



UNIVERSIDAD
DE GRANADA

Facultad de Ciencias

E.T.S. Ingenierías Informática y de Telecomunicación

DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y
MATEMÁTICAS

TRABAJO DE FIN DE GRADO

Metodologías Multivariantes para la Identificación de Patrones Biológicos

Presentado por:

Quintín Mesa Romero

Curso académico 20242025

Metodologías Multivariantes para la Identificación de Patrones Biológicos

Quintín Mesa Romero

Quintín Mesa Romero *Metodologías Multivariantes para la Identificación de Patrones Biológicos.*

Trabajo de fin de Grado. Curso académico 20242025.

**Responsable de
tutorización**

José Luis Romero Béjar
*Departamento de Estadística e
Investigación Operativa*

Doble Grado en
Ingeniería Informática y
Matemáticas

Facultad de Ciencias
E.T.S. Ingenierías
Informática y de
Telecomunicación

Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D./Dña. Quintín Mesa Romero

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 20242025, es original, entendido esto en el sentido de que no he utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a May 30, 2025

Fdo: Quintín Mesa Romero

1 Introducción

La biología, como disciplina científica, ha experimentado una evolución notable en las últimas décadas, pasando de enfoques cualitativos y descriptivos a un análisis más detallado y cuantitativo de los organismos vivos. Este cambio de paradigma se produjo a mediados del siglo XX con la llegada de la *biología molecular*, tras casi dos siglos de preeminencia del naturalismo basado en la observación y la contemplación. Este avance marcó el inicio de una nueva era en la que el desarrollo de ciertas herramientas tecnológicas permitió analizar los diversos y complejos niveles de organización de los organismos, generando grandes volúmenes de datos en periodos relativamente cortos: la *era de las ciencias ómicas* [1, 2].

En este contexto, las ciencias ómicas surgieron como un marco integrador que engloba el conocimiento derivado de la aplicación de tecnologías avanzadas para el estudio a nivel molecular de los distintos elementos que conforman los sistemas biológicos, como células, tejidos e individuos. Estas disciplinas no solo permiten analizar la complejidad interna de los organismos, sino también comprender las interacciones dinámicas entre sus componentes internos y los factores externos con los que estos interactúan. Ofrecen, en definitiva, una perspectiva holística del individuo, proporcionando una visión detallada del funcionamiento de sus células y de la influencia del entorno que las rodea [2].

El término *ómica* fue acuñado en la década de 1980 para referirse al estudio de *conjuntos de moléculas* específicas, como genes (genómica), transcripciones de ARN (transcriptómica), proteínas (proteómica) o metabolitos (metabolómica), entre otros. Estas disciplinas han evolucionado significativamente gracias a los avances tecnológicos que permiten abordar la complejidad inherente de los sistemas biológicos analizados. De hecho, este es el máximo distintivo de las ciencias ómicas: el uso de las llamadas *tecnologías ómicas*, herramientas de alto rendimiento diseñadas para generar grandes cantidades de datos en un solo experimento a partir de una única muestra. Este enfoque masivo en la obtención de datos, conocido como *Big Data*, ha transformado profundamente el análisis biológico, permitiendo explorar dinámicas moleculares con un gran nivel de detalle [2][3].

La integración de las ciencias ómicas con metodologías avanzadas de análisis, como las técnicas multivariantes y el aprendizaje automático, ha marcado un hito en la investigación biomédica, abriendo nuevas fronteras en la comprensión de los complejos sistemas biológicos. Estas metodologías, que permiten gestionar y analizar grandes volúmenes de datos con múltiples dimensiones, son fundamentales para descubrir patrones biológicos subyacentes que, de otro modo, podrían pasar desapercibidos

utilizando métodos tradicionales [4]. Técnicas multivariantes, como el análisis de componentes principales (PCA), el análisis clúster, el análisis factorial o el análisis discriminante, facilitan la identificación de relaciones y la reducción de la dimensionalidad en los datos, lo que es crucial para poder extraer información relevante de los voluminosos conjuntos de datos generados [5].

A medida que los volúmenes de datos generados por las tecnologías ómicas se incrementan, la *bioinformática* se ha consolidado como una disciplina esencial para procesar, gestionar y analizar dichos datos [6]. Facilita la identificación y visualización de patrones biológicos complejos a partir de grandes bases de datos, mediante el uso de algoritmos avanzados, herramientas computacionales y modelos estadísticos. Este enfoque es fundamental para descubrir asociaciones moleculares, determinar biomarcadores relevantes y comprender las bases genéticas de enfermedades [7]. En este sentido, las herramientas bioinformáticas, como los lenguajes de programación R y Python, entre otros, junto con plataformas especializadas como Bioconductor, permiten realizar análisis profundos de datos ómicos a gran escala, proporcionando los recursos necesarios para un manejo efectivo y preciso de la información biológica [8].

Además, la combinación de las ciencias ómicas con estas metodologías avanzadas ha mejorado significativamente nuestra comprensión de los procesos biológicos y ha facilitado el desarrollo de estrategias diagnósticas y terapéuticas innovadoras. En particular, las técnicas multivariantes y el aprendizaje automático han demostrado ser esenciales para la identificación de patrones biológicos en diversas enfermedades, desde el cáncer hasta trastornos neurodegenerativos, así como para predecir la respuesta a distintos tratamientos. Esta integración ha impulsado el avance hacia la medicina personalizada y de precisión, en la que los tratamientos se ajustan a las características individuales de cada paciente, haciéndolos mucho más eficientes y reduciendo efectos adversos.

En el presente trabajo, se explorará el uso de la transcriptómica, como ciencia ómica y las metodologías avanzadas de análisis de datos, como las técnicas multivariantes y el aprendizaje automático, para la identificación y clasificación de ciertos patrones biológicos. Se realizará una revisión teórica de las técnicas multivariantes más comunes, anteriormente mencionadas, aunque nos centraremos en una de ellas con el fin de proporcionar una base sólida para su aplicación práctica en datos ómicos. Posteriormente, se llevará a cabo una implementación realista y funcional para el análisis de datos biológicos, aplicando técnicas de aprendizaje automático para la identificación de patrones biológicos significativos. A través de estas metodologías avanzadas, se intentará simplificar los datos ómicos para poder extraer la información clave que permita clasificar y entender mejor los patrones biológicos, mejorando así la precisión de los modelos predictivos.

Parte I

DATOS ÓMICOS

2 Datos ómicos

A la información cuantitativa y cualitativa obtenida a partir de las tecnologías utilizadas en las distintas ciencias ómicas, se le denomina *datos ómicos*. Estos datos abarcan información genética (genómica), de expresión génica (transcriptómica), de proteínas (proteómica), metabolitos (metabolómica) y otras áreas emergentes dentro de las ciencias ómicas.

Una de sus características más relevantes es su *alta dimensionalidad*, lo que genera conjuntos de datos masivos y complejos. Esta naturaleza multidimensional y heterogénea de los datos ómicos presenta desafíos significativos en su procesamiento, análisis e interpretación.

En este capítulo, se presenta la estructura de los datos ómicos, destacando su naturaleza matricial en la que el número de características (variables) suele superar ampliamente el número de muestras. Se analiza el desafío estadístico que representan los datos de alta dimensión y se introduce el problema de la expresión diferencial. Además, se aborda la transcriptómica y las tecnologías de secuenciación de ARN (RNA-seq y scRNA-seq), describiendo la estructura de los datos que generan y los retos asociados a su manejo, dado su gran volumen y alta dimensionalidad.

La información presentada en las próximas dos secciones ha sido extraída de la fuente bibliográfica [9].

2.1 Estructura de los datos

Los datos con los que trabajaremos se caracterizan por tener una estructura parecida. Analizaremos un conjunto con pocas muestras frente al gran número de características que observaremos sobre ellas. Apreciamos aquí el carácter de alta dimensionalidad de los datos ómicos.

Las características que analizamos pueden ser de diferentes tipos, como el nivel de fluorescencia, en el caso de que estemos trabajando con microarrays¹, como los de ADN, metilación o proteínas, o el número de lecturas alineadas obtenidas en procedimientos de secuenciación. Estas características, pueden estar asociadas a un elemento de análisis o a un conjunto de muestras en un microarray. O bien, la información puede corresponder a un gen, un exón², una proteína o una región específica del genoma.

¹Microarray: La tecnología de microarrays permite estudiar la expresión de múltiples genes simultáneamente. Consiste en fijar miles de secuencias génicas en un chip de vidrio. Al exponer una muestra de ADN o ARN, el apareamiento de bases complementarias genera una señal luminosa medible, indicando los genes expresados en la muestra [10].

²Exón: un exón es una región del genoma que termina dentro de una molécula de ARN mensajero. Algunos exones son codificantes, es decir, contienen información para fabricar una proteína, mientras

Denotaremos por N al número de características observadas, que será un valor relativamente grande, del orden de miles. Como hemos mencionado anteriormente, estas características se observan sobre un conjunto reducido de individuos, del orden de las decenas, en el mejor de los casos. Sea entonces n el número de muestras sobre las que serán observadas las variables (características).

Por consiguiente, el problema se enmarca dentro del campo de la estadística de alta dimensión. Esta situación, donde N supera a n , contrasta con lo que se observa en los enfoques estadísticos convencionales, en los cuales suele ocurrir todo lo contrario: el número de muestras es mayor que el de variables. Aunque esta desigualdad presenta limitaciones, también abre un nuevo campo de investigación con retos que los métodos tradicionales no pueden resolver, lo que motiva el desarrollo de nuevos procedimientos que se explorarán más adelante.

Las características las almacenaremos en una matriz, que llamaremos *matriz de expresión*, dada por:

$$X = [x_{ij}]_{i=1,\dots,N,j=1,\dots,n} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}$$

donde x_{ij} cuantifica la característica i en la muestra j .

Nota. Observemos que en un contexto estadístico convencional, la matriz de datos sería la matriz transpuesta de la que vamos a estar utilizando.

En el supuesto de que x_{ij} esté asociado con un microarray de ADN, entonces, mide un nivel de fluorescencia, tomando valores positivos, aunque pudiera ser que, tras el procesamiento de los datos, se diera lugar a expresiones negativas. Por su parte, si se tratase de un dato obtenido mediante la técnica de secuenciación RNA-seq, que será introducida a continuación, tendríamos conteos; número de lecturas cortas que se alinean sobre un gen, exón o una zona genómica concretos. Un mayor número de lecturas será indicativo de una mayor expresión de dicha característica.

Los valores observados de una característica sobre el conjunto de todas las muestras (una fila de la matriz de expresión) son, en el ámbito de la transcriptómica, lo que se conoce como *perfil*, o de forma más general, perfil de expresión.

En la matriz X los valores correspondientes a las diferentes muestras son indepen-

que otros son no codificantes. Los genes del genoma están formados por exones e intrones, que son trozos muy grandes de ARN dentro de una molécula de ARN mensajero que interfieren con el código de los exones. Estos intrones se eliminan de la molécula de ARN para dejar una serie de exones unidos entre sí de manera que se puedan codificar los aminoácidos correctos [11][12].

dientes entre sí, aunque pueden haber sido obtenidos bajo condiciones distintas. Por lo tanto, no se trata de réplicas de una misma condición experimental, sino de observaciones independientes. Es decir, presentan independencia condicional. Sin embargo, las filas de X representan vectores que sí están relacionados. Por ejemplo, en una matriz de expresión génica⁵, los valores de expresión de las filas no son independientes, debido a que los genes tienden a actuar de manera coordinada.

