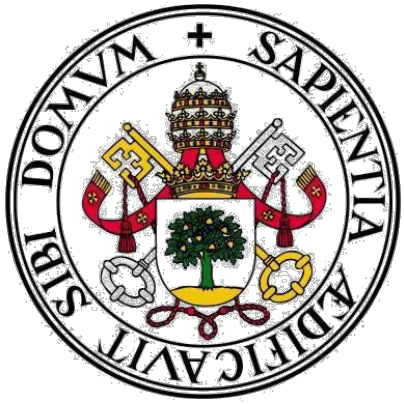

UNIVERSIDAD DE VALLADOLID

FACULTAD DE MEDICINA
ESCUELA DE INGENIERÍAS INDUSTRIALES

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA BIOMÉDICA

Análisis de datos de expresión en retina para la identificación de biomarcadores de DMAE: posible herramienta diagnóstica

Autor: D. David Blázquez García

Tutoras:

Dra. D. ª Itziar Fernández Martínez
Dra. D. ª Marita Hernández Garrido

TÍTULO:	Análisis de datos de expresión en retina para la identificación de biomarcadores de DMAE: posible herramienta diagnóstica
AUTOR:	D. David Blázquez García
TUTORAS:	Dra. D. ª Itziar Fernández Martínez Dra. D. ª Marita Hernández Garrido
DEPARTAMENTOS:	Estadística e Investigación Operativa Bioquímica y Biología Molecular y Fisiología

---

TRIBUNAL

PRESIDENTA:	Dra. D. ª Pilar Cidad Velasco
SECRETARIA:	Dra. D. ª Mercedes Durán Domínguez
VOCAL:	Dr. D. Iván Fernández Bueno
SUPLENTE 1:	Dra. D. ª Itziar Fernández Martínez
SUPLENTE 2:	Dra. D. ª Marita Hernández Garrido

FECHA	25/05/2023
CALIFICACIÓN:	

Agradecimientos

Quisiera aprovechar este espacio para expresar mi más sincero agradecimiento a las personas que han hecho posible que pueda realizar este Trabajo Fin de Grado.

En primer lugar, a mi familia, mis padres Miguel y Gema, y mi hermana Sofía. Su apoyo constante y sacrificios han sido fundamentales para llevarme hasta este momento. Gracias por creer siempre en mí. A mis amigas, Zaira, Laura, Irene, Silvia, Carol y Henar, les agradezco de todo corazón por estar a mi lado durante todo el camino. Gracias por escucharme, acompañarme en los peores momentos, celebrar los mejores y ayudarme en todo lo posible y más. Sin ellas no sé cómo habría llegado hasta aquí. Finalmente, también quiero expresar mi gratitud hacia mis tutoras, Marita e Itziar. Gracias por su orientación, sus conocimientos y su dedicación, no solamente sobre la elaboración de este trabajo, sino también sobre mi carrera profesional.

De nuevo, muchas gracias. Vuestra presencia ha sido fundamental para lograr este hito tan importante en mi vida.

Resumen

La Degeneración Macular Asociada a la Edad (DMAE) es una patología que se caracteriza por la aparición de alteraciones degenerativas y progresivas en el área central de la retina, la mácula, que conducen a un deterioro visual progresivo hasta la pérdida total de la visión. De hecho, esta enfermedad es la principal causa de ceguera legal en los países desarrollados en personas mayores de 55 años. Es por tanto muy importante poder realizar un diagnóstico temprano de la enfermedad para poner en marcha tratamientos que conserven la visión el máximo tiempo posible y evitar la progresión de la patología hacia los estadios más avanzados. De este modo, se plantea la posibilidad de estudiar la patología atendiendo a la información contenida en nuestros genes ya que, afortunadamente, el acceso a la información génica es a día de hoy una realidad gracias a técnicas de secuenciación de alto rendimiento, destacando sin duda entre todas el RNA-seq. De aquí surge la posibilidad de relacionar la expresión génica de una persona con su predisposición a enfermar. De este modo, el objetivo principal de este trabajo será la identificación de aquellos genes diferencialmente expresados en los pacientes de DMAE, con el fin de poder ayudar al diagnóstico de la enfermedad, se emplearán los resultados del análisis de expresión diferencial para el desarrollo de una aplicación web que permitiría clasificar a un individuo en un grupo de riesgo a partir de su perfil de expresión.

Palabras clave

Aplicación Web – Biomarcador – DMAE – Expresión diferencial – RNA-seq

Abstract

Age-related macular degeneration (AMD) is a pathology characterized by the appearance of progressive degenerative changes in the central area of the retina, the macula, leading to progressive visual impairment until total loss of vision. In fact, this disease is the main cause of legal blindness in developed countries in population aged 55 and over. It is therefore very important to be able to make an early diagnosis of the disease in order to implement treatments that preserve vision as long as possible and prevent the progression of the pathology to more advanced stages. This raises the possibility of studying the pathology by taking into account the information contained in our genes since, fortunately, access to genetic information is now a reality thanks to high-throughput sequencing techniques, with RNA-seq undoubtedly standing out among all of them. From this arises the possibility of relating a person's gene expression to his or her stage of the disease. Thus, the main objective of this work is the identification of those genes differentially expressed in AMD patients, in order to help in the diagnosis of the disease. To this end, the results of the differential expression analysis will be used for the development of a web application that would allow classifying an individual into a risk group based on his or her expression profile.

Keywords

AMD – Biomarker – Differential expression – RNA-seq – Web application

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

PARTE 1 INTRODUCCIÓN

Capítulo 1 - Motivación del estudio

Capítulo 2 - Degeneración Macular Asociada a la Edad
2.1 - Etiología
2.2 - Epidemiología
2.3 - Fisiopatología de la DMAE
2.4 - Técnicas de diagnóstico actuales
2.5 - Estrategias terapéuticas actuales

Capítulo 3 - Caracterización de la expresión génica
3.1 - Conceptos de expresión génica en el ser humano
3.2 - Síntesis de ARN
3.3 - Secuenciación de ARN (RNA-seq)
3.4 - Estudio de asociación del genoma completo (GWAS)
3.5 - Análisis de polimorfismos de nucleótido único (SNP)

Capítulo 4 - Revisión de estado sobre aplicaciones web para el diagnóstico precoz de la DMAE

Capítulo 5 - Hipótesis

Capítulo 6 - Objetivos

Capítulo 1 – Motivación del estudio

La patología de estudio de este trabajo es la Degeneración Macular Asociada a la Edad (DMAE), una enfermedad degenerativa progresiva que afecta al área central de la retina. La DMAE constituye la principal causa de ceguera en los países desarrollados en personas a partir de cincuenta años [1]. Aunque se debe indicar que no existe un consenso respecto a esta edad mínima, sí ha sido demostrada la relación de la DMAE con la edad, siendo más probable su aparición cuanto más avanzada es la edad del paciente. Debido al envejecimiento de la población mundial, el impacto de esta enfermedad será previsiblemente muy grande. De hecho, es la tercera causa principal de discapacidad visual a nivel mundial, con una prevalencia de 170 millones de pacientes [2], lo que la convierte en un importante problema de salud pública.

Desde el punto de vista genético, la DMAE es una enfermedad multifactorial o compleja, es decir que no se espera que se produzca por un único gen, sino por la contribución de múltiples genes y de diversos factores ambientales.

Se puede distinguir claramente en dos estadios: precoz y avanzada. Este último (DMAE avanzada) se establece principalmente en dos formas o tipos de DMAE: seca o atrófica, que es menos grave y más común; y húmeda, también llamada exudativa, neovascular o hemorrágica. Si bien es cierto que se conoce la existencia de etapas intermedias de la patología, su comprensión es aún limitada y la barrera con los estadios precoz y/o avanzado muy difusa.

En la actualidad, el diagnóstico y estadificación de la DMAE se basan en el examen clínico y en técnicas de imagen como la tomografía de coherencia óptica y la fotografía del fondo de ojo [3]. Sin embargo, como la etiología de la DMAE no se conoce en su totalidad, su desempeño es limitado, precisamente este hecho es lo que la convierte en un desafío que requiere el empleo de herramientas diferentes a las ordinarias para su abordaje. Si bien su aparición sí podría ser explicada en base a variaciones genéticas ya que, como se demostrará a lo largo de este trabajo, los pacientes de DMAE presentan una serie de biomarcadores diferenciales y significativos. Es decir, se requiere encontrar biomarcadores específicos que puedan ayudar en el diagnóstico y la estadificación de la DMAE. Los biomarcadores son indicadores mensurables de procesos biológicos o estados de enfermedad que pueden detectarse en muestras biológicas, como sangre o tejidos (e.g. el nivel de expresión de un gen) [2].

El objetivo principal de este trabajo es identificar biomarcadores específicos de la DMAE que puedan ayudar en el diagnóstico y clasificación por estadios de la enfermedad. Para lograrlo se propone, en primer lugar, utilizar datos de RNA-seq para obtener genes diferencialmente expresados relacionados con la DMAE. RNA-seq es una potente herramienta que permite detectar los niveles de expresión génica en una muestra. Comparando los niveles de expresión génica entre pacientes con DMAE y controles sanos, se puede identificar genes que se expresan diferencialmente en pacientes con DMAE. En segundo lugar, como los datos de RNA-seq son complejos y requieren un procesamiento y análisis cuidadosos para extraer información significativa, se desarrollará un protocolo que garantice la precisión y

reproducibilidad de los resultados. En tercer lugar, se identificarán los perfiles de expresión génica que caracterizan al paciente con DMAE en cada una de sus etapas. De esta manera, se podrá desarrollar una comprensión más completa de los mecanismos biológicos subyacentes de la DMAE. Así se justifica la necesidad de recurrir a la bioinformática para abordar este problema. Además, se propone la implementación de los resultados obtenidos en una aplicación web desarrollada a través del paquete de R `Shiny` [4], con el fin último de facilitar el acceso y comprensión del personal sanitario a este tipo de información fundamentada en cuestiones estadísticas, más alejada de la clínica habitual, pero que puede aportar un gran valor en el diagnóstico y estudio de la DMAE.

En conclusión, este trabajo pretende identificar biomarcadores específicos de la DMAE que puedan ayudar en el diagnóstico y la estadificación de la enfermedad. Mediante el uso de datos de RNA-seq, se pretende identificar genes diferencialmente expresados relacionados con la DMAE y desarrollar un protocolo para procesar y analizar este tipo de datos. Además, se describirán los perfiles de expresión génica que caracterizan al paciente con DMAE en cada uno de sus estadios y se desarrollará una aplicación web que proporcione una interfaz sencilla para que los profesionales sanitarios puedan acceder a los resultados del trabajo. Estos hallazgos tienen el potencial de mejorar el diagnóstico y el tratamiento de la DMAE, lo que conducirá a mejores resultados para los pacientes.

Capítulo 2 - Degeneración Macular Asociada a la Edad

2.1 - Etiología

A medida que aumenta la esperanza de vida, las enfermedades degenerativas como la DMAE constituyen una de las principales preocupaciones en la sociedad actual, ya que las complicaciones asociadas a las mismas, como es la pérdida de agudeza visual en el caso de la DMAE, comprometen altamente la calidad de vida. Como el propio nombre de la patología indica, la prevalencia de la DMAE aumenta con la edad, de hecho, es el principal factor de riesgo con el que se puede asociar.

La DMAE es una enfermedad degenerativa progresiva que afecta el área más importante de la retina, la mácula lútea: la parte central de la retina donde se forman, procesan y envían las imágenes que capta el ojo hacia el cerebro. Para intentar comprenderla se debe hablar antes brevemente de la estructura del ojo humano, la cual queda resumida en la Figura 1, donde se puede apreciar una sección horizontal de un ojo derecho visto desde arriba. Se deben destacar tres capas o túnicas [5]:

- Externa: formada por la esclera en la parte posterior, una capa fibrosa opaca que da un soporte estructural, y la córnea en la parte anterior, ésta es transparente para permitir la entrada de la luz al ojo, de hecho, es la principal estructura refractiva del ojo. La función esencial de esta capa es la protección de la estructura ocular frente a patógenos y agresiones externas, así como también el mantenimiento de la forma del globo ocular tras las contracciones de la musculatura ocular o variaciones de la presión intraocular.

- Media: se dispone entre la retina y la esclera, es la capa vascular y pigmentada del ojo. Constituida por la úvea que, a su vez, se distingue en úvea anterior y posterior. Su parte anterior está formada por dos elementos, el iris, un disco circular contráctil cuya función es análoga al diafragma de una cámara fotográfica (regula la cantidad de luz que pasa al interior del ojo); y el cuerpo ciliar, estructura encargada de producir y drenar el contenido que rodea las estructuras del segmento anterior del ojo: el humor acuoso, este fluido permite la nutrición de estas estructuras y mantiene la presión intraocular. La parte posterior de la úvea está constituida por la coroides, una capa delgada de tejido conectivo altamente vascularizado y pigmentado que está formada por cinco capas en la que se distribuyen los distintos vasos sanguíneos que irrigan el ojo: supracoroides, capa de grandes vasos, capa de vasos medianos, coriocapilar y su lámina basal, denominada membrana de Bruch. La coroides da soporte nutricional de la retina y, por ende, también a la mácula, por lo que está íntimamente relacionada con la DMAE, asimismo también juega un papel importante en la función inmune, la regulación de la presión intraocular y la termorregulación del ojo.

- Interna: solo presente en el segmento posterior del ojo, está formada por la retina, parte del sistema nervioso central, encargada de la fotorrecepción de la luz que entra a través del ojo y su transformación en energía eléctrica (impulsos

nerviosos) a través de los fotorreceptores de la retina, que a su vez está conectada con el cerebro por el nervio óptico. Se encuentran dos tipos de fotorreceptores en la retina: conos, responsables de la visión central, están especializados para trabajar con alta luminosidad, los hay sensibles a luz roja, azul o verde, percibiendo por ellos los colores; y bastones, especializados en condiciones de baja luminosidad, permiten distinguir formas y contrastes. La retina se encuentra externamente en contacto con la coroides e internamente con el contenido de la parte posterior del ojo, el humor vítreo, un gel viscoelástico transparente que ayuda a mantener el tono y forma del ojo, protege a la retina durante los movimientos oculares (ya que debido a su viscosidad puede amortiguar los golpes), almacena y transporta los metabolitos que requieren las estructuras oculares del segmento posterior.

Volviendo a la retina, esta posee una estructura muy compleja cuyo análisis no entra dentro de los objetivos de este trabajo. Aunque sí se va a diferenciar entre retina periférica y su zona central, donde como ya se adelantaba antes, se encuentra una zona de unos 5.5 mm de diámetro denominada mácula lútea [5], la cual presenta una pigmentación amarillenta con una mayor densidad de conos. A su vez, la mácula presenta una depresión central avascular de unos 1.5 mm de diámetros denominada fóvea, la cual tiene la mayor sensibilidad para la percepción de los detalles gracias a la alta densidad de conos dentro de la misma. Dado que la DMAE afecta principalmente a la zona macular, la cual es donde se concentra la mayoría de los fotorreceptores encargados de la visión central, es lógico que uno de los síntomas iniciales de la enfermedad sea la pérdida de visión central y, en su fase avanzada, la ceguera total del paciente.

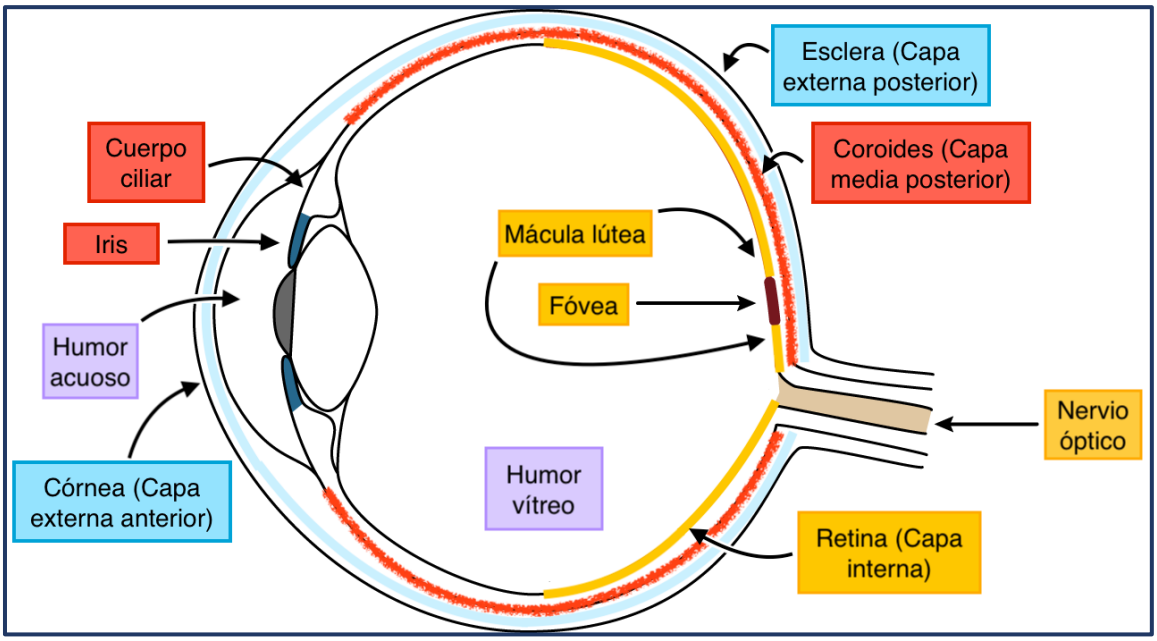

Figura 1 *Sección horizontal del ojo derecho visto desde arriba*

2.2 - Epidemiología

Se ha detectado que el desarrollo de esta enfermedad está relacionado con factores de riesgo genéticos y ambientales, siendo el único con una mayor evidencia la edad avanzada. Independientemente de padecer o no DMAE, con el envejecimiento hay una pérdida inherente de fotorreceptores, así como un adelgazamiento de la coroides acompañado con un aumento de su membrana basal, la membrana de Bruch [6]. Si bien, durante el transcurso de la DMAE, ocurren cambios patogénicos que desestabilizan la homeostasis ocular, los cuales acentúan estos fenómenos que ocurren de manera natural con el envejecimiento. Desgraciadamente las causas subyacentes a esta desestabilización son más bien desconocidas o no se comprenden completamente aún en la actualidad, llegando a ser la tercera causa principal de discapacidad visual, con una prevalencia de 170 millones pacientes mundialmente [2].

La información epidemiológica que se presenta a continuación se extrae de los resultados del meta-análisis "*Global prevalence of age-related macular degeneration an disease burden projection for 2020 and 2040: a systematic review and meta-analysis*" [7], realizado en colaboración por distintas entidades de Singapur en el año 2014. Se analizaron los casos de más de una centena de millar de pacientes de DMAE con edades comprendidas entre los 30 y 97 años. Además, el estudio realizó el análisis distinguiendo entre etnias (europea, africana, hispánica y asiática) y regiones (África, Asia, Europa, América del Sur y Caribe, América del Norte y Oceanía).

Tomando los resultados de prevalencia de todas las regiones geográficas consideradas que se observan en la Figura 2 y promediándolos, se determinó que la DMAE tiene una prevalencia en la población mundial del:

- 8.01% (Intervalo de Confianza del 95% (IC95%): 3.95, 15.49) en estadio temprano.
- 0.37% (IC95%: 0.18, 0.77) en estadio avanzado.

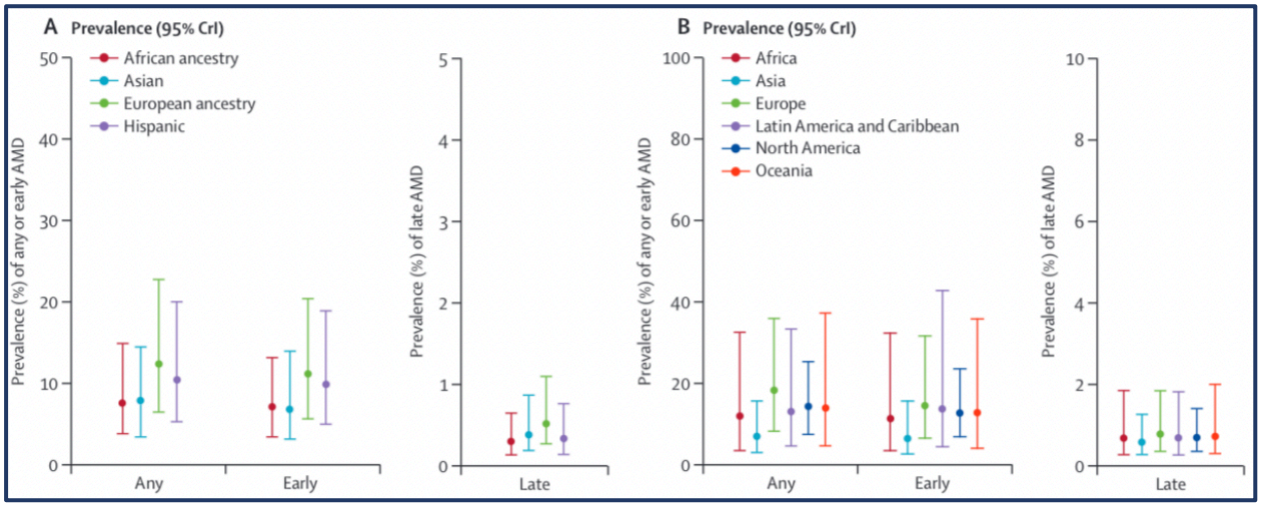

Figura 2 *Prevalencia de la DMAE por grupo étnico (A) y región geográfica (B). Tomada de Wong et al., 2014 [6]*

Del mismo modo, en la Figura 2 se observa que la prevalencia es mayor en poblaciones de etnia europea que en las de etnia asiática para DMAE temprana (11.2% vs 6.8%), sin embargo, las diferencias para DMAE avanzada no son tan importantes (0.5% vs 0.37%) [7]. Asimismo, la prevalencia en la población de ascendencia europea es mayor que en poblaciones africanas para todos los estadios, destacando que para la DMAE avanzada la prevalencia europea es del 12.3%, frente al 7.5% en la africana. El estudio también incorpora información sobre el tipo de DMAE (húmeda o seca), llegando a la conclusión de que no hay relación clara entre la prevalencia de la DMAE neovascular y el origen étnico; sin embargo, para el caso de la DMAE seca se observa que, de nuevo, las poblaciones de ascendencia europea tienen una mayor prevalencia que el resto, una prevalencia del 11% frente al 0.14% en poblaciones africanas, 0.21% en asiáticas y 0-16% en hispánicas.

En cuanto a la relación con la edad se observa en la Figura 3 una relación directamente proporcional entre prevalencia de la patología y edad del paciente [7]. El número de casos de DMAE aumenta con la edad para todos los grupos independientemente de origen étnico o región. Solamente destaca el importante incremento de los casos a partir de los 75 años en individuos de ascendencia europea, así como en los de las regiones de Europa y Oceanía. Con todo esto, se llega a concluir que la DMAE es más prevalente en individuos europeos, siendo mucho menos común en poblaciones asiáticas y africanas.

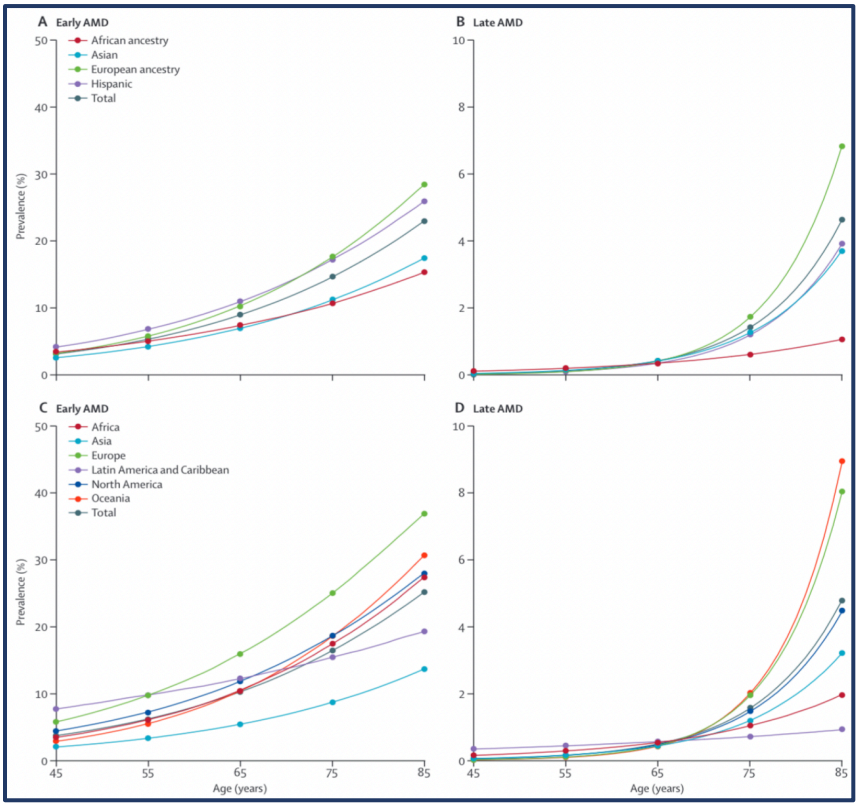

Figura 3 *Prevalencia de la DMAE con la edad según la etnia (A y B) y región (C y D). Tomada de Wong et al., 2014 [6]*

2.3 - Fisiopatología de la DMAE

Como se mencionó al principio del primer capítulo, la DMAE se distingue entre precoz y avanzada. La característica principal del primer estadio es la acumulación de drusas en la zona. Las drusas son depósitos extracelulares de color amarillento que inducen cambios degenerativos en diferentes estructuras del ojo como el EPR o la coroides, especialmente en la membrana de Bruch [8]. Inevitablemente, esta afectación de la retina concluirá con la destrucción de fotorreceptores en la zona de la mácula, causa de la pérdida de visión central. Como se irá justificando a lo largo de esta sección, el mayor desafío que plantea esta enfermedad es la falta de una comprensión profunda sobre los mecanismos fisiológicos que la dominan. Dada esta falta de información, solo se puede diferenciar claramente entre el primer y último estadio de la enfermedad, si bien se sabe que el estadio intermedio existe y sería un estado de transición entre los otros dos, su comprensión es aún bastante limitada [6].

- El estadio precoz es asintomático de progresión lenta, el cual se caracteriza por la aparición de drusas y alteraciones en estructuras como la membrana de Bruch o el epitelio pigmentario de la retina (EPR) [2].

- En el estadio avanzado la acumulación de drusas produce tal disrupción de la retina que la pérdida de visión es mucho más seria. La forma clínica más conocida de este estadio es la DMAE neovascular, también conocida como húmeda o exudativa, ya que se caracteriza por la aparición de nuevos vasos vasculares, exudados y acúmulo de otra clase de residuos en la zona de la retina. En contraparte, toda otra clase de DMAE en la que no haya neovascularización se denomina DMAE seca, atrófica o no exudativa [6].

Dentro de los dos tipos de DMAE avanzada (seca y húmeda), hay poco que se pueda profundizar sobre la DMAE seca, sus principales rasgos son los ya comentados, se trata del estadio avanzado de la DMAE que no acontece con neovascularización. Sin embargo, sería erróneo pensar que en la DMAE seca no se produce afectación de los coriocapilares, lo que no ocurre es la síntesis de nuevos vasos. De hecho, hay evidencias de que los coriocapilares quedan muy adelgazados en la DMAE seca en zonas denominadas de atrofia geográfica [9], la justificación principal de este hecho es el envejecimiento del propio tejido de la coroides a través de mecanismos como el estrés oxidativo, la inflamación persistente, la disfunción mitocondria, daños adquiridos en su material genético y senescencia celular [10].

Por su parte, la DMAE húmeda o neovascular se caracteriza por la transformación de los vasos sanguíneos en la coroides que irrigan la mácula mediante un proceso de neovascularización. Esta capa de capilares altamente anastomosados y fenestrados se conoce como coriocapilares [11]. Algunos de los cambios vasculares que se observan sobre los coriocapilares en la DMAE son la disminución de la densidad vascular, la pérdida de células endoteliales o fenestraciones anormales [12]. Cuando un proceso patológico (como por ejemplo un edema) ocurre, las estructuras suprayacentes se interrumpen, produciendo un desprendimiento focal de retina y la consecuente pérdida de visión [8]. Sin embargo, esta neovascularización coroidea es una reacción secundaria causada cuando la mácula ha sido dañada por otra alteración

patológica. Estas alteraciones inducen una respuesta inmune, la cual se acompaña con la acumulación de macrófagos y células gigantes en la zona, dando lugar a una situación de hipoxia en la zona, causa principal de la producción de factores de crecimiento endotelial vascular (VEGF) que, al mismo tiempo, inducen la neovascularización coroidea [8], [11]. Por ende, se justifica la existencia de una profunda relación entre DMAE húmeda y la regulación de factores proangiogénicos en la zona de la mácula.

En cualquier caso, la afectación de los coriocapilares compromete el transporte de nutrientes y la eliminación de productos de desecho, favoreciendo la aparición de drusas. Algunos de los componentes que se acumulan en las drusas son: albúmina, amiloide-β, apolipoproteína E (APOE), componentes del complemento, gránulos de melanina, fibrinógeno, inhibidores tisulares de metaloproteinasas (TIMPs), inmunoglobulinas, lípidos, lipofuscina, lipoproteínas, metaloproteinasas de la matriz (MMPs), organelas, pentraxinas, proteína C reactiva (PCR) y vitronectina [13]. El acúmulo de estas biomoléculas induce inflamación en el espacio subretiniano ya que se producen dos acciones clave en la DMAE:

- Activación del inflamasoma NLRP3: este complejo proteico intracelular estimula la producción de citoquinas (e.g. IL 1-β), consolida el proceso inflamatorio e induce la apoptosis. Este mecanismo justifica el daño celular percibido en la atrofia geográfica en la DMAE no exudativa [14].
- Atracción de microglía y macrófagos retinianos: en condiciones normales su acción debería mantener la homeostasis, pero, bajo las condiciones de la DMAE, sufren cambios funcionales como la fagocitosis ineficaz y la señalización deficiente, que acaban perpetuando el problema de las drusas. Asimismo, estas células producen un mediador de gran importancia para la retina: el leucotrieno B4, que estimula la producción y secreción de VEGF, es decir, agravando la neovascularización en los coriocapilares [15].

Para poder realizar una correcta determinación del nivel de severidad de DMAE, ya no solo el estadio en el que se encuentra la patología, sino la gravedad de la misma, se van a emplear los estándares descritos por *The Minnesota Grading System* (MGS) [16], el cual fue definido en base a los resultados obtenidos por el *Age-Related Eye Disease Study* (AREDS) [17]. De este modo se describen cuatro niveles de DMAE en función del tamaño de las drusas y el área de mácula afectada que se puedan evidenciar en la retina del paciente:

- MGS1 – Sujeto sano.
- MGS2 – Drusas de un tamaño menor a 125 μm y evidencia de anormalidades del EPR. Equivalente al estadio precoz.
- MGS3 – Drusas de un tamaño mayor a 125 μm y área afectada mayor a 180 μm pero sin afectar el centro de la mácula. Aquí se engloban las etapas intermedias entre el estadio precoz y la DMAE avanzada.
- MGS4 – DMAE avanzada con afectación del centro de la mácula o evidencias de neovascularización.

2.4 - Técnicas de diagnóstico actuales

El propósito de esta sección no es otro que la revisión de las técnicas empleadas en la actualidad para el diagnóstico de la DMAE. Aunque se podría recurrir a distintos enfoques para el diagnóstico de esta enfermedad (como el estudio de biomarcadores, por ejemplo), en la práctica clínica el diagnóstico de la DMAE resulta de las evidencias observadas tras la exploración oftalmológica del paciente, recurriendo a lo sumo a técnicas de imagen médica o, al menos, así es en el presente [3], [18], [19]. Por ende, se va a indicar a continuación el procedimiento más comúnmente empleado en clínica para el diagnóstico de la DMAE, destacando las principales limitaciones del mismo, lo cual llevará a justificar la necesidad de una herramienta con una naturaleza distinta que podría completar el diagnóstico.

- En primer lugar, se procede con la medición de la agudeza visual, mediante el empleo de optotipos y del test de rejilla de Amsler, una cuadrícula con un punto en el centro que permite evaluar la calidad de la visión central [18]. Para realizar este test se debe colocar al paciente a unos 35 cm de distancia de la misma, tapar un ojo y mirar fijamente el punto central, si ve líneas torcidas, dobladas, borrosas o que desaparecen en algún punto, tal como se ilustra en la Figura 4 [20], el resultado del test será positivo indicando que su visión central está afectada.

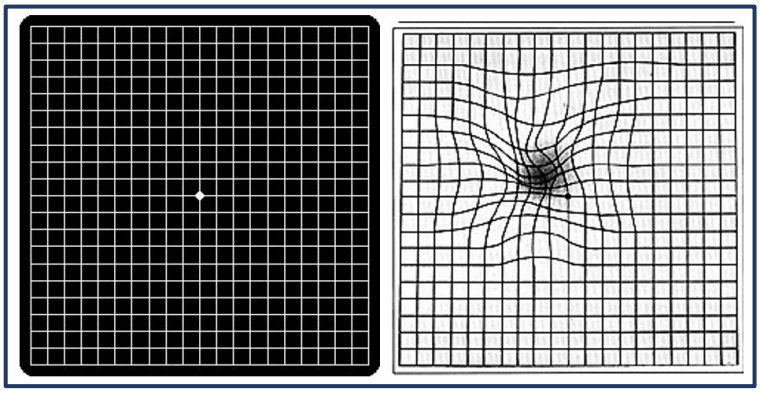

Figura 4 *Resultado del test de Amsler con visión normal (izquierda) frente a visión distorsionada (derecha). Tomada de Gómez-Ulla, 2022 [22]*

- En caso de que el paciente muestre indicios de pérdida de agudeza visual, se prosigue con la exploración del fondo de ojo mediante oftalmoscopia (haz de luz que permite estudiar el segmento posterior del ojo) [19]. De este modo, se puede detectar la presencia de drusas y confirmar el diagnóstico de DMAE.

- Adicionalmente, para llegar a un diagnóstico más específico que permita valorar el estadio y tipología de la enfermedad se recurre principalmente a dos técnicas de imagen:

  - Tomografía de Coherencia Óptica (OCT): se emplean ondas electromagnéticas del espectro infrarrojo (820-830 nm) para realizar una tomografía (un 'corte') de las estructuras oculares a partir de la reflexión de

las ondas sobre ellas, es una técnica idónea, ya que no es agresiva para el paciente y su sensibilidad es muy alta, aunque se debe tener en cuenta la disponibilidad de la tecnología, así como de los especialistas necesarios para realizarla [3].

- Angiografía con fluoresceína y verde de indocianina: una imagen radiológica de las estructuras vasculares del ojo que permite examinar el flujo sanguíneo de la retina y la coroides, por tanto, se considera una técnica más opcional a la que solo se recurre en caso de sospecha de DMAE neovascular [18].

Con todo ello, se debe notar que el protocolo de diagnóstico de la DMAE actualmente se basa en dos principios básicos: la pérdida de agudeza visual y la evidencia de rasgos característicos de la enfermedad (drusas o neovascularización coroidal) mediante imagen médica. Es decir, el diagnóstico no es posible durante las etapas iniciales de la patología, ya que en ese momento su visión central no se encontrará lo suficientemente afectada como para poder evidenciarse durante la prueba. Es más, dado que el paciente no presentará síntoma alguno que lo haga sospechar de la patología, no acudirá al especialista hasta un momento en el que la enfermedad esté mucho más avanzada. Por lo tanto, se pone aquí en manifiesto la necesidad de un diagnóstico precoz de la DMAE ya que, como se va a exponer en el siguiente apartado, de ello dependerá fuertemente el tratamiento y pronóstico del paciente.

2.5 - Estrategias terapéuticas actuales

Como se ha podido intuir en base a los apartados anteriores, el conocimiento respecto a los mecanismos que desencadenan la DMAE en sus etapas iniciales, así como en el caso no exudativo, es muy limitado. Hoy en día, solo se tiene una cierta comprensión del caso exudativo, es por ello por lo que las principales terapias se centran únicamente en este mismo. Aun así, lo primero que se debe hacer notar es que, en la actualidad, no existe una cura para la DMAE, solamente se dispone de terapias que pueden llegar a ralentizar su evolución. Por tanto, el tratamiento de la DMAE es muy dependiente de un diagnóstico realizado lo más pronto posible, ya que el tratamiento es más eficaz cuanto más incipiente es la lesión y menor es el tiempo de evolución. Ahora se pasa a desarrollar el tratamiento a seguir en función del tipo de DMAE. Como se venía mencionando, el caso de la DMAE no exudativa o seca es el menos estudiado, no existe un tratamiento en sí, solamente se procede recetando suplementos nutricionales en fases iniciales para asegurar la correcta nutrición de la retina a fin de frenar el progreso de la patología [18], [19].

En el caso de la DMAE exudativa o húmeda, aunque no se pueda revertir totalmente sus efectos, hay tratamientos para reducir su progresión, ya que suele ser más agresiva que el caso no exudativo, en concreto destacan dos terapias: las inyecciones intravítreas y la terapia fotodinámica (PDT) [18].

- Las inyecciones intravítreas (dentro de la cavidad vítrea) con fármacos anti-VEGF son el tratamiento más habitual para la DMAE exudativa, se busca inhibir el efecto de los VEGF que causan la neovascularización. Los fármacos más utilizados en estos casos son *ranibizumab*, *bevacizumab*, *aflibercept* o *brolucizumab* [19]. Este procedimiento es realizado en quirófano con anestesia tópica y su postoperatorio no es complicado. Su principal contrapartida es que debe repetirse periódicamente cada 1-3 meses hasta estabilizar la patología. Una alternativa que se ha desarrollado para este problema se trata de un sistema de administración portuaria implantable de *ranibizumab* [19], el cual administra de forma continuada el fármaco anti-VEGF en el humor vítreo.

