.

Revisando la validez estadística de las tablas de valores de activación de rutas obtenidos en los bucles de Hipathias realizados, observamos los gráficos de resumen y de rutas más alteradas obtenidos en R.

Antes es bueno considerar varios aspectos. Los miRNA seleccionados (de los 3 más sobreregulados y los 3 más reprimidos) tienen un número distinto de dianas y también, un factor de KD diferente. Los miRNA hsa-mir-934, hsa-mir-18a, hsa-mir-577 son los 3 más sobreregulados. Los miRNA cuyas dianas muestran sus rutas menos alteradas son hsa-mir-934 y hsa-mir-577. Casualmente, son los dos miRNA cuyo factor de KD es mayor, 1 y 0,61, respectivamente. Teniendo en cuenta que 1 es el valor máximo de la señal que se emite desde el receptor hasta el efector y que, los conteos sobre los que hemos realizado los Hipathias son de tejido sano, es razonable que aparezcan estos resultados. Por otro lado, los miRNA cuyas dianas muestran sus rutas (y su número) más alteradas son hsa-mir-18a y hsa-mir-342. Sus factores de KD se sitúan en el intermedio de los 6 que hemos considerado y siendo el valor para los reprimidos (hsa-mir-342) y el más bajo de los sobreregulados (hsa-mir-18a).

Hemos de señalar que, aunque estas observaciones son referidas a los gráficos de “top altered pathways”, en el resumen de resultados se puede ver que, en general, no son un

número elevado las rutas alteradas en ninguno de los casos. Esto es razonable ya que hemos simulado la actividad de 1 miRNA por cada uno de los análisis con Hipathias que hemos realizado, sobre tejido sano. Es interesante revisar las gráficas obtenidas con datos de TCGA y observar las diferencias entre número y rutas alteradas con respecto a estos 6 Hipathias.

También resulta llamativo, cómo dentro de la variedad de rutas alteradas que podemos observar en los 6 análisis mecanísticos realizados, existen dos rutas compartidas por al menos tres miRNA (hsa-mir-18a, hsa-mir-342 y hsa-mir-449) : "pathways in cancer" y "Thyroid hormone signaling pathway".

6.4. Análisis funcional basado en la sobrerrepresentación

Conociendo los genes diana de cada miRNA seleccionado, así como la lista de genes de referencia, realizamos 6 análisis funcionales mediante el empleo de análisis de sobrerrepresentación (ORA), uno por cada grupo de genes diana de cada miRNA seleccionado previamente, con el paquete *clusterProfiler*.

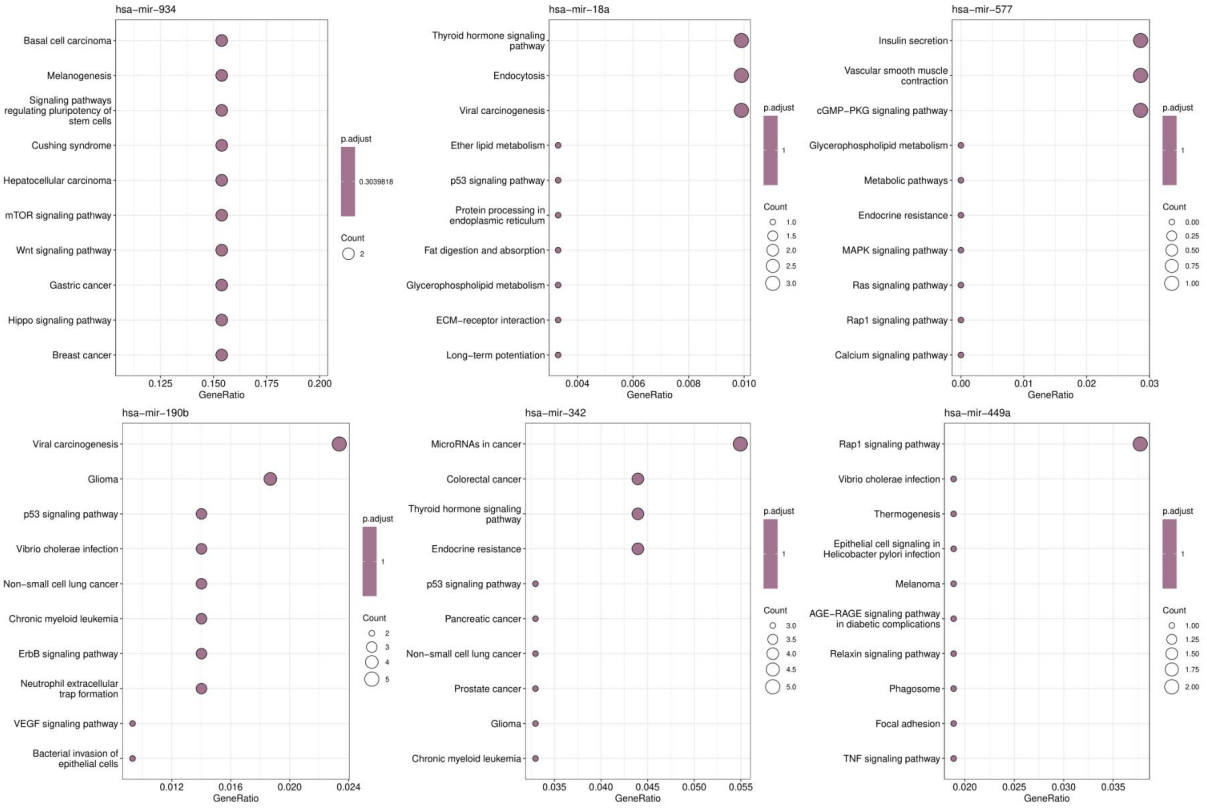


Figura 28: Dotplot de los resultados del análisis funcional basado en la sobrerrepresentación de los genes diana de los 6 miRNA seleccionados. Creación propia.

Aunque se pueden observar diferencias y parecen ser potencialmente resultados interesantes, al observar el p-valor, tanto en estos dotplot como barplot que mostramos a continuación, no podemos hacer nada más que señalar la falta de robustez en los resultados obtenidos en los análisis de sobrerrepresentación realizados.