Por lo general, los datos en las columnas de la matriz X , no pueden compararse directamente entre sí, por la presencia de diversos artefactos técnicos y ruido en la medición de la característica de interés. Es por ello que se han desarrollado técnicas para corregir estos problemas, denominadas como *técnicas de preprocesado*. Al aplicar estos métodos, los datos dejan de ser completamente independientes. No obstante, en la mayoría de los estudios este aspecto no se suele tener en cuenta. Tras la normalización, los datos siguen considerandose independientes por columnas (muestras) y dependientes por filas.

A la información o variables que describen y caracterizan a las muestras, las llamaremos *metadatos* o *variables fenotípicas*. En este contexto, el uso de este término es adecuado porque estas variables reflejan atributos medibles y observables de las muestras, lo que se conoce en el ámbito de la biología como *fenotipo* [14]. Normalmente tendremos varias variables fenotípicas. Llamaremos $y = (y_1, \dots, y_n)$ a los valores observados de una variable en las n muestras. Uno de los casos más típicos de variable fenotípica es cuando se tienen dos grupos de muestras: casos (individuos que tienen la enfermedad) y controles (no tienen la enfermedad o condición de interés). En este caso tendríamos $y_i = 1$, para un caso e $y_i = 0$, si es control. Si tuviéramos la situación en la que hay k grupos a comparar, con $k > 2$, entonces se utiliza $y_i \in \{1, \dots, k\}$ con $i = 1, \dots, n$. Hemos de recalcar que los valores y_k son arbitrarios y pueden tomar cualquier otro par de valores.

2.2 Problema Estadístico

Normalmente, las técnicas estadísticas utilizadas en muchos campos se basan en contextos en los que el número de muestras, n , es mayor que el de variables, N . Sin embargo, en el caso de los datos ómicos, esta relación se invierte, lo que obliga a ajustar estos procedimientos de manera que, en algunos casos, la adaptación resulta más o menos exitosa. En otras palabras, la falta de suficientes muestras para la cantidad de variables presentes, hace que sea extremadamente difícil encontrar un modelo que pueda capturar de manera precisa la relación entre las variables predictores y la

⁵Expresión génica: La expresión génica es el proceso por el cual la información codificada por un gen se usa para producir moléculas de ARN que codifican para proteínas o para producir moléculas de ARN no codificantes que cumplen otras funciones. La expresión génica actúa como un “interruptor” que controla cuándo y dónde se producen moléculas de ARN y proteínas y como un “control de volumen” para determinar qué cantidad de esos materiales se produce [13].

variable respuesta. Esto se debe a que no tenemos una cantidad adecuada de datos para entrenar de manera efectiva un modelo estadístico que pueda generalizarse de manera fiable a nuevas observaciones.

La dificultad de analizar datos de alta dimensionalidad resulta además de la conjunción de dos efectos.

En primer lugar, los espacios de alta dimensión tienen propiedades geométricas que son contra-intuitivas y alejadas de las propiedades que se pueden observar en espacios bidimensionales o tridimensionales.

En segundo lugar, las herramientas de análisis de datos suelen diseñarse teniendo en cuenta propiedades intuitivas y ejemplos en espacios de baja dimensión; por lo general, las herramientas de análisis de datos se ilustran mejor en espacios de dos y tres dimensiones, por razones obvias. El problema es que esas herramientas también se utilizan cuando los datos son de alta dimensión y más complejos. En este tipo de situaciones, perdemos la intuición del comportamiento de las herramientas y podemos sacar conclusiones erróneas sobre sus resultados, dificultando la construcción de modelos estadísticos precisos.

Por todo ello, en este contexto, las técnicas estadísticas utilizadas son mera aplicación de procedimientos diseñados para la situación antes comentada en la que el número de muestras es mayor que el de variables.

Uno de los principales retos que se abordarán es el análisis de expresión diferencial, que examina cómo varían los niveles de expresión génica entre distintas condiciones experimentales o grupos de individuos. En particular, se busca determinar si existe una relación entre el perfil de expresión génica y una variable fenotípica específica. Este enfoque, denominado análisis de expresión diferencial marginal, permite explorar asociaciones entre conjuntos de genes, organizados como grupos de filas dentro de la matriz de expresión, y la característica fenotípica de interés. A este tipo de análisis se le conoce también como análisis de conjuntos de genes o *gene set analysis*, y su objetivo es identificar patrones de expresión que puedan estar vinculados a determinados rasgos biológicos.

2.3 Transcriptómica: datos RNA-seq y single-cell RNA-seq

Entre las distintas tecnologías utilizadas en la generación de datos ómicos, la *transcriptómica* desempeña un papel fundamental en el análisis de la expresión génica, y en particular, las tecnologías de *RNA-seq* y *single-cell RNA-seq* han revolucionado este campo al permitir la cuantificación precisa de los niveles de ARN mensajero en diferentes condiciones biológicas. Estas técnicas producen datos con una estructura matricial compleja, caracterizada por un alto número de variables (genes) frente a un número reducido de muestras (individuos o células). Esta estructura plantea desafíos en términos de almacenamiento, procesamiento y análisis, debido a la alta

dimensionalidad de los datos generados. Nos centraremos en la transcriptómica por su capacidad para ofrecer una visión dinámica y profunda de la actividad celular, permitiendo identificar patrones de expresión génica que reflejan procesos biológicos clave.

En este apartado, se describirá la estructura de los datos obtenidos mediante RNA-seq y single-cell RNA-seq, y se discutirán los principales retos asociados a su manejo, desde las consideraciones técnicas hasta las implicaciones estadísticas y computacionales, que hacen de la transcriptómica un área idónea para aplicar metodologías multivariantes en la identificación de patrones biológicos.

2.3.1 ¿Qué es la transcriptómica? Tecnologías de secuenciación

La transcriptómica es la rama de la biología que estudia el conjunto completo de ARN (ácido ribonucleico) transcritos en una célula, tejido u organismo en un momento determinado, bajo condiciones específicas. A este conjunto se le denomina transcriptoma. Se centra en la cuantificación y caracterización de los distintos tipos de ARN, incluyendo ARN mensajero (mARN), ARN de transferencia (tRNA), ARN ribosomal (rRNA) y ARN no codificante (ncRNA), entre otros. Este campo ha evolucionado significativamente desde la formulación del dogma central de la biología molecular por Francis Crick en 1958, que estableció la transferencia de información genética desde el ADN al ARN y posteriormente a las proteínas.

A medida que la transcriptómica ha avanzado, se han ido desarrollando varias tecnologías para deducir y cuantificar el transcriptoma, basadas tanto en hibridación como en secuenciación. Los enfoques basados en hibridación, como los microarrays, pese a que son más económicos y tienen un alto rendimiento, dependen del conocimiento existente sobre la secuencia del genoma.

A diferencia de los métodos basados en microarrays, los enfoques basados en secuencias determinan directamente la secuencia de ARN. Inicialmente, se utilizó la secuenciación de Sanger de bibliotecas de ARN, pero era bastante costosa y de bajo rendimiento y generalmente no cuantitativa. Se desarrollaron métodos basados en etiquetas para superar estas limitaciones, pero tenían el inconveniente de que estaban basados en la costosa tecnología de secuenciación de Sanger, y una parte significativa de las etiquetas cortas no se podían asignar de forma única al genoma de referencia. Todas estas desventajas limitan el uso de la tecnología de secuenciación tradicional para anotar la estructura de los transcriptomas [15].

Tecnologías de secuenciación de ADN de alto rendimiento como *RNA-seq* y *single-cell RNA-seq* han emergido como herramientas clave para estudiar la expresión génica a gran escala. Estas técnicas permiten la cuantificación precisa de los niveles de ARN en diferentes condiciones biológicas, a diferencia de los métodos basados en microarrays, lo que proporciona información valiosa sobre la actividad celular y los mecanismos de

regulación genética. Sin embargo, los datos obtenidos mediante RNA-seq y single-cell RNA-seq poseen características específicas que influyen en su representación y análisis. Estas tecnologías generan grandes volúmenes de datos con una estructura matricial compleja, en la que el número de características (genes) supera ampliamente al número de muestras, lo que da lugar a retos significativos en términos de almacenamiento, procesamiento y análisis.

2.3.2 RNA-seq

El método RNA-seq (secuenciación de ARN) consiste en la conversión de una muestra de ARN (total o fraccionado) en una biblioteca de ADNc (ADN codificado), que luego es secuenciada utilizando tecnologías de secuenciación profunda. Genera un conjunto masivo de datos que consiste en lecturas cortas de ARN transcrito, secuencias que generalmente varían entre 30 y 400 pares de bases de longitud, representando fragmentos de transcritos provenientes de ARN mensajero (ARNm) o ARN no codificante. Estas secuencias se alinean con un genoma de referencia o con transcritos de referencia para mapear la estructura transcripcional y cuantificar la expresión génica. Una de las principales aplicaciones de RNA-seq es el análisis de expresión diferencial, mencionado en la sección previa y que abordaremos de forma práctica, que permite comparar los niveles de expresión de genes entre diferentes condiciones biológicas, como células tratadas frente a no tratadas, tejidos sanos frente a cancerosos o distintos estados del desarrollo. Esto ofrece una visión detallada de los cambios en la actividad génica y ayuda a identificar biomarcadores, rutas metabólicas alteradas o procesos reguladores clave. Además, RNA-seq no está limitado a detectar solo transcritos que corresponden a secuencias genómicas conocidas, lo que lo hace particularmente útil para organismos poco estudiados o cuando se carece de un genoma de referencia bien caracterizado [16].

En la secuenciación de ARN, las lecturas generadas a partir de las muestras de ARN se alinean contra un genoma de referencia o se ensamblan de nuevo para crear un *mapa transcripcional*. Si se dispone de un genoma de referencia, los datos se alinean para identificar la ubicación exacta de los transcritos en el genoma, permitiendo la cuantificación de la expresión génica. En casos donde no hay un genoma de referencia, las lecturas de ARN se ensamblan para generar una secuencia de contigs³ que luego se pueden anotar funcionalmente. Además de mapear las lecturas a un genoma, se deben identificar eventos de empalme (splicing⁴), que es crucial para detectar variantes de splicing alternativo. Este proceso es especialmente importante para genes que tienen varios exones, ya que las lecturas pueden cruzar estos empalmes y revelar alternativas de splicing que no son evidentes con tecnologías anteriores [19].

³Contig: Tramo de secuencia continua in silico generada por alineamiento de lecturas de secuencias solapantes [17].

⁴Splicing: el splicing o empalme, ocurre al final del proceso de transcripción e implica cortar y reorganizar secciones de ARNm [18].

Pese a las ventajas que la RNAseq tiene frente a tecnologías anteriores, los conjuntos de datos producidos son grandes y complejos y la interpretación no es sencilla. La interpretación de los datos de secuenciación de ARN depende de la cuestión científica de interés. El objetivo principal de muchos estudios biológicos es el perfil de expresión génica entre muestras, que es particularmente relevante, por ejemplo, para experimentos controlados que comparan la expresión en cepas de tipo salvaje y mutantes del mismo tejido, comparando células tratadas versus no tratadas, cáncer versus normal, etc. [20].

Por otra parte, los datos RNA-seq requieren estar en unos formatos específicos para su tratamiento. Formatos adecuados para almacenar secuencias tanto de ácidos nucleicos como de proteínas. Estos son: formato *FASTA* y *FASTQ*. La información que presentamos a continuación ha sido extraída de [9].