- La terapia fotodinámica se basa en el cierre de los vasos patológicos generados por la acción de los VEGF, durante la PDT se emplea un agente fotosensibilizante, como la *verteporfina* (*benzoporfirina* sintética), encapsulado en liposomas y se administra por infusión intravenosa. La *verteporfina* es un agente citotóxico que tiene un efecto sinérgico al combinarse con los anti-VEGF mencionados. Cuando se ha acumulado el agente en los capilares, se activa con un láser no térmico a 689 ± 3 nm de longitud de onda. De este modo, el agente activado crea especies reactivas de oxígeno (ROS), las cuales son citotóxicas, englobando y eliminando los capilares en los que se encuentren [21]. Sin embargo, esta terapia tiene una baja selectividad, ya que los liposomas son capaces de atravesar el epitelio discontinuo de capilares patológicos y sanos. Además, debido a la recurrencia de la angiogénesis, la PDT debe repetirse en algunos pacientes con relativa frecuencia.

Con todo ello, se prueba de nuevo la necesidad de un método de estimación del riesgo de aparición o padecer la enfermedad de forma incipiente, ya que los tratamientos actuales se centran en evitar el progreso de la DMAE a etapas más avanzadas. Sin embargo, los métodos de diagnóstico actuales no permiten un reconocimiento tan temprano de la DMAE, invalidando muchas de las terapias existentes. Por tanto, el desarrollo de una herramienta basada en marcadores de expresión puede ser la solución para complementar las técnicas actuales de diagnóstico de esta patología ya que, como se abordará en el siguiente capítulo, las técnicas que se emplean para el estudio de la expresión génica de una persona han avanzado enormemente en la actualidad y son cada vez más viables en la clínica.

Capítulo 3 - Caracterización de la expresión génica

3.1 - Conceptos de expresión génica en el ser humano

Antes de empezar a enunciar ninguna técnica relacionada con la expresión génica se debe aclarar una serie de conceptos que serán base de toda la teoría que se desarrollará posteriormente. En el ser humano, como todo ser vivo, es bien conocido que la información genética está contenida dentro de cada una de sus células en forma de biomoléculas, los ácidos nucleicos: ácido desoxirribonucleico (ADN) y ácido ribonucleico (ARN). A su vez, ambas biomoléculas se componen de nucleótidos unidos covalentemente mediante enlaces fosfodiéster [22]. Cada nucleótido está formado por una pentosa (2'-desoxi-D-ribosa o D-ribosa), uno o más fosfatos y una base nitrogenada, la cual se puede clasificar en función del compuesto del que deriva en:

- Purinas: adenina (A) y guanina (G) comunes a ADN y ARN.
- Pirimidinas: timina (T) y citosina (C) en el ADN; uracilo (U) y citosina (C) en el ARN.

Se debe anotar que esta clasificación es en función de las bases principales, pero los ácidos nucleicos pueden presentar también otras bases modificadas (e.g. Metilaciones). Es precisamente la secuencia específica de bases lo que determina la estructura de todo componente celular dentro del organismo. Las bases nitrogenadas pueden formar puentes de hidrógeno entre sus grupos carbonilo y amino de distintas cadenas, de modo que: la A se une con T o U y la G con C. Además, son hidrofóbicas y no tienen carga a pH celular, por lo que entre ellas se establecen interacciones hidrofóbicas que las apilan dando la estructura particular del ADN (Figura 5), descubierta en 1953 por James Watson y Francis Crick. Tras ciertos ajustes realizados posteriormente, se conoce que el ADN tiene una estructura helicoidal de doble cadena con 10.5 pares de bases por vuelta, es decir, cada molécula de ADN presenta dos cadenas o hebras de nucleótidos, las cuales son antiparalelas y complementarias entre sí de modo que en frente de cada A habrá una T y en frente de cada G una C [23]. De su propia estructura se deduce uno de los principios fundamentales de la conservación y transmisión del material genético: la replicación del ADN. Su principal característica es que se trata de una replicación semiconservadora, es decir, cada cadena de la doble hélice del ADN sirve como molde para la síntesis de una nueva, dando como resultado dos moléculas hijas de ADN, cada una compuesta de una hebra del ADN original y de una hebra complementaria nueva [22]. Existen puntos de origen concretos a partir de los cuales el proceso avanza bidireccionalmente, sintetizándose las nuevas hebras gracias a la acción de la ADN polimerasa, una enzima que cataliza la síntesis de ADN a partir de desoxirribonucleótidos.

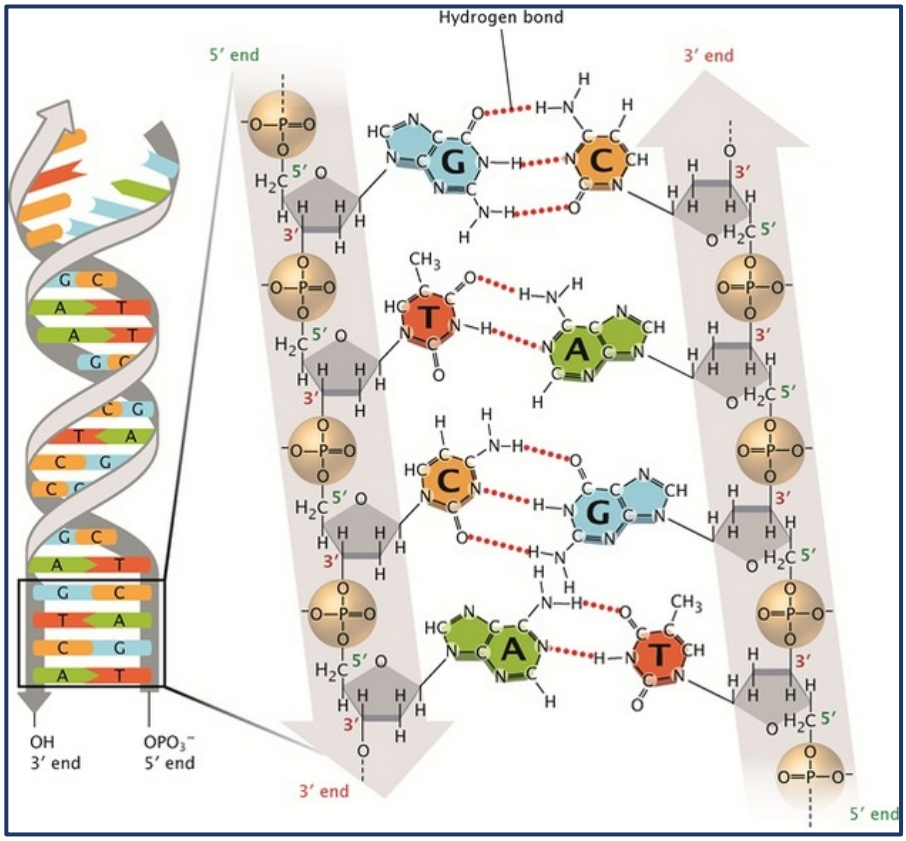

Figura 5 *Estructura del ADN. Tomada de A. Pray, 2008 [23]*

Por tanto, la función del ADN es el almacenamiento y transmisión de la información genética. En concreto, se denominará gen a todo el ADN que codifica la secuencia primaria de un producto génico final, sea polipéptido o ARN, con función estructural o catalítica [22]. Cada célula humana posee en torno a 3.070.128.600 pares de bases repartidos en 23 pares de cromosomas, las estructuras que se encuentran dentro del núcleo celular en las que se compacta el material genético; sin embargo, se estima que solo hay unos 20.000 genes que codifiquen proteínas [24]. El conjunto de todo el material genético de un ser vivo es denominado genoma. Además, el ADN contiene secuencias reguladoras que permiten identificar el principio o el final de los genes, influir en su transcripción o funcionar a modo de punto de inicio de la replicación. Sin embargo, también se encuentran intercalados entre los genes segmentos de ADN no codificantes, es decir, secuencias de ADN que no expresan ningún producto génico a las que se denominan intrones, y los segmentos codificantes exones. Se estima que, aproximadamente, solo el 1.5 % del ADN humano es codificante [22]. Dentro de la célula se pueden encontrar diferentes tipos de ARN, por lo que su función es algo más compleja, en esta sección solo se van a destacar los tres tipos de ARN principales: ARN ribosómico (rRNA), componente de los ribosomas, orgánulos encargados de la síntesis de proteínas; ARN mensajero (mRNA) encargado de, como su nombre indica, transportar la información genética desde los genes hasta los ribosomas, para ser traducido a proteína; finalmente, el ARN de transferencia (tRNA) participa en la traducción de la información del mRNA en aminoácido, componente elemental de las proteínas [22]. A partir de las secuencias de ADN se pueden obtener todos los tipos de ARN como se va a explicar a continuación.

3.2 - Síntesis de ARN

La expresión de la información genética requiere la síntesis de una molécula de ARN a partir de una de ADN gracias a la acción de la enzima ARN polimerasa, este proceso es conocido como transcripción. Durante este proceso solo se transcriben genes determinados, siendo la propia célula capaz de regular esta expresión según la circunstancia y momento. Se denomina transcriptoma al conjunto de todas las moléculas de ARN producidas en unas condiciones determinadas [22]. Existen secuencias reguladoras, así como secuencias que indican el principio y final de la transcripción. Las secuencias que indican el inicio se denominan promotoras, estas son muy variadas, pero hay unas más comunes: las secuencias consenso, ricas en residuos AT (e.g. -TATAAT- o -TTGACA-). El inicio de la transcripción solo emplea una de las hebras del ADN como molde, generando una nueva hebra de ARN antiparalela y de forma complementaria, apareando cada nucleótido de modo que se insertan residuos de U para aparearse con A en el ADN molde, A para T, C para G y viceversa. De este modo, una hebra de ADN es la hebra molde sobre la que se sintetiza la cadena de ARN, mientras que la otra es la hebra codificante, idéntica en secuencia de bases al ARN transcrito, pero con residuos de T en vez de U. Finalmente, la terminación de la síntesis del ARN está señalizada típicamente por secuencias autocomplementarias que forman una horquilla [22].

Sin embargo, solo parte del ARN de un gen codificante llegará a proteína. Una molécula de ARN recién sintetizada se denomina transcrito primario, el cual debe sufrir modificaciones hasta madurar en la molécula de ARN final. La modificación más importante que tiene lugar es la denominada como corte y empalme del ARN o *splicing*, en este proceso se eliminan los intrones y se unen los exones para formar una secuencia continua y específica para un polipéptido funcional. Pero la utilidad del corte y empalme del ARN no es solo la maduración del transcrito primario, sino que a su vez guarda una estrecha relación con la expresión génica, ya que la maduración diferencial del ARN puede dar lugar a múltiples productos a partir de un mismo gen. De hecho, en el 95% de los genes humanos tiene lugar un proceso de corte y empalme alternativo, por el cual exones concretos pueden ser incluidos o no en el ARN maduro, dando lugar a más de una proteína a partir de un mismo transcrito primario [22]. Se debe tener en cuenta que las técnicas de secuenciación de ARN no tienen en cuenta este ARN maduro que pasará a proteína, sino el transcriptoma inicial que incluye mucha más información.

3.3 - Secuenciación de ARN (RNA-seq)

La secuenciación de ARN (RNA-seq) es una técnica de secuenciación de nueva generación (NGS de sus siglas en inglés *Next-Generation Sequencing*) que permite estudiar la presencia y cantidad de RNA en una muestra biológica expresada como conteos [25]. Pero no solo aporta información cuantitativa respecto al nivel de expresión, ya que también permite la anotación de la muestra, esto es, la identificación de los transcritos expresados, así como de límites exón/intrón o los sitios de inicio transcripcional [26]. Esta técnica es por tanto adecuada para estudiar el perfil de expresión de pacientes de una determinada enfermedad y, de este modo, intentar comprender la relación entre patrones en la expresión génica y los estados fisiológicos de esos pacientes, en concreto para el caso de estudio, pacientes de DMAE.

Se debe indicar que existen más tecnologías para el estudio de los perfiles de expresión, siendo la tecnología de microarrays (*DNA microarray*) la más destacable [25]. Sin embargo, en la actualidad RNA-seq es dominante ya que, en comparación con el resto de tecnologías, requiere una cantidad pequeña de ARN para empezar a analizar, facilitando así la obtención de muestras; además, no se restringe el análisis a un conjunto de transcritos fijos, sino que permite analizar el transcriptoma; finalmente, no solo permite el estudio de la expresión a nivel de genes, sino que también de alelos individuales o variantes de transcripción. Un experimento habitual de RNA-seq genera miles de lecturas sobre secuencias cortas de material genético correspondientes a regiones codificantes. El procedimiento es como sigue [25]:

- En primer lugar, se obtiene ARNs implicados en la traducción (mRNA, tRNA o rRNA) y ARNs no codificantes de un conjunto de muestras asociadas con la patología de estudio mediante pequeñas biopsias.

- Se fragmenta el ARN en secuencias cortas de entre 200 y 300 pares de bases y se revierte a ADN complementario de doble cadena mediante la enzima retrotranscriptasa. En este paso se debe destacar que, al trabajar directamente con ADN de doble cadena, no se requiere hibridar con un complementario a diferencia de otras técnicas de análisis de expresión como los microarrays. Esa hibridación limita el número de muestras que se puede estudiar a aquellas que se unan a una sonda, sin embargo, con RNA-seq se tiene todo el transcriptoma disponible.

- Cada fragmento se alinea con el genoma de referencia contabilizando el número de lecturas coincidentes con las regiones del genoma que son transcritas, determinando así también el gen de procedencia. Para esta tarea se emplean métodos de alineación de lectura corta y que introducen huecos (*gapped short-read aligners*), algunos de los métodos más populares son *TopHat2*, STAR, *Stampy* o *GDNAP*. Después, las alineaciones se pueden procesar a regiones putativas del genoma mediante herramientas como *Cufflinks*, *StringTie* o *Trinity* [25].

- De este modo, se obtiene una matriz de conteo con tantas filas como genes asociados al estudio y columnas como muestras empleadas. Es decir, la información en la fila i-ésima y columna j-ésima será el número de lecturas mapeadas en el gen i y en la muestra *j*.

A la hora de interpretar los resultados de la matriz de conteo se deben tener una serie de consideraciones que causan variaciones en los conteos. Estas serán ilustradas a través del ejemplo de la Tabla 1, suponiendo que se parte de la siguiente matriz correspondiente a tres genes y dos muestras.

Tabla 1 *Ejemplo de matriz de conteo*

	Muestra 1	***Muestra 2***
Gen A	20	46
Gen B	654	1307
Gen C	1305	2580

En primer lugar, se nota que los genes en la muestra 2 tienen aproximadamente el doble de conteos que en la muestra 1, pudiendo implicar fluctuaciones en el número de veces que se lee cada nucleótido durante la secuenciación como se representa en la Figura 6. Por otra parte, se observa que el gen C tiene el doble de alineamientos que B en todas las muestras, lo cual puede ser debido a que el gen C se exprese con el doble de transcritos que el gen B o que ambos se expresen con el mismo número de transcritos, pero el gen C es el doble de largo que el B y, por tanto, produce el doble de fragmentos, por lo que en este caso un mayor conteo no significa una mayor expresión del gen, como se representa en la Figura 7.

Figura 6 *Gen con el doble de transcritos*

Figura 7 *Gen con el doble de fragmentos*

Finalmente, también se debe hacer notar una serie de posibles situaciones problemáticas que se pueden dar a la hora de analizar los resultados de un estudio de RNA-seq. A pesar de trabajarse con varias muestras a la vez, cabe la posibilidad de que una parte significante del genoma no se exprese en ninguna de las mismas; asimismo, habrá regiones transcritas que no sean codificantes o que, incluso si lo son, su marco abierto de lectura real todavía no haya sido identificado [25].

En pocas palabras, la tecnología RNA-seq proporciona una medida de la abundancia relativa de cada gen en cada muestra analizada y permite estudiar la importancia biológica de esta expresión para el caso.

Capítulo 4 - Revisión de estado sobre aplicaciones web para la ayuda al diagnóstico de la DMAE

Es de interés conocer el nivel de desarrollo de aplicaciones web para el diagnóstico de la DMAE en la actualidad a fin de saber las limitaciones de estas mismas y los puntos a mejorar con la propuesta de este trabajo. Sin embargo, esta revisión solo ha dado como resultado tres aplicaciones destinadas al diagnóstico precoz de la DMAE que hoy en día se encuentren completamente desarrolladas y comercializadas, siendo además ninguna de ellas fundamentada en el riesgo de padecer la patología a partir de la expresión génica de los pacientes. En primer lugar, se encuentra la aplicación *MaculaTester* (sitio web oficial: https://www.maculatester.com), disponible para iOS y Android por un precio de 2.99 $ o 3.49 €, básicamente se trata de una versión interactiva del test de rejilla de Amsler, uno de los principales métodos de diagnóstico de la DMAE como ya fue comentado. La principal ventaja respecto a los métodos tradicionales que presenta esta aplicación es la capacidad de almacenar los resultados obtenidos en el dispositivo electrónico, de este modo nunca se perderán pudiendo mostrárselos a su médico en cualquier momento; también permite comparar sus resultados actuales con todos los anteriores y estudiar la evolución de la enfermedad. Aun así, los creadores quieren aclarar que esta aplicación no es un método de diagnóstico de la DMAE, sino una herramienta para ayudar a detectar cambios en la función macular que no puede remplazar un examen médico realizado por un oftalmólogo u optómetra certificado. A fecha actual, esta aplicación no ha sido aún aprobada por la *Food and Drug Administration* (FDA), a diferencia de las otras dos aplicaciones que van a ser expuestas. La segunda aplicación existente es *Alleye* (sitio web oficial: https://alleye.io), disponible también para dispositivos iOS y Android de libre adquisición, pero con compras integradas dentro de la aplicación, su fin es detectar tempranamente indicios de DMAE y retinopatía diabética. Se fundamenta en la medición de la agudeza visual a través de tareas de alineamiento que el propio paciente puede realizar de manera autónoma, el usuario debe alinear un punto central junto a otros dos puntos externos sobre una línea recta fijada. Su principal rasgo es una interfaz tan intuitiva que permite un manejo muy sencillo para que el propio paciente pueda realizar su registro de manera autónoma. Finalmente, la única otra aplicación aprobada por la FDA para el diagnóstico precoz de la DMAE es mVT® (sitio web oficial: https://myvisiontrack.com), disponible en iOS y Android de manera gratuita. El test integrado consiste en una serie de figuras sobre las que el usuario debe seleccionar aquella cuya forma sea irregular. Aunque el fin de esta herramienta está más enfocado hacia labores de investigación que hacia un usuario final o entorno clínico.

Estas dos últimas aplicaciones han presentado una alta precisión en la detección de la DMAE en estudios clínicos con un área bajo la curva ROC (AUC) de 0.969 y 0.845 entre pacientes de DMAE y sujetos sanos jóvenes o de edad similar a los pacientes de DMAE, respectivamente [27]. Indicando con ello que la herramienta podría llegar a funcionar como método de *screening* de la enfermedad, aunque en la actualidad no ha sido implantada. Por tanto, a pesar de existir aplicaciones enfocadas al seguimiento de la evolución del paciente de DMAE, así como realizar un diagnóstico precoz de la misma, el usuario objetivo de estas aplicaciones es el propio paciente, mientras que en la propuesta de este trabajo no será el paciente el que la utilice, sino

que el enfoque es el de una herramienta de apoyo para el personal sanitario y la investigación sobre la enfermedad. Por ejemplo, podría ser de gran utilidad para que un médico de atención primaria decidiera si enviar o no al paciente al especialista ante la sospecha diagnóstica de padecer DMAE. En ese sentido, la propuesta de este trabajo es complementaria al resto de herramientas que existen en la actualidad. Para ello, se requiere un abordaje multidisciplinar que aún no han alcanzado las soluciones actuales, existiendo un gran margen de mejora una vez superadas las limitaciones señaladas mediante el uso combinado del análisis de la expresión génica y las tecnologías informáticas.

Capítulo 5 - Hipótesis

La DMAE es una enfermedad degenerativa de la zona central de la retina, que cursa con una pérdida progresiva de la agudeza visual de detalle. Es una enfermedad relacionada con el envejecimiento, con alta prevalencia y que representa una de las primeras causas de ceguera en mayores de 65 años. Aunque en la actualidad no existe cura, existen tratamientos que consiguen frenar la progresión de la enfermedad, por lo que un diagnóstico temprano es clave para el buen pronóstico visual del paciente. La hipótesis de este trabajo es que, a partir de técnicas de secuenciación de ARN, concretamente de RNA-seq, es posible identificar biomarcadores de la enfermedad, que permitan, tanto estimar el riesgo a padecer la patología, como clasificar su estadio, proporcionando una buena herramienta de ayuda a su diagnóstico, evolución e identificación de posibles dianas terapéuticas.

Capítulo 6 - Objetivos

Objetivo principal:

> Encontrar biomarcadores específicos de la DMAE que permitan su diagnóstico y clasificación por estadios.

Objetivos específicos:

> 1. Utilizar datos de RNA-seq para obtener genes diferencialmente expresados en relación a la DMAE.
>
> 2. Establecer un protocolo de tratamiento de datos de RNA-seq, para obtener información útil de este tipo de datos.
>
> 3. Identificar, describir y diferenciar los perfiles de expresión génica que caracterizan al paciente de DMAE en cada uno de sus estadios.
>
> 4. Desarrollar una aplicación web que ofrezca una interfaz sencilla para que profesionales sanitarios puedan diagnosticar y clasificar a un paciente a partir de su perfil de expresión.

PARTE 2 DESARROLLO

Capítulo 7 - Materiales
7.1 - Bases de datos
7.2 - Muestra de estudio
7.3 - Software utilizado

Capítulo 8 - Metodología
8.1 - Recopilación de datos y metadatos de RNA-seq
8.2 - Depuración y transformación de los datos de conteo
8.3 - Análisis exploratorio de datos de conteo
    8.3.1 - Normalización de datos de conteo
    8.3.2 - Distribución de datos de conteo
    8.3.3 – Análisis de Componentes Principales (PCA)
8.4 - Análisis de expresión diferencial
8.5 - Agrupación de genes
8.6 - Análisis de enriquecimiento
8.7 - Diseño de un clasificador
    8.7.1 - Definición de un modelo
    8.7.2 - Valoración de la capacidad discriminatoria del modelo
8.8 - Desarrollo de una aplicación web

Capítulo 9 - Resultados
9.1 - Depuración y transformación de los datos de conteo
9**.2** - Análisis exploratorio
9.3 - Análisis de expresión diferencial
9.4 - Agrupación de genes
9.5 - Análisis de enriquecimiento
9.6 - Diseño y ajuste del clasificador
9.7 – Aplicación web

Capítulo 7 - Materiales

7.1 - Bases de datos

Para realizar el análisis se debe acceder a información disponible para estudios bioinformáticos, hay distintas bases de datos a las que se puede recurrir con este fin. Las que se van a emplear en este trabajo son:

- *Gene Expression Omnibus* (GEO) es un repositorio público de datos de genómica funcional que admite conjuntos de datos basados en matrices y secuencias, administrada por el NCBI (*National Center for Biotechnology Information*), parte de la NLM (*National Library of Medicine*) de Estados Unidos. Además, proporciona herramientas para ayudar a los usuarios a consultar y descargar perfiles de expresión génica. De aquí se puede obtener el conjunto de muestras asociadas con la DMAE sobre las que se realiza el análisis de expresión.

- *Gene Ontology* (GO) es un sistema fundado por el NHGRI (*National Human Genome Research Institute, US National Institutes of Health*) estandarizado para la anotación de genes y sus productos con respecto a tres categorías basadas en procesos biológicos, funciones moleculares y componentes celulares. Los términos GO se organizan en una estructura jerárquica, en la que un término de un nivel superior representa una categoría biológica más amplia y los términos de niveles inferiores representan procesos o funciones biológicas más específicos.

- *Ensembl* es un buscador de genomas de vertebrados que apoya la investigación en genómica comparativa, evolución, variación de secuencias y regulación transcripcional. Fue resultado de una colaboración entre el *Wellcome Trust Sanger Institute* y el EBI (*European Bioinformatics Institute*). Permite anotar genes, calcular alineaciones múltiples, predecir la función reguladora y recopilar datos sobre enfermedades. De aquí se obtiene el genoma de referencia, correspondiente a la especie humana (*homo sapiens*), con código de *Ensembl* GRCh37.p13 y disponible de manera pública. Este genoma contiene 3098825702 pares de bases, 20805 genes codificantes y 22966 secuencias no codificantes. Su cariotipo o conjunto de cromosomas consiste en veintidós pares de cromosomas autosómicos, que se muestran ordenados en función de su longitud en la Figura 8, más dos cromosomas sexuales.

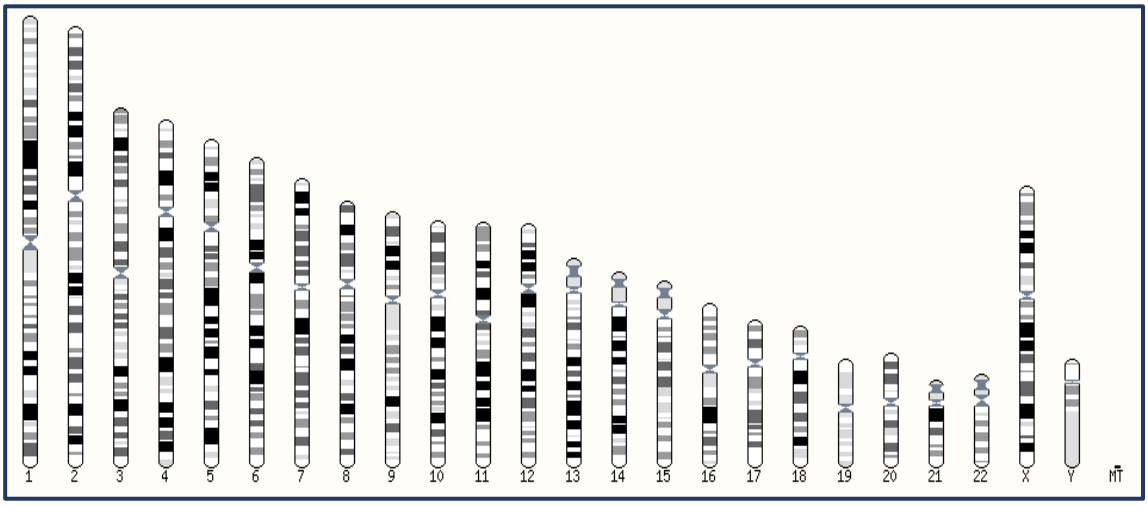

Figura 8 *Cariotipo GRCh37.p13*

7.2 - Muestra de estudio

Se van a emplear los datos de expresión génica disponibles en GEO con el código de identificación GSE115828 provenientes de 523 muestras *postmortem* de retina facilitadas por el *Minnesota Lions Eye Bank*, tras el consentimiento informado del propio donante o allegados. Se excluyeron aquellas muestras provenientes de sujetos que presentasen indicios de diabetes o glaucoma en su historial clínico, así como aquellas muestras en las que se encontrasen evidencias clínicas de retinopatía diabética, glaucoma, miopía magna o la presencia de partículas atípicas a través de imágenes de fondo de ojo; ya que no conviene incluir en el estudio a pacientes con patologías oculares de una índole similar a la de estudio. El grado de severidad de DMAE de cada muestra está establecido según el *Minnesota Grading System* [16], que queda recogido en la Tabla 2, de modo que también se descartaron aquellas muestras con un nivel MGS no concluyente o desconocido, ya que no pudieron ser clasificadas dentro del sistema debido a falta de información.

Tabla 2 *Minnesota Grading System [15]*

Grado de DMAE	**Evidencias**
MGS1	Máximo tamaño de drusa < C-0 y área total afectada < C-1
MGS2	▪ C-0 < Máximo tamaño de drusa < C-1 ▪ Área total afectada ≥ C-1 ▪ Evidencia de anormalidades del EPR
MGS3	▪ Máximo tamaño de drusa ≥ C-1 ▪ Máximo tamaño de drusa ≥ C-0 y área total > I-2 ▪ Máximo tamaño de drusa ≥ C-0 y área total > O-2 ▪ Atrofia > I-1 pero sin afectar el centro de la mácula
MGS4	▪ Atrofia con afectación del centro de la mácula ▪ Evidencias de DMAE neovascular
Medidas de referencia	C-0 ≈ 63 µm C-1 ≈ 125 µm I-1 ≈ 180 µm I-2 ≈ 360 µm O-2 ≈ 660 µm

Las muestras clasificadas como MGS1, al no presentar evidencias de DMAE, serán empleadas como controles. Por su parte, las muestras clasificadas como MGS2, MGS3 y MGS4 representarán los progresivos estadios de la enfermedad. Sobre las 523 retinas *postmorten* fue realizado un análisis de expresión génica mediante RNA-seq, el cual proporcionó, de media, 32.5 millones de lecturas con una coincidencia del 94% con el genoma de referencia (*homo sapiens*).

7.3 - *Software* utilizado

El crecimiento del uso de técnicas de secuenciación de nueva generación como RNA-seq en estudios biomédicos, ha provocado la aparición de proyectos que pretenden desarrollar nuevos métodos para modelizar y analizar los datos obtenidos por las mismas. Para implementar estos métodos, así como para almacenar, acceder y organizar la gran cantidad de datos disponibles, han sido necesarias nuevas herramientas bioinformáticas, computacionales y estadísticas. En el caso de los datos de expresión génica, para realizar un análisis más profundo se necesita disponer de software y entornos de desarrollo avanzado donde se puedan implementar o usar distintas herramientas y algoritmos computacionales y estadísticos de análisis de datos. Muchos de los programas o plataformas existentes son fáciles e intuitivos de utilizar, pero no suelen ser muy útiles para enfrentarse a análisis complejos, bien porque no tienen implementados muchos métodos, o simplemente porque no es posible automatizar tareas o desarrollar nuevos métodos basados en código propio. Frente a estos programas cerrados existe la posibilidad de analizar datos biológicos complejos utilizando software estadístico y lenguajes de programación avanzado, como Matlab o R, con librerías específicamente diseñadas para el análisis de datos de expresión génica.

Para este trabajo se ha decidido seleccionar R (https://www.r-project.org), cuya primera versión fue desarrollada por Robert Gentleman y Ross Ihaka en 1993, ya que es una potente herramienta de análisis estadístico y gráfico. Además, se trata de un *software* de libre distribución que se puede descargar directamente de su propia página oficial (https://cran.r-project.org/) y cuyo código se distribuye en forma de librerías agrupadas por paquetes listos para su utilización. Este trabajo se ha realizado a través de la interfaz *RStudio*, un Entorno de Desarrollo Integrado (IDE) de código abierto para R que puede instalarse gratuitamente de su sitio oficial (https://posit.co). Se ha empleado la versión más actual a la fecha de inicio de este trabajo R-4.2.2 lanzada el 31 de octubre de 2022, así como la versión 2023.03.0+386 "*Cherry Blossom*" de *RStudio* lanzada el 9 de marzo de 2023. En concreto, para el análisis de datos del ámbito de la bioinformática, R ofrece la posibilidad de trabajar con *Bioconductor* (https://bioconductor.org), un proyecto que comenzó en 2001 con el desarrollo de *software* abierto para el análisis e interpretación de datos genómicos. Actualmente, prácticamente la mayoría de los métodos disponibles en análisis de RNA-seq tiene su propio paquete en este entorno. Todos los métodos, análisis y resultados mostrados en este trabajo, se han desarrollado en este marco.

Finalmente, también se hará empleo del paquete de R `Shiny` [28], este permite construir aplicaciones web interactivas directamente a partir de los códigos desarrollados en R.

Capítulo 8 - Metodología

El análisis de los datos de RNA-seq va a seguir la siguiente línea de trabajo: al partir de unos datos que ya han sido depurados previo a su publicación en GEO, no se requiere de esta etapa inicial. Aún así serán analizados para un entendimiento más profundo de los datos de partida. El flujo de trabajo con RNA-seq puede seguir dos metodologías: cuantificar respecto un genoma de referencia o clasificar respecto a un transcriptoma [29]. En este trabajo, se va a realizar la primera de ellas puesto que es la información que se dispone y los resultados obtenidos por cualquiera de ellas son igualmente válidos. Los valores de expresión génica se estimarán como el número de conteos, cuyas unidades típicas son '*Reads Per Kilobase of transcript per Million reads mapped*' (RPKM), '*Fragments Per Kilobase of transcripts per Million reads mapped*' (FPKM) o '*Transcripts Per Million*' (TPM). Finalmente, un análisis de expresión diferencial para hallar los genes diferencialmente expresados para cada grupo de sujetos de estudio (MGS1-MGS4) que, además, presenten un perfil de expresión similar entre ellos. Una vez se tengan los genes diferencialmente expresados para los pacientes de DMAE según su severidad, se procederá a desarrollar un clasificador e integrarlo en una aplicación web desarrollada mediante el paquete de R `Shiny`. En la Figura 9 se presenta de manera esquemática los pasos del protocolo que se va a seguir en este análisis para tratar los datos de RNA-seq.

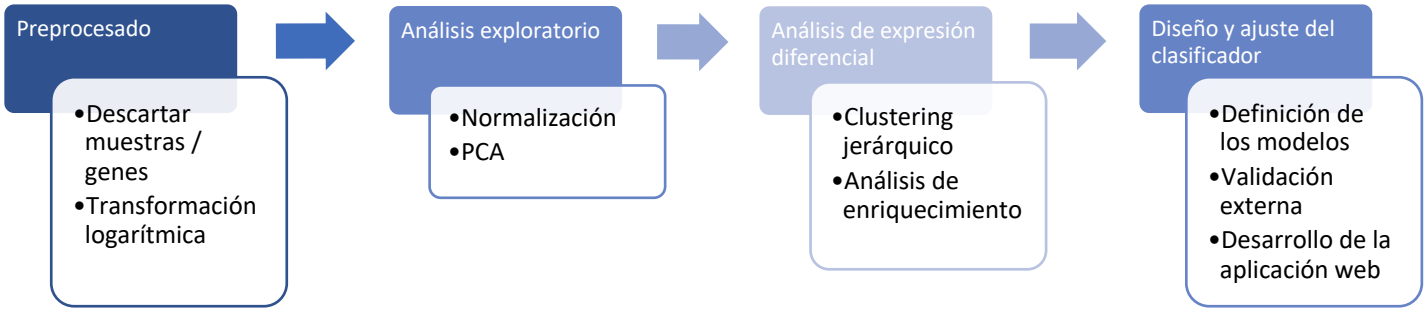

Figura 9 *Flujo de trabajo con RNA-seq empleado*

Todos los acrónimos empleados a lo largo del trabajo están recogidos en el Anexo I. Asimismo, todas las rutinas de R a las que se hace referencia a lo largo de los apartados de este capítulo y el siguiente están disponibles en el Anexo II.

8.1 - Recopilación de los datos y metadatos de RNA-seq

En la base de datos pública GEO (https://www.ncbi.nlm.nih.gov/geo/) se encuentran generalmente tres ficheros con información distinta acerca del estudio de RNA-seq: un archivo "(código de acceso)_DE_analysis.txt" con información relevante sobre los genes estudiados como el cromosoma en el que se ubican o si se trata de genes codificantes o no; otro archivo "(código de acceso)_RSEM_gene_counts.tsv" con la matriz de conteos sin procesar (una matriz de números enteros positivos); y un último archivo "(código de acceso)_series_matrix.txt", una matriz con toda la información fenotípica de cada muestra observada (e.g. sexo y edad del paciente al que pertenece la muestra). Una vez descargados se puede trabajar con los mismos.

8.2 - Depuración y transformación de los datos de conteo

Previo al propio análisis se debe realizar un preprocesado de los datos de conteo cuyo objetivo es eliminar las muestras y genes que no pueden ser incluidos en las etapas posteriores. Para empezar y como ya se adelantaba antes, las muestras que no han sido inequívocamente clasificadas en uno de los niveles MGS deben ser descartadas.

Además, se deben eliminar los genes poco informativos, es decir, aquellos que mayoritariamente tienen una expresión nula en todas las muestras. Teniendo en cuenta la distribución de los grupos, se descartan aquellos genes que no estén expresados en al menos el 90% de las muestras correspondientes a cada grupo, es decir, se descartarán los genes cuyo valor de expresión sea un cero en, al menos, el 90% de muestras de cada nivel.

Finalmente, entre las variables fenotípicas observadas, hay una de gran interés para este apartado: el número de integridad de ARN (RIN, del inglés *RNA Integrity Number*). El RIN es una medida directa de la calidad del RNA-seq, ya que un valor superior a 8 es indicativo de alta calidad (siendo aceptables valores entre 6 y 8 en caso de ARN fragmentado). Sin embargo, valores inferiores a 5 indican una mala integridad del RNA-seq [30]. Por lo que las muestras con un RIN inferior a ese mínimo deberán ser descartadas. Además, como se indicaba en los criterios de exclusión de la población de estudio, aquellas muestras que no hayan sido correctamente clasificadas dentro del MGS deberá ser igualmente descartadas.