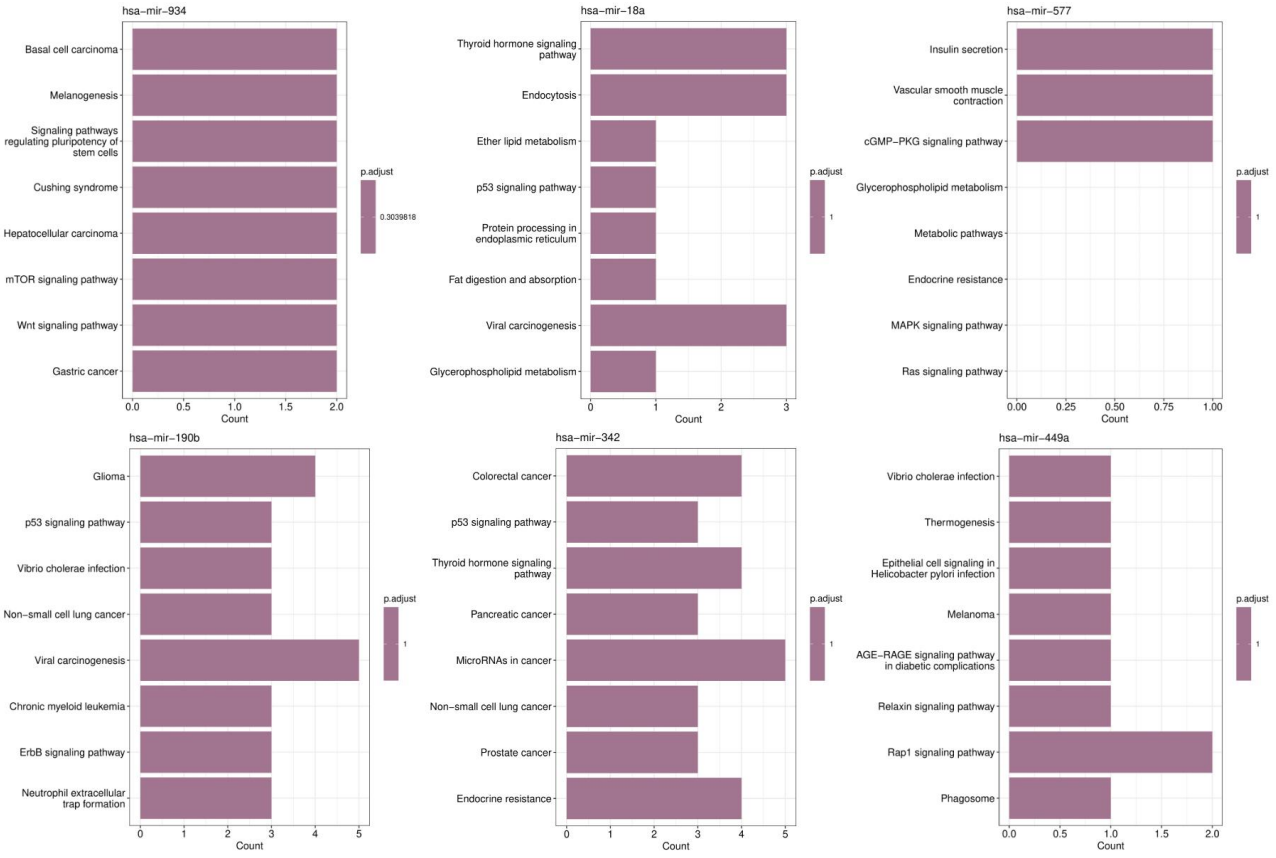


Figura 29: Barplot de los resultados del análisis funcional basado en la sobrerrepresentación de los genes diana de los 6 miRNA seleccionados. Creación propia.

7. Discusión

El objetivo de este trabajo ha sido el de establecer una metodología para la anotación e interpretación funcional de miRNAs desregulados. Hemos usado los datos de muestras de tejido de cáncer de mama del proyecto TCGA procedentes de GDC como ejemplo de uso. La descarga de datos desde R resulta ser conveniente cuando se trata de archivos de gran tamaño, aunque importar archivos descargados externamente tampoco resulta ser un

problema para R, que lee sin problemas una gran variedad de archivos diferentes.

Tras una limpieza y normalización de la matriz de conteos de los datos de TCGA, realizamos un análisis mecanístico con el objetivo de familiarizarnos con el método, con los requerimientos, con los resultados que podemos obtener y analizar el componente de señalización funcional del cáncer de mama luminal frente al resto de subtipos. Hipathia es uno de los métodos MPA desarrollados (2017), siendo el único de los métodos desarrollados hasta la fecha capaz de detectar cambios en la actividad de circuitos en células con cáncer, y postulándose como un método sensible y específico (Amadoz et al., 2019). Hipathia nos aporta, frente a métodos funcionales más tradicionales como ORA o GSEA, un enfoque mecanístico. Es por ello que, tomando unos pocos miRNAs se puede comprobar la variación en la actividad de rutas. Hipathia sólo requiere de expresión de genes y de la topología de los circuitos de señalización (proporcionados por KEGG) para poder simular una condición "mutada" o un knockdown o silenciamiento en nuestro caso, reduciendo la expresión de varios genes (asumiendo que el efecto en la transducción de una proteína inactiva es equivalente a su ausencia) (Peña-Chilet et al., 2019).

Conocer los circuitos que se encuentran alterados en el cáncer, en nuestro caso de mama, y las causas de ello pueden resultar ser de vital importancia para la supervivencia del paciente. Circuitos que desencadenan *hallmarks* de cáncer determinan una buena o mala prognosis de la enfermedad (Hidalgo et al., 2017). Un ejemplo puede ser uno de los miRNA empleados, hsa-mir-342, que se ha visto cómo su desregulación puede ser asociada a resistencia a tamoxifeno. O cómo el hsa-mir-18a puede ser un biomarcador potencial de cáncer de mama tipo basal (van Schooneveld et al., 2015).

Hemos empleado ORAs como anotación funcional ya que es un método relativamente fácil y rápido de aplicar con ánimo de comparar los resultados de los Hipathia y los ORAs con la expresión de los mRNA sanos de GTEX modificada. Emplear muestras de tejido sano, con unos niveles de expresión normales ayuda a poder observar alteraciones producidas por elementos reguladores, como es el caso de los miRNA procedentes del proyecto TCGA-BRCA, cuya expresión se encuentra alterada.

En el artículo de Borgmästars et al. (2019), ilustran la diferencia de niveles de expresión del miRNA precursor frente a las isoformas de miRNA maduro. Representan esos

niveles para hsa-mir-144 y las isoformas maduras hsa-mir-144-3p y hsa-mir-144-5p. Los niveles de expresión de la horquilla precursora hsa-mir-144 fueron más similares a la isoforma madura hsa-mir-144-5p que a hsa-mir-144-3p. En ese artículo también discuten sobre cómo muchas de las muestras de TCGA muestran expresión en hsa-mir-3p y no en hsa-mir-144-5p y cómo confiar en su precursor hsa-mir-144 hubiera causado falsos positivos en valores de expresión, ya los valores de expresión del precursor son más similares a los del miRNA maduro -5p en este caso. También comentan en este caso, cómo los datos de secuenciación de expresión de miRNA encontrados en TCGA no contenían información sobre los brazos -3p o -5p. Consiguieron desarrollar una idea para investigar esto aplicando un script de Python en R.

En nuestro caso, al no contar con la información de los brazos -3p y 5p de nuestros datos de miRNA de TCGA, debemos tener esto en consideración a la hora de interpretar los resultados. Además, al enlazar los miRNA significativos con genes dianas con la tabla de miRTarBase, que sí contaban con miRNA maduros tuvimos que adoptar la posición de quedarnos con el miRNA precursor, con la pérdida de información (y dianas) consecuente. Eso es algo a trabajar y mejorar en futuros estudios.

Quizás esta pérdida de información por trabajar con miRNA precursores y no maduros haya podido influir en la falta de robustez en los resultados de los ORAs. Aunque esto último puede deberse a que son grupos de genes pequeños y que el método no sea especialmente sensible (Amadoz et al., 2019).

Aún así, sí que hay rutas que aparecen destacadas en los dos métodos, la ruta de señalización de la hormona tiroidea y rutas de cáncer. La ruta de la hormona tiroidea es una de las rutas destacadas en el cáncer luminal tipo B, junto con otras rutas como la correspondiente a cAMP (también destacada en el análisis mecanístico de rutas de activación los genes diana de hsa-mir-449 y hsa-mir-18a) (Mejía-Pedroza et al., 2018). También aparecen rutas destacadas en el cáncer luminal tipo A, como la ruta de señalización FoxO (también destacada en el análisis mecanístico de rutas de activación los genes diana de hsa-mir-18a) (Mejía-Pedroza et al., 2018). Contar con la información correspondiente a todos los miRNA desregulados podría ampliar esta información.

Aunque ya con unos pocos miRNA se pueden observar variaciones en las rutas, estos 6 miRNA seleccionados son suficientes para entender la relevancia y el potencial beneficio de estudiar cómo la expresión desregulada de miRNA altera la activación de rutas de señalización en el cáncer. Sería interesante realizar un mapa de interacciones con todos los miRNA, tendríamos una imagen más completa de lo que ocurre a nivel celular.