1. **Formato FASTA:** basado en texto, es usado para representar secuencias de nucleótidos o de aminoácidos, ambos representados mediante una sola letra. Incluye símbolos para representar huecos (*gaps*) o posiciones desconocidas en la secuencia. Consta de dos líneas:

- a) La primera línea comienza con el símbolo `>`, junto con una descripción de la secuencia.
- b) La segunda, contiene la secuencia de bases o aminoácidos.

Pese a que no hay restricciones en el número de filas, el número de columnas no debería superar las 80.

2. **Formato FASTAQ:** es el más popular y consiste en cuatro líneas por lectura:

- a) La primera comienza con el carácter '@' y contiene el nombre de la secuencia. Opcionalmente, puede incluir una descripción.
- b) La segunda línea contiene la secuencia con las letras correspondientes, dependiendo del tipo de secuencia del que se trate (nucleótido o aminoácido).
- c) La tercera comienza con el carácter '+' y contiene información opcional sobre la secuencia.
- d) La cuarta y última línea cuantifica la calidad o confiabilidad de cada base en la secuencia recogida en la segunda línea, basada en el índice *Phred* y su codificación.

```

1      @SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
2      ACAGGGACGCCATCGAATCCGGATCNTNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
3      +SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
4      dee\edYcdc`bbY`S]bb_]Ua^BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
5

```

2.3.3 scRNA-seq (single-cell RNA-seq)

La secuenciación de ARN ha impulsado muchos descubrimientos e innovaciones en diversos campos en los últimos años. Por razones prácticas, la técnica RNA-seq suele realizarse en muestras que comprenden entre miles y millones de células. Sin embargo, esto ha dificultado la evaluación directa de la unidad fundamental de la biología: la célula. Es por esto que surge una variante a la RNA-seq: *scRNA-seq* (*single-cell RNA-seq*), que se ha extendido considerablemente hasta consolidarse como la opción principal en investigación de la diversidad celular, ya que posibilita el estudio individual de cada célula dentro de una misma muestra. Es una tecnología que permite la cuantificación y comparación de los transcriptomas de células individuales, logrando una disección de la expresión génica a resolución de una sola célula y describiendo moléculas de ARN con alta precisión y a escala genómica. Además, otro aspecto importante de esta técnica es que permite analizar la heterogeneidad celular; evaluar las similitudes y diferencias transcripcionales dentro de una población de células, lo que ha permitido una comprensión más detallada de los procesos moleculares al evidenciar la variabilidad existente dentro de una misma población celular.

Aunque existe una gran confianza en la utilidad general de scRNA-seq, hay una barrera técnica que debe considerarse cuidadosamente: el aislamiento efectivo de células individuales del tejido de interés, pues existe un riesgo potencial de que los protocolos utilizados para este proceso alteren los niveles de ARNm. Aunque las células vecinas pueden contribuir a mantener los estados celulares, scRNA-seq opera bajo el supuesto de que el aislamiento de células individuales no desencadena cambios transcriptómicos antes de la captura del ARNm. Además, esta técnica a menudo requiere el uso de marcadores específicos para distinguir con precisión diferentes poblaciones celulares, lo que puede añadir complejidad al diseño experimental. A esto se suman limitaciones relacionadas con el tiempo y los costos asociados, que pueden dificultar la realización de investigaciones a gran escala.

El análisis de datos provenientes de scRNA-seq supone un desafío computacional considerable debido a su alta dimensionalidad y a la presencia de ruido significativo. En el estudio de muestras heterogéneas, es fundamental identificar las distintas subpoblaciones celulares presentes, lo que requiere el desarrollo de algoritmos capaces de descomponer la mezcla de expresión génica obtenida. Para ello, se aplican técnicas de normalización y reducción de dimensionalidad, como el análisis de componentes principales (PCA), entre otras, que permiten segmentar las células en función de sus perfiles de expresión. No obstante, la presencia de efectos por lote y la variabilidad técnica pueden introducir sesgos en los resultados, lo que hace necesario implementar estrategias de corrección para garantizar la robustez del análisis y evitar la aparición de errores sistemáticos en la identificación de patrones biológicos.

Además, el volumen de datos generado por scRNA-seq plantea retos en términos de almacenamiento, gestión y escalabilidad computacional. La integración de millones

de lecturas individuales requiere estrategias eficientes de compresión y procesamiento distribuido, así como el uso de infraestructuras de alto rendimiento para análisis en estudios de gran escala. Dado que el aislamiento manual de células en el laboratorio puede ser costoso y técnicamente desafiante, se han propuesto enfoques computacionales que permitan inferir la composición celular sin necesidad de manipulación experimental. En este contexto, los métodos multivariantes desempeñan un papel clave en la extracción de información relevante, facilitando la identificación de estructuras en los datos sin necesidad de un conocimiento previo sobre los tipos celulares presentes en la muestra. La combinación de técnicas estadísticas y aprendizaje automático ha demostrado ser esencial para mejorar la precisión y fiabilidad de estos análisis en el ámbito biomolecular.

Dado que scRNA-seq es una variante de RNA-seq, hereda sus formatos de almacenamiento iniciales, principalmente FASTA y FASTQ, siendo este último el formato más crudo en el que se encuentran los datos de scRNA-seq.

Toda la información recogida en esta subsección ha sido extraída de las fuentes bibliográficas [21],[22],[23].

Parte II

FUNDAMENTOS MATEMÁTICOS

3 Técnicas multivariantes: fundamentos y desarrollo del análisis clúster

El análisis de datos en ciencias ómicas requiere metodologías capaces de manejar la complejidad inherente a los sistemas estudiados. En particular, las técnicas multivariantes han demostrado ser herramientas fundamentales para la exploración, modelado e interpretación de datos de alta dimensión. Estas metodologías permiten identificar relaciones entre variables, reducir la dimensionalidad y clasificar observaciones en función de patrones subyacentes.

Este capítulo está estructurado en dos secciones. En primer lugar, se presentarán las principales técnicas multivariantes, destacando su utilidad y objetivos dentro del análisis de datos. Posteriormente, se abordará en profundidad el análisis clúster, una técnica multivariante ampliamente utilizada para la identificación de patrones en grandes volúmenes de datos. Su aplicación en el ámbito biológico permite revelar estructuras subyacentes en datos complejos, facilitando la comprensión de procesos como la agrupación de expresiones génicas o la clasificación de organismos en función de sus características.

Dado que en el capítulo dedicado a los datos ómicos hemos introducido la matriz de datos ómicos X , que representa las N características medidas sobre n muestras, mantendremos esta notación en el desarrollo de los fundamentos matemáticos sobre los que se basa este trabajo. Así, consideraremos que la matriz de expresión $X \in \mathbb{R}^{N \times n}$ almacena las observaciones de nuestras variables, con filas representando las características y columnas las muestras.

3.1 Preliminares

La información aquí recogida se ha extraído de las fuentes [24, 25, 26].

El análisis multivariante es una herramienta clave para explorar y comprender la complejidad de los sistemas biológicos, económicos y sociales. Su capacidad para procesar múltiples variables simultáneamente permite identificar patrones ocultos en grandes volúmenes de datos.

Las técnicas multivariantes son fundamentales para abordar la complejidad de los datos en diversas disciplinas, incluyendo las ciencias ómicas, donde se requieren metodologías capaces de gestionar la alta dimensionalidad y variabilidad de los datos obtenidos. Estas herramientas permiten descubrir relaciones entre variables, reducir la

dimensionalidad y clasificar observaciones, lo que facilita la interpretación y el modelado de sistemas complejos.

El desarrollo del análisis multivariante se remonta a principios del siglo XX, cuando pioneros como Karl Pearson y R.A Fisher introdujeron técnicas fundamentales como el análisis de componentes principales y el análisis discriminante. Posteriormente, C.R. Rao y otros investigadores expandieron estos métodos, estableciendo bases matemáticas sólidas que han permitido su aplicación en un amplio espectro de disciplinas. Estas técnicas han ido desarrollándose exponencialmente con el avance de la computación, facilitando el procesamiento de grandes volúmenes de datos y dando lugar a análisis mucho más sofisticados en muchas áreas como la biología, la economía, las ciencias sociales, etc.

En términos generales, las metodologías multivariantes pueden dividirse en dos grandes enfoques: *descriptivo* e *inferencial*. El primero busca simplificar la estructura de los datos y revelar relaciones latentes entre variables, mientras que el segundo permite realizar pruebas de hipótesis considerando múltiples variables de manera simultánea, asegurando la validez estadística de los resultados. La elección de la técnica adecuada depende del tipo de datos y de la pregunta de investigación. A continuación, se presentan algunas de las principales metodologías multivariantes.

3.1.1 Análisis de Componentes Principales (PCA)

El *análisis de componentes principales (PCA)* fue introducido por primera vez por Karl Pearson a principios del siglo XX. El tratamiento formal de esta técnica se debe a Hotelling (1933) y Rao (1964). Su propósito era facilitar la comprensión de conjuntos de datos complejos mediante la reducción de su dimensionalidad, minimizando la pérdida de información. En PCA, un conjunto de N variables correlacionadas se transforma en un conjunto más pequeño de constructos hipotéticos no correlacionados llamados *componentes principales* (CP). Su objetivo es condensar la información proporcionada por dichas variables en unas pocas de ellas o en pocas combinaciones lineales de ellas (con máxima variabilidad).

Las componentes principales se definen como combinaciones lineales de las variables originales que capturan la mayor variabilidad posible en los datos. Matemáticamente, si Y es un vector de N variables observadas con media μ y matriz de covarianza Σ , las componentes principales Z_i se obtienen como:

$$Z_i = p_i'Y, \quad i = 1, 2, \dots, N$$

donde p_i es un vector de pesos o *cargas principales* que maximizan la varianza de Z_i bajo la restricción de que p_i tiene norma unitaria, es decir,

$$\max \text{Var}(Z_i) = p_i'\Sigma p_i, \text{ sujeto a } p_i'p_i = 1.$$

y tal que asegura que las componentes principales son ortogonales entre sí, es decir:

$$p_i' p_j = 0, \text{ para } i \neq j.$$

Así, garantizamos que las componentes principales Z_i y Z_j son intercorrelacionadas, es decir, su covarianza es cero para $i \neq j$.

Los vectores p_i son los autovectores de la matriz de covarianza Σ , y los valores propios λ_i , corresponden a la varianza explicada por cada componente principal. La transformación completa de los datos se expresa de la siguiente forma:

$$Z = P' Y$$

donde P es la matriz de autovectores de Σ , lo que garantiza que las componentes principales sean ortogonales entre sí y no correlacionadas, cada una con las anteriores.

Las CP se utilizan para descubrir e interpretar las dependencias que existen entre las variables y para examinar las relaciones que pueden existir entre los individuos. También son útiles para estabilizar las estimaciones, evaluar la normalidad multivariante y detectar valores atípicos.

3.1.2 Análisis factorial

El objetivo principal del *análisis factorial* (AF) es capturar la realidad de la manera más simple posible, identificando unas pocas variables latentes⁸ que definen esa realidad. Esta técnica multivariante busca explicar el comportamiento de las N variables en la matriz de datos X utilizando un número reducido de variables latentes, denominadas *factores*. Lo ideal es que toda la información contenida en X pueda ser representada mediante un número menor de factores. Esta técnica busca explicar las correlaciones entre las variables mediante la combinación lineal de dichos factores. Así, cada factor es una variable latente que influye en las variables observadas, y cuya presencia se infiere a partir de las correlaciones entre ellas.