Por otra parte, en el contexto de análisis de expresión génica, es muy habitual utilizar el logaritmo en base 2 (log2) de los datos de conteo; ya que esta transformación normalizará las distribuciones, por lo general muy asimétricas a la derecha, así como facilitará la interpretación de los resultados. Se debe notar que es posible encontrar valores nulos, por lo que se debe sumar una constante positiva antes de realizar la transformación logarítmica. De hecho, para medir los cambios de expresión en cada uno de los genes se empleará el *fold-change* (FC), definido como el cociente entre el nivel de expresión en la muestra de interés, respecto al

valor de expresión tomado como referencia. Un valor de FC en el rango (1,∞) indica sobre-expresión del gen en la muestra de interés respecto de la de referencia y un valor de FC en el rango [0,1) indica infra-expresión del gen en la muestra de interés respecto de la de referencia. La falta de simetría en este índice no facilita su interpretación. Sin embargo, en la escala logarítmica (logFC), dicho cociente se convierte en una diferencia, haciendo que un determinado valor y su recíproco sean simétricos. Valores positivos indicarán sobre-expresión, negativos indican infra-expresión y un logFC nulo indica que dicho gen se expresa a nivel constante. La elección de la base es dos para que así cambios de una unidad en la escala logarítmica doblen su valor en la escala original.

8.3 - Análisis exploratorio de los datos de conteo

Se va a trabajar con la transformación logarítmica de los datos de conteo. El efecto de esta transformación sobre un conjunto de datos se ilustra en la Figura 10 donde se representa la distribución del número de conteos de dos muestras antes y después de efectuar una transformación logarítmica en base 2.

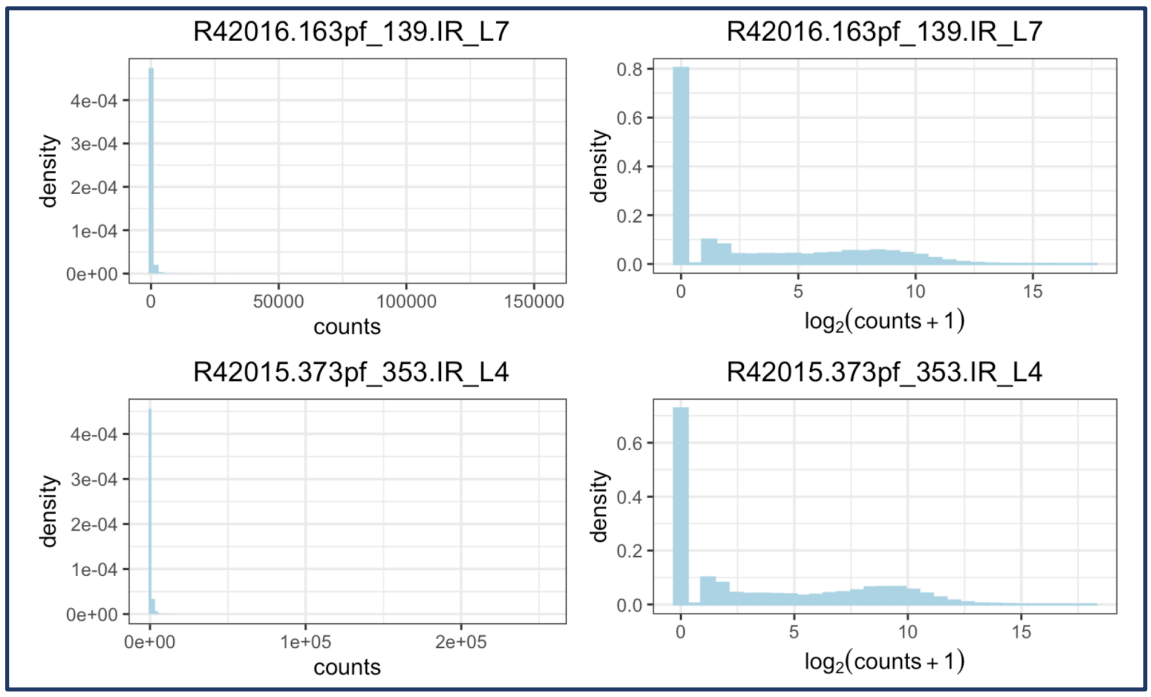

Figura 10 *Efecto de la transformación logarítmica sobre la distribución de los conteos*

Por otra parte, hay que notar que los datos de RNA-seq no son homocedásticos, es decir, su varianza depende de la cantidad observada, de hecho, para este caso la relación es directamente proporcional. Este comportamiento se corrige también mediante la transformación logarítmica ya comentada. Para ilustrarlo, se va a calcular los estadísticos de primer y segundo orden para los datos de conteo y de su transformación logarítmica, representados en la Figura 11. En definitiva, con una simple transformación logarítmica de los datos de conteo es suficiente para proceder con el resto del análisis.

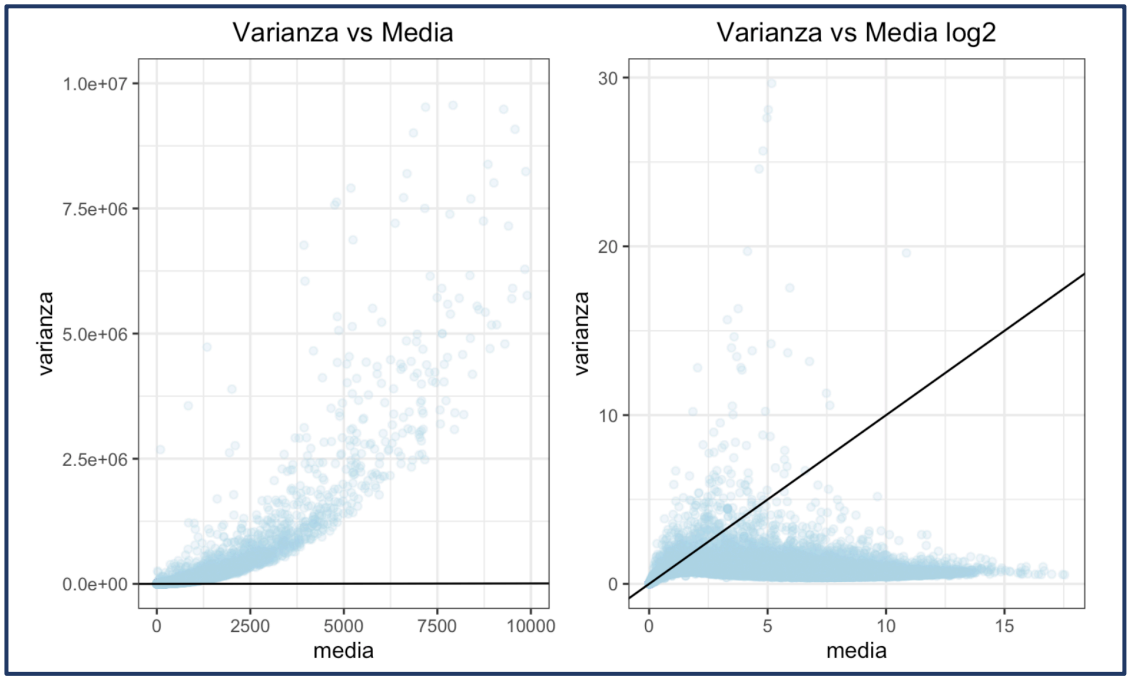

Figura 11 *Varianza frente a media de los datos de conteo*

8.3.1 - Normalización de datos de conteo

Se ha hablado del concepto "normalizar la distribución de los datos", normalizar en este contexto consiste en encontrar una escala común a todas las muestras y/o individuos para evitar falsos positivos, ya que un gen con una longitud menor puede mostrar un número menor de lecturas que otro con mayor longitud, sin que esto signifique que está diferencialmente expresado. Por tanto, en el análisis de datos de RNA-seq normalizar los datos de conteo es importante para minimizar el ruido técnico introducido durante el proceso de secuenciación con el fin de hacerlos comparables entre sí, un objetivo algo diferente a lo que se suele entender como normalización en términos estadísticos. En el contexto del análisis de expresión, el fin último de normalizar es identificar y eliminar las diferencias sistemáticas entre muestras que ocurren por fuentes de variabilidad distintas a la biológica.

Hay varios métodos de normalización, muchos de ellos basados en la estrategia general de, una vez elegido un nivel de referencia, expresar los conteos relativos a esa referencia. La cuestión de cuál es mejor, es una cuestión no resuelta aún. En este trabajo se van a utilizar tres métodos globales de normalización. En todos ellos, la estrategia es encontrar un factor de normalización para cada muestra:

- Normalización por el número total de lecturas (TC, del inglés *Total read Count*): consiste en dividir los datos de conteo para cada gen en una muestra por el número total de lecturas en dicha muestra. Es uno de los más utilizados, aunque no es muy sofisticado [31].

- Normalización por la log expresión relativa (RLE, del inglés *Relative Log Expression*): se calcula dividiendo el valor de conteo de cada gen y la mediana de expresión del mismo. Por ello a este método también se le llama mediana de los cocientes. Una vez que los datos han sido normalizados para cada gen, se busca la constante de normalización por muestra como la correspondiente mediana de los datos de conteo en la muestra [32].

- Normalización por la media recortada de los M-valores (TMM, del inglés *Trimmed Mean of M-values*): el método TMM se trata de un método global en el que cada muestra tendrá un factor de normalización denotado por $C_j, j = 1, \dots N$, donde N es el número de muestras. Se basa en la utilización de la media α% recortada, es decir, la media muestral obtenida tras eliminar el α% de las observaciones más extremas por arriba y el α% de las más extremas por abajo.
  Sea K_{gj} el conteo observado para el gen g $(g = 1, \dots, G)$ en la muestra j y $D_j = \sum_{g=1}^{G} K_{gj}$ el número total de lecturas en la muestra j, este método plantea un recorte doble, tanto en el *log fold change* del gen g para las muestras j y r, definido como,

$$M_g(j,r) = log_2\left({K_{gj}}/{D_j}\right) - log_2\left({K_{gr}}/{D_r}\right)$$

como en la intensidad media,

$$A_g(j,r) = \frac{1}{2}\left(log_2\left({K_{gj}}/{D_j}\right) + log_2\left({K_{gr}}/{D_r}\right)\right)$$

El factor de corrección para la muestra j se calcula como,

$$C_j = \frac{exp\left\{\frac{1}{N}\sum_{l=1}^{N} log\left(2^{TMM(j,r)}\right)\right\}}{2^{TMM(j,r)}}$$

donde $TMM(j,r)$ se calcula como una media ponderada de los valores de M, utilizando como pesos una aproximación a la inversa de su varianza, es decir,

$$\omega_g(j,r) = \left(\frac{D_j - K_{gj}}{D_j K_{gj}} + \frac{D_r - K_{gr}}{D_r K_{gr}}\right)^{-1}$$

y, por tanto,

$$TMM(j,r) = \frac{\sum_{g \in G^*} \omega_g(j,r) M_g(j,r)}{\sum_{g \in G^*} \omega_g(j,r)}$$

con G^* representando el conjunto de genes no recortados.

8.3.2 - Distribución de datos de conteo según su grupo

Una vez estudiada la distribución de los datos de conteo, se pasa a estudiar la distribución para cada grado de severidad de DMAE. Como ya se ha mencionado, cada individuo de la muestra está clasificado en uno de 4 grupos según los niveles del MGS. Para ello, se emplean herramientas gráficas como el diagrama de caja o *boxplot*, disponible en la librería `ggplot2` de R [33]. Para entenderlo, es necesario definir brevemente una serie de términos, que además se representan en la Figura 12:

- Primer cuartil (Q_1): valor para el cual el 25% de los datos son menores o iguales.
- Mediana o segundo cuartil: valor para el cual el 50% de los datos son menores o iguales, divide la distribución en dos partes iguales.
- Tercer cuartil (Q_3): valor para el cual el 75% de los datos son menores o iguales.
- Rango intercuartil (IQR): diferencia entre el tercer y primer cuartil ($Q_3 - Q_1$).

De esta manera, se define una regla para identificar *outliers*: toda observación demasiado alejada de la distribución de modo que su valor sea mayor que $Q_3 + 1.5 \cdot IQR$ o menor que $Q_1 - 1.5 \cdot IQR$.

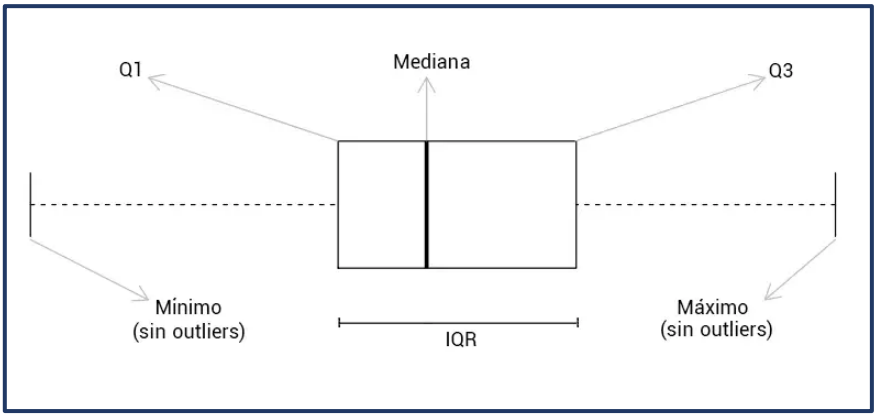

Figura 12 *Elementos de un boxplot. Tomada de H. Wickham, 2016 [33]*

8.3.3 - Análisis de Componentes Principales (PCA)

Para finalizar con el análisis exploratorio de los datos de conteo, se sugiere la búsqueda de posibles patrones de correlación entre los distintos grupos de muestras, para ello se emplea el Análisis de Componentes Principales (PCA), el cual permite encontrar componentes ortogonales representativas de los datos, cada una de estas componentes es una combinación lineal de las variables originales (los genes). Normalmente, se debe tener en consideración que en la matriz de conteo exista un ruido de fondo en alguna de las muestras debido a una mala secuenciación o algún error durante el proceso de obtención del conteo, si tal fuera el caso, se ocasionaría un resultado del PCA equivocado ya que dicha muestra o conjunto de muestras obtendría un valor demasiado relevante cuando en realidad su información es errónea. Sin embargo, uno de los objetivos logrados al normalizar los datos es precisamente la eliminación de este tipo de ruido, por lo que tras las etapas anteriores los datos ya están preparados para proceder con el PCA. Este método estadístico permite analizar datos de alta dimensión mediante su transformación a un espacio de menor dimensión [34]. Esto es, a partir de todas las variables de entrada (genes en este caso) el PCA permite encontrar unas pocas variables independientes entre sí obtenidas como combinación de estas primeras, de modo que se mejore la comprensión de los datos. Para realizar este tipo de análisis se deben seguir cinco pasos:

1. Normalización de los datos, como se explicó previamente en este capítulo.

2. Computación de la matriz de covarianza, esta es una matriz simétrica en la que el elemento en posición (i,j) es la covarianza entre los niveles de expresión de los genes i-ésimo y j-ésimo, una medida de la variación de forma conjunta de ambas variables respecto a sus medias. Si la covarianza es negativa, hay una relación negativa de modo que cuando el valor de conteo de un gen sube, el del otro baja y viceversa; si es positiva, la relación es positiva, ambos suben o bajan a la vez; si es nula, no hay relación entre los genes.

3. Obtención de autovalores, en esencia son números enteros positivos que representan la cantidad de varianza explicada por esa componente, además, tienen asociada una dirección definida en el nuevo espacio, dicha dirección es conocida como autovector.

4. Selección de las componentes principales, ya que habrá tantos pares de autovalores y autovectores como el número de variables y no todos ellos serán igual de importantes. Un criterio habitual para determinar el número de componentes consiste en seleccionar aquellas con un autovalor superior a la unidad. Además, se deberán ordenar los autovectores por su respectivo autovalor, de modo que la primera componente principal será la de mayor autovalor, la segunda componente la del segundo mayor autovalor, y así con el resto. Este orden coincide con el orden de importancia en cuanto a la variabilidad total que recogen de la muestra.

5. Transformación de los datos en el nuevo espacio, al haber hecho una selección previa de componentes, la dimensión de este nuevo espacio es menor que la del espacio original. Para ello, se deben reorientar los datos originales multiplicándolos por los autovalores seleccionados.

Se debe recordar que el resultado del PCA no es una modificación de los datos originales sino una nueva perspectiva de estos que puede aportar una mejor representación.

8.4 - Análisis de expresión diferencial

Como se lleva adelantando durante gran parte del trabajo, el punto clave del mismo consiste en localizar aquellos genes que se expresan diferencialmente en la DMAE, para ello, se va a recurrir a un modelo lineal generalizado (GLM, del inglés *General Linear Model*). Los GLMs, propuestos por McCullagh y Nelder [35], son modelos lineales que tratan de estimar la relación entre una variable respuesta Y, cuya distribución pertenece a la familia exponencial, con p variables explicativas, a través de una función de enlace. Dada una muestra de tamaño n, un GLM se compone de un predictor lineal de la forma,

$$\eta_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} \;\; i = 1, \dots, n$$

y dos funciones:

- La función de enlace que describe como la media de Y, μ_i, depende del predictor lineal, $g(\mu_i) = \eta_i$
- Una función de varianza que describe como la varianza depende de la media, $var(Y_i) = \phi V(\mu_i)$ donde el parámetro de dispersión ϕ es una constante.

Los GLMs generalizan los modelos de regresión lineal. En los modelos de regresión la distribución de la variable respuesta es $Y_i \sim N(x_i^T \beta, \sigma^2)$, es decir que,

$$\eta_i = x_i^T \beta = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

con función de enlace $g(\mu_i) = \mu_i$ y función de varianza $V(\mu_i) = 1$ y $\phi = \sigma^2$.

En R se puede utilizar la función `lm()` para ajustar modelos lineales y `glm()` para los GLM. En el caso de datos de RNA-seq el paquete `edgeR` [36] tiene implementada la función `glmFit()` para estos ajustes.

Previo al diseño del modelo, se deben estimar dos parámetros de dispersión: la variabilidad común entre todas las variables aleatorias del estudio (genes) y la dispersión específica de cada gen (ϕ_g). Ambos se pueden computar fácilmente con R mediante el estimador de máxima verosimilitud (MLE, del inglés *Maximum Likelihood Estimator*), esto es, el estimador que se obtiene maximizando la función de verosimilitud de la muestra $L(x_1, x_2, \dots, x_n|\theta)$, la cual asigna la probabilidad de

que se obtenga una muestra dependiendo del parámetro (o parámetros) θ. El MLE de θ es una función de la muestra, $\hat{\theta} = \hat{\theta}(X_1, X_2, \dots, X_n)$ tal que

$$L\left(x_1, x_2, \dots, x_n \middle| \hat{\theta}\right) = \max_{\theta \epsilon \Theta} L(x_1, x_2, \dots, x_n|\theta)$$

y que puede obtenerse resolviendo la ecuación,

$$\frac{\delta L(x_1, x_2, \dots, x_n|\theta)}{\delta \theta} = 0$$

Una vez estimados los parámetros de dispersión, se puede hallar el coeficiente de variación biológica (BCV, del inglés *Biological Coefficient of Variation*), que es la raíz cuadrada de la dispersión común. El BCV es un factor que influye en el número de genes diferencialmente expresados, de hecho, a mayor variación entre grupos respecto a la variación intragrupo más fácil será la detección de genes diferencialmente expresados. Se puede representar gráficamente mediante un plot BCV, un gráfico en el que se representa la raíz cuadrada de las dispersiones estimadas en el eje vertical frente al log2 de los conteos en el eje horizontal, donde cada punto representa un gen concreto y se representa mediante una línea roja la dispersión común. Se considera razonable un valor de variabilidad común en el intervalo [0.2- 0.4].

La formulación de los GLM queda determinada por las variables explicativas, edad, sexo, tratamiento, severidad de la patología… , que se consideren. Esta información se recoge en la matriz de diseño, una matriz con tantas filas como muestras y columnas como variables a incluir. Como en cualquier modelo lineal, se debe tener cuidado con el problema de la multicolinealidad. Este problema ocurre cuando las variables explicativas incluidas en el modelo están altamente correladas. Por ejemplo, dado un conjunto de muestras en la que algunas de ellas corresponden a pacientes tratados con uno de los dos tratamientos disponibles para una enfermedad, si se dispone de una variable fenotípica que recoja si una muestra ha recibido tratamiento o no y, a su vez, otras dos variables que indiquen si ha recibido la primera o la segunda opción de tratamiento; se cometería un error de diseño si en la matriz de diseño se incluyen las tres variables, ya que la primera es la suma de las otras dos, por lo que está altamente correlada con ellas. Los efectos de este problema en el ajuste del modelo son muy importantes, y en un caso extremo podría incluso no encontrarse una solución factible, por lo que es importante seleccionar bien qué variables explicativas incluir en los modelos. Adicionalmente, se debe considerar cuándo incluir un término constante o *intercept* (β_0), y cómo afecta al modelo subyacente, esto es un término adicional que no guarda relación con ninguna de las variables explicativas, pero ayuda a ajustar el modelo. Su inclusión proporciona un ajuste más flexible a los datos. Un modelo sin *intercept* sólo se recomendaría en casos en los que exista una razón biológica de peso por la que podría asociarse este mismo con un valor de expresión cero, pero tales contextos son raros en la modelización de la expresión génica [37].

Finalmente, los modelos ajustados por un gen se pueden emplear para resolver contrastes de hipótesis relacionados con sus parámetros. Como se van a realizar miles de contrastes de hipótesis durante el análisis (tantos como genes), con certeza van a aparecer muchos falsos positivos, lo cual es un grave problema a tener en cuenta. Para solventar el problema de comparaciones múltiples, se debe ajustar los p-valores. Uno de los métodos de ajuste más habituales en este ámbito es el método de Benjamini & Hochberg [38]. Una vez ajustados los p-valores, el criterio para seleccionar los genes diferencialmente expresados será aquellos cuyo p-valor ajustado sea significativo a nivel 0.05 y cuyo log *fold-change* sea mayor que 1. Finalmente, se pueden visualizar gráficamente los genes diferencialmente expresados que hayan salido resultado del análisis mediante *Volcano plots,* estos son gráficos que permiten comparar los niveles de expresión génica en dos condiciones diferentes. El gráfico muestra el logFC en abcisas y la significación estadística ($-\log_{10}$(p-valor)) en ordenadas. Asimismo, para comparar los conjuntos de genes seleccionados será útil emplear diagramas de Venn, representaciones de conjuntos que muestran las relaciones de intersección, inclusión y disyunción entre los grupos de genes que se han relacionado con cada comparación. De esta manera se podrá comparar de manera gráfica los resultados.

8.5 - Agrupación de genes

Como resultado del análisis de expresión diferencial se obtiene una lista de genes, sobre el cual, es muy interesante tratar de agrupar genes similares en cuanto a niveles de expresión. De este modo, se puede conseguir un entendimiento mayor del listado de genes diferencialmente expresados que se hayan agrupado debido a un patrón de expresión similar, ya que estos son normalmente regulados por los mismos mecanismos. Esto es, los genes co-expresados son (habitualmente) también genes co-regulados [39]. Una técnica sencilla de agrupación de genes por similitud se denomina *clustering* jerárquico. Como se van a comparar datos de expresión génica, se vuelve a remarcar la importancia de trabajar con los datos normalizados ya que, en caso contrario, los genes altamente expresados se agruparían conjuntamente, aunque presenten patrones de expresión diferentes. Con estos datos, se construirá un gráfico en forma de árbol conocido como dendrograma, en él se muestran los grupos que se han encontrado en el conjunto de datos, de modo que en el eje de abscisas se representan todos los datos ordenados por los grupos encontrados, y en ordenadas se representa el nivel de similitud o distancia que existe entre los grupos. El dendrograma se puede construir tanto para las muestras como para los genes diferencialmente expresados, de hecho, de esta manera se puede obtener el *heatmap* o mapa de calor del conjunto de datos, esto es, una representación visual en dos dimensiones del conjunto de datos en la que los genes se representan por columnas y las muestras por filas, todos ellos reordenados de acuerdo con los grupos obtenidos en sus respectivos dendrogramas.

El *clustering* jerárquico aglomerativo es un algoritmo de aprendizaje no supervisado que se utiliza para agrupar datos similares en *clusters*. El proceso comienza considerando cada variable (o muestra) como un *cluster* independiente y, a continuación, se combinan iterativamente pares de *clusters* hasta que todos los datos pertenecen a un único clúster. Para medir la no similitud entre los datos se debe utilizar una medida de distancia. En este caso, se utiliza la inversa de la correlación. Asimismo, como método de *clustering* se emplea la vinculación completa (*complete linkage method*), un método utilizado para medir la distancia entre *clusters*. Calcula la distancia entre dos *clusters* como la distancia máxima entre dos puntos cualesquiera (uno de cada *cluster*), es decir, la distancia entre las dos observaciones más alejadas de cada *cluster*.

Así pues, para realizar el método de *clustering* jerárquico aglomerativo con medida distancia la inversa de la correlación y método de vinculación completa implica los siguientes pasos:

1. Calcular la matriz de correlaciones de los datos.
2. Utilizar la matriz de correlación inversa como matriz de distancia.
3. Iniciar con cada dato individual como un *cluster* independiente.
4. Encontrar los dos *clusters* más cercanos según la medida de distancia.
5. Fusionar los dos *clusters* más cercanos en uno solo.
6. Volver a calcular la matriz de distancias entre los nuevos *clusters*.
7. Repetir 4-6 hasta que todos los datos pertenezcan al mismo *cluster.*

No hace falta implementar manualmente este algoritmo, ya que se encuentra disponible en R, bastará con emplear la función `hclust()` a la que se deben pasar los datos, así como indicar la medida de la distancia y el método de *clustering*.

Para determinar el número de grupos que razonablemente representen los patrones de expresión dentro del conjunto de genes, se debe atender a una medida del error cometido al variar el número de *clusters*. La más típicamente empleada es la suma de cuadrados de estimación de errores (SSE, del inglés *Sum of Squared Error*), que se define como la suma de los cuadrados de la distancia entre cada gen de un *cluster* y el "núcleo" de ese *cluster*. A medida que se aumenta el número de *cluster*, la distancia entre cualquier gen y su núcleo será cada vez menor ya que el propio *cluster* en sí será más pequeño. Sin embargo, a partir un determinado número de *cluster*, la SSE no disminuirá significativamente y ese será precisamente el número de *clusters* que se deben considerar como óptimo.

En conclusión, al agrupar los genes diferencialmente expresados que se han obtenido en el paso anterior, tendrán ahora además un perfil de expresión similar, pudiendo implicar que están regulados por los mismos mecanismos.

8.6 - Análisis de enriquecimiento

El análisis de enriquecimiento es un método ampliamente utilizado en bioinformática que ayuda a comprender la importancia biológica de un grupo de genes, como una lista de genes expresados diferencialmente, mediante la identificación de las vías funcionales y los términos de *Gene Ontology* (GO) que están sobre-representados o infra-representados en ese grupo de genes en comparación con lo que cabría esperar por azar. Su propósito es obtener información sobre los procesos biológicos, las funciones moleculares y los componentes celulares que se ven afectados por los cambios observados en la expresión génica. De modo que el análisis de enriquecimiento puede ayudar a:

- Identificar las vías y procesos biológicos que están significativamente enriquecidos en los genes diferencialmente expresados, proporcionando conocimientos sobre los mecanismos biológicos subyacentes que impulsan los cambios de expresión génica observados.

- Descubrir posibles nuevas funciones y vías que no se conocían previamente asociadas a este conjunto de genes.

- Facilitar la interpretación y validación de los resultados proporcionando una anotación funcional de los genes en términos de procesos biológicos, funciones moleculares y componentes celulares.

- Priorizar genes para su posterior validación experimental.

El procedimiento para realizar el análisis de enriquecimiento en GO a partir del listado de genes diferencialmente expresados se puede realizar a través de la función `gseGO()` de `clusterProfiler` [40]. Esta función implementa el método GSEA (del inglés *Gene Set Enrichment Analysis*), uno de los métodos más populares para realizar el enriquecimiento basado en una modificación del estadístico de Kolmogorov-Smirnov para dos muestras. De una manera sencilla, este método ordena los genes de toda la población en función de un criterio dado, en este caso su expresión diferencial, y estudia si la distribución de los genes pertenecientes al conjunto de genes obtenido es significativamente distinta de la distribución de los genes no pertenecientes a dicho conjunto [41]. Como recordatorio, se debe anotar los genes con términos GO (ya que esta información no está disponible en los ficheros de datos que se descargaron de GEO) y se debe determinar una ontología de GO a utilizar o incluso emplear las tres. Para determinar los genes sobre-representados e infra-representados, se realizarán miles de contrastes por lo que, al igual que en el análisis de expresión diferencial, se deberá realizar una corrección de p-valores. Finalmente, los resultados se presentan como una lista de términos GO significativamente enriquecidos. Para visualizarlos se utiliza el paquete `enrichplot` [42] de R y se representa tanto su *dotplot* como su mapa de enriquecimiento. El primero muestra los términos sobre-representados e infra-representados mediante puntos, el segundo organiza los términos en una red conectando categorías que tienen anotados genes en común.

8.7 - Diseño y ajuste de un clasificador

8.7.1 - Definición del modelo

Finalmente, se diseñará un método capaz de predecir que un paciente padezca DMAE y su nivel MGS asociado, a partir de los datos de expresión de los genes diferencialmente expresados que se han obtenido previamente. Para ello, se va a emplear un algoritmo de aprendizaje supervisado, en el que a partir de un conjunto de datos (cuya clase es conocida) se establezca un modelo que permita predecir la clase a la que pertenecería un nuevo paciente. En concreto, se va a hacer uso del algoritmo *Support Vector Machine* (SVM), un clasificador que se basa en encontrar la diferencia máxima entre clases en múltiples dimensiones de los datos. SVM es aplicable a problemas de regresión y clasificación, aunque se usa más comúnmente como modelo de clasificación para casos con límites no lineales entre clases. El conjunto total de muestras se divide en dos: un set de entrenamiento que servirá para ajustar los modelos y un set de test para validarlos externamente. Se sugiere emplear 2/3 de la muestra total seleccionados aleatoriamente para el set de entrenamiento y el restante para el set de test.

Se van a ajustar tres modelos diferentes:

1. El modelo que mejor distinga entre controles (MGS1) y patológicos (MGS2 + MGS3 + MGS4)
2. El modelo que mejor distinga entre niveles intermedios (MGS2 y MGS3)
3. El modelo que mejor distinga la DMAE avanzada (MGS4) respecto a todos los demás grupos

Como se ha dicho, la metodología que se va a emplear para el ajuste de estos modelos es SVM. Esta técnica tiene como objetivo obtener un hiperplano (subespacio de una dimensión menor a la del espacio que pertenece, como son, por ejemplo, las rectas en el plano bidimensional) óptimo capaz de separar lo mejor posible dos clases. Es decir, el modelo asignará a cada nueva observación un grupo u otro en función de a qué lado del hiperplano se localice. Para poder tratar con límites no lineales entre clases se requiere aumentar el espacio de variables para realizar la predicción, mediante el empleo de *kernels*, funciones que cuantifican la similitud entre dos observaciones en un nuevo espacio de mayor dimensión [38]. En este caso se elegirá la mejor entre un *kernel* lineal, radial, polinomial de grado tres y sigmoidal. Todo ello se podrá realizar gracias a la función `svm()` disponible en el paquete de R `e1071` [43].

Para cada uno de los tres modelos indicados, se deberá seleccionar los genes diferencialmente expresados más relevantes utilizando el algoritmo de mínima redundancia-máxima relevancia (mRMR del inglés *Minimum Redundancy – Maximum Relevance algorithm*) [44] que se basa en la idea del equilibrio entre "relevancia" respecto del grupo y "redundancia" entre genes. La relevancia y redundancia se cuantifican usando la información mutua (Mutual Information, MI). Se utilizará el paquete *mRMRe* [45] de R, en el que MI se estima como,

$$I(X,Y) = -\frac{1}{2} ln(1 - \rho(X,Y)^2)$$

donde ρ es el coeficiente de correlación entre las variables X e Y.

Sea Y la variable principal, el grupo de DMAE en este caso, y $X = \{x_1, \dots, x_n\}$ la expresión en el conjunto de genes, el primer elemento del conjunto seleccionado, S, será x_i, el gen con mayor MI con Y. En un segundo paso se añade a este conjunto x_j, con $i \neq j$, el gen que mejor equilibre una alta relevancia y una baja redundancia, es decir el que maximiza la expresión,

$$q = I(x_j, Y) - \frac{1}{|S|} \sum_{x_{k \in S}} I(x_j, x_k)$$

donde $|S|$ denota el cardinal de S.

Se continuará añadiendo genes al conjunto S hasta alcanzar el número que se haya pre-fijado. Para determinar el número óptimo de genes a incluir, se utilizará el método LOOCV (*Leave One Out Cross Validation*), eligiendo el número que maximice el área bajo la curva ROC (AUC), una métrica de la capacidad discriminatoria de los modelos predictivos que varía entre 0 y 1. Se define como el área bajo la curva ROC (del inglés, *Receiver Operating Characteristic*). La curva ROC es una representación gráfica del rendimiento de un clasificador binario a medida que varía el umbral de discriminación. Un AUC de 1 indica una clasificación perfecta, mientras que un AUC de 0.5 indica un clasificador aleatorio. A partir de esta métrica, la estimación del número óptimo de genes consiste en los siguientes pasos:

1. La muestra se divide en un set de entrenamiento de tamaño n-1, donde n es el número de muestras, y un set de test, consistente en la única muestra no incluida en el set de entrenamiento.

2. Se aplica el algoritmo mRMR en el set de entrenamiento para encontrar un conjunto de m genes con máxima relevancia y mínima redundancia.

3. Se ajusta el modelo predictivo (en la muestra de entrenamiento) utilizando como variables dependientes los genes seleccionados en 2.

4. Aplicar el modelo al set de test para predecir el tipo de muestra. Se calcula la probabilidad de pertenecer a una clase determinada.

5. Se vuelve a particionar la muestra completa en un set de entrenamiento y un set de test, y se repiten los pasos 2-4 hasta que todas las muestras sean una vez el set de test. La probabilidad predicha para cada muestra se utiliza para construir la curva ROC y calcular el AUC.

6. El número *m* de genes que se selecciona en el paso 2 es un parámetro fijado. Para decidir cuál es el mejor *m* se repiten los pasos 1-5 comenzando por uno y añadiendo el gen más informativo cada vez hasta llegar al número total de genes diferencialmente expresados. Además, es conveniente comprobar si la

incorporación de un mayor número de genes aporta una mejora importante en la capacidad discriminatoria del modelo, ya que serán preferible modelos más sencillos, con menor número de genes, puesto que su interpretación será más sencilla y serán más fácilmente generalizables. Por ello, se establece una tolerancia de 0.1 unidades en el valor del AUC, de forma que *m* será el mínimo número de genes cuyo modelo tenga una AUC que no sea mejorada en más de 0.1 unidades por modelos construidos a partir de un conjunto de genes mayor.

8.7.2 - Valoración de la capacidad discriminatoria del modelo

Los modelos establecidos se evalúan a partir de la muestra externa, el set de test. Además de la métrica de rendimiento AUC ya definida en el apartado anterior, se ha decidido acudir a la matriz de confusión y a las métricas que se relacionan con la misma dado su clara e intuitiva interpretación. La matriz de confusión es una tabla de doble entrada que compara el resultado obtenido por el clasificador con la clasificación real, en cada fila se representa el número de predicciones de cada clase, mientras que en cada columna se indican los valores reales. Se dirá que el un resultado es positivo si se diagnostica como patológico, y negativo en caso contrario. De este modo la matriz de confusión recoge cuatro posibles valores:

- Verdadero Positivo (VP): el valor predicho y el real son positivos. En este caso, el paciente padece DMAE (en el estadio correspondiente) y el clasificador así lo predice.
- Falso Positivo (FP): el valor predicho es positivo y el real es negativo. El clasificador da un resultado positivo cuando el paciente realmente está sano. En estadística se conoce como error de tipo I.
- Verdadero Negativo (VN): el valor predicho y el real con negativos. Es decir, el paciente está sano y el clasificador no lo detecta como patológico.
- Falso Negativo (FN): el valor predicho es negativo y el real positivo. El paciente es enfermo de DMAE (en el estadio correpondiente), pero el clasificador no lo detecta. Esto se conoce en estadística como error de tipo II.

A partir de estos cuatro valores se deducen una serie de métricas que van a permitir evaluar los resultados, ya que miden cuantitativamente la precisión de estos:

- Exactitud: representa cuántas predicciones han sido realizadas correctamente, tanto positivas como negativas. Cuanto mayor sea su valor, más cercano es el resultado a la realidad.

$$Exactitud = \frac{VP + VN}{(VP + FP + VN + FN)} \in [0,1]$$

- Sensibilidad o Tasa de Verdaderos Positivos (TVP): es la probabilidad de que un resultado positivo real dé positivo. Cuanto mayor es su valor, mayor es la capacidad para detectar la enfermedad entre la totalidad de enfermos.

$$Sensibilidad = \frac{VP}{(VP + FN)} \in [0,1]$$

- Especificidad o Tasa de Verdaderos Negativos (TVN): es la probabilidad de que un resultado negativo real dé un resultado negativo. Cuanto mayor es su valor, mayor es la capacidad para detectar pacientes sanos.

$$Especificidad = \frac{VN}{(VN + FP)} \in [0,1]$$

- Tasa de Falsos Positivos (TFP): es la probabilidad de que se dé un resultado positivo cuando el valor verdadero es negativo.

$$TFP = \frac{FP}{(FP + TN)} \in [0,1]$$

- Tasa de Falsos Negativos (TFN) o tasa de error: es la probabilidad de que se pase por alto un verdadero positivo.

$$TFN = \frac{FN}{(FN + VP)} \in [0,1]$$

8.8 - Desarrollo de la aplicación web

Para desarrollar la herramienta se empleará `Shiny`, un paquete de R que permite crear aplicaciones *web* interactivas directamente a partir de código R. El paquete `Shiny` funciona proporcionando dos componentes: la interfaz de usuario (UI) y el servidor.