Esto podría ser posible con un mayor conocimiento de los miRNA (sus mecanismo de regulación son complejos y actúan de manera diferente según el tejido (Borgmästars *et al.*, 2019) y unos medios computacionales a la altura. Por eso es interesante desarrollar los script necesarios, no sólo por facilitar la automatización de tareas (en vez de correr uno a uno los 6 análisis realizados por Hipathia en su página web) sino, porque pueden ser de ayuda para otras investigaciones, como una herramienta de partida que puede ser editada y mejorada según necesidades.

8. Conclusiones

1. Gracias a la integración de bases de datos diferentes (expresión alterada de miRNA en cáncer de mama, relación de miRNA y genes dianas y expresión de células de tejido sano) y al uso de un modelo mecanístico como es Hipahtia se puede estudiar el efecto sobre las rutas que puede tener un miRNA desregulado.
2. Hemos de ser cautelosos y conocedores del estado de maduración de los miRNA que empleamos para evitar falsos positivos que nos desvíen de ampliar y profundizar la investigación en este campo.
3. Se ha empleado un ORA, un análisis funcional clásico para comparar métodos y hemos podido observar que Hipathia es un modelo más sensible y específico que el ORA.
4. Desarrollar scripts automatizados facilita y agiliza la realización de procesos, aliviando la carga de la persona que se encuentre investigando y posibilitando un control más preciso y amplio del proceso que se está llevando a cabo.

9. Bibliografía

1. Amadoz, A., Hidalgo, M. R., Çubuk, C., Carbonell-Caballero, J., & Dopazo, J. (2019). A comparison of mechanistic signaling pathway activity analysis methods. *Briefings in bioinformatics*, *20*(5), 1655–1668. https://doi.org/10.1093/bib/bby040.
2. Blighe K, Rana S, Lewis M (2023). EnhancedVolcano: Publication-ready volcano plots with enhanced colouring and labeling. doi:10.18129/B9.bioc.EnhancedVolcano, R package version 1.20.0
3. Borgmästars, E., de Weerd, H.A., Lubovac-Pilav, Z. *et al.* miRFA: an automated pipeline for microRNA functional analysis with correlation support from TCGA and TCPA expression data in pancreatic cancer. *BMC Bioinformatics* 20, 393 (2019). https://doi.org/10.1186/s12859-019-2974-3
4. Colaprico A., Silva T. C., Olsen C., Garofano L., Cava C., Garolini D., Sabedot T. S., Malta T. M., Pagnotta S. M., Castiglioni I., Cevvarelli M., Bontempi G., Noushmehr H. (2015). *TCGAbiolinks*: an R/Bioconductor package for integrative analysis of TCGA data, *Nucleic Acids Research*, Volume 44, Issue 8, 5 May 2016, Page e71, https://doi.org/10.1093/nar/gkv1507.
5. Carlson M. (2019). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.8.2.
6. Carlson M. (2019). GO.db: A set of annotation maps describing the entire Gene Ontology. R package version 3.8.2.
7. Conesa, A., Madrigal, P., Tarazona, S. *et al.* A survey of best practices for RNA-seq data analysis. *Genome Biol* 17, 13 (2016). https://doi.org/10.1186/s13059-016-0881-8
8. Cubuk, C., Hidalgo, M. R., Amadoz, A., Pujana, M. A., Mateo, F., Herranz, C., Carbonell-Caballero, J., & Dopazo, J. (2018). Gene Expression Integration into Pathway Modules Reveals a Pan-Cancer Metabolic Landscape. *Cancer research*, *78*(21), 6059–6072. https://doi.org/10.1158/0008-5472.CAN-17-2705.
9. Grossman, Robert L., Heath, Allison P., Ferretti, Vincent, Varmus, Harold E., Lowy, Douglas R., Kibbe, Warren A., Staudt, Louis M. (2016) Toward a Shared Vision for Cancer Genomic Data. New England Journal of Medicine375:12, 1109-1112.

10. Hanahan D., Weinberg R. A. Cell, ISSN: 0092-8674, Vol: 100, Issue: 1, Page: 57-70.DOI:https://doi.org/10.1016/j.cell.2011.02.013.
11. Hidalgo, M. R., Cubuk, C., Amadoz, A., Salavert, F., Carbonell-Caballero, J., & Dopazo, J. (2017). High throughput estimation of functional cell activities reveals disease mechanisms and predicts relevant clinical outcomes. *Oncotarget*, *8*(3), 5160–5178. https://doi.org/10.18632/oncotarget.14107.
12. Hsi-Yuan Huang, Yang-Chi-Dung Lin, Shidong Cui, Yixian Huang, Yun Tang, Jiatong Xu, Jiayang Bao, Yulin Li, Jia Wen, Huali Zuo, Weijuan Wang, Jing Li, Jie Ni, Yini Ruan, Liping Li, Yidan Chen, Yueyang Xie, Zihao Zhu, Xiaoxuan Cai, Xinyi Chen, Lantian Yao, Yigang Chen, Yijun Luo, Shupeng LuXu, Mengqi Luo, Chih-Min Chiu, Kun Ma, Lizhe Zhu, Gui-Juan Cheng, Chen Bai, Ying-Chih Chiang, Liping Wang, Fengxiang Wei, Tzong-Yi Lee, Hsien-Da Huang, miRTarBase update 2022: an informative resource for experimentally validated miRNA–target interactions, Nucleic Acids Research, Volume 50, Issue D1, 7 January 2022, Pages D222–D230, https://doi.org/10.1093/nar/gkab1079.
13. Huber, W., Carey, J. V, Gentleman, R., Anders, S., Carlson, M., Carvalho, S. B, Bravo, C. H, Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, D. K, Irizarry, A. R, Lawrence, M., Love, I. M, MacDonald, J., Obenchain, V., Ole's, K. A, Pag'es, H., Reyes, A., Shannon, P., Smyth, K. G, Tenenbaum, D., Waldron, L., Morgan, M. (2015). "Orchestrating high-throughput genomic analysis with Bioconductor." Nature Methods, 12(2), 115–121. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.
14. Jafari, M., & Ansari-Pour, N. (2019). Why, When and How to Adjust Your P Values?. Cell journal, 20(4), 604–607. https://doi.org/10.22074/cellj.2019.5992.
15. Kanehisa, M., Furumichi, M., Sato, Y., Kawashima, M., & Ishiguro-Watanabe, M. (2023). KEGG for taxonomy-based analysis of pathways and genomes. *Nucleic acids research*, *51*(D1), D587–D592. https://doi.org/10.1093/nar/gkac963.
16. Loh, H. Y., Norman, B. P., Lai, K. S., Rahman, N. M. A. N. A., Alitheen, N. B. M., & Osman, M. A. (2019). The Regulatory Role of MicroRNAs in Breast Cancer. International journal of molecular sciences, 20(19), 4940. https://doi.org/10.3390/ijms20194940.