Matemáticamente, cada variable observada, $x_i \in \mathbb{R}^N$, se expresa como una combinación lineal de estos factores, más un término de error específico:

$$x_i = \sum_{l=1}^k q_{il} f_l + \mu_i + e_i, \quad i = 1, \dots, N.$$

donde, $q_{il} \in [q_{il}]_{N \times k}$ es una matriz de pesos, f_l , con $l = 1, \dots, k$ son los factores, μ_i denota la media de la variable x_i y e_i es la componente i -ésima del vector de errores aleatorios, $e_{N \times 1}$. El número de factores, k , debería ser siempre mucho más pequeño que N .

⁸Variable latente: variable no observable que se infiere a partir de un conjunto de variables observables utilizando un modelo matemático.

En definitiva, el modelo de análisis factorial asume que la variable observada x puede descomponerse en dos componentes: una parte explicada por los factores comunes y una parte específica de cada variable. Esto se expresa matricialmente de la siguiente forma:

$$x = \Lambda F + \psi$$

donde:

- Λ es la matriz de cargas factoriales de dimensión $N \times k$,
- F es el vector de factores de dimensión k .
- ψ es el vector de factores específicos o residuales de dimensión N

El análisis factorial fue desarrollado por Charles Spearman a principios del siglo XX para modelar la inteligencia humana, postulando que las puntuaciones en distintas pruebas estaban intercorrelacionadas debido a un único factor latente de inteligencia general. Su modelo de un solo factor fue posteriormente generalizado por Thurstone a múltiples factores.

El análisis de componentes principales (PCA) y el análisis factorial suelen confundirse porque ambos analizan la variación en un conjunto de variables a partir de la matriz de correlación o covarianza. Sin embargo, mientras que en el AF unas pocas variables latentes explican las correlaciones observadas, en el PCA se necesitan todos los componentes principales para describir completamente la variabilidad. Así, el PCA se centra en explicar la varianza total, mientras que el AF se enfoca en las relaciones entre las variables mediante factores comunes.

3.1.3 Análisis Discriminante

El *análisis discriminante* es una técnica multivariante que permite identificar un subconjunto de variables y funciones asociadas que maximicen la separación entre los grupos o poblaciones de estudio. Su objetivo principal es construir funciones discriminantes que describan y caractericen la separación de los grupos, evaluar el grado de diferenciación y analizar la contribución de cada variable a la discriminación.

Cuando estas funciones son combinaciones lineales de las variables originales, se denominan funciones discriminantes lineales (LDF). En particular, el análisis discriminante lineal de Fisher busca encontrar una combinación lineal de variables que maximice la separación entre grupos. Para dos grupos con medias μ_1 y μ_2 y una matriz de covarianza común Σ , la función discriminante de Fisher se define como:

$$L = a' y = \sum_{j=1}^N a_j y_j$$

donde a es el vector de coeficientes de la función discriminante e y es el vector de observaciones de las variables.

El vector a que maximiza la separación entre los grupos se obtiene como:

$$a_{\Sigma} = \Sigma^{-1}(\mu_1 - \mu_2)$$

Además, la *distancia de Mahalanobis*, que explicaremos con detalle en la siguiente sección, se emplea para medir la separación entre los centroides de los grupos:

$$D^2 = (\mu_1 - \mu_2)' S^{-1} (\mu_1 - \mu_2)$$

Si D^2 es significativo, implica una buena discriminación entre los grupos.

Este análisis tiene aplicaciones en diversos campos: en biología, Fisher (1936) lo utilizó para diferenciar especies de iris en función de características morfológicas; en la gestión de personal, permite clasificar profesionales según sus habilidades; en medicina, ayuda a distinguir entre individuos con alto o bajo riesgo de enfermedades; y en la industria, contribuye a identificar cuándo un proceso está bajo control o fuera de control.

Para el caso de múltiples grupos, la función discriminante se construye de manera que maximice la variabilidad entre los grupos en relación con la variabilidad dentro de los grupos, lo que se logra mediante una descomposición en valores propios.

Toda la información ha sido extraída de las fuentes bibliográficas [27, 28, 29, 30]

3.2 Análisis Clúster

3.2.1 Introducción

Un ser inteligente no puede tratar cada objeto que ve como una entidad única, diferente de cualquier otra en el universo. Debe categorizar los objetos para poder aplicar el conocimiento adquirido con tanto esfuerzo sobre objetos similares encontrados en el pasado al objeto en cuestión.

Steven Pinker, Cómo funciona la mente, 1997

Una de las habilidades básicas que poseemos los seres humanos es la de agrupar objetos similares para generar una clasificación. Esta idea de clasificar cosas similares en categorías es bastante primitiva. Nuestros antepasados prehistóricos debían ser capaces de darse cuenta de que muchos objetos tenían propiedades semejantes, como ser comestibles, venenosos, peligrosos, etc.

Organizar datos en grupos razonables es uno de los modos más fundamentales de comprensión y aprendizaje. De hecho, es vital para el desarrollo del lenguaje, el cual

consiste en palabras que nos ayudan a reconocer y analizar los diferentes tipos de eventos, objetos y personas con los que nos relacionamos. Por ejemplo, los sustantivos en una lengua son palabras que describen una clase de cosas que comparten unas características comunes; gatos, perros, caballos, etc., y dicho nombre agrupa a los individuos en grupos.

Además de ser una actividad conceptual humana básica, la clasificación es fundamental en muchas ramas de la ciencia. En biología, por ejemplo, la clasificación de organismos (*taxonomía*) ha sido una gran preocupación desde las primeras investigaciones. Aristóteles ya ideó un sistema para clasificar las especies del reino animal; comenzó dividiendo a los animales en dos grupos: los que tienen sangre roja (vertebrados) y los que carecen de ella (invertebrados). Además subdividió estos grupos según su forma de reproducirse: vivas, en huevos, en pupas, etc. Luego, Theophrastos escribió lo relativo a las plantas. Los libros resultantes estaban tan bien documentados y eran tan completos, tan profundos y de un alcance tan amplio que sentaron las bases de la investigación biológica durante siglos.

No fue ya hasta los siglos XVII y XVIII cuando los exploradores europeos crearon un nuevo programa similar de investigación y recolección bajo la dirección del sueco Linnaeus, quien estableció un sistema de clasificación que sentó las bases de la taxonomía moderna. Su método no solo organizaba a los seres vivos en categorías jerárquicas, sino que también reflejaba una idea más profunda: la clasificación es esencial para el conocimiento.

En este sentido, todo el conocimiento real que poseemos, depende de métodos con los que podamos distinguir lo similar de lo diferente. Cuanto mayor sea el número de distinciones naturales que un método comprenda, más clara será nuestra idea de las cosas. A medida que el número de objetos de estudio crece, la necesidad de desarrollar sistemas de clasificación más precisos se vuelve aún más necesaria.

En cierto nivel, un esquema de clasificación puede simplemente representar un método conveniente para organizar un gran conjunto de datos, de modo que se pueda comprender con mayor facilidad y la información se recupere de forma más eficiente. Si los datos pueden resumirse adecuadamente mediante un pequeño número de grupos de objetos, las etiquetas de grupo pueden proporcionar una descripción muy consisa de los patrones de similitudes y diferencias entre los datos. La necesidad de resumir conjuntos de datos de esta manera es cada vez más importante debido al creciente número de grandes bases de datos disponibles en muchas áreas de la ciencia, como la transcriptómica, que es la que nos ocupa en este trabajo, y la exploración de dichas bases de datos mediante *análisis clúster* y otras técnicas de análisis multivariante se denomina hoy día *minería de datos*.

Las técnicas numéricas de clasificación se originaron principalmente en las ciencias naturales, con el objetivo de liberar a la taxonomía de su naturaleza tradicionalmente

subjetiva. El objetivo era dar clasificaciones objetivas y estables. Objetivas en el sentido de que el análisis del mismo conjunto de organismos mediante la misma secuencia de métodos numéricos produce la misma clasificación; estables en cuanto a que la clasificación permanece igual ante la adición de organismos o de nuevas características que los describen.

Se le han dado muchos nombres a estas técnicas numéricas, dependiendo del área de aplicación. En biología, el término más extendido es el de *taxonomía numérica*. En psicología, se usa mucho el término *análisis Q*. En inteligencia artificial, el reconocimiento de patrones no supervisado es el término predilecto. Sin embargo, hoy en día, el *análisis clúster* es probablemente el término genérico para los procedimientos que buscan descubrir grupos en los datos.

La información recogida en esta introducción ha sido extraída de la fuente bibliográfica [31].

3.2.2 ¿Qué es el Análisis Cluster (AC)?

El *análisis cluster* puede definirse como el estudio formal de los algoritmos y métodos de clasificación de objetos. Un objeto es descrito por un conjunto de mediciones o bien de relaciones entre el objeto y otros objetos. No usa etiquetas de categoría que etiqueten objetos con identificadores previos. A diferencia del análisis discriminante, el análisis clúster no utiliza etiquetas predefinidas para clasificar los objetos, sino que busca descubrir estructuras en los datos de manera autónoma.

El objetivo del análisis cluster es agrupar objetos formando conjuntos (clusters) en los que los elementos dentro de cada uno sean lo más similares posible entre sí (baja variabilidad interna), mientras que los diferentes grupos sean lo más distintos entre sí (alta variabilidad entre ellos). Es, en definitiva, una técnica exploratoria que identifica patrones de similitud dentro de un conjunto de datos, agrupando elementos con características comunes mientras mantiene separadas las estructuras con mayores diferencias [32].

Un cluster puede entenderse como un conjunto de elementos que presentan una alta cohesión interna (homogeneidad) y una clara separación externa con respecto a otros grupos. Sin embargo, la definición formal de un cluster es difícil de establecer y depende en gran medida del juicio del usuario y del contexto en el que se aplica. Mientras que algunos métodos de análisis buscan identificar estructuras naturales en los datos, en muchas ocasiones el proceso de agrupamiento puede imponer una estructura artificial en la información. Esto resalta la importancia de interpretar con cautela los resultados de un análisis de clusters, ya que no siempre reflejan patrones inherentes a los datos, sino que pueden ser el resultado de los criterios específicos utilizados en la clasificación [31].

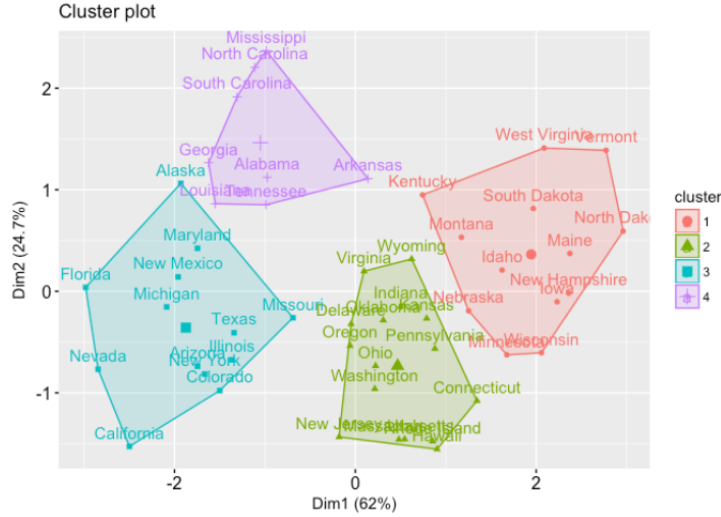


Figure 3.1: Ejemplo de clustering (Fuente: [33]).