La interfaz de usuario se crea mediante una serie de funciones de R que generan código HTML, CSS y JavaScript. Estas funciones permiten crear una amplia variedad de componentes interactivos, como botones, controles deslizantes y campos de entrada, que permiten al usuario interactuar con la aplicación.

El componente servidor de la aplicación `Shiny` es el responsable de procesar la entrada del usuario y generar la salida. Se construye utilizando código R que realiza los cálculos necesarios y genera las salidas resultantes, como gráficos o tablas. Este componente también es responsable de responder a los cambios en la interfaz de usuario y actualizar la salida en consecuencia.

Las aplicaciones creadas con `Shiny` pueden lanzarse de varias formas, como ejecutándolas localmente dentro de RStudio o desplegándolas en un servidor web. Una de las principales ventajas de `Shiny` es que permite a los desarrolladores crear aplicaciones web interactivas con relativamente poca experiencia en desarrollo *web*. El paquete proporciona un entorno que abstrae muchos de los detalles de bajo nivel del desarrollo web, lo que permite a los desarrolladores centrarse en la funcionalidad y las características de la aplicación. Además, el paquete se integra perfectamente con otros paquetes populares de R para el análisis y la visualización de datos, como `ggplot2` [46] y `dplyr` [47], lo que permite crear aplicaciones potentes e interactivas basadas en datos.

Capítulo 9 - Resultados

Como ya se ha adelantado, los datos con los que se va a trabajar están disponibles de manera pública en la base de datos GEO con el código de identificación GSE115828: "GSE115828_DE_analysis.txt", "GSE115828_RSEM_gene_counts.tsv" y "GSE115828_series_matrix.txt". Tras realizar una exploración básica de la matriz de conteo se observa que tiene unas dimensiones de 58051 filas (igual al número de genes) por 524 columnas (523 muestras ya que la primera columna contiene el nombre del gen) y no presenta ningún valor perdido o no registrado. La cabecera de esta matriz de conteo se representa en la Figura 13.

GeneID <chr>	R42015.419pf_1.IR_L7 <dbl>	R42015.490pf_100.IR_L2 <dbl>	R42016.137pf_101.IR_L5 <dbl>
1 ENSG00000000003	347	225.00	252
2 ENSG00000000005	2	0.00	0
3 ENSG00000000419	602	254.00	301
4 ENSG00000000457	826	422.99	510
5 ENSG00000000460	572	272.00	310
6 ENSG00000000938	47	21.00	12

6 rows | 1-5 of 524 columns

Figura 13 *Cabecera de la matriz de conteos sin procesar*

9.1 - Depuración y transformación de los datos de conteo

Para analizar la integridad de las medidas de ARN se van a analizar los valores de RIN registrados para cada muestra, teniendo en cuenta que hay 5 muestras para las cuales no se ha determinado su RIN. Contando el número de muestras dentro de un rango y dividiendo por el número total de muestras menos los valores no registrados (523 – 5 = 518 muestras) se observa que 97 muestras (18.72%) son de alta calidad, 501 muestras (96.72%) son aceptables y que sólo 6 muestras (0.01%) no presentan una buena integridad. Por tanto, se deben descartar por tener un número de integridad inferior a cinco (o no registrado).

De igual modo se pasa a estudiar la clasificación de las muestras (nivel MGS), se puede notar que hay muestras que no han sido clasificadas ("*no grade*"), así como valores perdidos ("*NA*") o sin un diagnóstico claro ("*2 or 3*"). Todas ellas deben ser descartadas al no poder clasificarse dentro de uno de los grupos de la patología, dejando por tanto 505 de las 523 muestras iniciales para proseguir con el análisis.

Respecto a la exploración inicial de los genes, se ha observado que 39199 (67.53%) de ellos son irrelevantes debido a que presentan un nivel de expresión demasiado bajo para el 90% de las muestras en cada nivel MGS. Por tanto, de las 523 muestras con 58051 genes cada una, el conjunto de datos se ha reducido, tras este análisis inicial, en 505 muestras con 18852 genes cada una. En la Tabla 3 se encuentra un descriptivo de la muestra de estudio en función de su nivel MGS resumiendo sus principales características. Se puede observar que el grupo de nivel MGS4 es considerablemente más pequeño que el resto, el valor RIN medio es equivalente entre los distintos niveles MGS, la edad media es cada vez mayor en función del estadio de la DMAE, lo cual es lógico que ocurra así. Además, la proporción de mujeres en las muestras es cada vez mayor en comparación con la de hombres.

Tabla 3 *Descriptivo de la muestra de estudio*

	Tamaño muestral	***RIN***			***Edad***			***Sexo***			
		Media	IC 95%		Rango: 47, 107			Hombres		Mujeres	
					Media	IC 95%					
			Inf.	Sup.		Inf.	Sup.	%	n	%	n
MGS1	124	7.41	5.13	9.69	73.90	70.07	77.73	49.19	61	50.81	63
MGS2	191	7.43	5.14	9.72	77.55	73.91	81.19	52.36	100	47.64	91
MGS3	125	7.39	5.11	9.67	84.29	81.11	87.46	40.00	50	60.00	75
MGS4	65	7.45	5.16	9.74	87.86	85.01	90.71	36.92	24	63.08	41

9.2 - Análisis exploratorio

En la Figura 14 se visualiza la distribución de los conteos en todas las muestras tras realizar su transformación logarítmica (notar la escala en ordenadas).

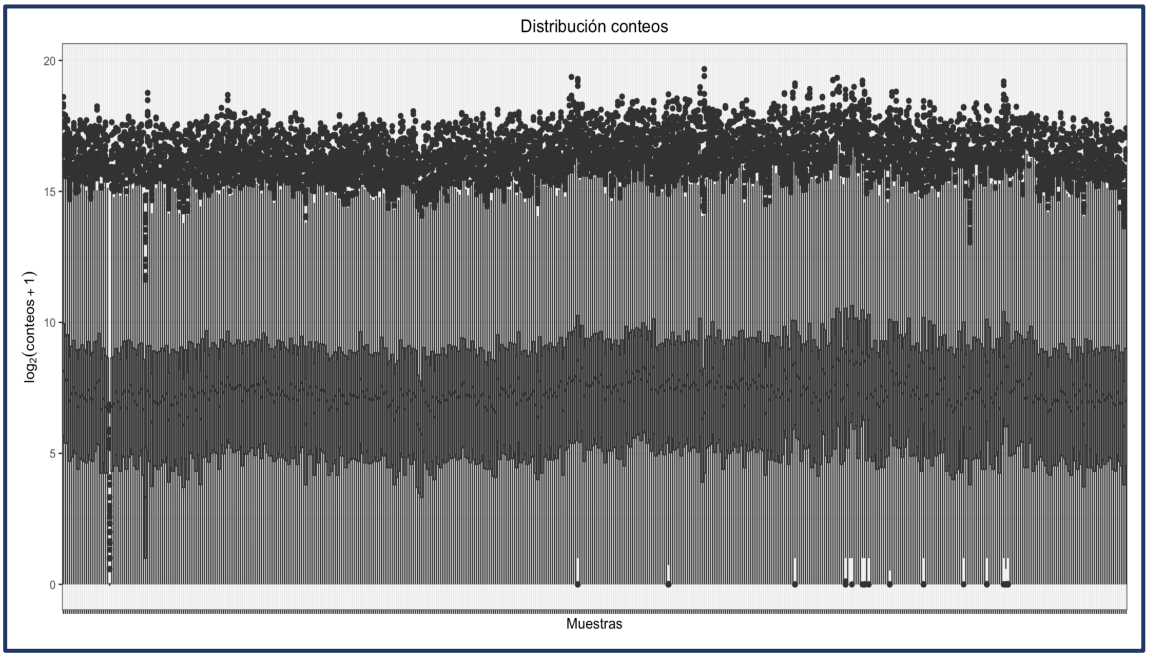

Figura 14 *Distribución conteos raw data*

Al ser una cantidad tan elevada de muestras, se dificulta su visualización. Por ello, se ha decidido extraer un conjunto aleatorio de muestras para cada nivel MGS de tamaño el 10% de su respectiva clase. En la Figura 15 se representa sus distribuciones a través de *boxplot* y gráfico de densidad. Aquí se puede observar diferencias entre las distribuciones por lo que se precisa normalizar los datos antes de proseguir con el análisis. Para ello, se ha probado a emplear los tres métodos de normalización indicados en la metodología del trabajo (TC, RLE y TMM) y su resultado se ilustra en la Figura 16, viendo que es igualmente válido emplear cualquiera de los tres métodos para normalizar la distribución de los conteos. De este modo, se decide seguir procediendo con los datos normalizados mediante el método TMM, ya que es el método de normalización más popular en el caso de que haya una mayoría de genes que no están diferencialmente expresados, lo más habitual en este tipo de análisis.

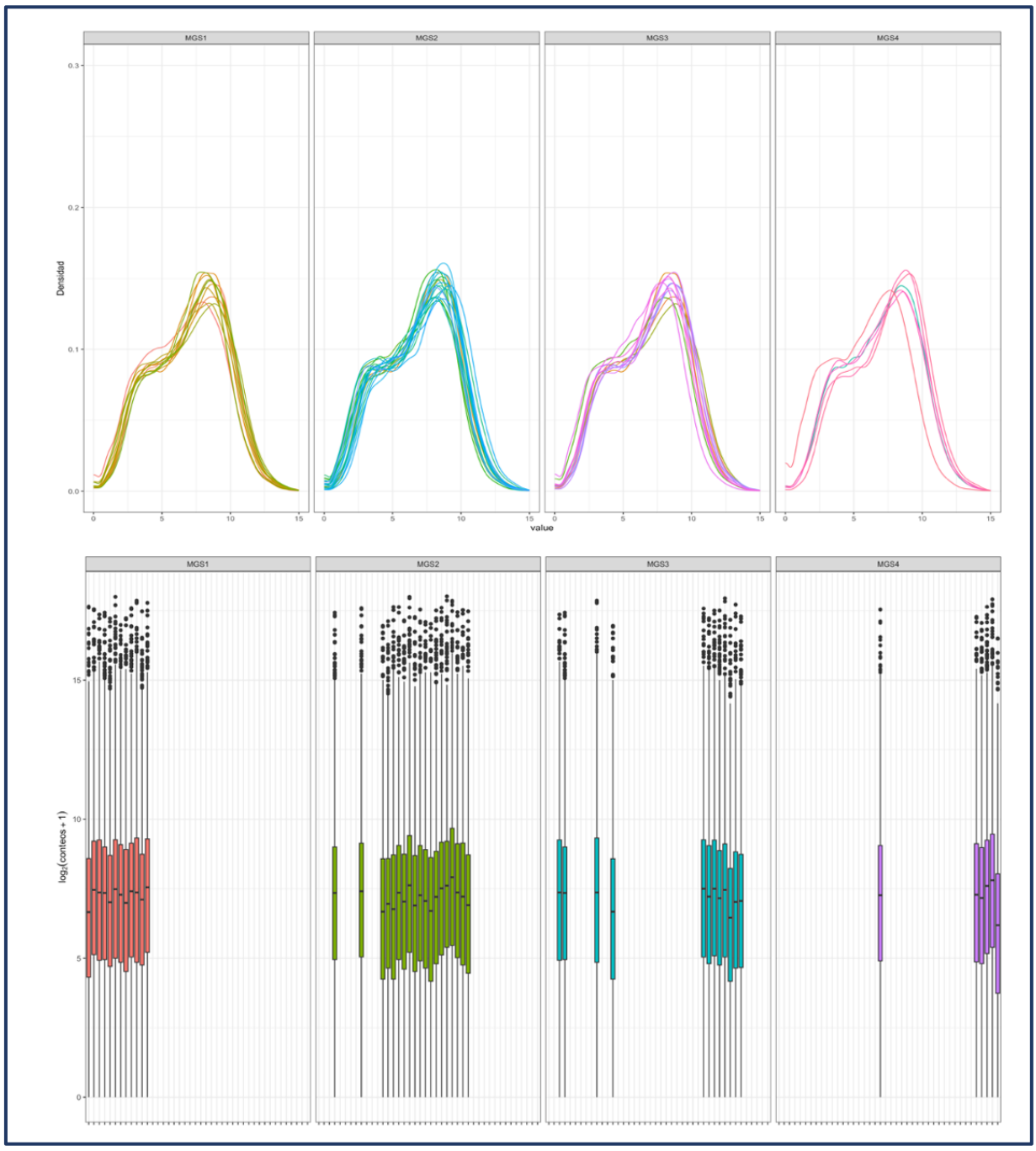

Figura 15 *Distribución de un conjunto aleatorio de muestras sin normalizar*

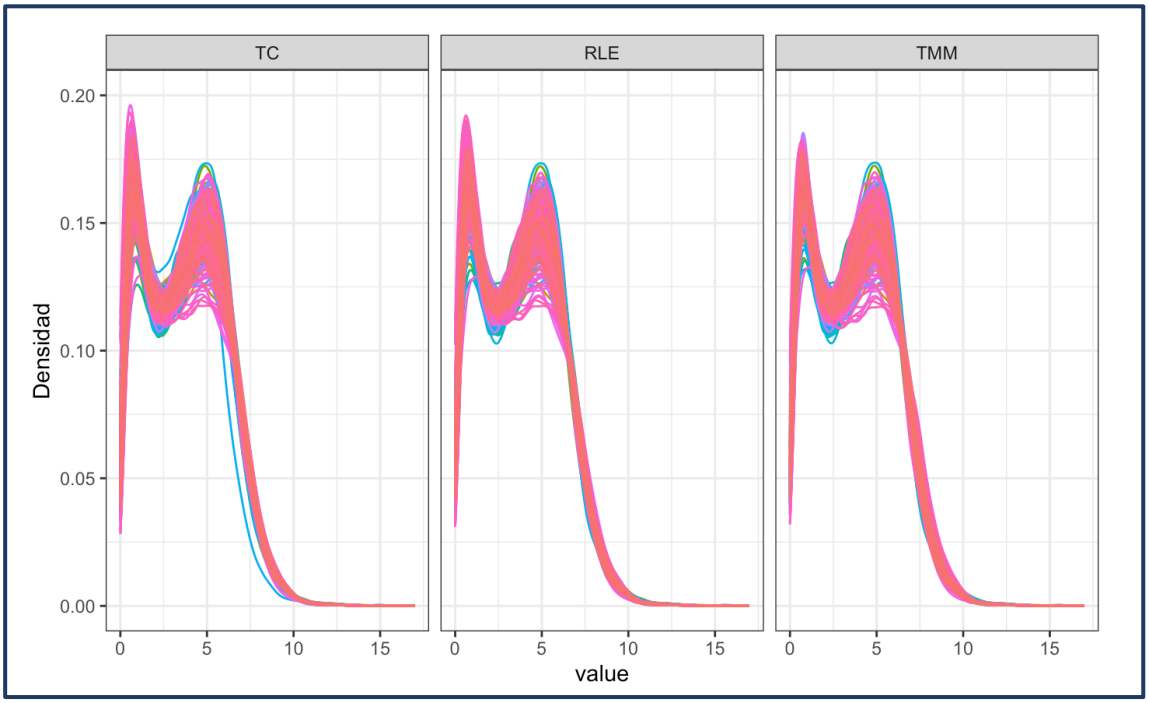

Figura 16 *Gráfico de densidad de todas las muestras comparando los tres métodos de normalización*

Los efectos de normalizar se presentan en la Figura 17 a través del mismo conjunto aleatorio de muestras que se empleaba previamente. Aquí se puede observar que las distribuciones entre niveles MGS son bastante similares. Sin embargo, en líneas generales, se puede comentar que por la longitud de las cajas (dada por el rango intercuartil) el número de conteos para el 50% de los genes en todos los grupos se encuentra aproximadamente entre 2 y 5 (en escala logarítmica en base 2 a la que se había sumado una unidad como constante). La línea horizontal dentro de cada caja representa la mediana de cada distribución que, de nuevo, es muy similar para todas y, al no situarse en el centro de la caja sino en su parte inferior, es un indicativo de asimetría positiva de la distribución. Esto no quiere decir que el lado inferior contenga más datos, sino que su dispersión es mayor que en una distribución más simétrica. Por la parte superior se observa que los valores máximos se encuentran en torno a $12 \approx 4095$ conteos, por lo que todos los puntos que se observan por encima son considerados *outliers*. Se debe tener en cuenta que, en este contexto, los valores atípicos no deben ser eliminados, ya que precisamente lo que se busca es encontrar aquellos genes diferencialmente expresados, una vez pasada la etapa de procesado inicial. La diferencia en la distribución de los datos antes y después de normalizar es clara, se observa que las muestras son comparables; disminuye la mediana, el rango intercuartil se reduce a más de la mitad que para los datos iniciales, es decir, que el número de conteos del 50 % de los genes ahora se encuentra en un intervalo más reducido; y, además, el valor máximo se reduce, observándose tras la normalización un mayor número de *outliers*.

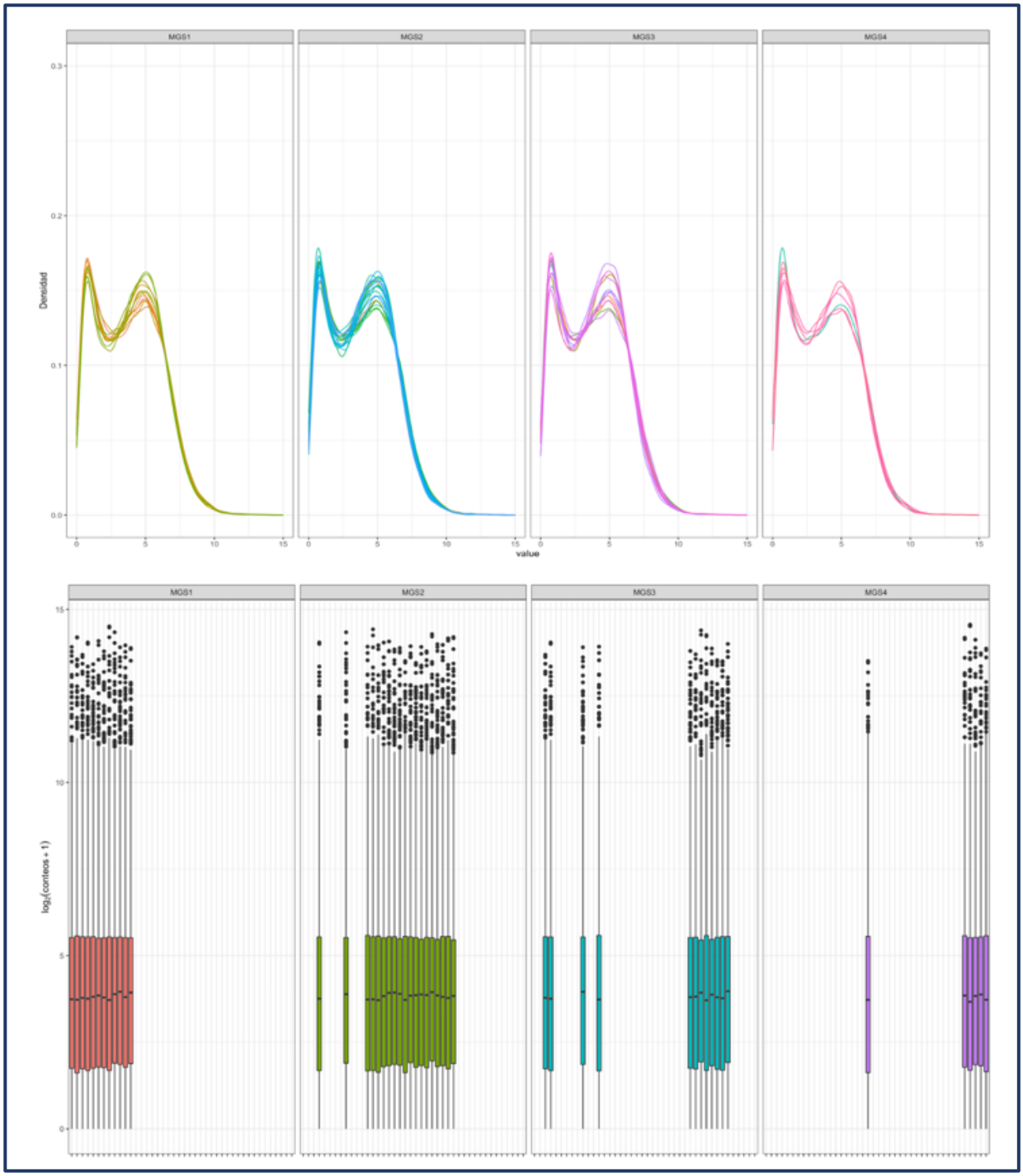

Figura 17 *Distribución de un conjunto aleatorio de muestras normalizadas mediante TMM*

Para finalizar el análisis exploratorio se va a realizar un Análisis de Componentes Principales (PCA). Como se explicó en la metodología, este método requiere la matriz de expresión normalizada, la cual acaba de ser obtenida mediante el método TMM. La obtención de los autovalores se hace de una manera muy sencilla a partir de la matriz de covarianza de los datos normalizados empleando la función `prcomp()`. En la Figura 18 se visualizan las proporciones de la variabilidad explicada por cada una de las diez primeras componentes principales. Con las dos primeras componentes principales se puede explicar el 92.7% de la variabilidad de los datos. Precisamente, estas dos primeras componentes son las que tienen un autovalor superior a la unidad (16.26 y 14.21, respectivamente), por lo que éstas serán las seleccionadas.

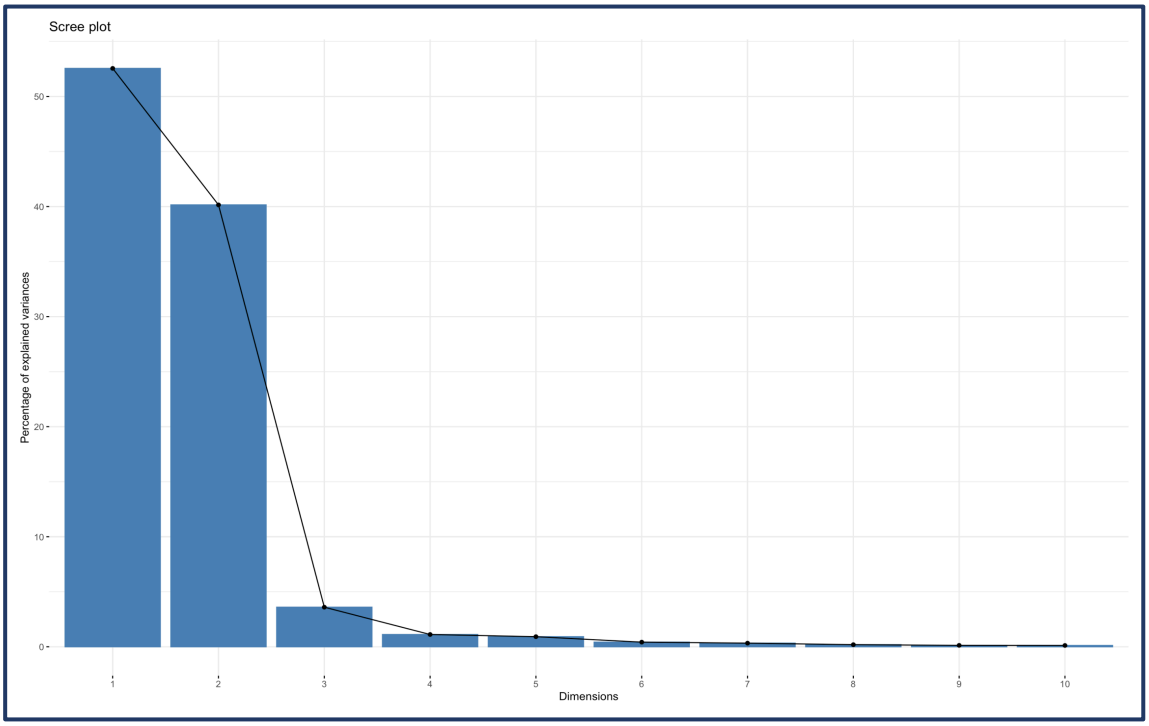

Figura 18 *Análisis de Componentes Principales*

De este modo, gracias al PCA se puede reducir la dimensión del problema de 18852 variables a tan solo 2 explicando prácticamente la totalidad de la variabilidad de los datos. Por último, en la Figura 19 se representa la totalidad de la muestra de estudio en el espacio formado por estas dos componentes principales. Al tomar dos componentes, los datos se presentan en un plano, en el que cada eje está representado por cada componente: la primera en abscisas y la segunda en ordenadas. Además, se ha separado las muestras por su nivel MGS para poder estudiar posibles diferencias entre los mismos. Desgraciadamente, no se puede evidenciar una distinción clara entre niveles MGS dentro del plano, aunque el PCA ha cumplido correctamente su objetivo de disminuir la dimensionalidad del problema.

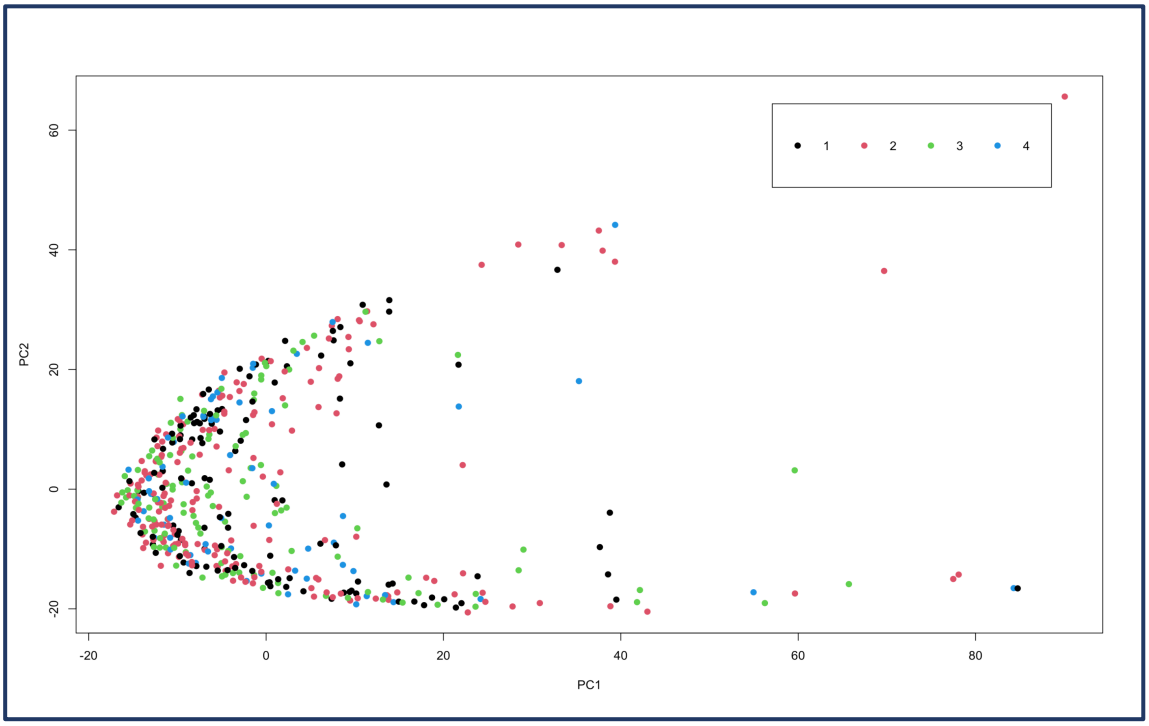

Figura 19 *Representación de las muestras en el espacio formado por las dos primeras componentes principales*

9.3 - Análisis de expresión diferencial

Para el análisis de expresión diferencial se va a emplear un GLM que incluye el efecto del grupo de severidad, la edad y el sexo, estas dos últimas variables con el objeto de corregir su posible efecto sobre la expresión diferencial entre grupos de DMAE. Para este modelo se va a partir de los datos normalizados mediante el método TMM y sobre ellos se estiman los parámetros de dispersión, de modo que la dispersión común estimada es de 0.1396448, por lo que el BCV es de 0.3737 (aproximando al cuarto decimal). Para la representación de las dispersiones estimadas se generó el plot BCV de la Figura 20, donde se puede apreciar la estimación de la dispersión común como la línea roja, como este valor se sitúa entre 0.2 y 0.4 se puede considerar que el estudio es razonable para el análisis de expresión diferencial.

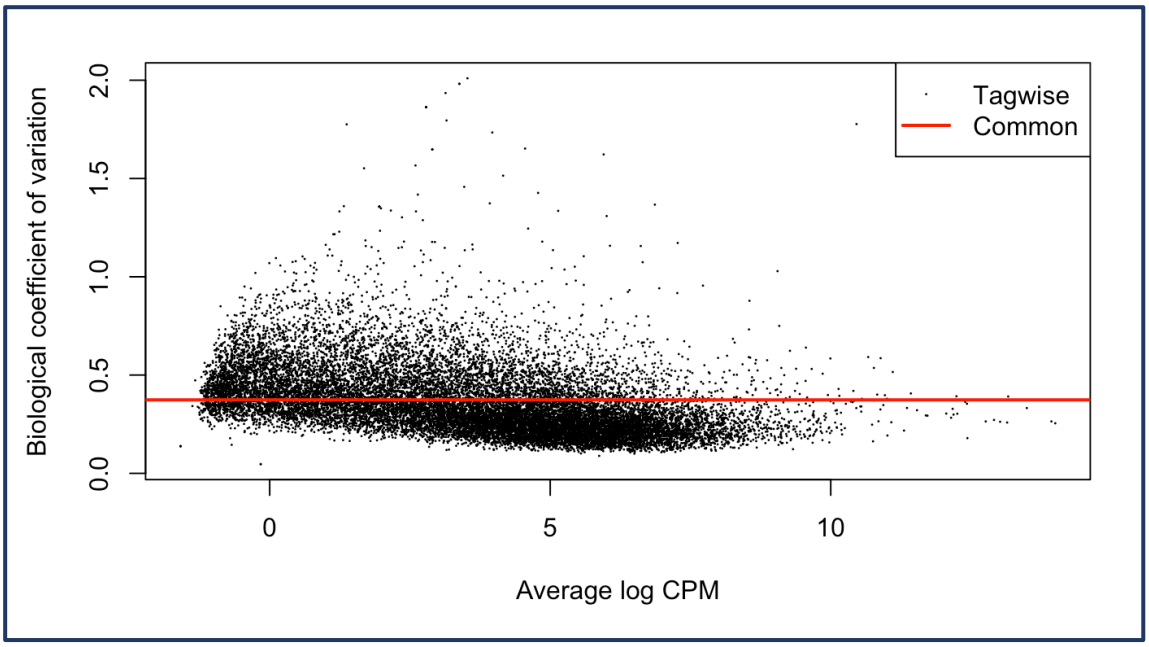

Figura 20 *Plot BCV de* ϕ_g

Tras construir la matriz de diseño, se realizan varios contrastes de comparación de expresión media entre dos grupos. En particular se evalúa la igualdad de expresión media entre las siguientes comparaciones dos a dos: MGS2 vs MGS1, MGS3 vs MGS1, MGS4 vs MGS1, MGS3 vs MGS2, MGS4 vs MGS2 y MGS4 vs MGS3. Esto es, con este análisis se van a extraer los genes que se expresan diferencialmente en un paciente de DMAE para cada uno de los niveles MGS frente a un paciente sano, además de (si existen) aquellos que se presenten entre pacientes de DMAE en distintos estadios. Aquí se realiza un ajuste de p-valores para solventar el problema de comparaciones múltiples y se seleccionan aquellos genes diferencialmente expresados para cada contraste obteniendo: 127 genes para las muestras MGS2, 163 genes para las muestras MGS3 y 156 genes para las muestras MGS4 respecto a un paciente sano. Además de 73 genes diferencialmente expresados para MGS3 frente a MGS2, 65 genes para MGS4 frente a MGS2 y otros 76 genes para MGS4 frente a MGS3. Se puede apreciar en la Figura 21 las diferencias

entre los conjuntos de genes seleccionados para cada uno de los niveles MGS a su respectivo *vulcano plot*, donde los puntos naranjas representan los genes diferencialmente expresados en ese conjunto. Adicionalmente, las relaciones entre estos conjuntos de genes se pueden entender fácilmente a través de los diagramas de Venn de la Figura 22. Se nota que, aunque hay varios genes comunes entre los conjuntos, cada nivel MGS presenta genes diferencialmente expresados que el resto de niveles no. Se representan tres diagramas distintos:

- En el primero se muestran los genes diferencialmente expresados para cada nivel MGS respecto a no padecer la enfermedad. Se observa que hay 102 genes identificados comunes a todos, por lo que pueden ser los más importantes, ya que identificarían la patología en sí, aunque no su estadio.

- En el segundo se comparan los genes diferencialmente expresados para el nivel MGS4 respecto a todos los demás niveles, de este modo se han encontrado 20 genes que lo podrían distinguir de MGS2 y 32 de MGS3, además de los ya indicados en el anterior diagrama con respecto a MGS1.

- En el tercero se representan los genes diferencialmente expresados para los niveles MGS2 y MGS3 para tratar de distinguir estos estadios intermedios ya que, tal como se introdujo anteriormente, su comprensión es bastante limitada en la actualidad. De hecho, se observa que la gran mayoría de genes son comunes a ambos niveles.

En total, teniendo en cuenta todos los contrastes realizados, se ha obtenido un listado de 220 genes diferencialmente expresados.