17. Luo, Weijun, Brouwer, Cory (2013). "Pathview: an R/Bioconductor package for pathway-based data integration and visualization." Bioinformatics, 29(14), 1830-1831. doi:10.1093/bioinformatics/btt285.
18. Mejía-Pedroza, R. A., Espinal-Enríquez, J., & Hernández-Lemus, E. (2018). Pathway-Based Drug Repositioning for Breast Cancer Molecular Subtypes. Frontiers in pharmacology, 9, 905. https://doi.org/10.3389/fphar.2018.00905.
19. Morgan M, Cheng Y (2023). Organism.dplyr: dplyr-based Access to Bioconductor Annotation Resources. R package version 1.28.0.
20. Mounir M, Lucchetta M, Silva TC, Olsen C, Bontempi G, Chen X, et al. (2019) New functionalities in the TCGAbiolinks package for the study and integration of cancer data from GDC and GTEx. PLoS Comput Biol 15(3): e1006701. https:// doi.org/10.1371/journal.pcbi.1006701.
21. Pagès H, Carlson M, Falcon S, Li N (2023). AnnotationDbi: Manipulation of SQLite-based annotations in Bioconductor. R package version 1.62.2, https://bioconductor.org/packages/AnnotationDbi.
22. Peng, Y., & Croce, C. M. (2016). The role of MicroRNAs in human cancer. *Signal transduction and targeted therapy*, *1*, 15004. https://doi.org/10.1038/sigtrans.2015.4
23. Peña-Chilet, M., Esteban-Medina, M., Falco, M. M., Rian, K., Hidalgo, M. R., Loucera, C., & Dopazo, J. (2019). Using mechanistic models for the clinical interpretation of complex genomic variation. *Scientific reports*, *9*(1), 18937. https://doi.org/10.1038/s41598-019-55454-7.
24. Perou, C. M., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., Lønning, P. E., Børresen-Dale, A. L., Brown, P. O., & Botstein, D. (2000). Molecular portraits of human breast tumours. Nature, 406(6797), 747–752. https://doi.org/10.1038/35021093.
25. Prat, A., & Perou, C. M. (2011). Deconstructing the molecular portraits of breast cancer. Molecular oncology, 5(1), 5–23. https://doi.org/10.1016/j.molonc.2010.11.003
26. Quévillon Huberdeau, M. and Simard, M.J. (2019), A guide to microRNA-mediated gene silencing. FEBS J, 286: 642-652. https://doi.org/10.1111/febs.14666.

27. Rian K, Hidalgo MR, Çubuk C, Falco MM, Loucera C, Esteban-Medina M, Alamo-Alvarez I, Peña-Chilet M, Dopazo J. Genome-scale mechanistic modeling of signaling pathways made easy: A bioconductor/cytoscape/web server framework for the analysis of omic data. Comput Struct Biotechnol J. 2021 May 15;19:2968-2978. doi: 10.1016/j.csbj.2021.05.022. PMID: 34136096; PMCID: PMC8170118. https://doi.org/10.1016/j.csbj.2021.05.022.
28. Robinson MD, McCarthy DJ, Smyth GK (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data." Bioinformatics, 26(1), 139-140. doi:10.1093/bioinformatics/btp616.
29. Silva, T. C., Colaprico, A., Olsen, C., D'Angelo, F., Bontempi, G., Ceccarelli, M., & Noushmehr, H. (2016). TCGA Workflow: Analyze cancer genomics and epigenomics data using Bioconductor packages. *F1000Research*, *5*, 1542. https://doi.org/10.12688/f1000research.8923.2.
30. van Schooneveld, E., Wildiers, H., Vergote, I. et al. Dysregulation of microRNAs in breast cancer and their potential role as prognostic and predictive biomarkers in patient management. Breast Cancer Res 17, 21 (2015). https://doi.org/10.1186/s13058-015-0526-y.
31. Wickham H, Bryan J (2023). _readxl: Read Excel Files_. R package version 1.4.3, <https://CRAN.R-project.org/package=readxl>.
32. Wickham, H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
33. Wilke C (2022). _ggridges: Ridgeline Plots in 'ggplot2'_. R package version 0.5.4, <https://CRAN.R-project.org/package=ggridges>.
34. Yu G, Wang L, Han Y, He Q (2012). "clusterProfiler: an R package for comparing biological themes among gene clusters." OMICS: A Journal of Integrative Biology, 16(5), 284-287. doi:10.1089/omi.2011.0118.

10. Anexos

Anexo 1: Índice de figuras

Anexo 2: Índice de tablas

Anexo 3: Lista de abreviaturas

BRCA	*Breast cancer gene*

FC	*Fold change*
FoxO	*Forkhead box O1*
FPKM	*Fragments per kilobase million*
GDC	*Genomic data commons*
GTEX	*Genotype-tissue Expression*
HER	*Human epidermal growth factor receptor*
HIF-1	*Hypoxia inducible factor 1 subunit alpha*
HIPATHIA	*High-throughput pathway interpretation and analysis*
KEGG	*Kyoto encyclopedia of genes and genome*
miRNA	*Micro RNA*
MPA	*Mechanistic pathway analysis*
mRNA	*Messenger RNA*
MTIs	*miRNA target interaction*
oncomiRs	*Oncogenic miRNA*
ORA	*Over-representation analysis*
PPAR	*Peroxisome proliferator activated receptor gamma*
pre-miRNA	*Precursor miRNA*
QLTs	*Quantitative trait locus*
RNA	*Ribonucleic acid*
RPKM	*Reads per kilobase million*
TCGA	*The cancer genome atlas program*
TMM	*Trimmed m-means*
TPM	*Transcripts per kilobase million*
tsmiRs	*Tumor supressor miRNA*

Anexo 4: Código completo en R y enlace de acceso

Con el enlace que mostramos a continuación se puede acceder a una carpeta alojada en Google Drive con el script completo en R

https://drive.google.com/drive/folders/1BPh2H2peQEGRGpIE_sd3myPt9MrALsgn?usp=drive_link