En la mayoría de las aplicaciones del AC se busca una partición de los datos en la que cada individuo u objeto pertenezca a un único cluster y el conjunto completo de clusters contenga a todos los individuos. Sin embargo, esto no siempre es así y, de hecho, en algunas circunstancias, la superposición de clusters puede ofrecer una solución más aceptable. Decimos que una respuesta aceptable del análisis cluster es que no se justifica la agrupación de los datos. El análisis cluster es un procedimiento objetivo; no están predefinidos, sino que se forman a medida que avanza el análisis.

Los datos básicos para la mayoría de las aplicaciones del análisis cluster se almacenan en una matriz $n \times p$, X , que contiene los valores de las variables que describen cada objeto que se va a agrupar, es decir,

$$X = [x_{ij}]_{i=1,\dots,n,j=1,\dots,p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

donde x_{ij} cuantifica la variable j en la muestra i . El AC tratará de desarrollar un esquema de clasificación que particionará las filas de X en k clusters [31].

3.2.3 Medidas de proximidad

Dado que el análisis cluster trata de identificar los vectores que son similares y agruparlos en clusters, es esencial contar con herramientas que permitan evaluar la cercanía o distancia entre los objetos que se están agrupando: las *medidas de proximidad*. Las decisiones sobre cómo se van a formar los clusters dependen directamente de las medidas de proximidad utilizadas, ya que estas definen cómo se calcula la similitud o diferencia entre los distintos elementos.

Si una medida de proximidad representa *similitud*, el valor de la medida incrementa cuanto más similares sean dos objetos. Alternativamente, si la medida de proximidad representa *disimilitud*, el valor de la medida disminuye a medida que dos objetos se vuelven más parecidos.

Medidas de Disimilaridad

Cuando todas las variables registradas son continuas, las proximidades entre los individuos generalmente se cuantifican mediante medidas de disimilaridad o medidas de distancia. Por ello, definiremos el concepto de *disimilaridad* análogamente al de distancia.

Definición 3.1. Sean $\Omega \subset \mathbb{R}^n$ un conjunto de puntos de \mathbb{R}^n y $x, y \in \Omega$ dos puntos cualesquiera de dicho conjunto. Entendemos por disimilaridad a toda aplicación $d : \Omega \times \Omega \rightarrow \mathbb{R}$ que satisface las siguientes propiedades:

- i) $d(x, y) \geq 0$
- ii) $d(x, y) = 0 \iff x = y$
- iii) $d(x, y) = d(y, x)$ (simétrica)

Se dice que la disimilaridad es *métrica* si satisface una cuarta propiedad:

$$\text{iv) } d(x, y) \leq d(x, z) + d(z, y) \forall z \in \Omega,$$

y se dirá *ultramétrica* si es métrica y además cumple:

$$\text{v) } d(x, y) \leq \max\{d(x, z), d(y, z)\}$$

A continuación presentamos algunas de las medidas de disimilaridad más usadas. Hemos de hacer notar que estas medidas normalmente se usan para medir cuán próximos están los individuos, pero si se quiere medir entre variables, también son válidas. Simplemente habría que trasponer la matriz X y trabajar con ella.

Toda la información que sigue a partir de este punto, ha sido extraída de las fuentes bibliográficas:[26, 29, 30, 31, 34].

Las medidas de disimilaridad se usan ante la necesidad de agrupar objetos con características similares. Una medida de disimilaridad adecuada es la distancia entre dos observaciones. Se considera una medida de disimilaridad por el hecho de que la distancia aumenta conforme dos individuos se alejan.

Una gran variedad de distancias pueden generarse mediante la norma¹⁰ $L_r, r \geq 1$,

¹⁰Norma: Una *norma* en un espacio vectorial X es una aplicación $\|\cdot\| : X \rightarrow \mathbb{R}$ que verifica:

$$d_{ij} = \|x_i - x_j\|_r = \left\{ \sum_{k=1}^p |x_{ik} - x_{jk}|^r \right\}^{\frac{1}{r}} \quad (1)$$

donde x_{ik} denota el valor de la k -ésima variable en el objeto i . A esta distancia se la conoce por el nombre de *distancia de Minkowski*

En lo que sigue, consideremos $x_i, x_j \in \mathbb{R}^p$ dos individuos de la población. Esto es, dos filas de la matriz X que contine los datos. Definimos las siguientes medidas de disimilaridad:

Definición 3.2. Definimos la *distancia euclídea* como la derivada de la norma L_2 (caso particular de la distancia de Minkowski cuando $r = 2$),

$$d_2(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{(x_i - x_j)'(x_i - x_j)} = \sqrt{\sum_{j=1}^p (x_j - y_j)^2}$$

De entre todas las medidas de disimilaridad, y de todas las métricas de Minkowski, la distancia Euclídea es la más común. Hemos de tener en cuenta que la distancia euclídea (y la euclídea al cuadrado) suelen calcularse a partir de datos en bruto, no de datos estandarizados. Esto presenta varias ventajas (p. ej., la distancia entre dos objetos cualesquiera no se ve afectada por la adición de nuevos objetos al análisis, que podrían ser valores atípicos). Sin embargo, las distancias pueden verse considerablemente afectadas por las diferencias de escala entre las dimensiones a partir de las cuales se calculan. Por ejemplo, si una de las dimensiones representa una longitud medida en centímetros y luego se convierte a milímetros (multiplicando los valores por 10), las distancias euclidianas o euclidianas al cuadrado resultantes (calculadas a partir de múltiples dimensiones) pueden verse considerablemente afectadas y, en consecuencia, los resultados de los análisis de conglomerados pueden ser muy diferentes.

Observación 3.3. La *distancia euclídea al cuadrado* puede resultar útil para asignar progresivamente mayor importancia a los objetos más separados. Además, al evitar las raíces cuadradas, los cálculos se simplifican.

Definición 3.4. Se denomina *distancia Manhattan, City Block, Hamming*, o d_1 , a la distancia derivada de la norma L_1 , que viene dada por

$$d_1(x_i, x_j) = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

(N.1) Desigualdad triangular: $\|x + y\| \leq \|x\| + \|y\| \forall x, y \in X$

(N.2) Homogeneidad por homotecias: $\|\lambda x\| = |\lambda| \|x\| \forall x \in X, \lambda \in \mathbb{R}$

(N.3) No degeneración: $x \in X, \|x\| = 0 \Rightarrow x = 0$. [35]

Se utiliza cuando todas las características (variables) son binarias y mide el número de características en las que dos individuos difieren.

Definición 3.5. Tomando límite cuando $r \rightarrow \infty$, la distancia derivada del límite de la norma L_r , llamada *norma del máximo*, L_∞ , es la que se conoce como *distancia de Chebychev*, y viene dada por

$$d_\infty(x_i, x_j) = \max_{k=1, \dots, p} |x_{ik} - x_{jk}|$$

Observación 3.6. La distancia euclídea la podemos ver como un caso particular de la distancia:

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)' M (x_i - x_j)}$$

cuando $M = I_p$, la matriz identidad de orden p .

Supongamos que M es la matriz de covarianzas de las variables (las columnas de la matrix X de datos), que se define de la siguiente forma:

$$M = \Sigma = \frac{1}{p} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ij} - \bar{x}_j)', \quad j = 1, \dots, p$$

Podemos observar que si $p \geq n$, la matriz de covarianzas Σ , es definida positiva y, por consiguiente, es invertible. Esto nos permite definir la siguiente medida de disimilaridad:

Definición 3.7. Definimos la *distancia de Mahalanobis* para individuos, como:

$$D_\Sigma(x_i, x_j) = \sqrt{(x_i - x_j)' \Sigma^{-1} (x_i - x_j)}$$

A diferencia de la distancia Euclídea, la de Mahalanobis es invariante frente a cambios de escala. En efecto, sea $C_{n \times n}$ una matriz no singular de orden n , entonces:

$$\begin{aligned} D_\Sigma(Cx_i, Cx_j) &= \sqrt{(Cx_i, Cx_j)' \left[\frac{1}{p} C(x_{ij} - \bar{x}_j)(x_{ij} - \bar{x}_j)' C' \right]^{-1} (Cx_i, Cx_j)} = \\ &= \sqrt{(x_i, x_j)' C' (C')^{-1} \left[\frac{1}{p} (x_{ij} - \bar{x}_j)(x_{ij} - \bar{x}_j)' \right]^{-1} C^{-1} C (x_i, x_j)} = \\ &= \sqrt{(x_i - x_j)' \Sigma^{-1} (x_i - x_j)} = D_\Sigma(x_i, x_j) \end{aligned}$$

La distancia de Mahalanobis no se usa comúnmente en técnicas de clustering porque calcula la matriz Σ considerando todos los individuos juntos, en lugar de tratar los objetos de cada clúster por separado. Además, su cálculo es más complejo que el de otras métricas. Sin embargo, puede aplicarse dentro de cada clúster en una fase específica del proceso [36].

Medidas de similaridad

Como en la subsección anterior, consideremos x_i, x_j dos vectores cualesquiera de \mathbb{R}^p .

Definición 3.8. Sea $\Omega \subset \mathbb{R}^n$ un conjunto de puntos de \mathbb{R} . Llamaremos *similaridad* a toda aplicación $s : \Omega \times \Omega \rightarrow \mathbb{R}$ que satisfaga las siguientes propiedades:

- i) $0 \leq s(x_i, x_j) \leq 1$
- ii) $s(x_i, x_j) = 1$ si y solo si $x_i = x_j$
- iii) $s(x_i, x_j) = s(x_j, x_i)$ (simétrica)

Las condiciones (i) y (ii) aseguran que toda similaridad es siempre positiva e idénticamente 1 cuando los objetos i y j sean iguales.

Observación 3.9. Podemos formar una disimilaridad a partir de una similaridad, sin más que hacer:

$$d_{ij} = 1 - s_{ij}$$

u otra función decreciente. Sin embargo, la diferencia de las disimilaridades, no constituye una métrica.

Naturalmente, podríamos pensar si esto también ocurre a la inversa, es decir, si dada una medida de disimilaridad d_{ij} , podemos construir una medida de similaridad, sin más que despejar s_{ij} en la anterior expresión:

$$s_{ij} = 1 / (1 + d_{ij})$$

Pero como d_{ij} no está acotada superiormente, $s_{ij} \in (0, 1]$, lo cual implica que $s_{ij} \neq 0$. Es por ello que no se pueden crear similaridades a partir de disimilaridades.

A continuación presentamos las principales medidas de similaridad que se usan en el ámbito del análisis cluster.

Definición 3.10. Definimos el *coeficiente de correlación de Pearson* entre los objetos x_i, x_j , como

$$q_{ij} = \frac{\sum_{k=1}^p (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{[\sum_{k=1}^p (x_{ik} - \bar{x}_i)^2 \sum_{k=1}^p (x_{jk} - \bar{x}_j)^2]}$$

donde $\bar{x}_i = \sum_{k=1}^p \frac{x_{ik}}{p}$ y $\bar{x}_j = \sum_{k=1}^p \frac{x_{jk}}{p}$.