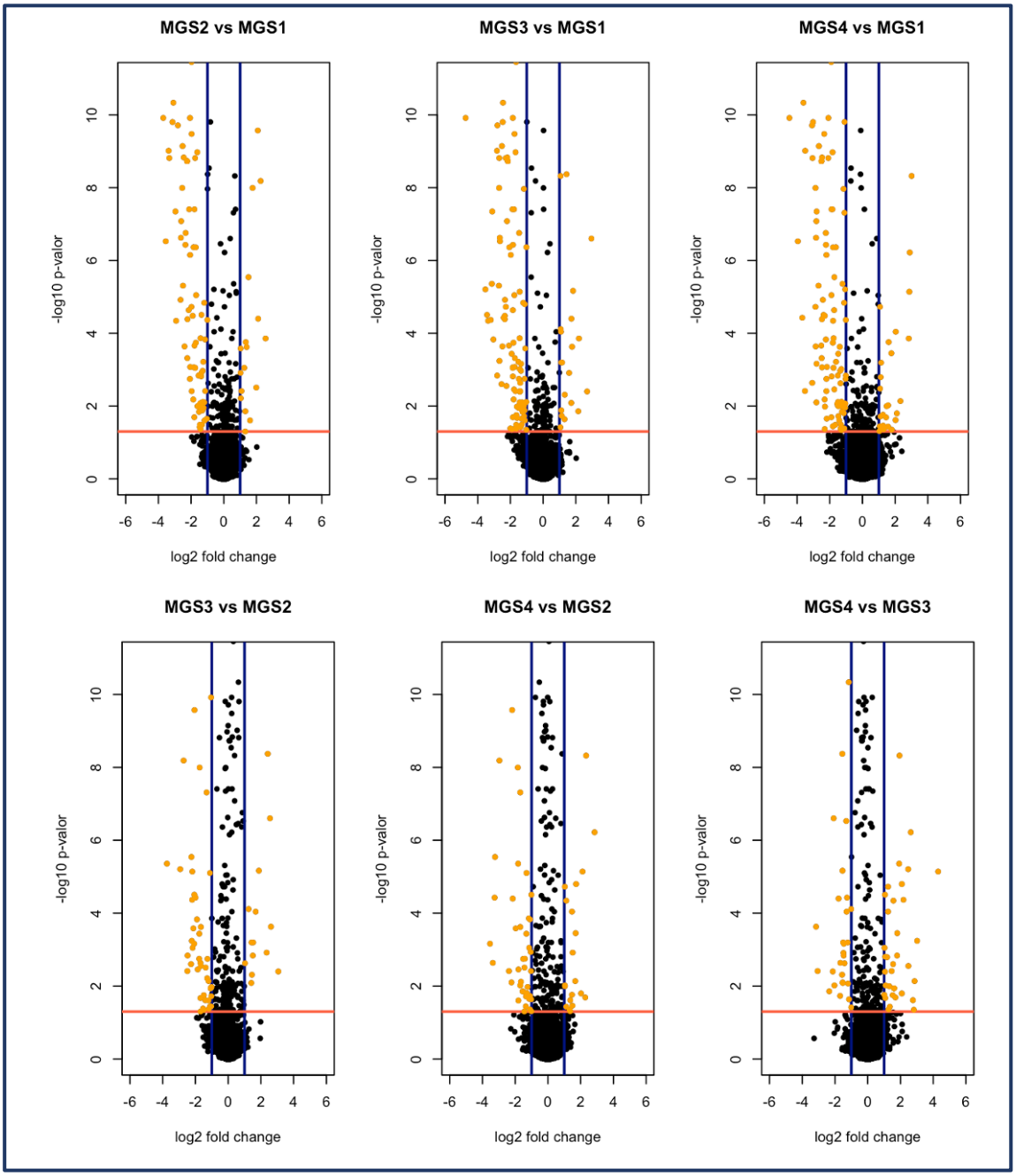

Figura 21 *Volcano plots de cada conjunto de genes diferencialmente expresados*

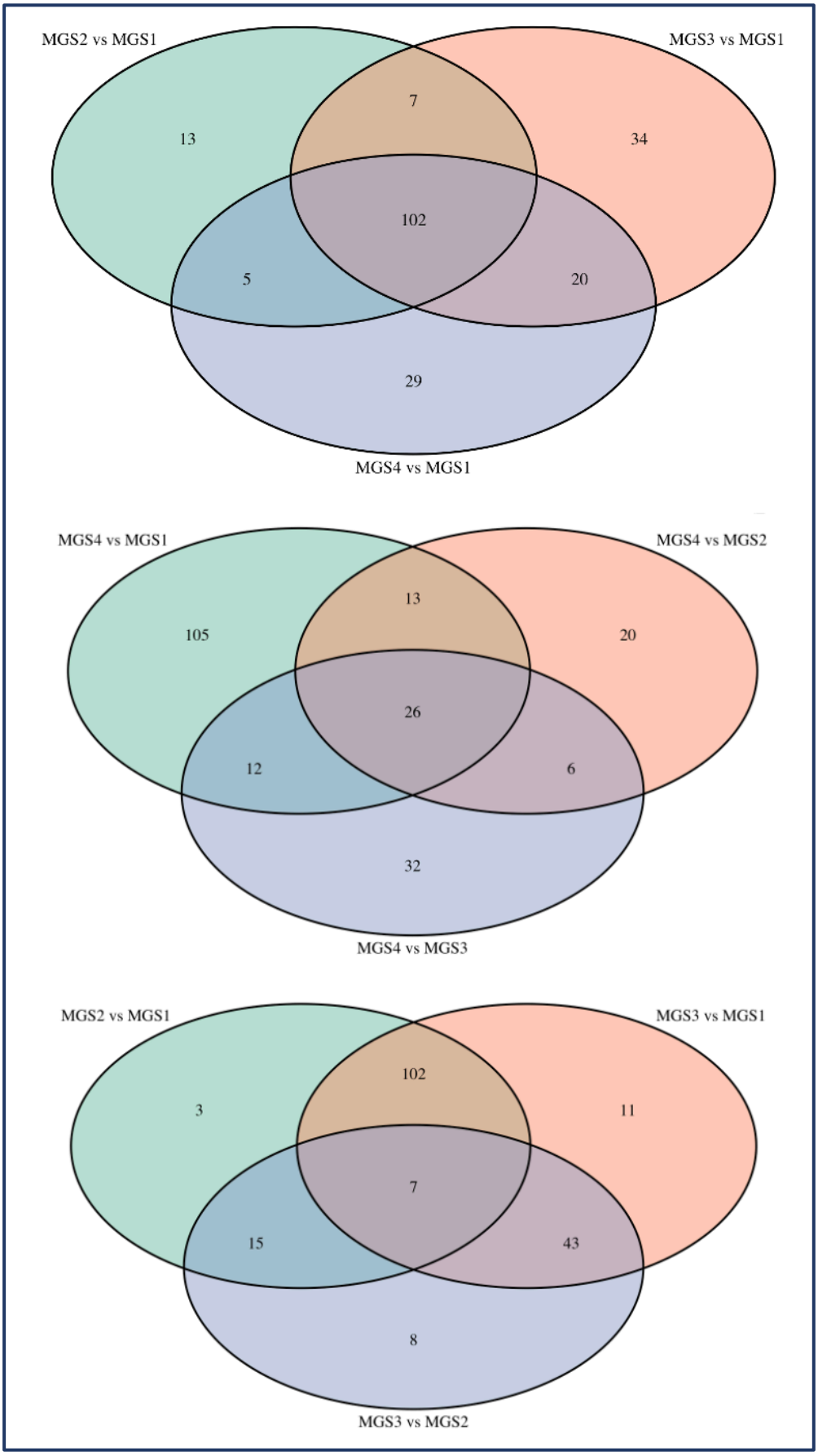

Figura 22 *Diagramas de Venn de los conjuntos de genes diferencialmente expresados*

9.4 - Agrupación de genes

Una vez obtenido el listado de genes diferencialmente expresados, se procede con su agrupación a fin de obtener aquellos que presenten un patrón de expresión más similar entre sí. De este modo, se podrá evaluar si existe un agrupamiento de genes/muestras relacionado con el nivel de expresión. Para ello, se ha aplicado *clustering* jerárquico aglomerativo sobre la expresión de los genes diferencialmente expresados resultado del paso anterior, se han tenido en consideración todos los genes que han resultado diferencialmente expresados para alguno de los contrastes, sumando un total de 220 genes. Para determinar el número de grupos óptimos se ha procedido estimando la SSE para distintos números de grupos, desde uno hasta veinte. Encontrando que a partir de diez la disminución del valor de la SSE no era tan pronunciando como hasta ese punto, como se aprecia en la Figura 23.

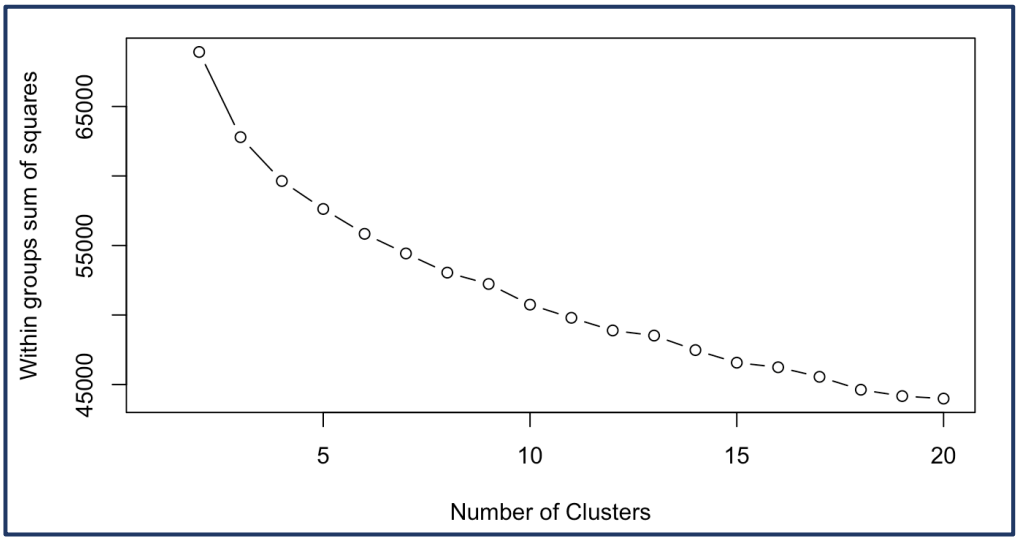

Figura 23 Evolución de la SSE con el número de grupos

Por tanto, se ha considerado un total de diez grupos para realizar *clustering* que se representan en la Figura 24 mediante un dedrograma y *heatmap*, no se aprecian grandes diferencias entre grupos. Por otra parte, en la Figura 25 se ha representado la distribución de los conteos para los genes diferencialmente expresados en los 10 grupos hallados. Aquí sí se aprecian diferencias en los perfiles de expresión entre los distintos grupos, así como la semejanza dentro del mismo grupo. Aunque se han representado las distribuciones distinguiendo entre los distintos niveles MGS, no se aprecian diferencias significativas entre ellos para los grupos encontrados, tan solo en los grupos 5 y 7 se presenta una diferencia notable entre las muestras MGS4 y el resto de niveles.

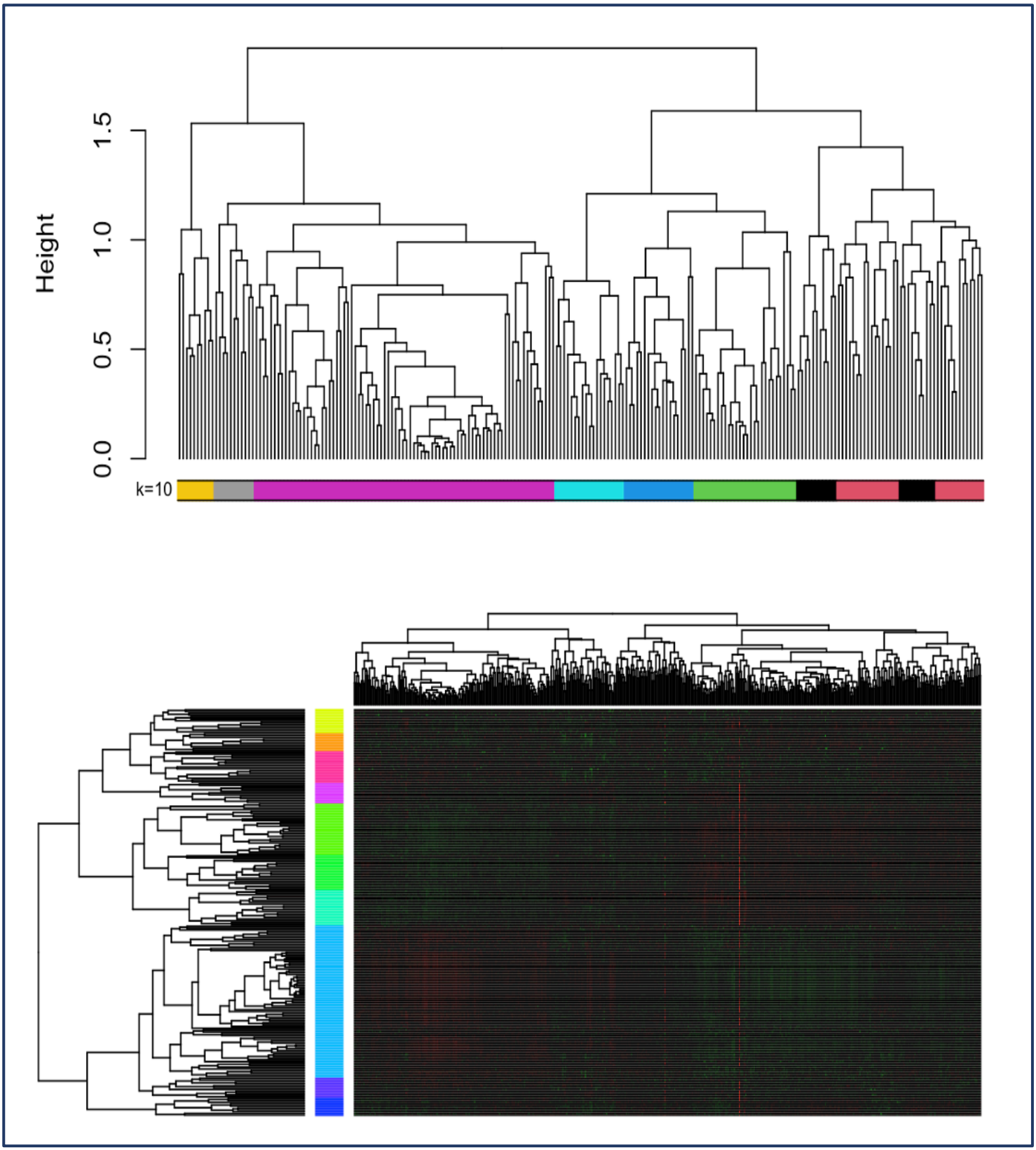

Figura 24 *Clustering de los genes diferencialmente expresados*

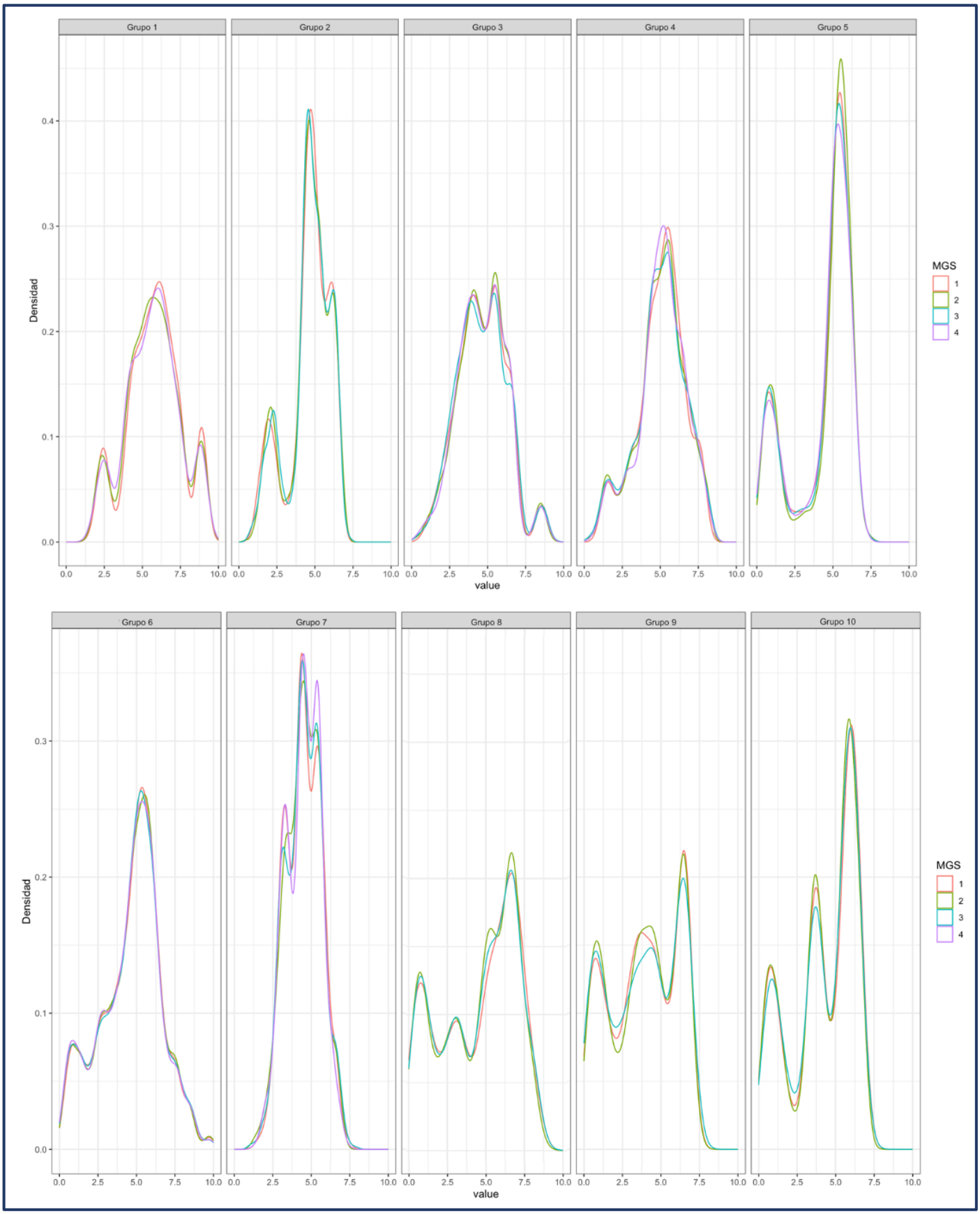

Figura 25 *Distribución de los conteos de los genes separados agrupados por clustering jerárquico*

9.5 - Análisis de enriquecimiento

Para proceder con el análisis de enriquecimiento se toma el listado completo de 220 genes y se analiza el resultado de cada uno de los contrastes considerados en el análisis de expresión diferencial: MGS2 vs MGS1, MGS3 vs MGS1, MGS4 vs MGS1, MGS3 vs MGS2, MGS4 vs MGS2 y MGS4 vs MGS3. De este modo, se puede visualizar en las Figuras 26-31 los procesos biológicos en los que los genes se encuentran implicados.

Atendiendo a las Figuras 26, 27 y 28, se observa un patrón común a todos los tipos de DMAE (MGS2, MGS3 y MGS4, respectivamente) respecto a los controles. Hay una supresión paulatina a medida que aumenta el estadio de la patología de genes vinculados con los siguientes procesos biológicos:

- El componente celular.
- El ensamblaje de proteínas.
- Enlaces.
- Función molecular.
- La entidad anatómica celular.
- La estructura anatómica intracelular.
- La regulación de procesos celulares.
- Procesos celulares.
- Procesos biológicos.
- Organelas.

En el caso de la DMAE avanzada (MGS4) se ha encontrado, respecto al resto de estadios, la activación de procesos relacionados con canales de transporte de cationes dependientes de voltaje, destacando el de Calcio, como su transportador transmembrana.

Se debe recordar que el objetivo de este trabajo no es el estudio de la relación de procesos biológicos con la DMAE, sino la obtención de biomarcadores diferencialmente representados en el paciente de DMAE. Este apartado tiene como fin completar los resultados obtenidos por el análisis de expresión diferencial, dejando su profundización como una posible línea futura, ya que estos grupos de genes podrían ser empleados como dianas para el estudio de la mecánica subyacente en la patología y de otras dianas para la investigación.

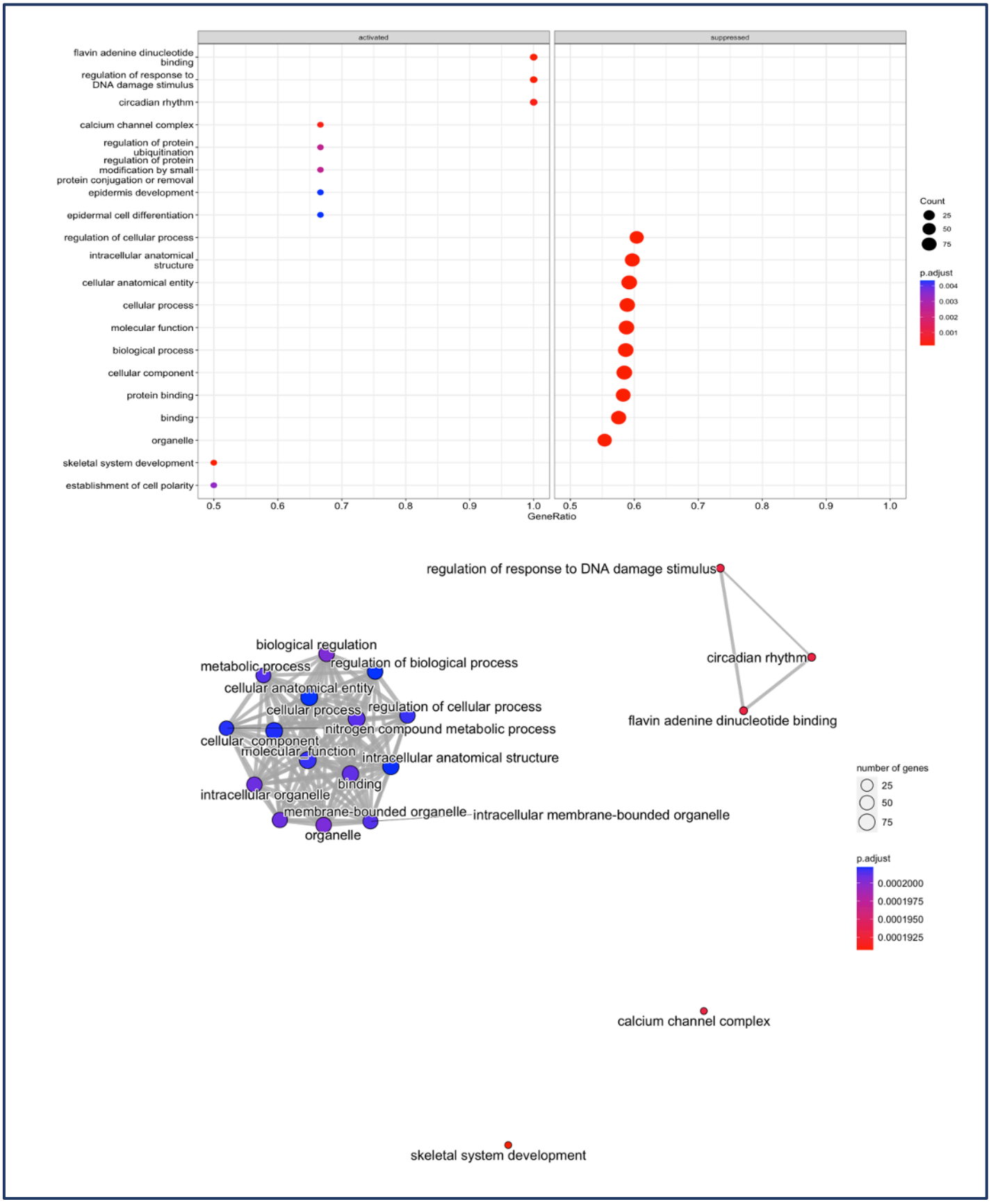

Figura 26 *Análisis de enriquecimiento contraste MGS2 vs MGS1*

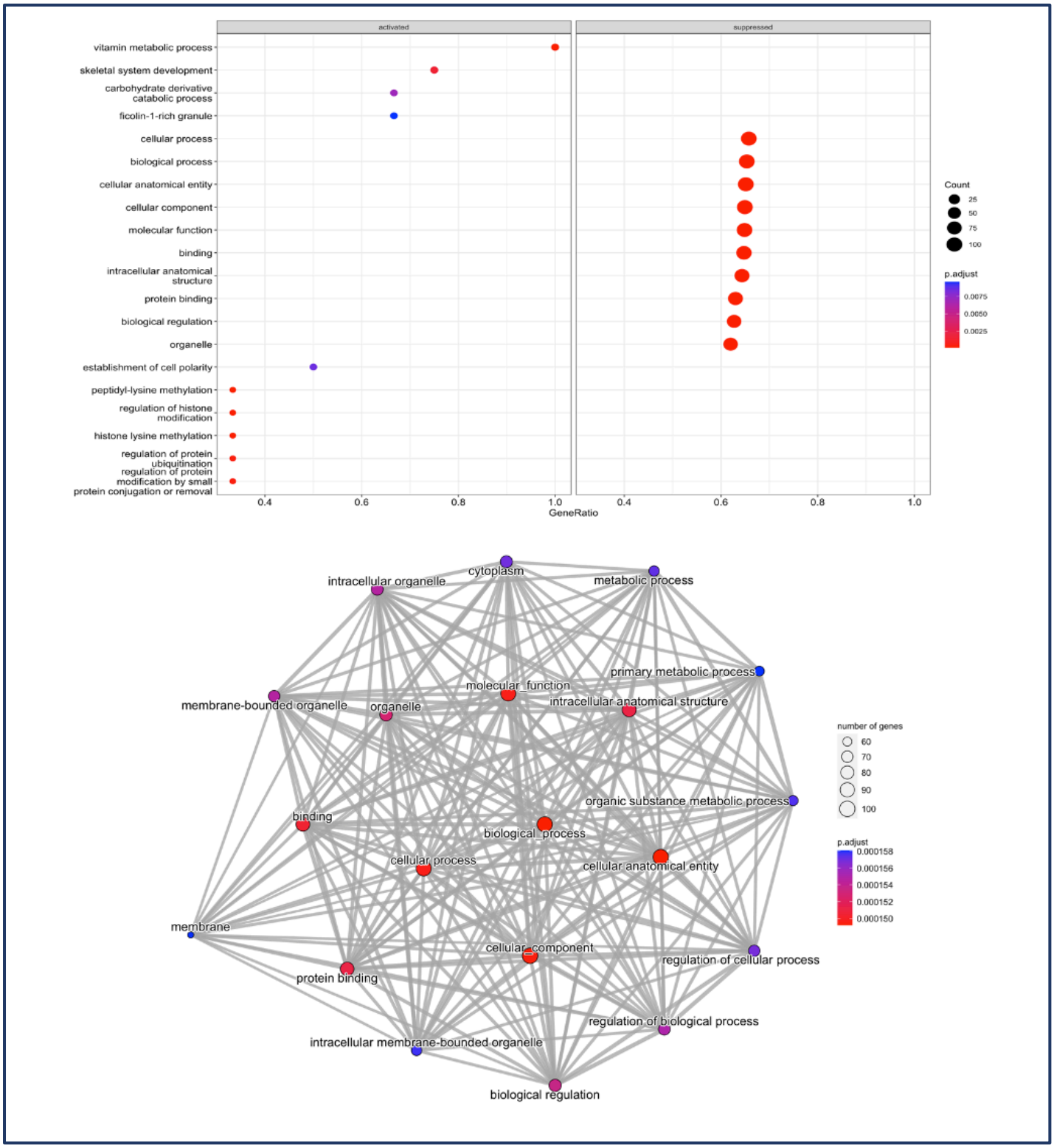

Figura 27 *Análisis de enriquecimiento contraste MGS3 vs MGS1*

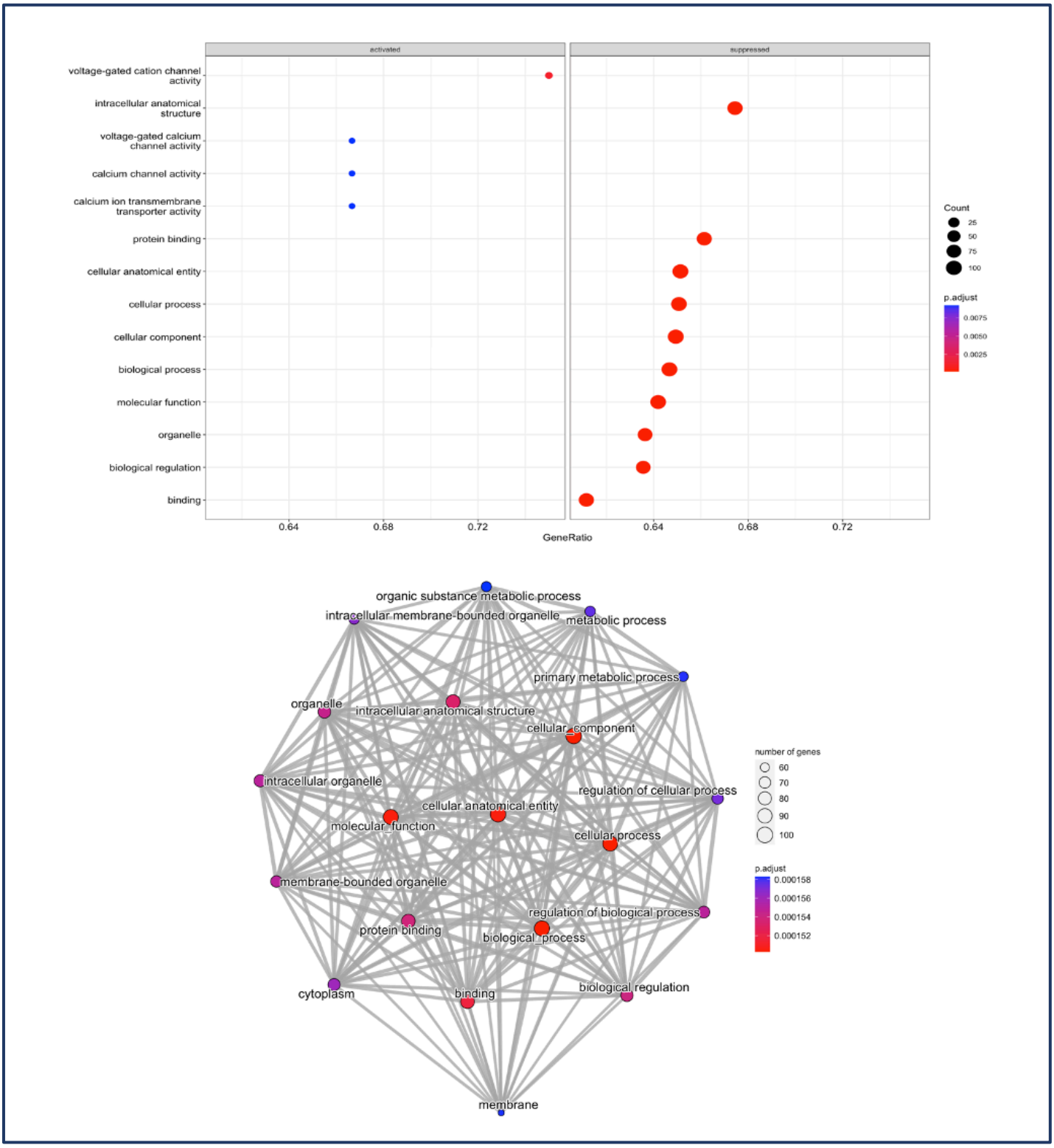

Figura 28 *Análisis de enriquecimiento contraste MGS4 vs MGS1*

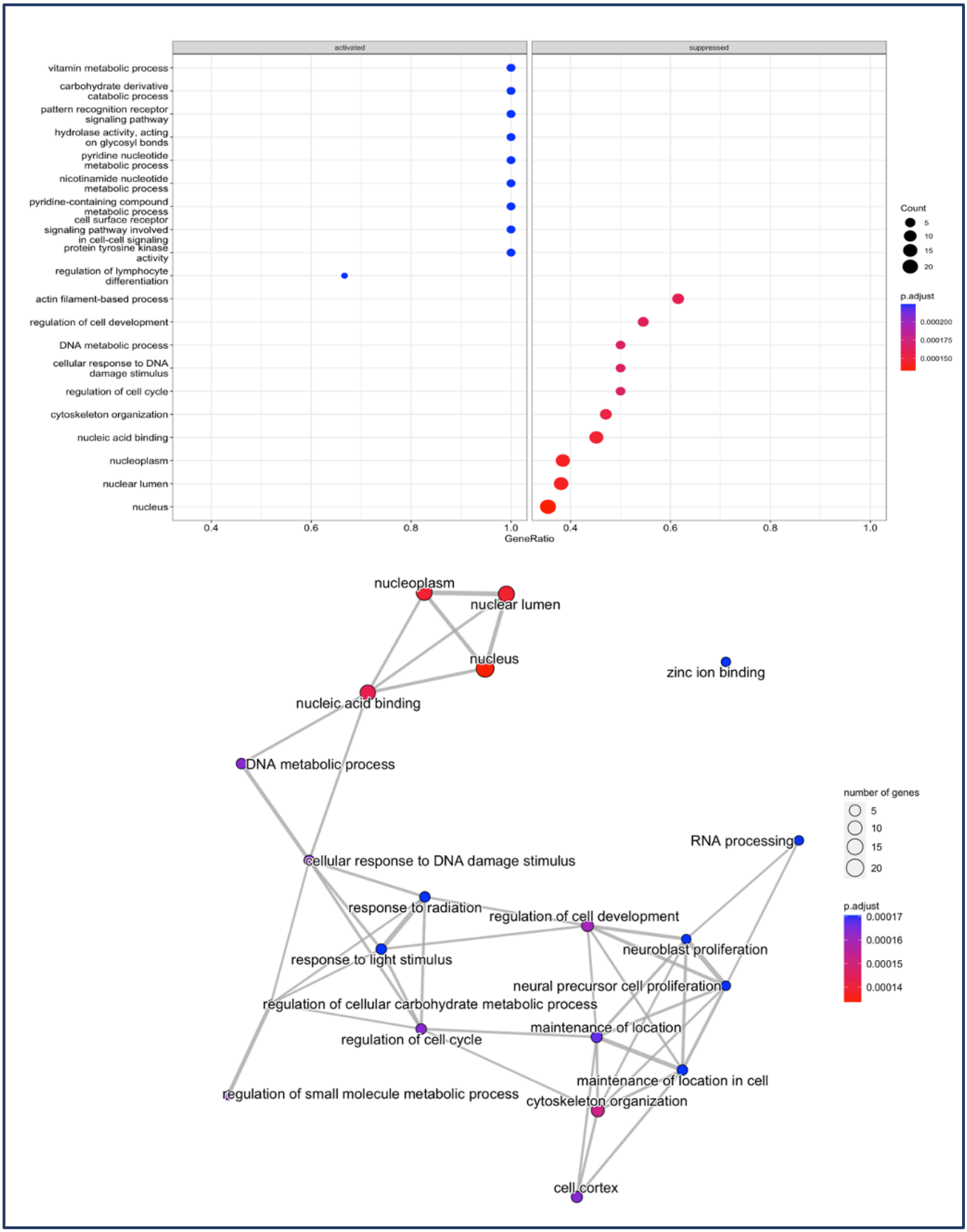

Figura 29 *Análisis de enriquecimiento contraste MGS3 vs MGS2*

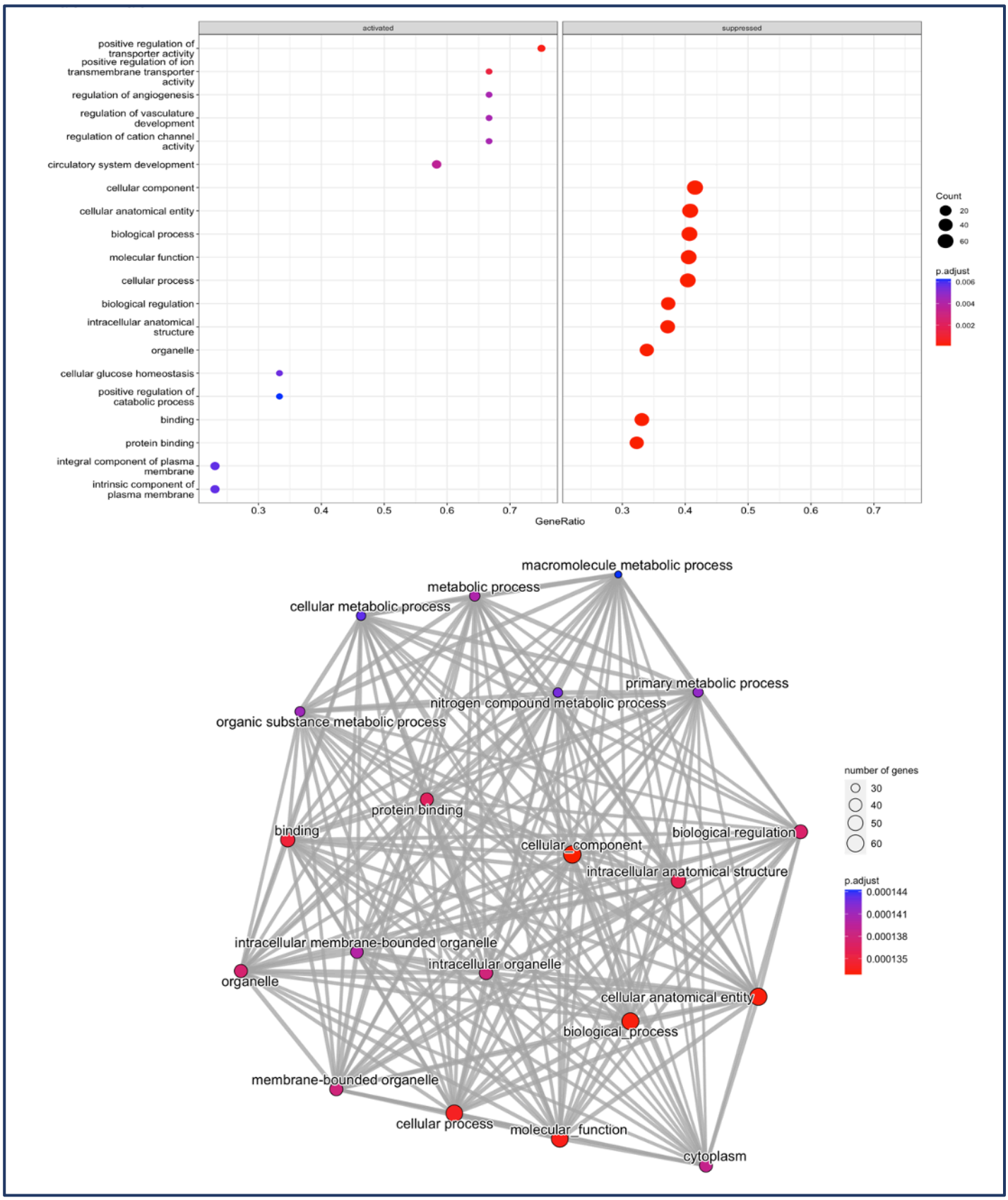

Figura 30 *Análisis de enriquecimiento contraste MGS4 vs MGS2*

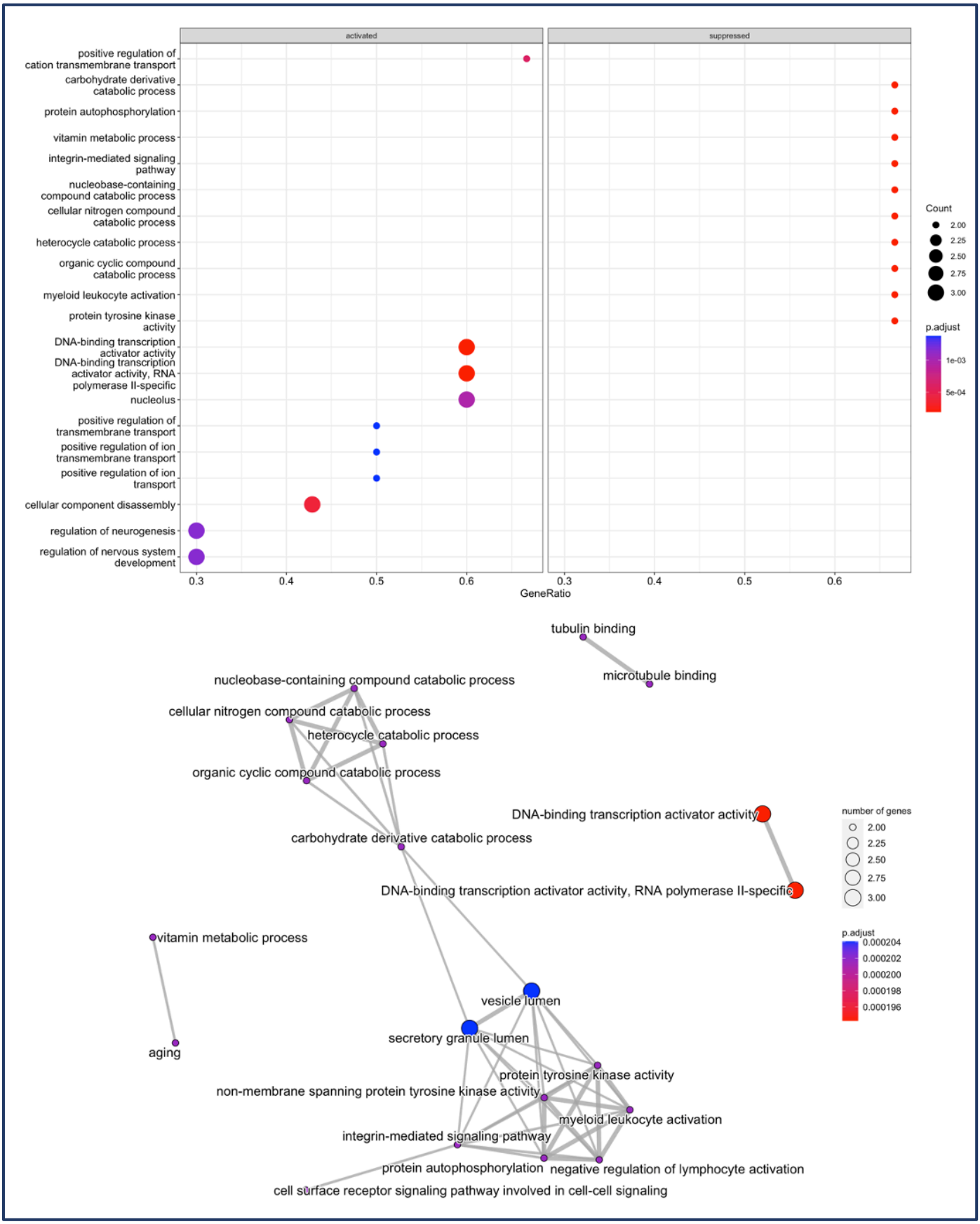

Figura 31 *Análisis de enriquecimiento contraste MGS4 vs MGS3*

9.6 - Diseño y ajuste del clasificador

La muestra total de 505 individuos se divide en dos sub-muestras, una muestra que servirá para ajustar los modelos (muestra de entrenamiento) y la otra para validar externamente estos modelos (muestra test). La primera está formada por 337 individuos (aproximadamente 2/3 de la muestra total) seleccionados aleatoriamente. La muestra test está formada por el 1/3 restante. Estas sub-muestras son comparables entre sí, ya que, dadas la información que se muestra en las Tablas 4-7, no hay evidencias suficientes de que sean distintas. Respecto al análisis de los genes, se debe indicar que en el Anexo II se incluyen los cálculos para el listado completo de los 220, donde se puede comprobar que todas estas variables también son comparables en ambas sub-muestras; sin embargo, en la Tabla 7 solo se han recogido los genes que son empleados posteriormente en los modelos para facilitar la lectura del manuscrito.