Además, también mostramos el script de R empleado para este trabajo.

```
##################################
###   TFM Elena Rodríguez      ###  DESARROLLO DE UN MÓDULO DE ANÁLISIS
TRNSCRIPTÓMICO
###   Máster bioinformática    ###  MECANÍSTICO E INTEGRATIVO
###   UEM                      ###
##################################

# BREVES CONSIDERACIONES PREVIAS

# Los datos proceden los siguientes repositiorios y bases de datos: GDC,
miRTarBase y GTEX

# Realizamos un análisis mecanístico con Hipathia para mostrar las
posibilidades y alcance de esta metodología
# Tras ello, el objetivo es probar una aproximación para estimar el papel
de los miRNA en las rutas metabólicas

# Los pasos a seguir serán: descarga de datos, preparación de matrices,
normalización, preparación hipathia,
# hipathia, creación de tabla miRNA-diana-valor_de_activación, selección
de miRNAs y dianas, ORAs, KD e hipatias


# LIBRERÍAS EMPLEADAS

# if (!require("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
#BiocManager::
install(c("tidyverse","TCGAbiolinks","edgeR","AnnotationDbi","org.Hs.eg.d
b","hipathia","GO.db","multiMiR","readxl","ggridges",
"BiocGenerics","clusterProfiler", "pathview", "ggplot2",
"EnhancedVolcano"))

#descarga de datos
library(TCGAbiolinks)
#normalización y análisis diferencial
library(edgeR)
#traducción genes
library(AnnotationDbi)
library(org.Hs.eg.db)
```

```
library(GO.db)
library(BiocGenerics)
library(dplyr)
#hipathia
library(hipathia)
#lectura formatos
library(readxl) #para leer archivos xlsx de excel
#ORA
library(clusterProfiler)
#Representación
library(pathview)
library(ggplot2)
library(ggridges)
library(EnhancedVolcano)
#edición de dataframes
library(scales)


# DESCARGA DE
DATOS_____________________________________________________________________
_____________________________________________________________________
_____________________


#Hacemos un condicional para que no se descargue los datos cada vez que
corramos el código
raw_gene_matrix_file = "raw_gene_matrix"
raw_mirna_file = "raw_miRNA_matrix"

#Vamos a hacer dos queries, una para los genes y otra para los miRNA


if(!file.exists(raw_gene_matrix_file)) {

  #Creamos query para datos de CONTEOS DE GENES

  query_TCGA_gene <- GDCquery(project = 'TCGA-BRCA',
                              data.category = 'Transcriptome Profiling',
                              data.type = 'Gene Expression
Quantification',
                              access = 'open',
                              experimental.strategy = 'RNA-Seq')

  getResults(query_TCGA_gene)


  #descargamos los datos - GDC download - conteo genes
```

```
  GDCdownload(query_TCGA_gene)


  #preparamos los datos
  tcga_sample_data <- GDCprepare(query_TCGA_gene, summarizedExperiment =
TRUE)
  tcga_gene_data <- tcga_sample_data
  raw_gene_matrix <- assay(tcga_gene_data,'unstranded')

  save(raw_gene_matrix, file = raw_gene_matrix_file)
  }else{
  load(raw_gene_matrix_file)
}

  if(!file.exists(raw_mirna_file)) {
  #Creamos query para datos de CONTEOS DE miRNA

  query_TCGA_miRNA <- GDCquery(project = 'TCGA-BRCA',
                               data.category = 'Transcriptome Profiling',
                               data.type = 'miRNA Expression
Quantification',
                               access = 'open',
                               experimental.strategy = 'miRNA-Seq')

  getResults(query_TCGA_miRNA)


  #descargamos los datos - GDC download - conteo miRNA
  GDCdownload(query_TCGA_miRNA)


  #preparamos los datos de miRNA
  raw_miRNA_matrix <- GDCprepare(query_TCGA_miRNA, summarizedExperiment =
TRUE)
  save(raw_miRNA_matrix, file = raw_mirna_file)
  }else{
  load(raw_mirna_file)
}

  #observamos dimensiones de las matrices de RNA y de miRNA

dim(raw_gene_matrix)
class(raw_gene_matrix)

dim(raw_miRNA_matrix)
class(raw_miRNA_matrix)
```

```
# PREPARAMOS LOS DATOS DESCARGADOS_____________________________________________
_____________________________________________________________________________
_____________________

  #filtramos la matriz de miRNA para quedarnos con los conteos
miRNA_matrix <- select(raw_miRNA_matrix,
matches(c('miRNA_ID','read_count')))

  #eliminamos "read_counts_" de los nombres de las columnas de la matriz
de datos miRNA
filter_miRNA <- sub('^read_count_', '', names(miRNA_matrix))

colnames(miRNA_matrix) <- c(filter_miRNA)

  #colocamos la columna "miRNA_ID" como rownames en la matriz de
miRNA_matrix
miRNA2_matrix <- miRNA_matrix[,-1]
rownames(miRNA2_matrix) <- miRNA_matrix[,1]

save(miRNA2_matrix, file = 'miRNA2_matrix')

 #ordenamos las columnas de las matrices
gene_matrix <- raw_gene_matrix[, c(order(colnames(raw_gene_matrix)))]
miRNA2_matrix <- miRNA2_matrix[, c(order(colnames(miRNA2_matrix)))]

 #modificamos las dos matrices para que tengan las mismas columnas

    #vemos si hay duplicados en las dos matrices
    sum(duplicated(colnames(miRNA2_matrix)))
    sum(duplicated(colnames(gene_matrix)))

    #nos quedamos con el nombre de la muestra TCGA-xx-xxxx
    #(el resto son datos que nos estorban para lo que queremos hacer)
    gene_new_cols <- sub('-...-...-....-..$', '', colnames(gene_matrix))
    miRNA_new_cols <- sub('-...-...-....-..$', '',
colnames(miRNA2_matrix))

    colnames(gene_matrix) <- c(gene_new_cols)
    colnames(miRNA2_matrix) <- c(miRNA_new_cols)

    #generamos un vector con las columnas modificadas que tienen en común
```

```
las dos matrices
    cols_intersection <- intersect(gene_new_cols, miRNA_new_cols)
    length(intersect(gene_new_cols, miRNA_new_cols))

    #filtramos las matrices con el vector común generado para quedarnos
con matrices con nombres de columnas idénticos
    mi_sub <- miRNA2_matrix[, cols_intersection]
    gen_sub <- gene_matrix[, cols_intersection]

    #comprobación de que el orden de las columnas es el deseado

    all(colnames(mi_sub) == colnames(gen_sub))

    #guardamos archivos
    save(mi_sub, file = 'mi_sub')
    save(gen_sub, file = 'gen_sub')



#
NORMALIZACIÓN_____________________________________________________________
_____________________________________________________________________
_____________________________________________

#normalizamos genes por tmm - EdgeR

  #GEN
y<-DGEList(gen_sub)
y<-calcNormFactors(y, method = "TMM")
gen_norm <- cpm(y,log=TRUE,prior.count = 3)

  #guardamos archivos
save(gen_norm, file = 'gen_norm')

#la matriz de miRNA será normalizada previo análisis diferencial después
de trabajar con Hipathia


# PREPARACIÓN
HIPATHIA__________________________________________________________________
_____________________________________________________________________
_____________________________

#Hipathia necesita dos objetos:

    #una matriz de datos normalizados (que ya tenemos)
```

```
    #una matriz de diseno (que será con ER+ y ER-)


#Para ello, necesitamos que el número y las columnas de la matriz de
datos coincidan con el número y las filas de los datos clínicos
#También necesitamos tener las condiciones para el análisis, en este
caso, ER+ y ER-
#Por último, la matriz de datos necesita tener los nombres de filas en
formato entrezid


#ADECUAMOS MATRICES

  #cargamos la matriz de datos clínicos
  clinicalh<- read.table("clinical_info_TCGA-BRCA.tsv", sep = "\t",