Observación 3.11. Dado que $q_{ij} \in [-1, 1]$, es claro que no satisface la primera condición necesaria para ser una similaridad. No obstante, esto cambia si consideramos en su lugar, el valor absoluto del mismo, $|q_{ij}|$, o bien tomando $1 - q_{ij}^2$.

Presentamos ahora otra medida de similaridad para las filas de X .

Definición 3.12. Definimos el *coseno del ángulo* θ entre los vectores x_i, x_j como

$$\cos\theta = c_{ij} = \frac{x_i'x_j}{\|x_i\|\|x_j\|}$$

Observación 3.13. Podemos ver que estamos en la misma situación que antes, $c_{ij} \in [-1, 1]$, lo cual indica que no verifica la propiedad 1 de las medidas de similaridad. La solución a esto es la misma que en el caso anterior: trabajar con $\|q_{ij}\|$ o $\|1 - q_{ij}^2\|$.

Llegados a este punto en el que ya sabemos medir cómo de diferentes o similares son los objetos que queremos agrupar, es el momento de introducir distintas estrategias o algoritmos de agrupación, que pueden ser jerárquicas o no jerárquicas.

3.2.4 Métodos jerárquicos

La información recogida en esta subsección y en las sucesivas subsubsecciones se ha obtenido principalmente de las fuentes bibliográficas [26, 29, 32, 37, 38, 39].

Los *métodos jerárquicos* para el análisis cluster representan un intento de encontrar buenos clusters en los datos mediante la combinación o división de clusters, con el propósito de minimizar (maximizar) una medida de disimilaridad (similaridad). Hay dos tipos de métodos jerárquicos: los *aglomerativos* y los *disociativos*. Los aglomerativos comienzan con clusters que contienen un solo objeto y sucesivamente van combinando clusters hasta que todos forman un único cluster. Los disociativos, por su parte, hacen justo lo contrario: comienzan con todos los objetos agrupados en un único cluster, a continuación dividen dicho cluster en dos clusters separados, y así sucesivamente hasta formar clusters con un solo objeto.

Observación 3.14. Aunque se le suele prestar más atención a los métodos aglomerativos, los disociativos proporcionan clusterings más sofisticados y robustos.

Dendrograma

El resultado final de todo método jerárquico es lo que se conoce como *dendrograma* o *diagrama en árbol jerárquico*. Es una representación gráfica del clústering que generalmente se dibuja en sentido inverso, comenzando desde el último cluster que se ha formado y que engloba a todos los objetos. En el punto de similaridad donde dos clusters se combinan para dar lugar al cluster final, este se divide en dos clusters padres y así sucesivamente; la solución con k clusters se obtiene fusionando algunos clusters de la solución con $(k + 1)$.

El dendrograma puede dibujarse horizontal o verticalmente, queda a elección de cada uno; ambas formas proporcionan la misma información. En nuestro caso, lo consideraremos en vertical. Nos permite determinar la altura del criterio de unión a la que los clusters se combinan para formar un nuevo cluster más grande. Los elementos

similares se combinan a alturas bajas, mientras que los elementos más disímiles se combinan a mayor altura en el dendrograma. Por lo tanto, es la diferencia de alturas la que define la proximidad de los individuos entre sí. Cuanto mayor sea la distancia entre las alturas a las que se combinan los clusters, más fácilmente podremos identificar la estructura sustancial de los datos.

Cortando el dendrograma a una altura adecuada, podremos obtener una partición de los datos en un número específico de grupos. Si dibujamos una línea horizontal a una altura adecuada, el número, k , de líneas verticales intersecadas por dicha línea horizontal determina una solución con k clusters. Por tanto, estamos diciendo que cada intersección de la horizontal con una de las k líneas verticales representa un cluster y los elementos que están por debajo de dicha intersección serán los elementos que conforman el cluster.

Nos podemos preguntar cómo saber a qué altura cortar el dendrograma. La respuesta es que depende; deberemos cortar a aquella altura que mejor represente los datos de entrada.

Nota 1. Cabe remarcar que, si bien las distancias verticales son importantes a la hora de determinar una solución, las horizontales son totalmente irrelevantes.

Ejemplo 3.15. Vemos qué información nos puede dar este dendrograma sobre el número de clusters, aproximado, en los que podríamos agrupar un conjunto de datos:

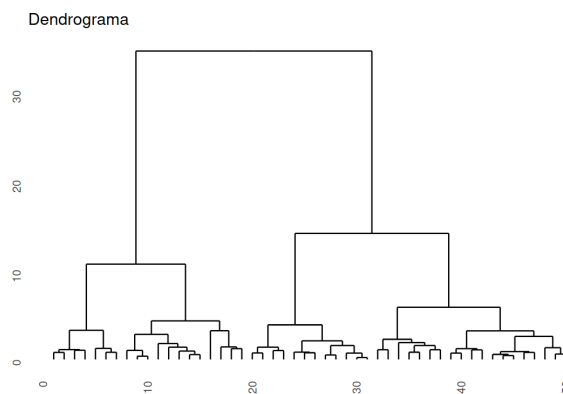


Figure 3.2: Ejemplo de dendrograma (Fuente: [39]).

Aclaremos que en el eje vertical tenemos la *distancia euclídea* que existe entre cada grupo y en el eje horizontal, los datos. Las líneas verticales son los agrupamientos que, a medida que vamos subiendo, van disminuyendo, hasta acabar en uno solo. Observamos que hay una gran separación entre las ramas en un entorno alrededor de 10, por lo que, si trazáramos una línea horizontal a una altura por encima o por debajo de 10, identificaríamos 3 y 4 clusters, respectivamente.

No podemos afirmar nada acerca del número de clusters óptimo con esta información. Habría que utilizar otros métodos, tanto visuales como matemáticos que nos permitan hacer una afirmación acerca de k un poco más rotunda.

Métodos jerárquicos aglomerativos

A continuación, vamos a analizar los principales métodos jerárquicos aglomerativos. Hemos de aclarar que ninguno de los que presentaremos es más eficiente que otro. Esto dependerá de la naturaleza de los datos y el enfoque que tenga el análisis.

Estrategia de la distancia mínima o máxima similaridad

Del inglés *single linkage*, traducido como *encadenamiento simple*, y más conocida como la estrategia del *vecino más cercano* es una estrategia que se basa en combinar dos clusters que tengan distancia mínima (o máxima similaridad), siendo esta la distancia mínima entre sus componentes.

Sean C_i y C_j dos clusters cualesquiera con n_i, n_j elementos, respectivamente, y sean $x_k^i \in C_i, x_m^j \in C_j$, con $k = 1, \dots, n_i, m = 1, \dots, n_j$ dos elementos de dichos clusters. La distancia entre C_i y C_j viene dada por

$$d(C_i, C_j) = \min_{\substack{k=1, \dots, n_i \\ m=1, \dots, n_j}} \{d(x_k^i, x_m^j) \mid x_k^i \in C_i \text{ y } x_m^j \in C_j\}$$

donde $d(x_k^i, x_m^j)$ es la distancia Euclídea, generalmente, o cualquier otra distancia entre los elementos x_k^i, x_m^j de los clusters.

Equivalentemente, en un contexto en el que usemos similaridad, se unirán dos clusters C_i, C_j si

$$s(C_i, C_j) = \max_{\substack{k=1, \dots, n_i \\ m=1, \dots, n_j}} \{s(x_k^i, x_m^j) \mid x_k^i \in C_i \text{ y } x_m^j \in C_j\}$$

Estrategia de la distancia máxima o mínima similaridad

Conocido también como el método del encadenamiento completo, al contrario que en la estrategia anterior, esta se basa en calcular la distancia máxima entre los clusters teniendo en cuenta la distancia máxima entre sus componentes. Esto es, ponemos el foco de atención en los elementos más dispares de los clusters, es decir, los que están a mayor distancia o son menos similares.

Sean entonces C_i y C_j dos clusters cualesquiera con n_i, n_j elementos, respectivamente y sean $x_k^i \in C_i, x_m^j \in C_j$, con $k = 1, \dots, n_i, m = 1, \dots, n_j$ dos elementos de dichos clusters. Entonces, la distancia entre dichos clusters viene dada por:

$$d(C_i, C_j) = \max_{\substack{k=1, \dots, n_i \\ m=1, \dots, n_j}} \{d(x_k^i, x_m^j) \mid x_k^i \in C_i \text{ y } x_m^j \in C_j\}$$

Equivalentemente, si lo miramos desde el punto de vista de la similaridad, se unirán dos clusters C_i, C_j si

$$s(C_i, C_j) = \min_{\substack{k=1, \dots, n_i \\ m=1, \dots, n_j}} \{s(x_k^i, x_m^j) \mid x_k^i \in C_i \text{ y } x_m^j \in C_j\}$$

Estrategia de la distancia o similitud promedio

Mientras que en las dos estrategias anteriores la combinación de dos clusters en uno dependía únicamente de un único par de objetos dentro de cada cluster, y se usaba la distancia máxima o mínima, la estrategia de la distancia promedio calcula la distancia entre dos clusters como el promedio de las disimilaridades en cada cluster. Dependiendo de si consideramos el tamaño de los clusters en el promedio o no, podemos distinguir entre dos estrategias diferentes: la no ponderada y la ponderada.

Consideremos en lo que sigue dos clusters C_i, C_j con n_i, n_j elementos, y x_k^i, x_m^j dos elementos de C_i, C_j , respectivamente con $k = 1, \dots, n_i, m = 1, \dots, n_j$.

- **No ponderada**

Supongamos, sin pérdida de generalidad, que C_i está constituido por dos clusters C_{i_1} y C_{i_2} con n_{i_1}, n_{i_2} elementos respectivamente, la distancia entre C_i y C_j viene dada por

$$d(C_i, C_j) = \frac{d(C_{i_1}, C_j) + d(C_{i_2}, C_j)}{2}$$

Observamos que esta distancia es independiente de los tamaños de los clusters involucrados. Esto quiere decir que las distancias respecto a C_j de C_{i_1} y C_{i_2} tienen el mismo peso.