Tabla 4 *Análisis comparativo de las sub-muestras según la Edad*

Edad	***N***	***Media***	***DT***	***IC 95% media***		***Med.***	***Min.***	***Max.***
				Inf.	***Sup.***			
Entrenamiento	337	79.22	3.22	74.89	83.55	80	47	107
Test	168	80.01	3.18	74.52	86.50	81	54	103
Total	505	79.65	3.21	76.14	83.16	81	47	107

Nota: DT = Desviación Típica; Med. = Mediana; Min. = Mínimo; Max. = Máximo

Tabla 5 *Análisis comparativo de las sub-muestras según el Sexo*

Sexo	***Muestra Entrenamiento***				***Muestra Test***			
	n	***%***	***IC 95% para %***		***n***	***%***	***IC 95% para %***	
			Inf.	***Sup.***			***Inf.***	***Sup.***
Femenino	183	54.3	48.98	59.62	87	51.8	44.23	59.34
Masculino	154	45.7	40.38	51.02	81	48.2	40.66	55.77
Total	337	100	.	.	168	100	.	.

Tabla 6 *Análisis comparativo de las sub-muestras según el Nivel MGS*

Nivel MGS	***Muestra Entrenamiento***				***Muestra Test***			
	n	***%***	***IC 95% para %***		***n***	***%***	***IC 95% para %***	
			Inf.	***Sup.***			***Inf.***	***Sup.***
MGS1	87	25.8	21.14	30.49	37	22.0	15.76	28.29
MGS2	122	36.2	31.07	41.33	69	41.1	33.63	48.51
MGS3	86	25.5	20.87	30.17	39	23.2	16.83	29.60
MGS4	42	12.5	8.94	17.49	23	13.7	8.49	18.89
Total	337	100	.	.	168	100	.	.

Tabla 7 *Análisis comparativo de las sub-muestras según los genes*

Variable	*Muestra*	*Media*	*DT*	*IC 95% media*		*Med.*	*Min.*	*Max.*
				Inf.	*Sup.*			
ENSG00000000003	Train	4.13	0.66	2.01	6.26	4.13	0.00	5.24
	Test	4.15	0.62	2.02	6.28	4.13	3.01	5.63
	Total	4.14	0.65	2.01	6.27	4.13	0.00	5.63
ENSG00000002079	Train	0.06	0.29	0.00	0.32	0.00	0.00	0.34
	Test	0.06	0.30	0.00	0.32	0.00	0.00	0.38
	Total	0.06	0.29	0.00	0.32	0.00	0.00	0.38
ENSG00000003147	Train	2.73	0.66	0.99	4.45	2.76	0.00	4.50
	Test	2.67	0.61	0.95	4.39	2.70	1.43	3.45
	Total	2.71	0.65	0.98	4.44	2.74	0.00	4.50
ENSG00000004846	Train	0.01	0.16	0.00	0.12	0.00	0.00	0.22
	Test	0.01	0.16	0.00	0.12	0.00	0.00	0.22
	Total	0.01	0.16	0.00	0.12	0.00	0.00	0.22
ENSG00000005001	Train	0.02	0.24	0.00	0.17	0.00	0.00	0.53
	Test	0.02	0.22	0.00	0.17	0.00	0.00	0.30
	Total	0.02	0.23	0.00	0.17	0.00	0.00	0.53
ENSG00000005102	Train	0.03	0.25	0.00	0.21	0.00	0.00	0.36
	Test	0.03	0.24	0.00	0.21	0.00	0.00	0.31
	Total	0.03	0.25	0.00	0.21	0.00	0.00	0.36
ENSG00000005379	Train	5.72	0.83	3.24	8.20	5.77	0.00	7.44
	Test	5.71	0.78	3.23	8.19	5.71	4.06	7.06
	Total	5.71	0.82	3.23	8.19	5.73	0.00	7.44
ENSG00000005421	Train	0.03	0.27	0.00	0.21	0.00	0.00	0.38
	Test	0.04	0.28	0.00	0.25	0.00	0.00	0.42
	Total	0.04	0.27	0.00	0.25	0.00	0.00	0.42
ENSG00000005469	Train	6.41	0.75	3.79	9.03	6.45	0.00	7.64
	Test	6.48	0.64	3.85	9.11	6.48	5.33	7.45
	Total	6.43	0.72	3.81	9.05	6.45	0.00	7.64
ENSG00000005471	Train	0.70	0.55	0.00	1.59	0.68	0.00	1.75
	Test	0.71	0.51	0.00	1.61	0.70	0.20	1.57
	Total	0.70	0.54	0.00	1.59	0.68	0.00	1.75
ENSG00000005981	Train	0.09	0.33	0.00	0.41	0.06	0.00	0.46
	Test	0.09	0.34	0.00	0.41	0.00	0.00	0.66
	Total	0.09	0.33	0.00	0.41	0.00	0.00	0.66
ENSG00000006606	Train	0.87	0.60	0.00	1.86	0.86	0.00	2.43
	Test	0.94	0.63	0.00	1.97	0.91	0.09	2.99
	Total	0.89	0.61	0.00	1.89	0.87	0.00	2.99
ENSG00000006042	Train	4.65	0.64	2.40	6.90	4.65	0.00	6.49
	Test	4.65	0.58	2.40	6.90	4.62	3.78	6.07
	Total	4.65	0.62	2.40	6.90	4.64	0.00	6.49
ENSG00000007312	Train	0.06	0.30	0.00	0.32	0.00	0.00	0.48
	Test	0.05	0.30	0.00	0.29	0.00	0.00	0.50
	Total	0.06	0.30	0.00	0.32	0.00	0.00	0.50
ENSG00000007908	Train	0.35	0.75	0.00	0.98	0.19	0.00	5.57
	Test	0.33	0.74	0.00	0.94	0.14	0.00	4.66
	Total	0.34	0.75	0.00	0.96	0.17	0.00	5.57
ENSG00000008516	Train	0.09	0.33	0.00	0.41	0.06	0.00	0.49
	Test	0.10	0.35	0.00	0.44	0.08	0.00	0.60
	Total	0.09	0.34	0.00	0.41	0.06	0.00	0.60

Nota: Train = Muestra de entrenamiento; DT = Desviación Típica; Med. = Mediana; Min. = Mínimo; Max. = Máximo

Mediante el método LOOCV, se irán tomando genes del listado que ha sido hallado en el análisis de expresión diferencial, para encontrar los mejores modelos que, a partir de los niveles de expresión, permitan:

1. Distinguir entre patológicos (MGS2 + MGS3 + MGS4) y controles (MGS1)

Se parte del listado de 102 genes diferencialmente expresados comunes a los contrastes MGS2 vs MGS1, MGS3 vs MGS1 y MGS4 vs MGS1. Esto es, el conjunto de genes diferencialmente expresados para un paciente de DMAE independientemente de su estadio frente a un control. Se calcula el AUC variando el número de genes y *kernel* empleado en el modelo, representando el resultado en la Figura 32.

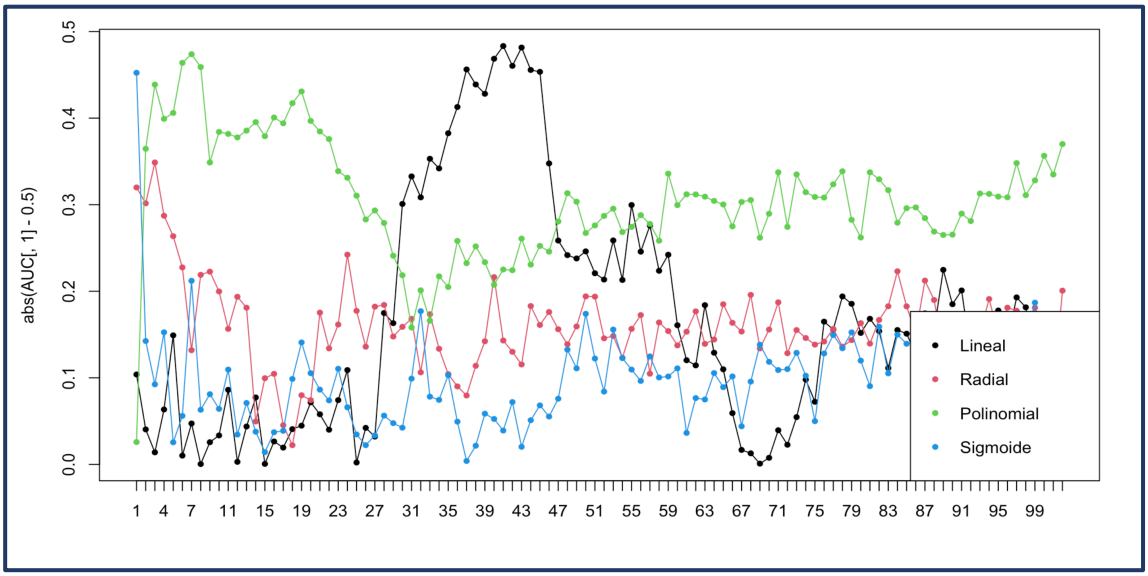

Figura 32 *AUC del primer modelo en función del número de genes incluidos y kernel empleado*

De aquí, se va a comparar la capacidad predictiva de los modelos más pequeños con mayor AUC, en base a los resultados reflejados en las matrices de confusión de las Tablas 8, 9 y 10.

- *Kernel* sigmoide con 1 gen

  ENSG00000004846

Tabla 8 Matriz de confusión kernel sigmoide 1 gen

		Real	
		DMAE	***Sano***
Predicción	***DMAE***	213	43
	Sano	37	44

- *Kernel* polinómico de grado 3 con 6 genes

  ENSG00000002079 ENSG00000004846 ENSG00000005001
  ENSG00000005102 ENSG00000006042 ENSG00000007312

Tabla 9 *Matriz de confusión kernel polinómico 6 genes*

		Real	
		DMAE	***Sano***
Predicción	***DMAE***	242	50
	Sano	8	37

- *Kernel* polinómico de grado 3 con 7 genes

  ENSG00000002079 ENSG00000004846 ENSG00000005001
  ENSG00000005102 ENSG00000005471 ENSG00000006042
  ENSG00000007312

Tabla 10 *Matriz de confusión kernel polinómico 7 genes*

		Real	
		DMAE	***Sano***
Predicción	***DMAE***	240	38
	Sano	10	49

Las medidas de discriminación se resumen en la Tabla 11.

Tabla 11 *Medidas de discriminación del modelo Patológico vs Control*

Patológico	***Sigmoide 1 gen***			***Polinómico 6 genes***			***Polinómico 7 genes***		
	Medida	***IC 95% Medida***		***Medida***	***IC 95% Medida***		***Medida***	***IC 95% Medida***	
		Inf.	***Sup.***		***Inf.***	***Sup.***		***Inf.***	***Sup.***
AUC	0.71	0.66	0.76	0.83	0.79	0.87	0.86	0.83	0.90
Sensibilidad	0.85	0.81	0.89	0.97	0.95	0.99	0.96	0.94	0.98
Especificidad	0.51	0.46	0.56	0.43	0.38	0.48	0.56	0.51	0.61
Exactitud	0.79	0.75	0.83	0.83	0.79	0.87	0.86	0.83	0.90

Siguiendo el criterio del modelo de dimensión más pequeña con mejor AUC se selecciona el modelo con *kernel* polinómico con 6 genes ya que mejora el AUC en más de 0.1 unidades respecto al sigmoide y, aunque disminuye el AUC respecto al polinómico de 7 genes, esta diferencia es de 0.03 unidades.

2. Distinguir entre niveles intermedios (MGS3 frente a MGS2)

Se parte del listado de 7 genes diferencialmente expresados comunes a los contrastes MGS2 vs MGS1, MGS3 vs MGS1 y MGS3 vs MGS2. Esto es, el conjunto de genes diferencialmente expresados para un paciente de DMAE en estadio intermedio MGS2 frente a uno MGS3. Se calcula el AUC variando el número de genes y *kernel* empleado en el modelo, representando el resultado en la Figura 33. Este caso es bastante más claro y solo se va a comparar la capacidad predictiva del modelo con *kernel* polinomial de grado 3 con 3 genes con el de 4 genes en base a los resultados recogidos en las Tablas 12 y 13

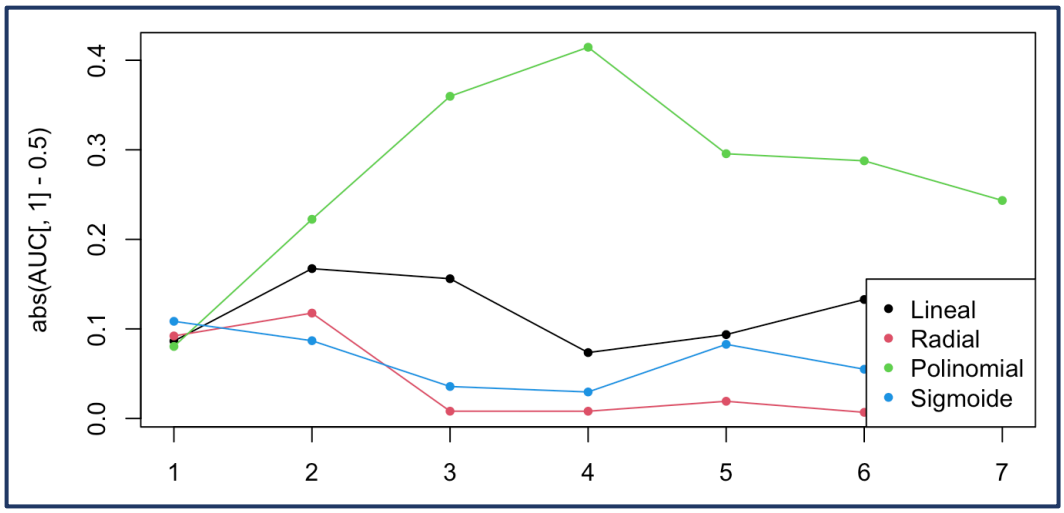

Figura 33 *AUC del segundo modelo en función del número de genes incluidos y kernel empleado*

- *Kernel* polinómico de grado 3 con 3 genes

ENSG00000003147 ENSG00000005379 ENSG00000005981

Tabla 12 *Matriz de confusión kernel polinómico 3 genes*

		Real	
		MGS3	***Control***
Predicción	***MGS3***	48	42
	Control	38	209

Nota: Control = MGS1 + MGS3 + MGS4

- *Kernel* polinómico de grado 3 con 4 genes

  ENSG00000003147 ENSG00000005379 ENSG00000005469 ENSG00000005981

Tabla 13 *Matriz de confusión kernel polinómico 4 genes*

		Real	
		MGS3	***Control***
Predicción	***MGS3***	55	47
	Control	31	204

Nota: Control = MGS1 + MGS3 + MGS4

Las medidas de discriminación se resumen en la Tabla 14.

Tabla 14 *Medidas de discriminación del modelo Intermedio vs Control*

Intermedio	***Polinómico 3 genes***			***Polinómico 4 genes***		
	Medida	***IC 95% Medida***		***Medida***	***IC 95% Medida***	
		Inf.	***Sup.***		***Inf.***	***Sup.***
AUC	0.73	0.68	0.77	0.77	0.73	0.81
Sensibilidad	0.56	0.51	0.61	0.64	0.59	0.69
Especificidad	0.81	0.77	0.85	0.81	0.77	0.85
Exactitud	0.77	0.73	0.81	0.77	0.73	0.81

En este caso, el mejor modelo es el de *kernel* polinómico con 3 genes, ya que la inclusión de un gen más no mejora el AUC por más de 0.1 unidades.

3. Distinguir entre DMAE avanzada (MGS4) y el resto de estadios (MGS1 + MGS2 + MGS3)

Se parte del listado de 26 genes diferencialmente expresados comunes a los contrastes MGS4 vs MGS1, MGS4 vs MGS2 y MGS4 vs MGS3. Esto es, el conjunto de genes diferencialmente expresados para un paciente de DMAE avanzada (MGS4) frente a cualquier otro estadio. Se calcula el AUC variando el número de genes y *kernel* empleado en el modelo, representando el resultado en la Figura 34. De nuevo, se va a comparar la capacidad predictiva de tres modelos con mejor AUC en base a los resultados de las Tablas 15, 16 y 17.

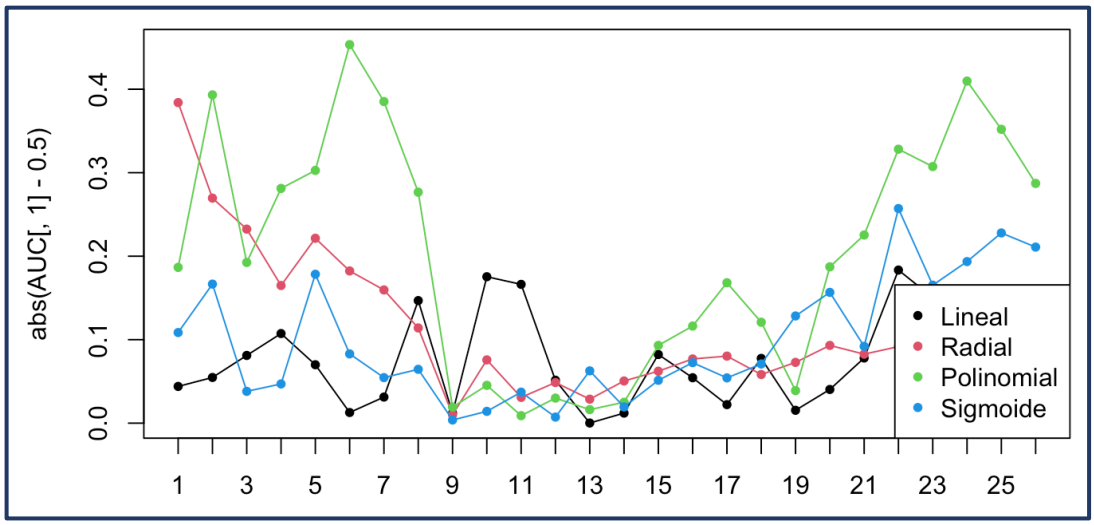

Figura 34 *AUC del tercer modelo en función del número de genes incluidos y kernel empleado*

- *Kernel* radial con 1 gen:

  ENSG00000005421

Tabla 15 *Matriz de confusión kernel radial 1 gen*

		Real	
		MGS4	***Control***
Predicción	***MGS4***	13	0
	Control	29	295

Nota: Control = MGS1 + MGS2 + MGS3

- *Kernel* polinómico de grado 3 con 2 genes:

  ENSG00000005421 ENSG00000006606

Tabla 16 *Matriz de confusión kernel polinómico 2 genes*

		Real	
		MGS4	***Control***
Predicción	***MGS4***	8	1
	Control	34	294

Nota: Control = MGS1 + MGS2 + MGS3

- *Kernel* polinómico de grado 3 con 6 genes

  ENSG00000000003 ENSG00000005421 ENSG00000005471
  ENSG00000006606 ENSG00000007908 ENSG00000008516

Tabla 17 *Matriz de confusión kernel polinómico 6 genes*

		Real	
		MGS4	***Control***
Predicción	***MGS4***	30	17
	Control	12	278

Nota: Control = MGS1 + MGS2 + MGS3

Las medidas de discriminación se resumen en la Tabla 18.

Tabla 18 *Medidas de discriminación del modelo Avanzado vs Control*

Avanzado	*Radial 1 gen*			*Polinómico 2 genes*			*Polinómico 6 genes*		
	Medida	*IC 95% Medida*		*Medida*	*IC 95% Medida*		*Medida*	*IC 95% Medida*	
		Inf.	*Sup.*		*Inf.*	*Sup.*		*Inf.*	*Sup.*
AUC	0.73	0.68	0.78	0.71	0.66	0.76	0.83	0.79	0.87
Sensibilidad	0.31	0.26	0.36	0.19	0.15	0.23	0.71	0.66	0.76
Especificidad	1.00	1.000	1.00	0.99	0.98	1.00	0.94	0.92	0.97
Exactitud	0.91	0.88	0.94	0.89	0.86	0.92	0.91	0.88	0.94

Se selecciona el modelo con *kernel* polinómico con 6 genes ya que mejora el AUC por más de 0.1 unidades respecto a las otras dos opciones.

Por tanto, el clasificador estará conformado por tres modelos cuyas capacidades discriminatorias estimadas mediante validación externa quedan resumidas en las Tablas 19-22:

- Patológico (MGS2 + MGS3 + MGS4) vs Control (MGS1), el modelo con *kernel* polinómico con 6 genes: ENSG00000002079, ENSG00000004846, ENSG00000005001, ENSG00000005102, ENSG00000006042, ENSG00000007312.

Tabla 19 *Matriz de confusión Modelo Patológico vs Control en la muestra de test*

		Real	
		DMAE	***Control***
Predicción	***DMAE***	110	20
	Control	21	17

- Intermedio (MGS3) vs Control (MGS1 + MGS2 + MGS4), el modelo con *kernel* polinómico con 3 genes: ENSG00000003147, ENSG00000005379, ENSG00000005981.

Tabla 20 *Matriz de confusión Modelo Intermedio vs Control en la muestra de test*

		Real	
		MGS3	***Control***
Predicción	***MGS3***	79	32
	Control	29	28

- Avanzado (MGS4) vs Control (MGS1 + MGS2 + MGS3), el modelo con *kernel* polinómico con 6 genes: ENSG00000000003, ENSG00000005421, ENSG00000005471, ENSG00000006606, ENSG00000007908, ENSG00000008516.

Tabla 21 *Matriz de confusión Modelo Avanzado vs Control en la muestra de test*

		Real	
		MGS4	***Control***
Predicción	***MGS4***	14	37
	Control	9	108

Las medidas de discriminación del clasificador se recogen en la Tabla 22. Atendiendo al AUC, el peor modelo es el que clasifica los estadios intermedios (MGS2 vs MGS3), aunque presenta una buena sensibilidad por lo que puede ser empleado para el clasificador; los otros dos modelos mantienen una AUC superior a 0.5 en la muestra de test. Del modelo Avanzado vs Control se debe hacer notar su baja sensibilidad, teniendo en cuenta que en la muestra de test solo hay 23 individuos con nivel MGS4 (13.7% de la muestra).

Tabla 22 *Validación externa de los 3 modelos seleccionados*

Clasificador	***Patológico***			***Intermedio***			***Avanzado***		
	Medida	***IC 95% Medida***		***Medida***	***IC 95% Medida***		***Medida***	***IC 95% Medida***	
		Inf.	***Sup.***		***Inf.***	***Sup.***		***Inf.***	***Sup.***
AUC	0.67	0.60	0.74	0.55	0.47	0.63	0.71	0.64	0.78
Sensibilidad	0.84	0.79	0.90	0.73	0.66	0.80	0.61	0.54	0.68
Especificidad	0.46	0.39	0.54	0.47	0.39	0.55	0.74	0.67	0.81
Exactitud	0.76	0.70	0.83	0.64	0.57	0.71	0.73	0.66	0.80

La clasificación final que combina los tres modelos diseñados es reflejada en la Tabla 23.

Tabla 23 *Matriz de confusión Clasificador*

		Real		
		Sano MGS1	***Intermedio MGS2 + MGS3***	***Avanzado MGS4***
Predicción	***Sano MGS1***	17	10	0
	Intermedio MGS2 + MGS3	17	88	13
	Avanzado MGS4	3	10	10

Finalmente, en la Tabla 24 se muestran los resultados de aplicar las métricas para evaluar la capacidad discriminatoria del clasificador sobre la muestra externa.

Tabla 24 *Validación externa del clasificador*

	Sano MGS1			***Intermedio MGS2 + MGS3***			***Avanzado MGS4***		
	Medida	***IC 95% Medida***		***Medida***	***IC 95% Medida***		***Medida***	***IC 95% Medida***	
		Inf.	***Sup.***		***Inf.***	***Sup.***		***Inf.***	***Sup.***
Sensibilidad	0.46	0.38	0.54	0.82	0.76	0.88	0.43	0.36	0.50
Especificidad	0.75	0.68	0.82	0.45	0.37	0.53	0.72	0.65	0.79
TFP	0.09	0.05	0.13	0.53	0.45	0.61	0.11	0.06	0.16
TFN	0.54	0.47	0.62	0.19	0.13	0.25	0.57	0.50	0.64
Exactitud	0.70	0.63	0.77	0.70	0.63	0.77	0.70	0.63	0.77

9.7 – Aplicación web

En este apartado se presenta el modo funcionamiento de la aplicación web desarrollada, cuyo *script* se puede encontrar en el Anexo II. Para ejecutar la aplicación, lo único que se debe realizar es ejecutar dicho *script* en R. De este modo, la guía de usuario para utilizar la aplicación de predicción DMAE desarrollada es:

- Paso 1. Carga de datos de RNA-seq

1. Al iniciar la aplicación, podrá ver un panel central con información sobre la aplicación y los genes empleados para la predicción tal como se muestra en la Figura35.
2. Se mostrará un menú lateral en el lado izquierdo de la pantalla.
3. Seleccione la opción "Carga RNA-seq" en el menú lateral.
4. Aparecerá un cuadro de carga de archivos.
5. Haga clic en el botón "Suba sus datos en formato csv" para seleccionar el archivo CSV que contiene los datos de RNA-seq que desea analizar. Una vez se hayan cargado correctamente, se mostrará el mensaje “Se han cargado los datos, para visualizar sus resultados vaya a la sección del menú lateral.” al fondo del panel.

   - Tenga en cuenta que solo se permite la carga de archivos en formato CSV.
   - Asegúrese de que sus datos estén organizados con muestras en filas y genes en columnas en el archivo CSV.

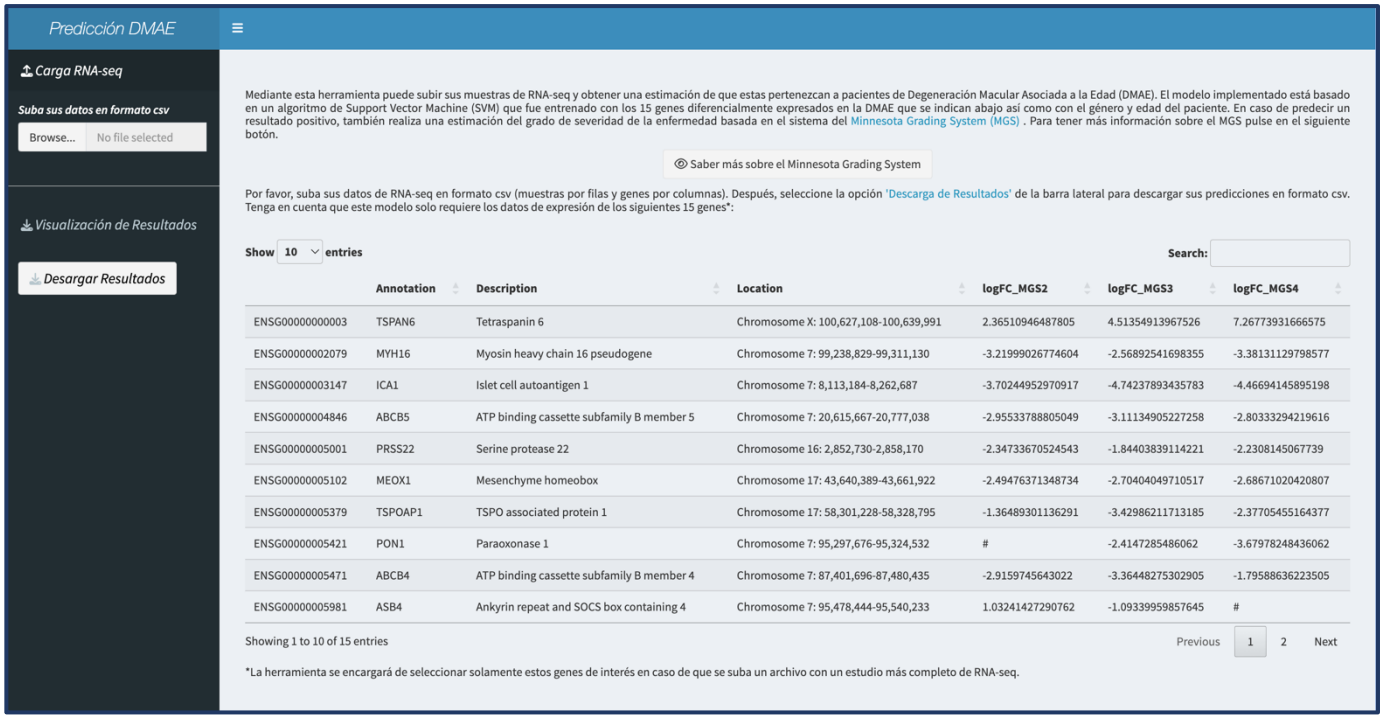

Figura 35 *Vista inicial de la aplicación*

- Paso 2. Visualización de Resultados

1. Después de cargar los datos, seleccione la opción "Visualización de Resultados" en el menú lateral.
2. Esto cambiará el contenido del panel central al igual que se muestra en la Figura 36, donde ahora visualizará una representación gráfica de los resultados.
   - Se representan 4 gráficos en los que cada punto representa una de las muestras introducidas y cuyo color está asociado con el nivel de severidad de DMAE (verde para sano, naranja para DMAE intermedia y rojo para DMAE avanzada).
   - En cada eje se representa la predicción realizada por cada uno de los modelos, de modo que en la primera representación se representan las predicciones de los tres modelos definidos en el apartado anterior en un espacio tridimensional; en el segundo, un plano con las predicciones del primer y segundo modelo; en el tercero las del primero y el tercero; y en el último, las del segundo y tercero.
   - Al fondo del panel se muestra una tabla con los valores numéricos de las predicciones realizadas por los tres modelos para cada una de las muestras, así como la clasificación final estimada.
3. Verá un botón "Descargar Resultados" en la parte superior de la página.
4. Haga clic en el botón "Descargar Resultados" para descargar las predicciones realizadas por la aplicación en formato CSV.
   - Las predicciones realizadas incluirán una estimación de si las muestras pertenecen a pacientes con DMAE y del grado de severidad de la enfermedad basada en el sistema del Minnesota Grading System (MGS), de modo que se distingue entre: sano (MGS1), intermedio (MGS2 + MGS3) y avanzado (MGS4).
   - El archivo CSV descargado contendrá los resultados de todas las muestras cargadas.

- Información adicional sobre la aplicación

- La aplicación emplea los modelos indicados en la sección anterior entrenados con los 15 genes diferencialmente expresados en la DMAE, así como con el género y edad del paciente.
- La selección de genes de interés se realiza automáticamente, en caso de que se cargue un archivo con un estudio más completo de RNA-seq.
- Para obtener más información sobre el sistema MGS, haga clic en el botón "Saber más sobre el *Minnesota Grading System*" en la página de carga de datos.

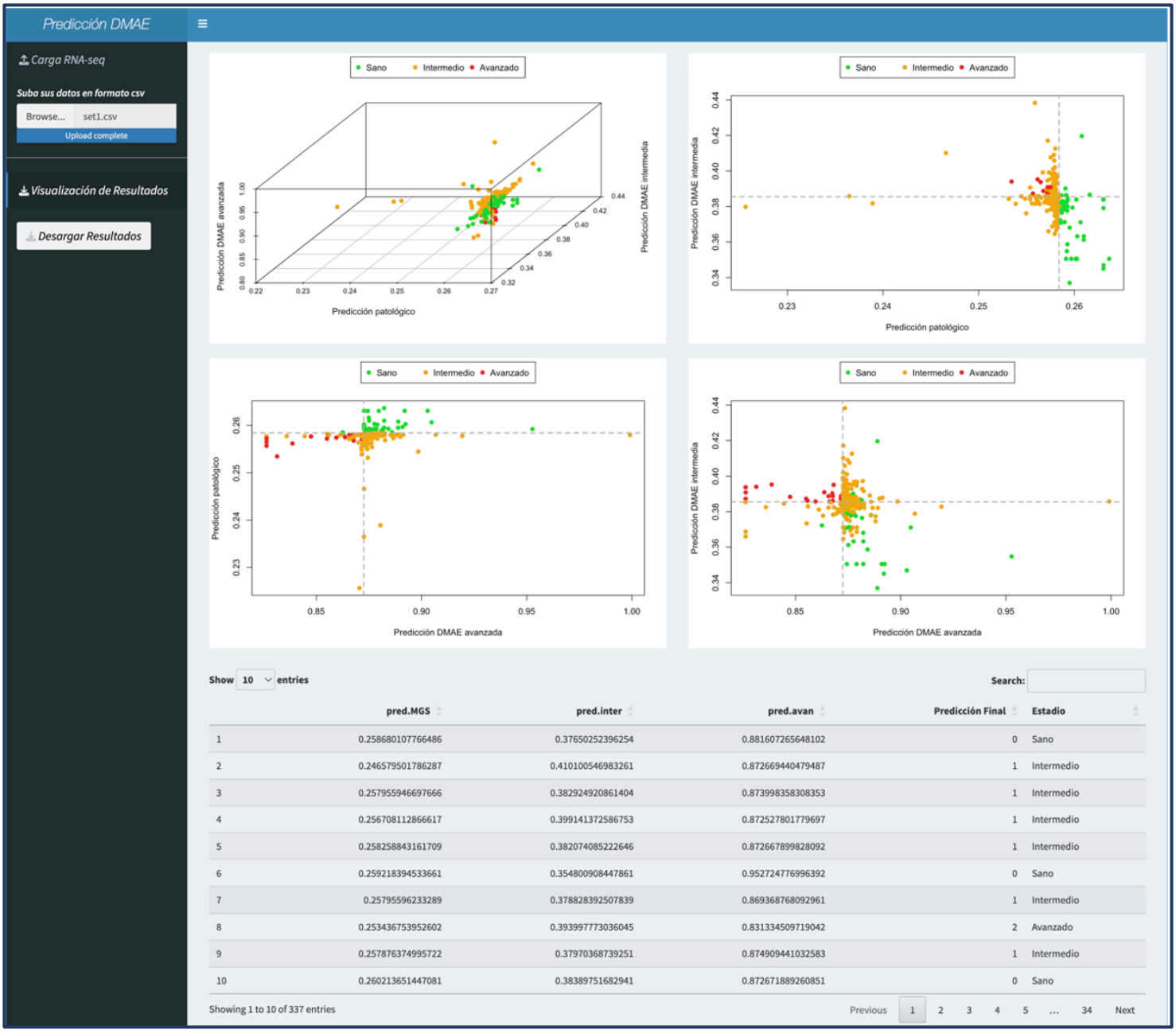

Figura 36 *Visualización de Resultados de la aplicación*

PARTE 3 CONCLUSIÓN

Capítulo 10 - Análisis y discusión de los resultados

10.1 - Perfil de expresión del paciente de DMAE
10.2 - Valoración del clasificador
10.3 - Aplicación web

Capítulo 11 - Grado de consecución de los objetivos

Capítulo 12 - Limitaciones

Capítulo 13 - Líneas futuras

Capítulo 10 - Análisis y discusión de los resultados

Este trabajo se ha enfocado en la identificación de biomarcadores específicos de la DMAE que puedan ayudar en su diagnóstico y estadificación. Para lograr esto, se ha utilizado información de expresión génica a partir de un análisis de RNA-seq para identificar genes diferencialmente expresados relacionados con la DMAE. Con ello, se ha desarrollado una metodología para procesar y analizar este tipo de datos, además de obtenerse los perfiles de expresión génica que caracterizan a los pacientes con DMAE; finalmente, se ha desarrollado una aplicación web que proporciona una interfaz sencilla para que los profesionales sanitarios puedan acceder a los resultados del trabajo.