na.strings = "NAN")
  dim(clinicalh)

  #Comprobamos que hay muestras coincidentes entre ambos data.frames
  sum(colnames(gen_norm)%in%clinicalh$bcr_patient_barcode)
  length(colnames(gen_norm)%in%clinicalh$bcr_patient_barcode)

  #Comprobamos si todas est?n en el mismo orden - aún aquí no lo están
  all(colnames(gen_norm)==clinicalh$bcr_patient_barcode)

  #comprobar siempre si hay duplicados, vamos a ver cuantos hay
  sum(duplicated(clinicalh$bcr_patient_barcode))

  #Vamos a ver los duplicados
  clinicalh[duplicated(clinicalh$bcr_patient_barcode),c(1:3)]

  #Sacamos los nombres coincidentes
  sanames<-intersect(colnames(gen_norm),clinicalh$bcr_patient_barcode)

  #Seleccionamos las columnas de la matriz por nombre,
  #en este caso el vector de nombres que hemos generado, sanames
  gen_norm1<-gen_norm[,sanames]

  #Eliminamos los duplicados con el operador inverso ! y el vector lógico
de duplicados
  clinicalh<-clinicalh[!duplicated(clinicalh$bcr_patient_barcode),]

  #Cambiamos los nombres de fila por los nombres de muestra
  rownames(clinicalh)<-clinicalh$bcr_patient_barcode

  #Seleccionamos por nombre del vector snames las filas del data.frame
```

```
  clinicalh<-clinicalh[sanames,]

  #Comprobamos el orden de todas las muestras
  all(rownames(clinicalh)==colnames(gen_norm1))

  #Comprobamos dimensiones
  dim(clinicalh)
  dim(gen_norm1)

  #guardamos archivos
  save(clinicalh, file = 'clinicalh')
  save(gen_norm1, file = 'gen_norm1')


#ER+ ER-

  #A partir de clinicalh hacemos una matriz con las filas formato tcga y
columna con estado ER, como un dataframe
  clinicalh_df <- as.data.frame(clinicalh)
  design2 <- clinicalh_df[c("er_status_by_ihc")]

  #guardamos archivos
  save(design2, file = 'design2')



#TRADUCCIÓN DE ENSEMBLE A ENTREZ ID

  #Hipathia necesita que las filas estén en formato ENTREZ.ID y las
nuestras están en ENSEMBL

  #miramos los formatos que acepta, keytypes como entrada y columns como
salida
  keytypes(org.Hs.eg.db)
  columns(org.Hs.eg.db)

  #quitamos los decimales de los nombres de columnas de la matriz
gen_norm1 (que ha pasado por normalización y design)
  genrows <- rownames(gen_norm1)
  gen_sub2 <- sub('\\.[0-9]*$', '', genrows)
  rownames(gen_norm1) <- c(gen_sub2)

  gen_sub_rown <- rownames(gen_norm1)

  #creamos una tabla con 64070 filas de ensembl y entrezid
  gen_hi <- AnnotationDbi::select(org.Hs.eg.db,
```

```
keys=gen_sub_rown,columns="ENTREZID", keytype="ENSEMBL")

  #ponemos la columna que nos interesa al principio de nuestra matriz
gen_norm1
  gen_norm2 <- cbind(gen_hi$ENTREZID,gen_norm1)
  #cambiamos los nombres de fila por los de la columna con entrezid
  #aparece una x delante de los nombres, pero no es relevante
  gen_norm3 <- gen_norm2[,-1]
  rownames(gen_norm3) <- gen_norm2[,1]


#
HIPATHIA_____________________________________________________________________
_____________________________________________________________________
_____________________________________


  #nos aseguramos de que Hipathia pueda leer los conteos normalizados
como cifras
  num_gen_norm3 <- matrix(as.numeric(gen_norm3), ncol=ncol(gen_norm3))
  rownames(num_gen_norm3) <- rownames(gen_norm3)
  colnames(num_gen_norm3) <- colnames(gen_norm3)

  design2[design2=="[Not Evaluated]"] <- "NotEvaluated"

  #generamos el objeto summarizedExperiment que necesita hipathia para
hacer su función
  ER_sumexp <- SummarizedExperiment(assays=SimpleList(raw=num_gen_norm3),
                                    colData=design2)
  save(ER_sumexp, file = 'ER_sumexp')

  #cargamos rutas de humanos
  pathways <- load_pathways(species = "hsa")

  #hipathia hace lo suyo
  hidataER <- hipathia(ER_sumexp, pathways, uni.terms = TRUE, GO.terms =
TRUE,
                       decompose = FALSE, verbose=TRUE)
  save(hidataER, file = 'hidataER')

  #DAcomp
  DadataER <- DAcomp(hidataER, design2$er_status_by_ihc, "Positive",
"Negative")
  save(DadataER, file = 'dadataER')

  #Vemos los datos
  DAoverview(DadataER)
```

```
  DAsummary(DadataER)
  DAtop(DadataER)
  #Obtenemos un report
  ERreport <- DAreport(DadataER,pathways)
  visualize_report(ERreport)
  #Guardamos los valores de activación de los circuitos para cada muestra
  path_vals <- get_paths_data(DadataER, matrix = TRUE)
  save(path_vals, file = 'path_vals')

  #Con esto, separamos las rutas según estén reguladas
  upaths <- path_vals[path_vals$`UP/DOWN`== "UP",]
  dpaths <- path_vals[path_vals$`UP/DOWN`== "DOWN",]



# ANÁLISIS DIFERENCIAL DE
miRNA_____________________________________________________________________
_____________________________________________________________________
___________

  #Para el análisis diferencial, seguimos los pasos indicados por el
paquete de edgeR (salvo exploración/representación)
  #el análisis diferencial es de las muestras hormonopositivas vs
negativas de miRNA
  #estos son: filtrado y normalización, matriz de diseño, estimación de
la dispersión y expresión diferencial (ya tenemos las matrices de
partida)


    #matriz de diseño

      #Exploramos la variable
      table(clinicalh$er_status_by_ihc)
      #Seleccionamos los casos que nos interesan, positivo y negativo, de
ER

ERpn<-clinicalh[clinicalh$er_status_by_ihc=="Negative"|clinicalh$er_statu
s_by_ihc=="Positive",]
      head(ERpn)
      levels(ERpn)
      #Comprobamos si nos queda algun valor perdido
      sum(is.na(clinicalh$er_status_by_ihc)) #cero
      #la variable principal del diseño tiene que ser un factor
(ERpos_neg)
      ERpnfactor<-factor(clinicalh$er_status_by_ihc,levels =
c("Positive","Negative"))
```

```
      mi_norm_fac <-mi_sub[,rownames(ERpn)]
      all(rownames(ERpn)==colnames(mi_norm_fac))

      #designER<-model.matrix(~ERpn)
      designER <- model.matrix(~ ERpnfactor, data = clinicalh)

      #filtrado y normalización, es importante calcular los factores
antes del análisis (o no habrá factores de dispersión necesarios)
      t<-DGEList(mi_norm_fac)
      t<-calcNormFactors(t)

    #análisis diferencial

      #Nos quedamos con aquellas muestras con un número de conteos mayor
a 3 en almenos 3 muestras
      #Este criterio puede modificarse dependiendo de los datos, debe
hacerse siempre sobre la matriz de conteos
      #Usamos el método de edgeR, filterByExpr, que permite la seleccion
por grupos
      keep<-rowSums(t$counts>3)> 3
      table(keep)
      keep<-rowSums(mi_norm_fac>3)>3
      sum(!keep)
      keep<-filterByExpr(t, designER)
      sum(!keep)
      t1<-t[keep,,keep.lib.sizes=TRUE]

      #Comprobamos el numero de genes que nos quedamos
      str(t1)

      #Calculamos la dispersion
      t1<-estimateDisp(t1)


      #Ajustamos el modelo, en este caso un modelo linear generalizado
con test QL
      fit<-glmQLFit(t1,designER)
      qlf <- glmQLFTest(fit,coef = "ERpnfactorNegative") #Tenemos que
indicar el nombre de la columna que queremos usar como variable
principal,
                                                  #en este caso
ERNegative
                                                  #qlf <-
glmQLFTest(fit) si no lo definimos, el modelo evalua cada uno de los
posibles contrastes
                                                  #de manera similar a
```

```
un ANOVA

      #Sacamos los resultados del estadistico, aquellos genes mas