- **Ponderada**

A diferencia de la anterior estrategia, esta distancia es el promedio ponderado de las distancias de las componentes del cluster con respecto a las del otro. Supongamos que C_i está constituido por dos clusters C_{i_1}, C_{i_2} con n_{i_1}, n_{i_2} elementos tales que $n_i = n_{i_1} + n_{i_2}$ y tomemos los elementos $x_z^{i_1} \in C_{i_1}, x_w^{i_2} \in C_{i_2}$, donde $z = 1, \dots, n_{i_1}$ y $w = 1, \dots, n_{i_2}$. Entonces, la distancia promedio pondera entre C_i, C_j viene dada por

$$d(C_i, C_j) = \frac{\sum_{k=1}^{n_i} \sum_{m=1}^{n_j} d(x_k^i, x_m^j)}{n_i n_j} = \frac{1}{(n_{i_1} + n_{i_2}) n_j} \sum_{k=1}^{n_{i_1} + n_{i_2}} \sum_{m=1}^{n_j} d(x_k^i, x_m^j) =$$

$$\begin{aligned}
 &= \frac{1}{(n_{i_1} + n_{i_2})n_j} \sum_{z=1}^{n_{i_1}} \sum_{m=1}^{n_j} d(x_z^{i_1}, x_m^j) + \frac{1}{(n_{i_1} + n_{i_2})n_j} \sum_{w=1}^{n_{i_2}} \sum_{m=1}^{n_j} d(x_w^{i_2}, x_m^j) = \\
 &= \frac{n_{i_1}}{(n_{i_1} + n_{i_2})n_j n_{i_1}} \sum_{z=1}^{n_{i_1}} \sum_{m=1}^{n_j} d(x_z^{i_1}, x_m^j) + \frac{n_{i_2}}{(n_{i_1} + n_{i_2})n_j n_{i_2}} \sum_{w=1}^{n_{i_2}} \sum_{m=1}^{n_j} d(x_w^{i_2}, x_m^j) = \\
 &= \frac{n_{i_1}}{n_{i_1} + n_{i_2}} d(C_{i_1}, C_j) + \frac{n_{i_2}}{n_{i_1} + n_{i_2}} d(C_{i_2}, C_j) = \\
 &= \frac{n_{i_1} d(C_{i_1}, j) + n_{i_2} d(C_{i_2}, C_j)}{n_{i_1} + n_{i_2}}
 \end{aligned}$$

Método del centroide

En este método, para determinar cuán semejantes son dos clusters, vamos a poner el foco de atención en los *centroides* de los mismos, es decir, en los vectores de medias de las variables observadas sobre los individuos que conforman el cluster. Entonces, los centroides de los clusters C_i, C_j son

$$\bar{x}^i = \frac{\sum_{k=1}^{n_i} x_k^i}{n_i} = \begin{bmatrix} \bar{x}_1^i \\ \bar{x}_2^i \\ \vdots \\ \bar{x}_{n_i}^i \end{bmatrix} \quad \text{y} \quad \bar{x}^j = \frac{\sum_{m=1}^{n_j} x_m^j}{n_j} = \begin{bmatrix} \bar{x}_1^j \\ \bar{x}_2^j \\ \vdots \\ \bar{x}_{n_j}^j \end{bmatrix}$$

respectivamente, y el cuadrado de la distancia Euclídea entre ambos centroides es $d^2(C_i, C_j) = \|\bar{x}^i - \bar{x}^j\|^2$. Se persigue entonces el objetivo de encontrar clusters que hagan mínima la distancia entre sus centroides, para, consecuentemente, combinarlos en un nuevo cluster, que llamaremos C_t con centroide $\bar{x}^t = \frac{(n_i \bar{x}^i + n_j \bar{x}^j)}{n_i + n_j}$. Se calcularían de nuevo las distancias entre dicho centroide y el de los demás clusters. Continuaríamos el proceso hasta obtener un único cluster.

El método del centroide se denomina *método de la mediana* si se utiliza un promedio no ponderado de los centroides: $\bar{x}^t = \frac{(\bar{x}^i + \bar{x}^j)}{2}$. Este método es preferible cuando $n_i \gg n_j$ o viceversa.

Calculamos el cuadrado de la distancia Euclídea entre el cluster C_t y el centroide \bar{x}^u de un tercer cluster C_u de la siguiente forma:

$$d^2(C_t, C_u) = \left(\frac{n_i}{n_i + n_j} \right) d^2(C_i, C_j) + \left(\frac{n_j}{n_i + n_j} \right) d^2(C_j, C_u) - \left(\frac{n_i n_j}{n_i + n_j} \right) d^2(C_i, C_j)$$

Método de Ward

El *método de Ward*, también conocido como *método de la suma incremental de cuadrados*, es un método jerárquico aglomerativo cuya filosofía la podemos resumir de la siguiente forma:

Supongamos que tenemos m elementos que queremos agrupar. Comenzamos con m clusters, cada uno de ellos con un único individuo. A continuación, estudiamos la similitud de dichos clusters y, el par de clusters más similares se combinan, reduciéndose el número de clusters en uno. Se sigue este proceso hasta que todos los clusters estén fusionados. El objetivo de este método es, en definitiva, encontrar en cada iteración, los dos clusters tales que su fusión genere el menor incremento en el valor total de la suma de los cuadrados de las diferencias entre cada elemento y el centroide del cluster.

Para desarrollar formalmente el método, vamos a establecer la siguiente notación:

- Notaremos por x_{ij}^k al valor de la j -ésima variable observada sobre el i -ésimo individuo dentro del k -ésimo cluster. Suponemos que dicho cluster estará formado por n_k individuos.
- Al centroide del k -ésimo cluster lo denotaremos como hicimos en el método del centroide: m^k , con n componentes, m_j^k .
- E_k será la suma de los cuadrados de las distancias (distancia euclídea) de cada individuo del cluster k al centroide (los errores del cluster k). Esto es,

$$E_k = \sum_{i=1}^{n_k} \sum_{j=1}^n (x_{ij}^k - m_j^k)^2 = \sum_{i=1}^{n_k} \sum_{j=1}^n (x_{ij}^k)^2 - n_k \sum_{j=1}^n (m_j^k)^2$$

- Llamaremos E a la suma de los cuadrados de los errores de todos los clusters en su conjunto. Es decir, suponiendo que hubiera h clusters

$$E = \sum_{k=1}^h E_k$$

Observación 3.16. Notemos que al principio del método, $E_k = 0, \forall k \in \{1, \dots, m\}$ pues hay m clusters compuestos por un solo individuo.

En cada etapa, buscaremos los dos clusters cuya fusión minimice el incremento en E .

Supongamos que en la siguiente etapa son C_p y C_q los dos clusters que se fusionan en uno nuevo, C_t . Entonces, el incremento de E , ΔE_{pq} , vendrá dado por

$$\begin{aligned} \Delta E_{pq} &= E_t - E_p - E_q = \\ &= \left[\sum_{i=1}^{n_t} \sum_{j=1}^n (x_{ij}^t)^2 - n_t \sum_{j=1}^n (m_j^t)^2 \right] - \left[\sum_{i=1}^{n_p} \sum_{j=1}^n (x_{ij}^p)^2 - n_p \sum_{j=1}^n (m_j^p)^2 \right] - \left[\sum_{i=1}^{n_q} \sum_{j=1}^n (x_{ij}^q)^2 - n_q \sum_{j=1}^n (m_j^q)^2 \right] = \end{aligned}$$

3 Técnicas multivariantes: fundamentos y desarrollo del análisis clúster

$$\stackrel{n_t=n_p+n_q}{=} n_p \sum_{j=1}^n (m_j^p)^2 + n_q \sum_{j=1}^n (m_j^q)^2 - n_t \sum_{j=1}^n (m_j^t)^2$$

Ahora, por un lado sabemos por el apartado anterior que el centroide de un cluster C_c se calcula como

$$m^c = \frac{\sum_{j=1}^{n_c} x_j^c}{n_c}$$

Aplicando esto al cluster C_p , tenemos:

$$m^p = \frac{\sum_{j=1}^{n_p} x_j^p}{n_p} \iff n_p m^p = \sum_{j=1}^{n_p} x_j^p$$

Para el cluster C_q :

$$m^q = \frac{\sum_{j=1}^{n_q} x_j^q}{n_q} \iff n_q m^q = \sum_{j=1}^{n_q} x_j^q$$

Cuando fusionamos los clusters C_p y C_q , tenemos un cluster C_t de $n_t = n_p + n_q$ elementos, con centroide

$$m^t = \frac{\sum_{j=1}^{n_t} x_j^t}{n_t}$$

Como C_t tiene todos los puntos de C_p y C_q , podemos escribir

$$\sum_{j=1}^{n_t} x_j^t = \sum_{j=1}^{n_p} x_j^p + \sum_{j=1}^{n_q} x_j^q$$

Por lo que, si sustituimos en las expresiones de $n_p m^p$ y $n_q m^q$:

$$\sum_{j=1}^{n_t} x_j^t = n_p m^p + n_q m^q$$

Multiplicando por n_t , obtenemos:

$$n_t m^t = n_p m^p + n_q m^q$$

Como esto es una igualdad vectorial, la igualdad se verifica componente a componente:

$$n_t m_j^t = n_p m_j^p + n_q m_j^q \text{ con } j = 1, \dots, n$$

Partiendo entonces de la igualdad $n_t m_j^t = n_p m_j^p + n_q m_j^q$, elevando al cuadrado ambos miembros obtenemos

$$n_t^2 (m_j^t)^2 = n_p^2 (m_j^p)^2 + n_q^2 (m_j^q)^2 + 2n_p n_q m_j^p m_j^q =$$

3 Técnicas multivariantes: fundamentos y desarrollo del análisis clúster

$$= n_p^2(m_j^p)^2 + n_q^2(m_j^q)^2 + n_p n_q (2m_j^p m_j^q) = n_p^2(m_j^p)^2 + n_q^2(m_j^q)^2 + n_p n_q ((m_j^p)^2 + (m_j^q)^2 - (m_j^p - m_j^q)^2)$$

de donde se obtiene

$$n_t^2(m_j^t)^2 = n_p(n_p + n_q)(m_j^p)^2 + n_q(n_p + n_q)(m_j^q)^2 - n_p n_q (m_j^p - m_j^q)^2$$

Dividiendo por n_t^2 se obtiene

$$(m_j^t)^2 = \frac{n_p}{n_t}(m_j^p)^2 + \frac{n_q}{n_t}(m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2$$

Por tanto, si sustituimos esta expresión en la del incremento, ΔE_{pq} , nos queda lo siguiente:

$$\begin{aligned} \Delta E_{pq} &= n_p \sum_{j=1}^n (m_j^p)^2 + n_q \sum_{j=1}^n (m_j^q)^2 - n_t \sum_{j=1}^n \left[\frac{n_p}{n_t}(m_j^p)^2 + \frac{n_q}{n_t}(m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2 \right] \iff \\ \iff \Delta E_{pq} &= n_p \sum_{j=1}^n (m_j^p)^2 + n_q \sum_{j=1}^n (m_j^q)^2 - n_p \sum_{j=1}^n (m_j^p)^2 - n_q \sum_{j=1}^n (m_j^q)^2 + \frac{n_p n_q}{n_t} \sum_{j=1}^n (m_j^p - m_j^q)^2 \\ \Rightarrow \Delta E_{pq} &= \frac{n_p n_q}{n_t} \sum_{j=1}^n (m_j^p - m_j^q)^2 \end{aligned}$$

Obtenemos por consiguiente que el menor incremento de los errores cuadráticos es proporcional al cuadrado de la distancia euclídea de los centroides de los clusters fusionados.