Los hallazgos de este trabajo tienen el potencial de mejorar el diagnóstico y el tratamiento de la DMAE. Al identificar biomarcadores específicos de la enfermedad, los médicos podrían diagnosticarla con mayor precisión y comenzar un tratamiento más temprano, lo que podría mejorar la calidad de vida de los pacientes y reducir la progresión de la enfermedad. En los siguientes apartados se analizará en mayor detalle estos hallazgos mencionados.

10.1 - Perfil de expresión del paciente de DMAE

En este trabajo se aporta una metodología para saber qué genes se deben analizar cuando se estudie la DMAE. En la Tabla 25, se indica un resumen de la información más relevante de los 15 genes que se han empleado en los modelos de clasificación de la DMAE, identificados por su código Ensembl, así como los logFC para los contrastes MGS2 vs MGS1, MGS3 vs MGS1 y MGS4 vs MGS1 obtenidos en la etapa de análisis de expresión diferencial. Atendiendo a las tres últimas columnas de esta misma tabla, se pueden conocer los genes presentes en cada uno de los tres modelos diseñados y que, por tanto, se encuentran diferencialmente expresados en un paciente de DMAE en cualquier estadio, específicamente para la DMAE intermedia o para la DMAE avanzada, respectivamente.

Tabla 25 *Información de los 15 genes empleados en los modelos*

Anotación Ensembl	*Nombre del gen*	*Descripción*	*Localización*	*logFC MGS2*	*logFC MGS3*	*logFC MGS4*	*Modelo Patológico*	*Modelo Intermedio*	*Modelo Avanzado*
ENSG00000000003	TSPAN6	*Tetraspanin 6*	X: 100,627,108-100,639,991	2.37	4.51	7.27			X
ENSG00000002079	MYH16	*Myosin heavy chain 16 pseudogene*	7: 99,238,829-99,311,130	-3.22	-2.57	-3.38	X		
ENSG00000003147	ICA1	*Islet cell autoantigen 1*	7: 8,113,184-8,262,687	-3.70	-4.74	-4.47		X	
ENSG00000004846	ABCB5	*ATP binding cassette subfamily B member 5*	7: 20,615,667-20,777,038	-2.96	-3.11	-2.80	X		
ENSG00000005001	PRSS22	*Serine protease 22*	16: 2,852,730-2,858,170	-2.35	-1.84	-2.23	X		
ENSG00000005102	MEOX1	*Mesenchyme homeobox*	17: 43,640,389-43,661,922	-2.50	-2.70	-2.69	X		
ENSG00000005379	TSPOAP1	*TSPO associated protein 1*	17: 58,301,228-58,328,795	-1.36	-3.43	-2.38		X	
ENSG00000005421	PON1	*Paraoxonase 1*	7: 95,297,676-95,324,532	####	-2.41	-3.68			X
ENSG00000005471	ABCB4	*ATP binding cassette subfamily B member 4*	7: 87,401,696-87,480,435	-2.92	-3.36	-1.80			X
ENSG00000005981	ASB4	*Ankyrin repeat and SOCS box containing 4*	7: 95,478,444-95,540,233	1.03	-1.09	####		X	
ENSG00000006042	TMEM98	*Transmembrane protein 98*	17: 32,927,910-32,945,106	-1.25	-1.52	-1.62	X		
ENSG00000006606	CCL26	*C-C motif chemokine ligand 26*	7: 75,769,533-75,789,896	####	-1.40	-3.07			X
ENSG00000007312	CD79B	*CD79b molecule*	17: 63,928,738-63,932,336	-1.43	-1.37	-1.51	X		
ENSG00000007908	SELE	*Selectin E*	1: 169,722,640-169,764,705	####	####	-1.52			X
ENSG00000008516	MMP25	*Matrix metallopeptidase 25*	16: 3,046,062-3,060,726	####	####	1.26			X

Los perfiles de expresión génica descritos en este apartado podrían proporcionar información valiosa sobre la patogenia de la DMAE y, por lo tanto, ayudar en el desarrollo de nuevas terapias para la misma. Asimismo, se debe indicar, como valor añadido, que la metodología empleada para obtener estos perfiles se podría aplicar a estudios de otras patologías, ya que la DMAE en sí es irrelevante para la metodología, el análisis se podría aplicar fuera cual fuera la enfermedad de estudio.

10.2 - Valoración del clasificador

El clasificador resultante no presenta unos resultados excelentes que pudieran permitir su empleo como herramienta diagnóstica en clínica; sin embargo, sí son unos resultados razonables que cumplen las expectativas de este Trabajo Fin de Grado. No se ha logrado distinguir bien las muestras entre los distintos estadios. En las muestras empleadas para el diseño del clasificador hay que destacar su gran desequilibrio: se disponía de muchas más muestras de DMAE intermedia que del resto (aproximadamente el 60 % tanto en la sub-muestra de entrenamiento como en la de test). Esto justifica la alta sensibilidad en la clase mayoritaria y una baja sensibilidad en la minoritaria. Con ello en mente, no es posible alcanzar una clasificación por estadios a través de este análisis basado exclusivamente en datos de RNA-seq, pero lo que sí se ha logrado es una posible técnica de *screening* de la DMAE. Es decir, este clasificador sí es capaz de discriminar las muestras patológicas de las sanas, llegando a sobre-diagnosticar la enfermedad, lo cual es positivo en herramientas cuyo fin es el diagnóstico precoz para la prevención de la patología.

10.3 - Aplicación web

La aplicación web desarrollada en este trabajo permite diagnosticar enfermos de DMAE a partir de los datos de expresión de los genes comentados en el Capítulo 10.1, lo que puede ser de gran utilidad para los profesionales sanitarios que buscan una herramienta de diagnóstico rápida de la DMAE. Sin embargo, la aplicación no es capaz de distinguir adecuadamente entre los diferentes estadios de la enfermedad, lo que limita su utilidad clínica. A pesar de esta limitación, sigue siendo una herramienta valiosa para la investigación de la DMAE, ya que proporciona una forma fácil de acceder a los resultados del trabajo. Los investigadores pueden utilizar la aplicación para analizar grandes conjuntos de datos de pacientes. En última instancia, la aplicación web desarrollada sirve para hacer útil la información a un sanitario, no es el valor en sí de este trabajo.

Capítulo 11 - Grado de consecución de los objetivos

A continuación, se va a analizar el grado de consecución de cada uno de los objetivos planteados en base a los resultados obtenidos.

El objetivo principal del trabajo "Encontrar biomarcadores específicos de la DMAE que permitan su diagnóstico y clasificación por estadios" ha sido cumplido prácticamente en su totalidad, ya que efectivamente se ha obtenido biomarcadores específicos de la DMAE, en concreto, un listado de 15 genes diferencialmente expresados:

ENSG00000000003	ENSG00000002079	ENSG00000003147
ENSG00000004846	ENSG00000005001	ENSG00000005102
ENSG00000005379	ENSG00000005421	ENSG00000005471
ENSG00000005981	ENSG00000006042	ENSG00000006606
ENSG00000007312	ENSG00000007908	ENSG00000008516

Este listado de genes ha permitido además el desarrollo de un clasificador para el diagnóstico de la DMAE; sin embargo, la clasificación por estadios en base a estos biomarcadores no ha sido adecuadamente lograda debido a las limitaciones de la muestra de estudio que se comentarán más adelante.

En referencia a los 4 objetivos específicos planteados en este trabajo:

1. Utilizar datos de RNA-seq para obtener genes diferencialmente expresados en relación a la DMAE.

2. Establecer un protocolo de tratamiento de datos de RNA-seq, para obtener información útil de este tipo de datos.

3. Identificar, describir y diferenciar los perfiles de expresión génica que caracterizan al paciente de DMAE en cada uno de sus estadios.

4. Desarrollar una aplicación web que ofrezca una interfaz sencilla para que profesionales sanitarios puedan diagnosticar y clasificar a un paciente a partir de su perfil de expresión.

Los objetivos 3 y 4 acaban de ser tratados en los Capítulos 10.1 y 10.3, respectivamente. Por su parte, se debe indicar que los dos primeros objetivos específicos han sido alcanzados satisfactoriamente, ya que el análisis de expresión diferencial ha aportado el listado de genes indicado aquí a través del protocolo que se ha aplicado en la metodología del trabajo, en resumen, este protocolo de tratamiento de datos de RNA-seq consiste en:

I. Preprocesado de los datos de conteo
   a. Descartar muestras mal registradas y genes poco expresados
   b. Transformación logarítmica de los datos de conteo

II. Análisis exploratorio de los datos
   a. Normalización mediante TMM
   b. Análisis de Componentes Principales

III. Análisis de expresión diferencial
   a. *Clustering* jerárquico
   b. Análisis de enriquecimiento

IV. Diseño y ajuste del clasificador
   a. Definición de los modelos
   b. Validación externa

Capítulo 12 - Limitaciones

Aunque a lo largo del trabajo se ha hablado de un estadio intermedio entre los dos conocidos (inicial y avanzado), se desconoce qué mecanismos conducen de un estadio a otro, lo cual dificulta la clasificación por estadios de la DMAE y la búsqueda de biomarcadores específicos para cada uno de los estadios concretos. Por ello, se decidió emplear el sistema de clasificación MGS a fin de tener una clasificación clara entre las muestras para poder realizar el estudio. Sin embargo, es importante tener en cuenta que este sistema de clasificación no es ampliamente utilizado en la práctica clínica y no es tan conocido como el sistema de clasificación publicado por AREDS (*Age-Related Eye Disease Study*) y el sistema de clasificación de la DMAE según la Clasificación Internacional de la Enfermedad (CIE).

- El sistema de clasificación de AREDS divide la DMAE en cuatro categorías al igual que el MGS: ausencia de DMAE, DMAE temprana DMAE intermedia y DMAE avanzada. Se basa en la presencia de características específicas en el fondo de ojo, como drusas, cambios en la capa de células epiteliales pigmentarias de la retina y la presencia de neovascularización coroidea, pero carece de unas pautas medibles como las que presenta el MGS [17].

- El sistema de clasificación de la CIE es un sistema de codificación utilizado en medicina para clasificar y codificar enfermedades. La DMAE se identifica en la CIE bajo el código H35.3 y se divide en cuatro categorías:

  H35.30: DMAE no especificada.
  H35.31: DMAE temprana.
  H35.32: DMAE intermedia.
  H35.33: DMAE avanzada.

  Este sistema de clasificación se basa principalmente en el grado de gravedad de la enfermedad y no proporciona detalles específicos sobre las características clínicas o el seguimiento de la enfermedad [48].

Por tanto, aunque el sistema de clasificación del MGS es menos conocido y utilizado, proporcionan una forma más concreta de evaluar y estadificar la DMAE, requerimiento que se perseguía en este trabajo.

Por otra parte, en este trabajo solo se ha tenido acceso a un estudio de RNA-seq, lo cual condiciona la solidez del análisis; sin embargo, la inclusión de más datos no sería un problema en caso de disponer de ellos. De igual modo ocurre con la posibilidad de añadir un análisis de Polimorfismos de nucleótido único (SNPs) para mejorar la predicción. Si en este trabajo no se ha realizado es precisamente por ausencia de datos de esta naturaleza disponibles en bases públicas. Asimismo, se debe recordar que las muestras de este estudio se han obtenido mediante biopsias de retina, este método es altamente invasivo e inviable su realización en clínica, por ello, se emplearon muestras *postmortem*; sin embargo, existen varias soluciones a este asunto, ya que si se lograse extrapolar los resultados a otros tejidos, el método sería más adecuado para la clínica; además, es posible que las técnicas de obtención de biopsias avancen a tal nivel que realizar la extracción de una única célula de retina sobre un paciente sea factible, de modo que toda la metodología planteada en este trabajo sería perfectamente aplicable.

Capítulo 13 – Líneas futuras

Retomando la idea con la que se cerró el capítulo anterior, se debe indicar que, en la actualidad, la aplicación del método descrito en este trabajo para el diagnóstico y clasificación por estadios de la DMAE no se podría llevar a cabo en clínica, pero el futuro de este tipo de tecnología es prometedor. Su avance en los últimos años ha tenido un ritmo vertiginoso y no parece que haya alcanzado su máximo desarrollo aún. Es una posibilidad cada vez más real que en las clínicas se apliquen métodos de diagnóstico basados en analizar la presencia de biomarcadores específicos para una patología. Por tanto, el valor real que personalmente creo que aporta este Trabajo Fin de Grado es de carácter científico por el desarrollo de estas metodologías y no tanto la herramienta desarrollada en sí, ya que como se ha comentado previamente, la capacidad discriminatoria del clasificador no es tan buena y hay muchas limitaciones a la hora de obtener muestras por los métodos empleados en este estudio. El clasificador obtenido puede ser valioso para los investigadores que buscan desarrollar nuevas terapias para la enfermedad o mejorar la eficacia de las terapias existentes. Es más, los genes encontrados podrían ser ejemplos de dianas para la investigación de la mecánica subyacente en la DMAE. Su potencial en clínica podría ser mayor, en caso de mejorar la extracción de biopsias de retina o si se pudieran extrapolar estos resultados a otro tipo de tejido más accesible. Además, se podría incorporar la información genética en otras herramientas de clasificación de la enfermedad, basadas en factores más clínicos, mejorando su capacidad discriminatoria.

En conclusión, este trabajo tiene el potencial de mejorar el entendimiento de la Degeneración Macular Asociada a la Edad a largo plazo. Al estudiar y aplicar conceptos de bioinformática, se ha logrado adquirir un valioso aprendizaje personal. La bioinformática brinda herramientas y técnicas para analizar grandes conjuntos de datos genómicos y comprender mejor las bases moleculares de la DMAE. Este enfoque puede llevar a avances significativos en la identificación de biomarcadores, la comprensión de los mecanismos subyacentes de la enfermedad y, en última instancia, al desarrollo de nuevas estrategias de prevención y tratamiento para mejorar la salud visual de los pacientes con DMAE.

BIBLIOGRAFÍA

[1] W. L. Wong *et al.*, «Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis», *Lancet Glob Health*, vol. 2, n.º 2, feb. 2014, doi: 10.1016/S2214-109X(13)70145-1.

[2] K. L. Pennington y M. M. Deangelis, «Epidemiology of age-related macular degeneration (AMD): associations with cardiovascular disease phenotypes and lipid factors», doi: 10.1186/s40662-016-0063-5.

[3] J. M. R. Moreno, F. Cabrera López, A. G. Layana, J. García, A. Luis, y A. Barquet, «Protocolo de diagnóstico, seguimiento y recomendaciones generales en la degeneración macular asociada a la edad (DMAE) precoz e intermedia: consenso de un panel de expertos».

[4] Chang W *et al.*, «Shiny: Web Application Framework for R_. R package version 1.7.4»,. 2022.

[5] D. Mª y C. P. Marín, «Óptica Fisiológica: El sistema óptico del ojo y la visión binocular».

[6] D. Ardeljan y C. C. Chan, «Aging is not a disease: distinguishing age-related macular degeneration from aging», *Prog Retin Eye Res*, vol. 37, pp. 68-89, 2013, doi: 10.1016/J.PRETEYERES.2013.07.003.

[7] W. L. Wong *et al.*, «Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis», *Lancet Glob Health*, vol. 2, n.º 2, feb. 2014, doi: 10.1016/S2214-109X(13)70145-1.

[8] M. van Lookeren Campagne, J. Lecouter, B. L. Yaspan, y W. Ye, «Mechanisms of age-related macular degeneration and therapeutic opportunities», *Journal of Pathology*, vol. 232, n.º 2, pp. 151-164, ene. 2014, doi: 10.1002/PATH.4266.

[9] M. Sastre-Ibáñez, A. Barreiro-González, R. Gallego-Pinazo, R. Dolz-Marco, y B. García-Armendariz, «Atrofia geográfica: etiopatogenia y terapias actuales», *Arch Soc Esp Oftalmol*, vol. 93, n.º 1, pp. 22-34, ene. 2018, doi: 10.1016/J.OFTAL.2017.07.004.

[10] Z. Ungvari, S. Tarantini, A. J. Donato, V. Galvan, y A. Csiszar, «Mechanisms of vascular aging», *Circ Res*, vol. 123, n.º 7, pp. 849-867, 2018, doi: 10.1161/CIRCRESAHA.118.311378/FORMAT/EPUB.

[11] E. H. Sohn *et al.*, «Choriocapillaris Degeneration in Geographic Atrophy», *Am J Pathol*, vol. 189, n.º 7, pp. 1473-1480, jul. 2019, doi: 10.1016/J.AJPATH.2019.04.005.

[12] B. Lee, J. Ahn, C. Yun, S. W. Kim, y J. Oh, «Variation of Retinal and Choroidal Vasculatures in Patients With Age-Related Macular Degeneration», *Invest Ophthalmol Vis Sci*, vol. 59, n.º 12, pp. 5246-5255, oct. 2018, doi: 10.1167/IOVS.17-23600.

[13] A. Kauppinen, J. J. Paterno, J. Blasiak, A. Salminen, y K. Kaarniranta, «Inflammation and its role in age-related macular degeneration», *Cellular and Molecular Life Sciences*, vol. 73, n.º 9, pp. 1765-1786, may 2016, doi: 10.1007/S00018-016-2147-8.

[14] N. Piippo *et al.*, «Decline in cellular clearance systems induces inflammasome signaling in human ARPE-19 cells», *Biochim Biophys Acta Mol Cell Res*, vol. 1843, n.º 12, pp. 3038-3046, 2014, doi: 10.1016/J.BBAMCR.2014.09.015.

[15] F. Sasaki *et al.*, «Leukotriene B4 promotes neovascularization and macrophage recruitment in murine wet-type AMD models», *JCI Insight*, vol. 3, n.º 18, sep. 2018, doi: 10.1172/JCI.INSIGHT.96902.
[16] T. W. Olsen y X. Feng, «The Minnesota Grading System of eye bank eyes for age-related macular degeneration», *Invest Ophthalmol Vis Sci*, vol. 45, n.º 12, pp. 4484-4490, dic. 2004, doi: 10.1167/IOVS.04-0342.
[17] A. S. Lindblad *et al.*, «The Age-Related Eye Disease Study (AREDS): Design Implications AREDS Report No. 1», *Control Clin Trials*, vol. 20, n.º 6, p. 573, 1999, doi: 10.1016/S0197-2456(99)00031-8.
[18] «DMAE, Degeneración Macular Asociada a la Edad | ICR», 2020.
[19] S. Mehta, «Degeneración macular asociada a la edad (DMAE) - Trastornos oftálmicos - Manual MSD versión para público general».
[20] Gómez-Ulla, «Test de Amsler | Instituto Oftalmológico Gómez-Ulla». 2022.
[21] Vargas A y Delie F, «Uso potencial de nanopartículas biodegradables en la terapia fotodinámica de enfermedades oculares», *Archivo Sociedad Española de Oftalmología*, vol. 84, pp. 169-176, 2009.
[22] D. L. Nelson, *Lehninger : principios de bioquímica*, 7ª ed. Barcelona: Omega, 2019.
[23] L. A. Pray, «Discovery of DNA Double Helix: Watson and Crick», *Nature*, vol. Education 1(1):100, 2008.
[24] Genetic Alliance; The New York-Mid-Atlantic Consortium for Genetic and Newborn Screening Services., *Cómo entender la genética: Una guía para pacientes y profesionales médicos en la región de Nueva York y el Atlántico Medio.* Washington (DC): Genetic Alliance, 2009.
[25] A. D. Baxevanis, G. D. Bader, y D. S. Wishart, *Bioinformatics*, 4ª ed. Hoboken, New Jersey: Wiley, 2020.
[26] Z. Wang, M. Gerstein, y M. Snyder, «RNA-Seq: a revolutionary tool for transcriptomics», *Nat Rev Genet*, vol. 10, n.º 1, pp. 57-63, ene. 2009, doi: 10.1038/NRG2484.
[27] M. K. Schmid, M. A. Thiel, K. Lienhard, R. O. Schlingemann, L. Faes, y L. M. Bachmann, «Reliability and diagnostic performance of a novel mobile app for hyperacuity self-monitoring in patients with age-related macular degeneration», *Eye 2019 33:10*, vol. 33, n.º 10, pp. 1584-1589, may 2019, doi: 10.1038/s41433-019-0455-6.
[28] W. Chang *et al.*, «shiny: Web Application Framework for R». 15 de diciembre de 2022.
[29] M. H. H. Withanage, H. Liang, y E. Zeng, «RNA-Seq Experiment and Data Analysis», *Methods Mol Biol*, vol. 2418, pp. 405-424, 2022, doi: 10.1007/978-1-0716-1920-9_22.
[30] J. Sian-Hülsmann, C. M. Monoranu, E. Grünblatt, y P. Riederer, «Neurochemical markers as potential indicators of postmortem tissue quality», *Handb Clin Neurol*, vol. 150, pp. 119-127, ene. 2018, doi: 10.1016/B978-0-444-63639-3.00009-8.
[31] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, y B. Wold, «Mapping and quantifying mammalian transcriptomes by RNA-Seq», *Nat Methods*, vol. 5, n.º 7, pp. 621-628, jul. 2008, doi: 10.1038/NMETH.1226.

[32] S. Anders y W. Huber, «Differential expression analysis for sequence count data», *Genome Biol*, vol. 11, n.º 10, pp. 1-12, oct. 2010, doi: 10.1186/GB-2010-11-10-R106/COMMENTS.
[33] Hadley Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
[34] M. Ghorbani y E. K. P. Chong, «Stock price prediction using principal components», *PLoS One*, vol. 15, n.º 3, 2020, doi: 10.1371/JOURNAL.PONE.0230124.
[35] P. McCullagh y J. A. Nelder, «Generalized Linear Models», 1989, doi: 10.1007/978-1-4899-3242-6.
[36] M. D. Robinson, D. J. McCarthy, y G. K. Smyth, «<tt>edgeR</tt> : a Bioconductor package for differential expression analysis of digital gene expression data», *Bioinformatics*, vol. 26, n.º 1, pp. 139-140, ene. 2010, doi: 10.1093/bioinformatics/btp616.
[37] C. Law, K. Zeglinski, X. Dong, M. Alhamdoosh, G. K Smyth, y M. E Ritchie, «A guide to creating design matrices for gene expression experiments», *Bioconductor*. 17 de noviembre de 2020.
[38] Benjamini Y y Hochberg Y, «Controlling the false discovery rate: a practical and powerful approach to multiple testing.», *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, pp. 289-300, 1995.
[39] «08 Cluster analysis », *Introduction to RNA-seq*. 2020.
[40] T. Wu *et al.*, «clusterProfiler 4.0: A universal enrichment tool for interpreting omics data», *The Innovation*, vol. 2, n.º 3, p. 100141, ago. 2021, doi: 10.1016/j.xinn.2021.100141.
[41] A. Subramanian *et al.*, «Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles», *Proc Natl Acad Sci U S A*, vol. 102, n.º 43, pp. 15545-15550, oct. 2005, doi: 10.1073/PNAS.0506580102.
[42] Y. Guangchuang, «enrichplot: Visualization of Functional Enrichment Result». 2023.
[43] Meyer D, Dimitriadou E, Hornik K, Weingessel A, y Leisch F, «e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien». 2023.
[44] C. Ding y H. Peng, «Minimum redundancy feature selection from microarray gene expression data», *J Bioinform Comput Biol*, vol. 3, n.º 2, pp. 185-205, abr. 2005, doi: 10.1142/S0219720005001004.
[45] N. De Jay, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, y B. Haibe-Kains, «mRMRe: an R package for parallelized mRMR ensemble feature selection», *Bioinformatics*, vol. 29, n.º 18, pp. 2365-2368, sep. 2013, doi: 10.1093/BIOINFORMATICS/BTT383.
[46] H. Wickham, «Elegant Graphics for Data Analysis», *Springer-Verlag New York*, 2016.
[47] Hadley Wickham, Romain François, y Lionel Henry, «A Grammar of Data Manipulation. R package version 1.1.2».
[48] M. DE Sanidad y C. Y. Bienestar Social, «Diagnósticos y Procedimientos NÚMERO 4. 2º SEMESTRE 2017 UNIDAD TÉCNICA DE CODIFICACIÓN CIE-10-ES».

ANEXO I - GLOSARIO DE ABREVIATURAS Y ACRÓNIMOS

ADN	**Á**cido **D**esoxirribo**n**ucleico
AREDS	*Age-Related Eye Disease Study*
ARN	**Á**cido **R**ibo**n**ucleico
AUC	**Á**rea **B**ajo la **C**urva ROC
CIE	**C**lasificación **I**nternacional de la **E**nfermedad
DMAE	**D**egeneración **M**acular **A**sociada a la **E**dad
EBI	*European Bioinformatics Institute*
EPR	**E**pitelio **P**igmentario de la **R**etina
FDA	*Food and Drug Administration*
FC	*Fold Change*
GEO	*Gene Expression Omnibus*
GLM	*General Linear Model*
GO	*Gene Ontology*
IC	**I**ntervalo de **C**onfianza
IQR	**R**ango **I**nter**c**uartil
KEGG	*Kyoto Encyclopedia of Genes and Genomes*
LOOCV	*Leave One Out Cross Validation*
MGS	*Minnesota Grading System*
mRMR	*Minimum Redundancy – Maximum Relevance algorithm*
mRNA	**ARN M**ensajero
NCBI	*National Center for Biotechnology Information*
NGS	**S**ecuenciación de **N**ueva **G**eneración
NHGRI	*National Human Genome Research Institute*
OCT	**T**omografía de **C**oherencia **Ó**ptica
PDT	**T**erapia **F**oto**d**inámica
PMI	*Post-Mortem Interval*
RIN	*RNA Integrity Number*
RNA-seq	**S**ecuenciación de **ARN**
ROS	**E**species **R**eactivas de **O**xígeno
SVM	*Support Vector Machine*
rRNA	**ARN R**ibosómico
tRNA	**ARN** de **T**ransferencia
VEGF	**F**actor de **C**recimiento **E**ndotelial **V**ascular