importantes para el modelo, mas significativos
      pvalues<-topTags(qlf, n= Inf,
adjust.method="BH",sort.by="PValue",p.value=1)

      #Por comodidad lo convertimos a un data.frame
      p_values<-as.data.frame(pvalues)
      alpha<-0.05
      sig<-p_values[p_values$FDR<alpha,]

      #sacamos la lista de miRNA significativos
      list_miRNA<-rownames(sig)
      length(list_miRNA)

      #guardamos la lista de miRNA significativos en forma de dataframe
      sig_miRNA <- as.data.frame(list_miRNA, stringAsFactors=FALSE)
      save(sig_miRNA, file = 'sig_miRNA')

      #Obtención de los miRNA upregulados y reprimidos
      #Creamos una nueva variable con los genes expresados
significativamente y ordenados de forma decreciente.
      mi_row<-rownames(sig)
      mi_Exp<-sig$logFC
      names(mi_Exp)<-mi_row
      mi_Exp<-sort(mi_Exp,decreasing = T)
      ud_miRNA <- as.data.frame(mi_Exp)

      #Usamos la funcion del FoldChange para obtener los upregulados y
reprimidos
      upmiRNA<-rownames(sig[sig$logFC>0,])
      downmiRNA<-rownames(sig[sig$logFC<0,])
      length(upmiRNA)
      length(downmiRNA)

      #con esto guardamos los miRNA según estén regulados
      dmiRNA <- as.data.frame(downmiRNA)
      umiRNA <- as.data.frame(upmiRNA)
      save(dmiRNA, file = 'dmiRNA')
      save(umiRNA, file = 'umiRNA')

      #Obtención de los 20 genes mas significativos
      #Usamos la funcion head que nos dara los 20 primeros miRNA
      head(mi_Exp, 20)
      head(umiRNA, 20)
```

```
      head(dmiRNA, 20)

      #Representamos los resultados
       # Smear
       plotSmear(qlf, de.tags=rownames(sig))
       abline(h=c(-1,1), col="blue")

       # Histograma pvalues
       hist(p_values$PValue, breaks=50, col="grey", border="white")

      pdf("volcano-plot.pdf")
      EnhancedVolcano(sig, x='logFC', y='PValue', lab=rownames(sig))
      dev.off()

#HACEMOS UNA TABLA QUE RELACIONE
miRNA-FC-GEN_DIANA_____________________________________________________
_____________________________________________________________________
  #
    #las fuentes son la tabla resultado del análisis diferencial (sig) y
el dataframe de miRTarBase

      #Abrimos el archivo de la báse de datos miRTarBase, el
correspondiente de la especie humana
      hsa_MTI <- read_excel("hsa_MTI.xlsx")

      #modificamos la columna de miRNA para que pase de ser
hsa-miR-134-3p o hsa-miR-13-3p a hsa-mir-134 y hsa-mir-13
      col_hsa <- sub('R','r', hsa_MTI$miRNA)
      col_hsa <- sub('-3p$','', col_hsa)
      col_hsa <- sub('-5p$','', col_hsa)   #Hacemos esto dado que los
miRNA obtenidos en GDC no son maduros, asumiendo la pérdida de
información consecuente

      hsa_MTI$miRNA <- col_hsa

      #extraemos las columnas de interés y hacemos una nueva tabla con
ellas
      targetmi <- hsa_MTI[,c(2,4,5)]

      #eliminamos duplicados

      small_targetmi <- targetmi%>%distinct(miRNA,`Target Gene`,`Target
Gene (Entrez ID)`, .keep_all = TRUE)

      #filtramos la tabla con los genes diana con los miRNA
      sig$miRNA <- rownames(sig)
```

```
      mi_fc_target <- merge(sig, small_targetmi, by = "miRNA")

     #preparamos la columna de FC

      # Calculamos el FC no logarítmico, y lo escalamos entre 0 y 1
      # Esto hace que un miRNA downregulado tienda a 0, un upregulado
tienda a 1
      mi_fc_target <- mi_fc_target %>% mutate(FC = 2^logFC)
      mi_fc_target <- mi_fc_target %>% mutate(FC_scaled = rescale(FC, to
= c(0, 1)))
      mi_fc_target <- select(mi_fc_target, 1, 2, 9, 10, 3, 4, 5, 6, 7, 8)
      mi_fc_target2 <- mi_fc_target[,c(1,4,9,10)]

        # Selección de dianas en base a mi_fc_target
      mi_fc_target_counts <- mi_fc_target %>% group_by(miRNA) %>%
summarise(count = n())


      # SELECCIÓN DE miRNA Y
DIANAS_____________________________________________________________
_____________________________________________________________

      #vemos los miRNA desregulados más significativos
      head(umiRNA, 3)
      head(dmiRNA, 3)

      #seleccionamos sus dianas
      #Creamos un vector con los 6 miRNA más desregulados, up y down)
      top6_miRNA <- c("hsa-mir-934", "hsa-mir-18a", "hsa-mir-577",
"hsa-mir-190b", "hsa-mir-342", "hsa-mir-449a")
      desmiRNA <- as.data.frame(top6_miRNA)

      #Hacemos un bucle para seleccionar las dianas de cada uno, y su
factor de silenciamiento

      #Creamos una lista para almacenar los resultados
      results_list <- vector("list", length(top6_miRNA))

      for (gen_rowname in rownames(desmiRNA)) {
        print(gen_rowname)
        gen_to_search <- desmiRNA[gen_rowname,]
        print(gen_to_search)
        hsa_gene <- (mi_fc_target %>% filter(miRNA == gen_to_search))
        hsa_gene_list <- hsa_gene$`Target Gene`
        uniquegen <- unique(hsa_gene_list)
        # Factor de silenciamiento: es el mismo para todos los genes del
```

```
miRNA
        mirna_fc <- hsa_gene$FC_scaled[1]

        #Almacenamos los resultados en la lista
        results_list[[as.integer(gen_rowname)]] <- list(gen_to_search,
length(uniquegen), uniquegen, mirna_fc)
      }
      #
      #Convertimos la lista de resultados en una matriz de listas
      pathcounts <- do.call(rbind, results_list)

      #Extraemos vectores para cada miRNA
      gt_mir_934 <- unlist(pathcounts[1,3])
      gt_mir_18a <- unlist(pathcounts[2,3])
      gt_mir_577 <- unlist(pathcounts[3,3])
      gt_mir_190b <- unlist(pathcounts[4,3])
      gt_mir_342 <- unlist(pathcounts[5,3])
      gt_mir_449 <- unlist(pathcounts[6,3])



# MATRIZ DE DATOS DE MAMA NORMALES DE
GTEX_____________________________________________________________________
_____________________________________________
      gct_file <-
read.table("gene_reads_2017-06-05_v8_breast_mammary_tissue.gct.gz", skip
= 2, header = TRUE)

      #Preparamos la matriz para ser normalizada
      gct <- gct_file[,-1]
      rownames(gct) <- gct$Name
      gct_gen_description <- gct_file[,c(2,3)] #por si necesito la
descripción en algún momento
      gct[,c('Name', 'Description')] <- list(NULL)

      #Normalizamos por TMM
      q<-DGEList(gct)
      q<-calcNormFactors(q, method = "TMM")
      gct_norm <- cpm(q,log=TRUE,prior.count = 3)

      save(gct_norm, file = 'gct_norm') #guardamos archivos

      #Revisamos que los genes de Gtex estén en hipathia
      list_gens_hipathia <- as.data.frame(pathways[["all.genes"]]) #con
esto tenemos los genes
```

```
      #Quitamos el decimal de los rownames de gct_norm y
gct_gen_description
      gct_new_rownames <- sub('..$', '', rownames(gct_norm))
      rownames(gct_norm) <- c(gct_new_rownames)

      gct_gen_description$Name <- gct_new_rownames #quitamos el decimal
también en este data.frame

      #Ponemos la columna de ensembl de gct_gen_description en gct_norm
      gct_norm2 <- cbind(gct_gen_description$Name,gct_norm)

      #Traducimos a entrez id los ensembl de gct
      gct_eid <- AnnotationDbi::select(org.Hs.eg.db,
keys=gct_new_rownames,columns="ENTREZID", keytype="ENSEMBL")
      gct_eid2 <- gct_eid %>% na.omit() #con esto eliminamos las filas
con NA. Pasamos de 56200 obs. a 25614 obs.

      #para filtrar la matriz de datos gct vamos a ver primero cuáles
están incluidos en hipathia y luego vamos a filtrar nuestra matriz con el