ANEXO II - RUTINAS DE R

```
setwd("/Users/blazwel/Desktop/TFG/Códigos")

summary_data_set <- read.table(file.path("/Users/blazwel/Desktop/TFG/Códigos", "GSE115828_DE_analysis.txt"), header=TRUE)

rawCountTable <- read.table(file.path("/Users/blazwel/Desktop/TFG/Códigos", "GSE115828_RSEM_gene_counts.tsv"), header=TRUE)

head(rawCountTable)
```

```
library(GEOquery)

SampleInfo <- getGEO(filename="GSE115828_series_matrix.txt",GSEMatrix = TRUE,getGPL = FALSE) #Retrieve matrix data and store it in R object

MGS_level <- SampleInfo$`mgs_level:ch1`

sex <- SampleInfo$`Sex:ch1`

age <- SampleInfo$`age:ch1`

rin <- as.numeric(SampleInfo$`rin:ch1`) # RNA integrity number

n <- length(MGS_level)

n1 <- length(which(MGS_level == 1))

n2 <- length(which(MGS_level == 2))

n3 <- length(which(MGS_level == 3))

n4 <- length(which(MGS_level == 4))
```

```
low.RIN.samples <- c(which(rin=='N/A'), which(rin<5))

# Elimino las 12 muestras descartadas

count.aux <- rawCountTable[,2:524]

count.aux <- count.aux[-low.RIN.samples]

rawCountTable <- data.frame(GeneID = rawCountTable$GeneID, count.aux)

MGS_level <- (MGS_level[-low.RIN.samples])

age <- age[-low.RIN.samples]

sex <- sex[-low.RIN.samples]

rin <- rin[-low.RIN.samples]


missing.values <- c(which(MGS_level == 'NA'), which(MGS_level == 'no grade'), which(MGS_level == '2 or 3'))

# Descarto las 6 muestras sin clasificar

count.aux <- rawCountTable[,2:512]

count.aux <- count.aux[-missing.values]
```

```
rawCountTable <- data.frame(GeneID = rawCountTable$GeneID, count.aux)
MGS_level <- (MGS_level[-missing.values])
age <- age[-missing.values]
sex <- sex[-missing.values]
rin <- rin[-missing.values]
keep <- ''
for(gen in 1:58051){
  keep1 <- length(which(rawCountTable[gen,which(MGS_level == 1)]!=0))/
n1
  if(keep1 >= 0.9){
      keep2 <- length(which(rawCountTable[gen,which(MGS_level == 2)]!=
0))/n2
      if(keep2 >= 0.9){
      keep3 <- length(which(rawCountTable[gen,which(MGS_level == 3)]!=
0))/n3
      if(keep3 >= 0.9){
         keep4 <- length(which(rawCountTable[gen,which(MGS_level == 4)
]!=0))/n4
         if(keep4 >= 0.9){
           keep <- c(keep,gen)
         }
         }
      }
 }
}
keep <- keep[2:length(keep)]


matriz.conteo <- rawCountTable[]
matriz.conteo <- data.frame(matriz.conteo[keep,])
dim(matriz.conteo)
```

```
library(ggplot2)
library(gridExtra)
library(reshape)
log2CountTable <- log2(matriz.conteo[,2:506] + 1)
df.raw <- reshape::melt(as.matrix(log2CountTable), id = rownames(matri
z.conteo))
```

```
names(df.raw)[1:2] <- c('id', 'sample')

Distribucion_conteo <- ggplot(df.raw, aes(x = sample, y = value)) + ge
om_boxplot() + theme_bw() + ggtitle("Distribución conteos") + xlab("Mu
estras") + ylab(expression(log[2](conteos + 1))) + theme(axis.text.x =
element_blank()) +  theme(plot.title = element_text(hjust = 0.5))

Distribucion_conteo


set.seed(124)


MGS1.cols <- sample(length(which(MGS_level==1)), 0.1*length(which(MGS_
level==1)), replace = FALSE)

log2MGS1 <-  log2(matriz.conteo[,MGS1.cols] + 1)

df.MGS1 <- reshape::melt(as.matrix(log2MGS1), id = rownames(matriz.con
teo))

names(df.MGS1)[1:2] <- c('id', 'sample')

df.MGS1$group <- rep('MGS1', nrow(df.MGS1))


MGS2.cols <- sample(length(which(MGS_level==2)), 0.1*length(which(MGS_
level==2)), replace = FALSE)

log2MGS2 <-  log2(matriz.conteo[,MGS2.cols] + 1)

df.MGS2 <- reshape::melt(as.matrix(log2MGS2), id = rownames(matriz.con
teo))

names(df.MGS2)[1:2] <- c('id', 'sample')

df.MGS2$group <- rep('MGS2', nrow(df.MGS2))


MGS3.cols <- sample(length(which(MGS_level==3)), 0.1*length(which(MGS_
level==3)), replace = FALSE)

log2MGS3 <-  log2(matriz.conteo[,MGS3.cols] + 1)

df.MGS3 <- reshape::melt(as.matrix(log2MGS3), id = rownames(matriz.con
teo))

        names(df.MGS3)[1:2] <- c('id', 'sample')

df.MGS3$group <- rep('MGS3', nrow(df.MGS3))
```

```
MGS4.cols <- sample(length(which(MGS_level==4)), 0.1*length(which(MGS_
level==4)), replace = FALSE)

log2MGS4 <-  log2(matriz.conteo[,MGS4.cols] + 1)

df.MGS4 <- reshape::melt(as.matrix(log2MGS4), id = rownames(matriz.con
teo))

names(df.MGS4)[1:2] <- c('id', 'sample')

df.MGS4$group <- rep('MGS4', nrow(df.MGS4))


df.all <- rbind(df.MGS1,df.MGS2,df.MGS3,df.MGS4)

df.all$group <- factor(df.all$group, levels = c('MGS1','MGS2','MGS3','
MGS4'))


ggplot(df.all, aes(x = value, colour = sample)) + geom_density(show.le
gend = FALSE) + theme_bw() + facet_grid(. ~ group)+ ylab('Densidad')
+  theme(plot.title = element_text(hjust = 0.5)) + ylim(c(0,0.3)) + xl
im(c(0,15))
```

```
ggplot(df.all, aes(x = sample, y = value, fill = group)) + geom_boxplo
t(show.legend = FALSE) + theme_bw() + facet_grid(. ~ group)+  ylab(exp
ression(log[2](conteos + 1))) + theme(axis.text.x = element_blank()) +
theme(plot.title = element_text(hjust = 0.5)) + xlab("")
```

```
if (!require("BiocManager", quietly = TRUE))

    install.packages("BiocManager")

BiocManager::install("edgeR")


matriz.conteo.dge <- edgeR::DGEList(as.matrix(matriz.conteo))

normalizacion.TMM <- data.frame(normalizacion.TMM, row.names = rowname
s(matriz.conteo))


TMM.MGS1 <- reshape::melt(as.matrix(normalizacion.TMM[,MGS1.cols]), id
= rownames(normalizacion.TMM))

names(TMM.MGS1)[1:2] <- c('id', 'sample')

TMM.MGS1$group <- rep('MGS1', nrow(df.MGS1))


TMM.MGS2 <- reshape::melt(as.matrix(normalizacion.TMM[,MGS2.cols]), id
= rownames(normalizacion.TMM))

names(TMM.MGS2)[1:2] <- c('id', 'sample')

TMM.MGS2$group <- rep('MGS2', nrow(df.MGS2))
```

```
TMM.MGS3 <- reshape::melt(as.matrix(normalizacion.TMM[,MGS3.cols]), id
= rownames(normalizacion.TMM))

names(TMM.MGS3)[1:2] <- c('id', 'sample')

TMM.MGS3$group <- rep('MGS3', nrow(df.MGS3))


TMM.MGS4 <- reshape::melt(as.matrix(normalizacion.TMM[,MGS4.cols]), id
= rownames(normalizacion.TMM))

names(TMM.MGS4)[1:2] <- c('id', 'sample')

TMM.MGS4$group <- rep('MGS4', nrow(df.MGS4))


df.all.TMM <- rbind(TMM.MGS1,TMM.MGS2,TMM.MGS3,TMM.MGS4)

df.all.TMM$group <- factor(df.all.TMM$group, levels = c('MGS1','MGS2',
'MGS3','MGS4'))


ggplot(df.all.TMM, aes(x = value, colour = sample)) + geom_density(sho
w.legend = FALSE) + theme_bw() + facet_grid(. ~ group)+ ylab('Densidad
') +  theme(plot.title = element_text(hjust = 0.5)) + ylim(c(0,0.3)) +
xlim(c(0,15))
```

```
ggplot(df.all.TMM, aes(x = sample, y = value, fill = group)) + geom_bo
xplot(show.legend = FALSE) + theme_bw() + facet_grid(. ~ group)+  ylab
(expression(log[2](conteos + 1))) + theme(axis.text.x = element_blank(
)) +  theme(plot.title = element_text(hjust = 0.5)) + xlab("")
```

```
library(factoextra)

corr.matrix <- cor(normalizacion.TMM)

myPCA <- prcomp(corr.matrix, center = TRUE, scale. = TRUE)

fviz_eig(myPCA)

pca1 <- princomp(corr.matrix, scores = TRUE, cor = TRUE)

pca1$sdev[1:10]^2 # Autovalores

myPCA.summary <- summary(myPCA)

cum.proportion <- myPCA.summary$importance[3,]

cum.proportion[1:10]

MGS_level <- as.factor(MGS_level)

plot(myPCA$x[,1:2], col=as.numeric(MGS_level), pch=19)

legend('topright', inset = 0.05, levels(MGS_level), pch=19, col=1:4, h
oriz= TRUE)
```

```
#  The level associated with the control group is not listed first, it
can be changed to the first or reference level
MGS_level <- as.numeric(MGS_level)
MGS_level <- as.factor(MGS_level)
MGS_level <- relevel(MGS_level, ref=1)


matriz.GLM <- matriz.conteo[,2:506]
rownames(matriz.GLM) <- matriz.conteo$GeneID
matriz.GLM <- edgeR::DGEList(matriz.GLM, group = MGS_level)
matriz.GLM <- edgeR::calcNormFactors(matriz.conteo.dge, method = 'TMM'
)


age <- as.numeric(age)
sex <- replace(sex, which(sex=='F'),0)
sex <- replace(sex, which(sex=='M'),1)
sex <- as.numeric(sex)


matriz.GLM$samples$replicate <- MGS_level
matriz.GLM$samples$age <- age
matriz.GLM$samples$sex <- sex
design.matrix <- model.matrix(~ 0 + matriz.GLM$samples$replicate + age
+ sex)
colnames(design.matrix) <- c('MGS1','MGS2','MGS3','MGS4','Age','Sex')
matriz.GLM <- edgeR::estimateCommonDisp(matriz.GLM)
matriz.GLM <- edgeR::estimateTagwiseDisp(matriz.GLM)
plotBCV(matriz.GLM)
```

```
contrast.matrix <- makeContrasts(MGS2-MGS1,MGS3-MGS1,MGS4-MGS1,MGS3-MG
S2,MGS4-MGS2,MGS4-MGS3,levels=c(colnames(design.matrix)))

fit <- edgeR::glmFit(matriz.GLM, design.matrix)


test <- glmLRT(fit, contrast = contrast.matrix)

res <- topTags(test, n =nrow(matriz.GLM$counts))


rows.MGS2 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS2...MGS1) >
1

selected.genes.MGS2 <- res$table[rows.MGS2,]

selected.genes.MGS2$updown <- factor(ifelse(selected.genes.MGS2$logFC.
MGS2...MGS1 > 0, 'up', 'down'))

rownames(selected.genes.MGS2) <- matriz.conteo$GeneID[rows.MGS2]

nrow(selected.genes.MGS2)
```

```
rows.MGS3 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS3...MGS1) >
1

selected.genes.MGS3 <- res$table[rows.MGS3,]

selected.genes.MGS3$updown <- factor(ifelse(selected.genes.MGS3$logFC.
MGS3...MGS1 > 0, 'up', 'down'))

rownames(selected.genes.MGS3) <- matriz.conteo$GeneID[rows.MGS3]

nrow(selected.genes.MGS3)
```

```
rows.MGS4 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS4...MGS1) >
1

selected.genes.MGS4 <- res$table[rows.MGS4,]

selected.genes.MGS4$updown <- factor(ifelse(selected.genes.MGS4$logFC.
MGS4...MGS1 > 0, 'up', 'down'))

rownames(selected.genes.MGS4) <- matriz.conteo$GeneID[rows.MGS4]

nrow(selected.genes.MGS4)
```

```
rows.MGS32 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS3...MGS2)
> 1

selected.genes.MGS32 <- res$table[rows.MGS32,]

selected.genes.MGS32$updown <- factor(ifelse(selected.genes.MGS32$logF
C.MGS3...MGS2 > 0, 'up', 'down'))

rownames(selected.genes.MGS32) <- matriz.conteo$GeneID[rows.MGS32]

nrow(selected.genes.MGS32)
```

```
rows.MGS42 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS4...MGS2)
> 1

selected.genes.MGS42 <- res$table[rows.MGS42,]

selected.genes.MGS42$updown <- factor(ifelse(selected.genes.MGS42$logF
C.MGS4...MGS2 > 0, 'up', 'down'))

rownames(selected.genes.MGS42) <- matriz.conteo$GeneID[rows.MGS42]

nrow(selected.genes.MGS42)
```

```
rows.MGS43 <- res$table$FDR < 0.05 & abs(res$table$logFC.MGS4...MGS3)
> 1

selected.genes.MGS43 <- res$table[rows.MGS43,]

selected.genes.MGS43$updown <- factor(ifelse(selected.genes.MGS43$logF
C.MGS4...MGS3 > 0, 'up', 'down'))

rownames(selected.genes.MGS43) <- matriz.conteo$GeneID[rows.MGS43]

nrow(selected.genes.MGS43)
```

```
gene_list <- data.frame(GEO,row.names = GEO)

gene_list$Annotation <- Annotation

gene_list$Description <- Description

gene_list$Location <- Location

gene_list$logFC_MGS2 <- logFC2

gene_list$logFC_MGS3 <- logFC3

gene_list$logFC_MGS4 <- logFC4


save(gene_list, "gene_list.Rdata")
```

```
par(mfrow=c(1,3))
plot(res$table$logFC.MGS2...MGS1, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS2 vs MGS1')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS2$logFC.MGS2...MGS1,
       -log10(selected.genes.MGS2$FDR),
       pch=20, col='orange')
plot(res$table$logFC.MGS3...MGS1, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS3 vs MGS1')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS3$logFC.MGS3...MGS1,
       -log10(selected.genes.MGS3$FDR),
       pch=20, col='orange')

plot(res$table$logFC.MGS4...MGS1, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS4 vs MGS1')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS4$logFC.MGS4...MGS1,
       -log10(selected.genes.MGS4$FDR),
       pch=20, col='orange')
```

```
par(mfrow=c(1,3))

plot(res$table$logFC.MGS3...MGS2, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS3 vs MGS2')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS32$logFC.MGS3...MGS2,
       -log10(selected.genes.MGS32$FDR),
       pch=20, col='orange')

plot(res$table$logFC.MGS4...MGS2, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS4 vs MGS2')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS42$logFC.MGS4...MGS2,
       -log10(selected.genes.MGS42$FDR),
       pch=20, col='orange')

plot(res$table$logFC.MGS4...MGS3, -log10(res$table$FDR), pch=20,
     xlim = c(-6,6), ylim = c(0,11),
     xlab= 'log2 fold change', ylab = '-log10 p-valor',
     main='MGS4 vs MGS3')
abline(v=c(-1,1),lwd=2,col='navy')
abline(h=-log10(0.05),lwd=2,col='tomato')
points(selected.genes.MGS43$logFC.MGS4...MGS3,
       -log10(selected.genes.MGS43$FDR),
       pch=20, col='orange')
```

```
if(!('VennDiagram' %in% installed.packages()))
  install.packages('VennDiagram')


library('VennDiagram')


grid.newpage()
vd <- venn.diagram(x = list('MGS2 vs MGS1' = rownames(selected.genes.MGS2), 'MGS3 vs MGS1' = rownames(selected.genes.MGS3), 'MGS4 vs MGS1' = rownames(selected.genes.MGS4)), fill = RColorBrewer::brewer.pal(3,'Set2')[1:3], filename = NULL)
grid.draw(vd)


grid.newpage()
vd.MGS4 <- venn.diagram(x = list('MGS4 vs MGS1' = rownames(selected.genes.MGS4), 'MGS4 vs MGS2' = rownames(selected.genes.MGS42), 'MGS4 vs MGS3' = rownames(selected.genes.MGS43)), fill = RColorBrewer::brewer.pal(3,'Set2')[1:3], filename = NULL)
grid.draw(vd.MGS4)


grid.newpage()
vd.MGS32 <- venn.diagram(x = list('MGS2 vs MGS1' = rownames(selected.genes.MGS2), 'MGS3 vs MGS1' = rownames(selected.genes.MGS3), 'MGS3 vs MGS2' = rownames(selected.genes.MGS32)), fill = RColorBrewer::brewer.pal(3,'Set2')[1:3], filename = NULL)
grid.draw(vd.MGS32)
```

```
hc <- hclust(as.dist((1-cor(scaledata, method="spearman"))/2), method=
"complete") # Clusters columns by Spearman correlation.

hr <- hclust(as.dist((1-cor(t(scaledata), method="spearman"))/2), meth
od="complete")

tree = as.dendrogram(hr, method="average")

genes.group = cutree(hr, k=10)

table(genes.group)

clustColBar <- rainbow(length(unique(genes.group)), start=0.1, end=0.9
)

clustColBar <- clustColBar[as.vector(genes.group)]


gplots::heatmap.2(scaledata,

          Rowv=as.dendrogram(hr),

          Colv=as.dendrogram(hc),

          col=redgreen(100),

          scale="row",

          margins = c(7, 7),

          cexCol = 0.7,

          labRow = F,

          main = "Heatmap",

          trace = "none",

          RowSideColors=clustColBar,

          key = FALSE)
```

```
normalizacion.TMM <- data.frame(normalizacion.TMM, row.names = rowname
s(matriz.conteo))

normalizacion.TMM.DE <- normalizacion.TMM[listado.genes,]


G1 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'1'),]), id = rownames(normalizacion.TMM))

names(G1)[1:2] <- c('id', 'sample')

G1$group <- rep('Grupo 1', nrow(G1))

G1 <- MGS_level[which(genes.group=='1')]

aux <- rep(g1[1], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g1)) {

  aux <- rep(g1[v], 503)

  MGS <- append(MGS,aux)

}

G1$MGS <- MGS


G2 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'2'),]), id = rownames(normalizacion.TMM))

names(G2)[1:2] <- c('id', 'sample')

G2$group <- rep('Grupo 2', nrow(G2))

G2 <- MGS_level[which(genes.group=='2')]

aux <- rep(g2[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g2)) {

  aux <- rep(g2[v], 503)

  MGS <- append(MGS,aux)

}

G2$MGS <- MGS


G3 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'3'),]), id = rownames(normalizacion.TMM))

names(G3)[1:2] <- c('id', 'sample')

G3$group <- rep('Grupo 3', nrow(G3))

G3 <- MGS_level[which(genes.group=='3')]

aux <- rep(g3[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g3)) {
```

```
  aux <- rep(g3[v], 503)
  MGS <- append(MGS,aux)
}
G3$MGS <- MGS


G4 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'4'),]), id = rownames(normalizacion.TMM))
names(G4)[1:2] <- c('id', 'sample')
G4$group <- rep('Grupo 4', nrow(G4))
G4 <- MGS_level[which(genes.group=='4')]
aux <- rep(g4[2], 503)
MGS <- factor(aux, levels = levels(MGS_level))
for (v in 2:length(g4)) {
  aux <- rep(g4[v], 503)
  MGS <- append(MGS,aux)
}
G4$MGS <- MGS


G5 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'5'),]), id = rownames(normalizacion.TMM))
names(G5)[1:2] <- c('id', 'sample')
G5$group <- rep('Grupo 5', nrow(G5))
G5 <- MGS_level[which(genes.group=='5')]
aux <- rep(g5[2], 503)
MGS <- factor(aux, levels = levels(MGS_level))
for (v in 2:length(g5)) {
  aux <- rep(g5[v], 503)
  MGS <- append(MGS,aux)
}
G5$MGS <- MGS



G.all <- rbind(G1,G2,G3,G4,G5)
ggplot(G.all, aes(x = value, colour = MGS)) + geom_density(show.legend
= TRUE) + theme_bw() + facet_grid(. ~ group)+ ylab('Densidad') +  them
e(plot.title = element_text(hjust = 0.5)) + xlim(0,10)
```

```
G6 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'6'),]), id = rownames(normalizacion.TMM))

names(G6)[1:2] <- c('id', 'sample')

G6$group <- rep('Grupo 6', nrow(G6))

g6 <- MGS_level[which(genes.group=='6')]

aux <- rep(g6[1], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g6)) {

  aux <- rep(g6[v], 503)

  MGS <- append(MGS,aux)

}

G6$MGS <- MGS


G7 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'7'),]), id = rownames(normalizacion.TMM))

names(G7)[1:2] <- c('id', 'sample')

G7$group <- rep('Grupo 7', nrow(G7))

g7 <- MGS_level[which(genes.group=='7')]

aux <- rep(g7[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g7)) {

  aux <- rep(g7[v], 503)

  MGS <- append(MGS,aux)

}

G7$MGS <- MGS


G8 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'8'),]), id = rownames(normalizacion.TMM))

names(G8)[1:2] <- c('id', 'sample')

G8$group <- rep('Grupo 8', nrow(G8))

g8 <- MGS_level[which(genes.group=='8')]

aux <- rep(g8[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g8)) {

  aux <- rep(g8[v], 503)

  MGS <- append(MGS,aux)

}

G8$MGS <- MGS
```

```
G9 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group==
'9'),]), id = rownames(normalizacion.TMM))

names(G9)[1:2] <- c('id', 'sample')

G9$group <- rep('Grupo 9', nrow(G9))

g9 <- MGS_level[which(genes.group=='9')]

aux <- rep(g9[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g9)) {

  aux <- rep(g9[v], 503)

  MGS <- append(MGS,aux)

}

G9$MGS <- MGS


G10 <- reshape::melt(as.matrix(normalizacion.TMM.DE[which(genes.group=
='10'),]), id = rownames(normalizacion.TMM))

names(G10)[1:2] <- c('id', 'sample')

G10$group <- rep('Grupo 10', nrow(G10))

g10 <- MGS_level[which(genes.group=='10')]

aux <- rep(g10[2], 503)

MGS <- factor(aux, levels = levels(MGS_level))

for (v in 2:length(g10)) {

  aux <- rep(g10[v], 503)

  MGS <- append(MGS,aux)

}

G10$MGS <- MGS



G.all <- rbind(G6,G7,G8,G9,G10)

ggplot(G.all, aes(x = value, colour = MGS)) + geom_density(show.legend
= TRUE) + theme_bw() + facet_grid(. ~ group)+ ylab('Densidad') +  them
e(plot.title = element_text(hjust = 0.5)) + xlim(0,10)
```

```r
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("clusterProfiler")
BiocManager::install("org.Hs.eg.db") # humano
BiocManager::install("enrichplot")
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
require("DOSE")
# MGS2-MGS1
gene_list <- res$table$logFC.MGS2...MGS1
names(gene_list) <- rownames(selected.genes.MGS4)
gene_list <- sort(gene_list,decreasing = TRUE)

gse <- clusterProfiler::gseGO(geneList=gene_list,
                              ont = 'ALL',
                              keyType = "ENSEMBL",
                              nPerm = 10000,
                              minGSSize = 3,
                              maxGSSize = 800,
                              OrgDb = org.Hs.eg.db,
                              pvalueCutoff = 0.05,
                              pAdjustMethod = "none")
dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.sign)

# MGS4-MGS1
gene_list <- res$table$logFC.MGS4...MGS1
names(gene_list) <- rownames(selected.genes.MGS4)
gene_list <- sort(gene_list,decreasing = TRUE)

gse <- clusterProfiler::gseGO(geneList=gene_list,
                              ont = 'ALL',
                              keyType = "ENSEMBL",
                              nPerm = 10000,
                              minGSSize = 3,
                              maxGSSize = 800,
```

```
                                OrgDb = org.Hs.eg.db,
                                pvalueCutoff = 0.05,
                                pAdjustMethod = "none")
dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.sign)
x2 <- pairwise_termsim(gse)
emapplot(x2, showCategory=20)

# MGS3-MGS2
gene_list <- res$table$logFC.MGS3...MGS2
names(gene_list) <- rownames(selected.genes.MGS4)
gene_list <- sort(gene_list,decreasing = TRUE)

gse <- clusterProfiler::gseGO(geneList=gene_list,
                                ont = 'ALL',
                                keyType = "ENSEMBL",
                                nPerm = 10000,
                                minGSSize = 3,
                                maxGSSize = 800,
                                OrgDb = org.Hs.eg.db,
                                pvalueCutoff = 0.05,
                                pAdjustMethod = "none")
dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.sign)
x2 <- pairwise_termsim(gse)
emapplot(x2, showCategory=20)
# MGS4-MGS2
gene_list <- res$table$logFC.MGS4...MGS2
names(gene_list) <- rownames(selected.genes.MGS4)
gene_list <- sort(gene_list,decreasing = TRUE)

gse <- clusterProfiler::gseGO(geneList=gene_list,
                                ont = 'ALL',
                                keyType = "ENSEMBL",
                                nPerm = 10000,
                                minGSSize = 3,
                                maxGSSize = 800,
                                OrgDb = org.Hs.eg.db,
                                pvalueCutoff = 0.05,
```

```
                                pAdjustMethod = "none")

dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.sign)
x2 <- pairwise_termsim(gse)
emapplot(x2, showCategory=20)
# MGS4-MGS3
gene_list <- res$table$logFC.MGS4...MGS3
names(gene_list) <- rownames(selected.genes.MGS4)
gene_list <- sort(gene_list,decreasing = TRUE)

gse <- clusterProfiler::gseGO(geneList=gene_list,
                                ont = 'ALL',
                                keyType = "ENSEMBL",
                                nPerm = 10000,
                                minGSSize = 3,
                                maxGSSize = 800,
                                OrgDb = org.Hs.eg.db,
                                pvalueCutoff = 0.05,
                                pAdjustMethod = "none")

dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.sign)
x2 <- pairwise_termsim(gse)
emapplot(x2, showCategory=20)
```

```
selected.genes <- normalizacion.TMM
rownames(selected.genes) <- matriz.conteo$GeneID
matriz.DMAE <- data.frame(t(selected.genes))
matriz.DMAE$age <- age
matriz.DMAE$sex <- sex

set.seed(020708)
n <- ceiling(2*505/3)
train.DMAE.index <- sample(1:505,size=n,replace=FALSE)
train.DMAE.set <- matriz.DMAE[train.DMAE.index, ]
test.DMAE.set <- matriz.DMAE[-train.DMAE.index, ]

if(!('e1071' %in% installed.packages()))
  install.packages('e1071')
if(!('mRMRe' %in% installed.packages()))
  install.packages('mRMRe')
if(!('cvAUC' %in% installed.packages()))
  install.packages('cvAUC')
if(!('pROC' %in% installed.packages()))
  install.packages('pROC')
library(e1071)
library(mRMRe) # seleccion de características
library(cvAUC)
library(pROC)
```

```
#################################################################
########### Modelo MGS2 vs MGS3 ##############################
#################################################################
set.seed(1)
DMAE.names <- rows.MGS3 & rows.MGS2
DMAE.names <- DMAE.names & rows.MGS32
#length(which(DMAE.names==TRUE)) # Cojo los 7 genes comunes
df<-(train.DMAE.set[,DMAE.names])
gr<-as.factor(MGS_level[train.DMAE.index])
df$gr<-factor(1*(is.element(gr,c("2","3"))),levels=0:1,labels=c("Contr
ol","MGS"),ordered=TRUE)
## Seleccion del "mejor modelo" en cada caso
GENES<-PREDICHOS.LINEAL<-PREDICHOS.RADIAL<-PREDICHOS.POLI<-PREDICHOS.S
IG<-vector("list",ncol(df)-1)
AUC<-NULL
for(nn in 1:(ncol(df)-1)){


  PREDICHOS<-matrix(NA,nrow=nrow(df),ncol=4)
  for(i in 1:nrow(df)) {
    train<-df[-i,]
    test<-df[i,]


    ## seleccion de genes
    data <- mRMR.data(data = train)
    dataensemble <- mRMR.ensemble(data = data, target_indices = (ncol(
train)),
                                  solution_count = 1,
                                  feature_count = nn,method="exhaustive")
    genes<-colnames(df)[solutions(dataensemble)[[1]]]
    GENES[[nn]]<-cbind(GENES[[nn]],genes)


    ## svm lineal
    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="li
near",probability=TRUE)
    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]
    PREDICHOS[i,1]<-pred
```

```
    ## svm radial

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="ra
dial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,2]<-pred


    ## svm polinomial de grado 3

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="po
lynomial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,3]<-pred


    ## svm sigmoide

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="si
gmoid",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,4]<-pred

  }

  AUC<-rbind(AUC,sapply(1:ncol(PREDICHOS),function(j) cvAUC(PREDICHOS[
,j], df[,ncol(df)])$cvAUC))

}

rownames(AUC)<-genes

plot(abs(AUC[,1]-0.5),ylim=range(abs(AUC-0.5)),pch=20,xlab="",xaxt="n"
,main="Modelo: Intermedio vs Control")

lines(1:nrow(AUC),abs(AUC[,1]-0.5))

axis(1,at=1:nrow(AUC))

points(abs(AUC[,2]-0.5),col=2,pch=20)

lines(1:nrow(AUC),abs(AUC[,2]-0.5),col=2)

points(abs(AUC[,3]-0.5),col=3,pch=20)

lines(1:nrow(AUC),abs(AUC[,3]-0.5),col=3)

points(abs(AUC[,4]-0.5),col=4,pch=20)

lines(1:nrow(AUC),abs(AUC[,4]-0.5),col=4)

legend("bottomright",pch=20,col=1:4,c("Lineal","Radial","Polinomial","
Sigmoide"))
```

```
#################################################################
########### Modelo Avanzado ###############################
#################################################################
set.seed(1)
DMAE.names <- rows.MGS4 & rows.MGS42
DMAE.names <- DMAE.names & rows.MGS43
#length(which(DMAE.names==TRUE)) # Cojo los 26 genes comunes
df<-(train.DMAE.set[,DMAE.names])
gr<-as.factor(MGS_level[train.DMAE.index])
df$gr<-factor(1*(is.element(gr,c("4"))),levels=0:1,labels=c("Control",
"MGS"),ordered=TRUE)
## Seleccion del "mejor modelo" en cada caso
GENES<-PREDICHOS.LINEAL<-PREDICHOS.RADIAL<-PREDICHOS.POLI<-PREDICHOS.S
IG<-vector("list",ncol(df)-1)
AUC<-NULL
for(nn in 1:(ncol(df)-1)){


  PREDICHOS<-matrix(NA,nrow=nrow(df),ncol=4)
  for(i in 1:nrow(df)) {
    train<-df[-i,]
    test<-df[i,]


    ## seleccion de genes
    data <- mRMR.data(data = train)
    dataensemble <- mRMR.ensemble(data = data, target_indices = (ncol(
train)),
                                  solution_count = 1,
                                  feature_count = nn,method="exhaustive")
    genes<-colnames(df)[solutions(dataensemble)[[1]]]
    GENES[[nn]]<-cbind(GENES[[nn]],genes)


    ## svm lineal
    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="li
near",probability=TRUE)
    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]
    PREDICHOS[i,1]<-pred
```

```
    ## svm radial

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="ra
dial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,2]<-pred


    ## svm polinomial de grado 3

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="po
lynomial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,3]<-pred


    ## svm sigmoide

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="si
gmoid",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,4]<-pred


  }

  AUC<-rbind(AUC,sapply(1:ncol(PREDICHOS),function(j) cvAUC(PREDICHOS[
,j], df[,ncol(df)])$cvAUC))

}

rownames(AUC)<-genes

plot(abs(AUC[,1]-0.5),ylim=range(abs(AUC-0.5)),pch=20,xlab="",xaxt="n"
,main="Modelo: Avanzado vs Control")

lines(1:nrow(AUC),abs(AUC[,1]-0.5))

axis(1,at=1:nrow(AUC))

points(abs(AUC[,2]-0.5),col=2,pch=20)

lines(1:nrow(AUC),abs(AUC[,2]-0.5),col=2)

points(abs(AUC[,3]-0.5),col=3,pch=20)

lines(1:nrow(AUC),abs(AUC[,3]-0.5),col=3)

points(abs(AUC[,4]-0.5),col=4,pch=20)

lines(1:nrow(AUC),abs(AUC[,4]-0.5),col=4)

legend("bottomright",pch=20,col=1:4,c("Lineal","Radial","Polinomial","
Sigmoide"))
```

```
#################################################################

########### Modelo DMAE ##############################

#################################################################

set.seed(1)

DMAE.names <- rows.MGS2 & rows.MGS3

DMAE.names <- DMAE.names & rows.MGS4

#length(which(DMAE.names==TRUE)) # Cojo los 102 genes comunes

df<-(train.DMAE.set[,DMAE.names])

gr<-as.factor(MGS_level[train.DMAE.index])

df$gr<-factor(1*(is.element(gr,c("2","3","4"))),levels=0:1,labels=c("C
ontrol","DMAE"),ordered=TRUE)

## Seleccion del "mejor modelo" en cada caso

GENES<-PREDICHOS.LINEAL<-PREDICHOS.RADIAL<-PREDICHOS.POLI<-PREDICHOS.S
IG<-vector("list",ncol(df)-1)

AUC<-NULL

for(nn in 1:(ncol(df)-1)){


  PREDICHOS<-matrix(NA,nrow=nrow(df),ncol=4)

  for(i in 1:nrow(df)) {

    train<-df[-i,]

    test<-df[i,]


    ## seleccion de genes

    data <- mRMR.data(data = train)

    dataensemble <- mRMR.ensemble(data = data, target_indices = (ncol(
train)),

                                  solution_count = 1,

                                  feature_count = nn,method="exhaustive")

    genes<-colnames(df)[solutions(dataensemble)[[1]]]

    GENES[[nn]]<-cbind(GENES[[nn]],genes)


    ## svm lineal

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="li
near",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,1]<-pred
```

```
    ## svm radial

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="ra
dial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,2]<-pred


    ## svm polinomial de grado 3

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="po
lynomial",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,3]<-pred


    ## svm sigmoide

    model <- svm(gr~.,data=train[,c("gr",genes),drop=FALSE],kernel="si
gmoid",probability=TRUE)

    pred<-attr(predict(model,test[,,drop=FALSE],probability=TRUE),"pro
babilities")[,levels(train$gr)[2]]

    PREDICHOS[i,4]<-pred


  }

  AUC<-rbind(AUC,sapply(1:ncol(PREDICHOS),function(j) cvAUC(PREDICHOS[
,j], df[,ncol(df)])$cvAUC))


}

rownames(AUC)<-genes

plot(abs(AUC[,1]-0.5),ylim=range(abs(AUC-0.5)),pch=20,xlab="",xaxt="n"
,main="Modelo: DMAE vs Control")

lines(1:nrow(AUC),abs(AUC[,1]-0.5))

axis(1,at=1:nrow(AUC))

points(abs(AUC[,2]-0.5),col=2,pch=20)

lines(1:nrow(AUC),abs(AUC[,2]-0.5),col=2)

points(abs(AUC[,3]-0.5),col=3,pch=20)

lines(1:nrow(AUC),abs(AUC[,3]-0.5),col=3)

points(abs(AUC[,4]-0.5),col=4,pch=20)

lines(1:nrow(AUC),abs(AUC[,4]-0.5),col=4)

legend("bottomright",pch=20,col=1:4,c("Lineal","Radial","Polinomial","
Sigmoide"))
```

```r
if(!('e1071' %in% installed.packages()))
  install.packages('e1071')
if(!('caret' %in% installed.packages()))
  install.packages('caret')

library('e1071')
library('caret')
```

```r
#####################################################
####### MODELO DMAE
#####################################################

# Kernel polinomial de grado 3 con 7 genes
set.seed(400)
genes_names <-  c("ENSG00000004846", "ENSG00000005001", "ENSG00000005102", "ENSG00000002079", "ENSG00000007312", "ENSG00000006042", "ENSG00000005471")
matriz.DMAE <- (train.DMAE.set[,genes_names])
matriz.DMAE$age <- age[train.DMAE.index]
matriz.DMAE$sex <- sex[train.DMAE.index]
gr<-as.factor(MGS_level[train.DMAE.index])
matriz.DMAE$MGS <- factor(1*(is.element(gr,c("1"))),levels=0:1,labels=c("DMAE","Control"),ordered=TRUE)

matriz.test <- (test.DMAE.set[,genes_names])
matriz.test$age <- age[-train.DMAE.index]
matriz.test$sex <- sex[-train.DMAE.index]
grt<-as.factor(MGS_level[-train.DMAE.index])
matriz.test$MGS <- factor(1*(is.element(grt,c("1"))),levels=0:1,labels=c("DMAE","Control"),ordered=TRUE)
model.MGS <- svm(MGS ~ ., data = matriz.DMAE,
                 probability = TRUE,
                 kernel = "polynomial",
                 )

test.MGS.predictions <- predict(model.MGS, matriz.DMAE, type='class')
#caret::confusionMatrix(test.MGS.predictions, as.factor(matriz.DMAE$MGS))
```

```
pred<-attr(predict(model.MGS,matriz.DMAE,probability=TRUE),"probabilit
ies")[,levels(matriz.DMAE$MGS)[2]]
cvAUC(pred, matriz.DMAE$MGS)$cvAUC

PRED<-data.frame(pred,matriz.DMAE$MGS)
library(rpart)
tree <- rpart(matriz.DMAE$MGS ~ ., data = PRED)
summary(tree)

DMAE <- pred < 0.2584702 #  to the left DMAE

CLAS.FINAL<-rep("Control",nrow(PRED))
CLAS.FINAL[DMAE]<-"DMAE"

# filas predicho / real columnas
table(factor(CLAS.FINAL,levels=levels(PRED[,2])),PRED[,2])
PRED<-data.frame(pred,matriz.DMAE$MGS)
library(rpart)
tree <- rpart(matriz.DMAE$MGS ~ ., data = PRED)
summary(tree)

DMAE <- pred < 0.2584963 #  to the left DMAE

CLAS.FINAL<-rep("Control",nrow(PRED))
CLAS.FINAL[DMAE]<-"DMAE"

# filas predicho / real columnas
table(factor(CLAS.FINAL,levels=levels(PRED[,2])),PRED[,2])

PRED<-data.frame(pred,matriz.DMAE$MGS)
library(rpart)
tree <- rpart(matriz.DMAE$MGS ~ ., data = PRED)
summary(tree)
```

```
## MODELOS DEFINITIVOS ##

### VALIDACIÓN EXTERNA ###


# 1. Kernel polinómico de grado 3 con 6 genes

set.seed(400)

genes_names <- c("ENSG00000004846", "ENSG00000005001", "ENSG0000000510
2", "ENSG00000002079", "ENSG00000007312", "ENSG00000006042")

matriz.DMAE <- (train.DMAE.set[,genes_names])

matriz.DMAE$age <- age[train.DMAE.index]

matriz.DMAE$sex <- sex[train.DMAE.index]

gr<-as.factor(MGS_level[train.DMAE.index])


matriz.DMAE$MGS <- factor(1*(is.element(gr,c("1"))),levels=0:1,labels=
c("DMAE","Control"),ordered=TRUE)

matriz.test <- (test.DMAE.set[,genes_names])

matriz.test$age <- age[-train.DMAE.index]

matriz.test$sex <- sex[-train.DMAE.index]

grt<-as.factor(MGS_level[-train.DMAE.index])

matriz.test$MGS <- factor(1*(is.element(grt,c("1"))),levels=0:1,labels
=c("DMAE","Control"),ordered=TRUE)

model.MGS <- svm(MGS ~ ., data = matriz.DMAE,

                  probability = TRUE,

                  kernel = "polynomial",

                  )


test.MGS.predictions <- predict(model.MGS, matriz.test, type='class')

#caret::confusionMatrix(test.MGS.predictions, as.factor(matriz.test$MG
S))


pred.MGS<-attr(predict(model.MGS,matriz.test,probability=TRUE),"probab
ilities")[,2]

cvAUC(pred.MGS, matriz.test$MGS)$cvAUC
```

```
PRED<-data.frame(pred.MGS,matriz.test$MGS)
library(rpart)
tree <- rpart(matriz.test$MGS ~ ., data = PRED)
#summary(tree)

DMAE <- pred.MGS < 0.2584 #  to the left DMAE

CLAS.FINAL<-rep("Control",nrow(PRED))
CLAS.FINAL[DMAE]<-"DMAE"

# filas predicho / real columnas
table(factor(CLAS.FINAL,levels=levels(PRED[,2])),PRED[,2])
save(model.MGS, file = "model_MGS.Rdata")
```

```
# 2. Kernel polinómico de grado 3 con 3 genes
set.seed(400)
genes_names <-  c("ENSG00000003147", "ENSG00000005379", "ENSG000000059
81")
matriz.DMAE <- (train.DMAE.set[,genes_names])
matriz.DMAE$age <- age[train.DMAE.index]
matriz.DMAE$sex <- sex[train.DMAE.index]
gr<-as.factor(MGS_level[train.DMAE.index])
matriz.DMAE$MGS <- factor(1*(is.element(gr,c("1","4"))),levels=0:1,lab
els=c("MGS3","Control"),ordered=TRUE)

matriz.test <- (test.DMAE.set[,genes_names])
matriz.test$age <- age[-train.DMAE.index]
matriz.test$sex <- sex[-train.DMAE.index]
grt<-as.factor(MGS_level[-train.DMAE.index])
matriz.test$MGS <- factor(1*(is.element(grt,c("1","4"))),levels=0:1,la
bels=c("MGS3","Control"),ordered=TRUE)
model.Intermedio <- svm(MGS ~ ., data = matriz.DMAE,
                  probability = TRUE,
                  kernel = "polynomial",
                  )
```

```
test.MGS.predictions <- predict(model.Intermedio, matriz.test, type='class')

#caret::confusionMatrix(test.MGS.predictions, as.factor(matriz.test$MGS))


pred.inter<-attr(predict(model.Intermedio,matriz.test,probability=TRUE),"probabilities")[,1]

cvAUC(pred.inter, matriz.test$MGS)$cvAUC

PRED<-data.frame(pred.inter,matriz.test$MGS)

library(rpart)

tree <- rpart(matriz.test$MGS ~ ., data = PRED)

#summary(tree)


DMAE <- pred.inter < 0.3854839 #  to the left


CLAS.FINAL<-rep("Control",nrow(PRED))

CLAS.FINAL[DMAE]<-"MGS3"

table(factor(CLAS.FINAL,levels=levels(PRED[,2])),PRED[,2])

save(model.Intermedio, file = "model_Intermedio.Rdata")
```

```r
# 3. Kernel polinómico con 6 genes

set.seed(400)

genes_names <-  c("ENSG00000000003","ENSG00000005421","ENSG00000005471
","ENSG00000006606","ENSG00000007908","ENSG00000008516")

matriz.DMAE <- (train.DMAE.set[,genes_names])

matriz.DMAE$age <- age[train.DMAE.index]

matriz.DMAE$sex <- sex[train.DMAE.index]

gr<-as.factor(MGS_level[train.DMAE.index])

matriz.DMAE$MGS <- factor(1*(is.element(gr,c("1","2","3"))),levels=0:1
,labels=c("MGS4","Control"),ordered=TRUE)


matriz.test <- (test.DMAE.set[,genes_names])

matriz.test$age <- age[-train.DMAE.index]

matriz.test$sex <- sex[-train.DMAE.index]

grt<-as.factor(MGS_level[-train.DMAE.index])

matriz.test$MGS <- factor(1*(is.element(grt,c("1","2","3"))),levels=0:
1,labels=c("MGS4","Control"),ordered=TRUE)

model.Avanzado <- svm(MGS ~ ., data = matriz.DMAE,

                  probability = TRUE,

                  kernel = "polynomial",

                  )


test.MGS.predictions <- predict(model.Avanzado, matriz.test, type='cla
ss')

#caret::confusionMatrix(test.MGS.predictions, as.factor(matriz.test$MG
S))


pred.avan<-attr(predict(model.Avanzado,matriz.test,probability=TRUE),"
probabilities")[,1]

cvAUC(pred.avan, matriz.test$MGS)$cvAUC

PRED<-data.frame(pred.avan,matriz.test$MGS)

library(rpart)

tree <- rpart(matriz.test$MGS ~ ., data = PRED)

#summary(tree)


DMAE <- pred.avan < 0.8724653


CLAS.FINAL<-rep("Control",nrow(PRED))

CLAS.FINAL[DMAE]<-"MGS4"
```

```
# filas predicho / real columnas
table(factor(CLAS.FINAL,levels=levels(PRED[,2])),PRED[,2])
save(model.Avanzado, file = "model_Avanzado.Rdata")
```

```
## Clasificacion Final
gr <- MGS_level[-train.DMAE.index]
gr <- replace(gr, which(gr=='3'),'2')
gr<-factor(gr,levels=c(1,2,4),labels=c("Sano","Intermedio","Avanzado")
)
PRED<-data.frame(pred.MGS,pred.inter,pred.avan,gr)
```

```
## Shiny: ui
library(shiny)
library(shinydashboard)
library(shinythemes)
library(DT)

load("~/Desktop/TFG/Códigos/gene_list.Rdata")

dashboardPage(
  skin = "blue",

  dashboardHeader(
    title = tags$em("Predicción DMAE", style = "text-align:justify;col
or:#ffffff;font-size:100%"),
    titleWidth = 250
  ),

  dashboardSidebar(
    width = 250,

    sidebarMenu(
      menuItem(
        tags$em("Carga RNA-seq", style = "font-size:120%"),
        icon = icon("upload"),
        tabName = "data"
      ),
      fileInput(
        'file1',
        em('Suba sus datos en formato csv ', style = "text-align:cente
r;color:#ffffff;font-size:100%;bold"),
        multiple = FALSE,
        accept = c('.csv')
      ),
      hr(),
      menuItem(
        tags$em("Visualización de Resultados", style = "font-size:120%
"),
        icon = icon("download"),
```

```
        tabName = 'MGS'
      ),
      br(),
      column(width = 12,
             downloadButton(
               "downloadData",
               em('Desargar Resultados', style = "text-align:center;co
lor:black;color:black;font-size:120%;bold")
             ),)
    )
  ),


  dashboardBody(tabItems(
    tabItem(
      tabName = "data",


      fluidPage(
        br(),


        tags$h5(
          "Mediante esta herramienta puede subir sus muestras de RNA-s
eq y obtener una estimación de que estas
                           pertenezcan a pacientes de Degeneración Macu
lar Asociada a la Edad (DMAE). El modelo implementado está basado
                           en un algoritmo de Support Vector Machine (S
VM) que fue entrenado con los 15 genes diferencialmente expresados
                           en la DMAE que se indican abajo así como con
el género y edad del paciente. En caso de predecir un resultado
                           positivo, también realiza una estimación del
grado de severidad de la enfermedad basada en el sistema del",
          tags$span("Minnesota Grading System (MGS)", style =
                      "color:#2596be;bold"),
          tags$span(
            ". Para tener más información sobre
                           el MGS pulse en el siguiente botón.",
            style = "text-align:justify"
          ), style = "text-align:justify"
        ),
```

```
        fluidRow(column(
          width = 12,align="center",
          actionButton("show", "Saber más sobre el Minnesota Grading System", icon("eye"))
        )),

        tags$h5(
          "Por favor, suba sus datos de RNA-seq en formato csv (muestras por filas y genes por columnas).
                                    Después, seleccione la opción",
          tags$span("'Descarga de Resultados'", style = "color:#2596be;bold"),
          tags$span("de la barra lateral para descargar sus prediccions en formato csv."),
          tags$span(" Tenga en cuenta que este modelo solo requiere los datos de expresión de los siguientes 15 genes*:"),style = "text-align:justify"),
        br(),

        datatable(gene_list[, 2:7]),

        tags$h5(
          "*La herramienta se encargará de seleccionar solamente estos genes de interés en caso de que se suba
                                    un archivo con un estudio más completo de RNA-seq.",
          style = "font-size:100%"
        ),
      ),

      fluidRow(column(
        width = 12,
        uiOutput("sample_input_data_heading"),
      )),

    ),

    tabItem(
      tabName = "MGS",
```

```
      fluidPage(

        fluidRow(column(width = 6,
          plotOutput("plot3d")),
          column(width = 6,
          plotOutput("plotMGS_inter"))),
        br(),
        fluidRow(column(width = 6,
          plotOutput("plotMGS_avan")),
          column(width = 6,
          plotOutput("plotinter_avan"))),
        br(),
        fluidRow(column(
          width = 12,
          dataTableOutput("sample_predictions"),
        )),

      )
    )
  ))
)
```