resultado
      hipathia_entrezid <- list_gens_hipathia$`pathways[["all.genes"]]`
#vector de genes de hipathia

      gct_eid3 <- filter(gct_eid2, gct_eid2$ENTREZID %in%
c(hipathia_entrezid)) #se filtra con el vector de genes de hipathia

      gct_hipathia <- c(gct_eid3$ENSEMBL) #vector de genes

      matching_rows <- which(gct_norm2[,1] %in% gct_hipathia) #este es un
vector con los valores en común de gct_norm2 y gct_hipathia

      gct_norm3_hip <- gct_norm2[matching_rows, ]

      #comprobamos que los genes dianas están en hipathia

        #con esto traducimos entrezid a symbol, que es el formato que
presentan las dianas
        list_gens_hipathia2 <- AnnotationDbi::select(org.Hs.eg.db,
keys=hipathia_entrezid,columns="SYMBOL", keytype="ENTREZID")
        symbol_hipathia <- c(list_gens_hipathia2$SYMBOL)

        #con setdiff comprobamos los valores que pueden no estar
incluidos en symbol_hipathia (1o el vector diana -> el valor que puede
aparecer es el que falta en el segundo vector)
        setdiff(gt_mir_342, symbol_hipathia)
        setdiff(gt_mir_18a, symbol_hipathia) # CTGF
```

```
        setdiff(gt_mir_190b, symbol_hipathia)
        setdiff(gt_mir_342, symbol_hipathia)
        setdiff(gt_mir_449, symbol_hipathia)
        setdiff(gt_mir_577, symbol_hipathia) # CTGF, G6PC

        #gt_mir_18a y gt_mir_577 tienen genes que no tiene hipathia. Los
eliminamos de los vectores

        gt_mir_18a2 <- gt_mir_18a[! gt_mir_18a %in% c('CTGF')]
        gt_mir_5772 <- gt_mir_577[! gt_mir_577 %in% c('CTGF', 'G6PC')]

     #Ahora ya tenemos la matriz de conteos de gct normalizada por TMM y
filtrada por los genes que controla Hipathia
     #También hemos cotejado los vectores de los genes diana con los
genes que controla Hipathia



# ORAs e
Hipathias_____________________________________________________________
_____________________________________________________________
_____________________

        #Preparaciones previas

        #Traducimos vectores de dianas y matriz de conteos
(gct_norm3_hip) a entrez id para realizar los ORA e Hipathia

        #vectores de dianas
        gt_vectors <- list(gt_mir_934, gt_mir_18a2, gt_mir_190b,
gt_mir_342, gt_mir_449, gt_mir_5772)
        gt_eid_list <- list()

        for (key_vector in gt_vectors) {
          result <- AnnotationDbi::select(org.Hs.eg.db, keys =
key_vector, columns = "ENTREZID", keytype = "SYMBOL")
          gt_eid_list <- append(gt_eid_list, list(result$ENTREZID))
        }

        #matriz de conteos
        gct_n3h <- gct_norm3_hip[,1]
        gtc_n3h_eid <- AnnotationDbi::select(org.Hs.eg.db, keys =
gct_n3h, columns= "ENTREZID", keytype="ENSEMBL")
        gct_norm3h_eid <- cbind(gtc_n3h_eid$ENTREZID, gct_norm3_hip)

        gct_eid_vector <- gct_norm3h_eid[,1]
```

```
      #ORA
        ora_vectors <- gt_eid_list
        ora_results <- list()

        pdf("ora-plots.pdf")
        for (i in 1:length(ora_vectors)) {
          print(i)
          target_genes <- unlist(ora_vectors[i])
          miRNA_name <- top6_miRNA[i]
          result_ora <- enrichKEGG(target_genes, organism = "hsa",
keyType = "kegg", pvalueCutoff = 1, qvalueCutoff=1,  pAdjustMethod =
"BH", gct_eid_vector)

          print(dotplot(result_ora, title=miRNA_name))
          print(barplot(result_ora, title=miRNA_name))
          ora_results <- append(ora_results, list(result_ora))
        }
        dev.off()

      #Hipathia y knockdowns

        #cargamos rutas de humanos
        pathways <- load_pathways(species = "hsa")

        #definimos la función que va a ejecutar el knockdown: replicando
los datos de entrada sanos
        #con muestras acabadas en _KD, así como creamos las matrices de
diseño
        autoKD<-function(x,genes,kd_factor) {
          c<-x
          colnames(c)<-paste(colnames(x),"KD",sep="_")
          c[rownames(c)%in%genes,]<-c[rownames(c)%in%genes,]*kd_factor
          r<-cbind(x,c)

d<-data.frame("type"=c(rep("normal",times=ncol(x)),rep("kd",times=ncol(x)
)),stringsAsFactors = F,row.names = c(colnames(x),colnames(c)))
          #return(list("design"=d,"expMatrix"=r))
          return(SummarizedExperiment(assays=SimpleList(raw=r),
                                      colData=d))
        }


        gen_kd_data<-function(kd_candidate) {
          # Para cada muestra de miRNA, corremos la función de knockdown
para generar las matrices de diseño y genes correspondientes.
```

```
          #buscamos los entrezids correspondientes a los Symbol obtenidos
de HSA
          kd_genes <- AnnotationDbi::select(org.Hs.eg.db,
keys=unlist(kd_candidate[3]), columns="ENTREZID",
keytype="SYMBOL")$ENTREZID
          kd_factor <- unlist(kd_candidate[4])

          #adaptamos la columna de genes normlizada para que los rownames
sean el entrezid
          gct_genes_to_kd <- gct_norm3h_eid
          rownames(gct_genes_to_kd) = NULL
          rownames(gct_genes_to_kd) <- gct_genes_to_kd[,1]
          gct_genes_to_kd <- gct_genes_to_kd[,-c(1,2)]

          gct_genes_to_kd_numeric <- matrix(as.numeric(gct_genes_to_kd),
ncol=ncol(gct_genes_to_kd))
          rownames(gct_genes_to_kd_numeric) <- rownames(gct_genes_to_kd)
          colnames(gct_genes_to_kd_numeric) <- colnames(gct_genes_to_kd)
          return(list(miRNA=unlist(kd_candidate[1]),
design=autoKD(gct_genes_to_kd_numeric, kd_genes, kd_factor)))
        }

        #kd_data_design <- gen_kd_data(kd_577)
        kd_data_designs <- apply(as.data.frame(pathcounts), 1,
gen_kd_data)

        run_hipathia<-function(row) {
          row<-unlist(row)
          miRNAname <- row$miRNA
          print(row$design)
          print(paste("running hipathia for miRNA: ", miRNAname))
          hidataERKD <- hipathia(row$design, pathways, uni.terms = TRUE,
GO.terms = TRUE,
                         decompose = FALSE, verbose=TRUE)
          save(hidataERKD, file = paste0('hidataERKD-', miRNAname))

          return(c(miRNA=miRNAname, design=row$design,
hidata=hidataERKD))
        }

        results <- lapply(kd_data_designs, run_hipathia)

        genDacomp<-function(row){
          dadata<-DAcomp(row$hidata, colData(row$design)$type, "kd",
"normal")
          path_vals <- get_paths_data(dadata, matrix = TRUE)
```

```
    save(path_vals, file = paste("path-vals", row$miRNA, sep="-"))
    return(list(miRNA=row$miRNA, design=row$design,
hidata=row$hidata, dacomp=dadata, path_vals=path_vals))
  }
  resultsDacomp <- lapply(results,genDacomp)


  genReports<-function(row){
    dirName<-paste("report", row$miRNA, sep="-")
    Dadata<-row$dacomp

    #dir.create(dirName)
    #DAreport(Dadata,pathways,output_folder=dirName,path="./")
    # Open pdf file
    pdf(file=paste(dirName, "pdf", sep=".") )
    # create a 2X2 grid
    par( mfrow= c(2,2) )
    DAoverview(Dadata)
    DAsummary(Dadata)
    # DAtop(Dadata)
    dev.off()
  }

  run_path_vals<-function(row) {
    row<-unlist(row)
    path_vals_mirna <- get_paths_data(DadataER, matrix = TRUE)
    save(path_vals, file = paste0('pathvals-', miRNA_name))
    return(path_vals_mirna)
  }

  path_vals_mirna<- lapply(results, run_path_vals)



##############################
###                        ###
###         FIN            ###
###                        ###
##############################
```