Veamos cómo calcular los incrementos a partir de otros ya previamente calculados. Sea C_t el cluster que resulta de fusionar C_p y C_q y consideremos ahora otro cluster C_r distinto a los demás. Entonces, por lo visto anteriormente, el incremento en E que se produce al fusionar C_r con C_t , viene dado por

$$\Delta E_{rt} = \frac{n_r n_t}{n_r + n_t} \sum_{j=1}^n (m_j^r - m_j^t)^2$$

Como $n_t m^t = n_p m^p + n_q m^q$ y $n_t = n_p + n_q$, y sabemos también que $(m_j^t)^2 = \frac{n_p}{n_t}(m_j^p)^2 + \frac{n_q}{n_t}(m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2$, deducimos que

$$\begin{aligned} (m_j^r - m_j^t)^2 &= (m_j^r)^2 + (m_j^t)^2 - 2m_j^r m_j^t = \\ &= (m_j^r)^2 + \frac{n_p}{n_t}(m_j^p)^2 + \frac{n_q}{n_t}(m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2 - 2m_j^r \frac{n_p m_j^p + n_q m_j^q}{n_t} = \end{aligned}$$

$$\begin{aligned}
 &= \frac{n_p(m_j^r)^2 + n_q(m_j^q)^2}{n_t} + \frac{n_p}{n_t}(m_j^p)^2 + \frac{n_q}{n_t}(m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2 - 2m_j^r \frac{n_p m_j^p + n_q m_j^q}{n_t} = \\
 &= \frac{n_p}{n_t}(m_j^r - m_j^p)^2 + \frac{n_q}{n_t}(m_j^r - m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2
 \end{aligned}$$

Obtenemos por tanto,

$$\begin{aligned}
 \Delta E_{rt} &= \frac{n_r n_t}{n_r + n_t} \sum_{j=1}^n (m_j^r - m_j^t)^2 = \\
 &= \frac{n_r n_t}{n_r + n_t} \sum_{j=1}^n \left[\frac{n_p}{n_t}(m_j^r - m_j^p)^2 + \frac{n_q}{n_t}(m_j^r - m_j^q)^2 - \frac{n_p n_q}{n_t^2}(m_j^p - m_j^q)^2 \right] = \\
 &= \frac{n_r n_p}{n_r + n_t} \sum_{j=1}^n (m_j^r - m_j^p)^2 + \frac{n_q n_r}{n_r + n_t} \sum_{j=1}^n (m_j^r - m_j^q)^2 - \frac{n_r n_p n_q}{n_t(n_r + n_t)} (m_j^p - m_j^q)^2 = \\
 &= \frac{1}{n_r + n_t} \sum_{j=1}^n \left[n_r n_p (m_j^r - m_j^p)^2 + n_r n_q (m_j^r - m_j^q)^2 - \frac{n_r n_p n_q}{n_p + n_q} (m_j^p - m_j^q)^2 \right] = \\
 &= \frac{1}{n_r + n_t} [(n_r + n_p) \Delta E_{rp} + (n_r + n_q) \Delta E_{rq} - n_r \Delta E_{pq}]
 \end{aligned}$$

Nota 2. Se puede demostrar que esta relación no depende de la forma específica en que se mide la distancia, siempre que esta se defina a partir de una norma inducida por un producto escalar o que satisfaga la ley del paralelogramo.

Ofrecemos a continuación un pequeño ejemplo de aplicación de este método aglomerativo, de elaboración propia aunque basado en la fuente bibliográfica [32].

Ejemplo 3.17. Consideremos los siguientes datos de 5 genotipos sobre los que se observan dos variables X_1 y X_2 .

Genotipo	X_1	X_2
G_1	7	10
G_2	8	10
G_3	6	5
G_4	3	2
G_5	11	10

Table 3.1: Ejemplo método de Ward

En la primera iteración del método, contamos con 5 clusters, cada uno compuesto por un único punto (cada punto de nuestro dataset en un cluster):

A continuación, barajamos todas las parejas que podemos formar con estos 5 elementos, para estudiar qué pareja de clusters generaría el menor ΔE al fusionarse. Estudiemos entonces las $\binom{5}{2} = 10$ combinaciones posibles.

3 Técnicas multivariantes: fundamentos y desarrollo del análisis clúster

Partición	Centroides	E_k	E	ΔE
G_1, G_2, G_3, G_4, G_5	los propios puntos	$E_{G_1} = E_{G_2} = E_{G_3} = E_{G_4} = E_{G_5} = 0$	0	0

Table 3.2: Nivel 0

Partición	Centroides	E_k	E	ΔE
$(G_1, G_2), G_3, G_4, G_5$	$C_{G_1G_2} = (7.5, 10)$	$E_{G_1G_2} = 0.5$	0.5	0.5
$(G_1, G_3), G_2, G_4, G_5$	$C_{G_1G_3} = (6.5, 7.5)$	$E_{G_1G_3} = 13$	13	13
$(G_1, G_4), G_2, G_3, G_5$	$C_{G_1G_4} = (5, 6)$	$E_{G_1G_4} = 40$	40	40
$(G_1, G_5), G_2, G_3, G_4$	$C_{G_1G_5} = (9, 10)$	$E_{G_1G_5} = 8$	8	8
$(G_2, G_3), G_1, G_4, G_5$	$C_{G_2G_3} = (7, 7.5)$	$E_{G_2G_3} = 14.5$	14.5	14.5
$(G_2, G_4), G_1, G_3, G_5$	$C_{G_2G_4} = (5.5, 6)$	$E_{G_2G_4} = 44.5$	44.5	44.5
$(G_2, G_5), G_1, G_3, G_4$	$C_{G_2G_5} = (9.5, 10)$	$E_{G_2G_5} = 4.5$	4.5	4.5
$(G_3, G_4), G_1, G_2, G_5$	$C_{G_3G_4} = (4.5, 3.5)$	$E_{G_3G_4} = 9$	9	9
$(G_3, G_5), G_1, G_2, G_4$	$C_{G_3G_5} = (8.5, 7.5)$	$E_{G_3G_5} = 25$	25	25
$(G_4, G_5), G_1, G_2, G_3$	$C_{G_4G_5} = (7, 6)$	$E_{G_4G_5} = 64$	64	64

Table 3.3: Nivel 1

Observamos que se han de fusionar los elementos G_1 y G_2 . En este nivel, la configuración de clusters es $(G_1, G_2), G_3, G_4, G_5$.

A continuación, calculemos los incrementos que daría lugar las $\binom{4}{2} = 6$ posibles combinaciones.

Partición	Centroides	E_k	E	ΔE
$(G_1, G_2, G_3), G_4, G_5$	$C_{G_1G_2G_3} = (7, 8.33)$	$E_{G_1G_2G_3} = 18.67$ $E_{G_4} = E_{G_5} = 0$	18.67	18.17
$(G_1, G_2, G_4), G_3, G_5$	$C_{G_1G_2G_4} = (6, 7.33)$	$E_{G_1G_2G_4} = 56.67$ $E_{G_3} = E_{G_5} = 0$	56.67	56.17
$(G_1, G_2, G_5), G_3, G_4$	$C_{G_1G_2G_5} = (8.67, 10)$	$E_{G_1G_2G_5} = 8.67$ $E_{G_3} = E_{G_4} = 0$	8.67	8.17
$(G_3, G_4), (G_1, G_2), G_5$	$C_{G_3G_4} = (4.5, 3.5)$ $C_{G_1G_2} = (7.5, 10)$	$E_{G_3G_4} = 9$ $E_{G_1G_2} = 0.5, E_{G_5} = 0$	9	8.5
$(G_3, G_5), (G_1, G_2), G_4$	$C_{G_3G_5} = (8.5, 7.5)$	$E_{G_3G_5} = 25$ $E_{G_1G_2} = 0.5, E_{G_4} = 0$	25.5	25
$(G_4, G_5), (G_1, G_2), G_3$	$C_{G_4G_5} = (7, 6)$	$E_{G_4G_5} = 64$ $E_{G_1G_2} = 0.5, E_{G_3} = 0$	64.5	64

Table 3.4: Nivel 2

Inferimos de esta iteración que el genotipo G_5 deberá fusionarse con G_1 y G_2 . De esta forma, nos queda la siguiente configuración: $(G_1, G_2, G_5), G_3, G_4$.

Consideremos entonces las $\binom{3}{2} = 3$ posibles combinaciones que podemos hacer con los elementos de la configuración que se nos ha quedado:

3 Técnicas multivariantes: fundamentos y desarrollo del análisis clúster

Partición	Centroides	E_k	E	ΔE
$(G_1, G_2, G_5, G_3), G_4$	$C_{G_1 G_2 G_3 G_5} = (8, 8.75)$	$E_{G_1 G_2 G_3 G_5} = 32.75$ $E_{G_4} = 0$	32.75	24.08
$(G_1, G_2, G_5, G_4), G_3$	$C_{G_1 G_2 G_4 G_5} = (7.25, 8)$	$E_{G_1 G_2 G_4 G_5} = 80.75$ $E_{G_3} = 0$	80.75	72.08
$(G_1, G_2, G_5), (G_3, G_4)$	$C_{G_1 G_2 G_5} = (8.67, 10)$ $C_{G_3 G_4} = (4.5, 3.5)$	$E_{G_1 G_2 G_5} = 8.67$ $E_{G_3 G_4} = 9$	17.67	9

Table 3.5: Nivel 3

Observamos que los elementos G_3 y G_4 se han de combinar en un cluster independiente. Es por ello que en este nivel la configuración de clusters que resulta es $(G_1, G_2, G_5), (G_3, G_4)$.

En la siguiente iteración, la única posibilidad es que todos los elementos estén en un único cluster: $(G_1, G_2, G_5, G_3, G_4)$:

Partición	Centroides	E_k	E	ΔE
$(G_1, G_2, G_3, G_4, G_5)$	$C_{G_1 G_2 G_3 G_4 G_5} = (7, 7.4)$	$E_{G_1 G_2 G_3 G_4 G_5} = 89.2$	89.2	71.53

Table 3.6: Nivel 4

Finalmente, presentamos el dendrograma que resulta de haber aplicado el método de Ward a los datos de partida:

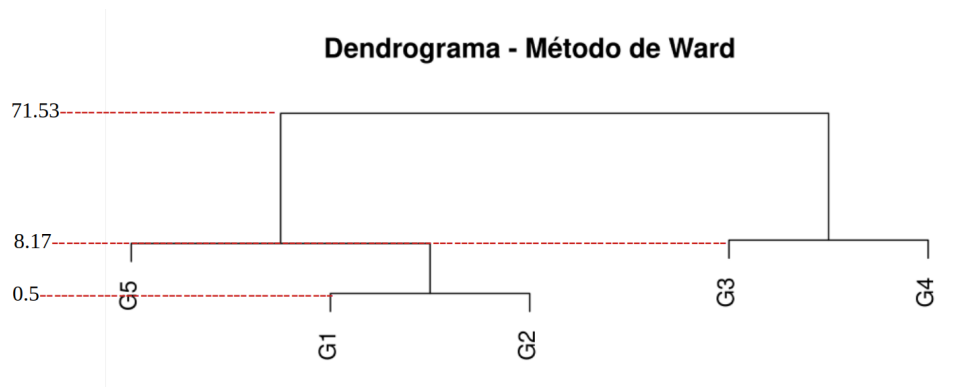


Figure 3.3: Dendrograma resultante (Fuente: elaboración propia).

Métodos disociativos

Como ya apuntamos al comienzo de la subsección, un método jerárquico *disociativo* comienza con un único cluster constituido por todos los individuos de nuestro conjunto de datos y lo divide en dos clusters. Sucesivamente, en cada iteración, uno de los clusters se divide en dos subclusters hasta conseguir un cluster por elemento.

Aunque la forma de calcular la distancia entre clusters es igual que en los aglomerativos, la diferencia está en que como se parte de un único cluster, se va a intentar maximizar las distancias o, equivalentemente, minimizar las similitudes. En definitiva, lo que se persigue es separar los elementos más disímiles en clusters diferentes.

Los métodos disociativos son generalmente de dos clases: *monotéticos* y *politéticos*. En un enfoque monotético, la división de un grupo en dos subgrupos se basa en una sola variable y resultan útiles cuando las variables son de tipo binario, mientras que el enfoque politético utiliza las p variables para realizar la división.

Tienen la misma desventaja que los aglomerativos: una vez realizada la partición, un elemento no puede trasladarse a otro grupo al que no pertenece en el momento de la partición. En cambio, si se buscan grupos más grandes, a veces se prefiere el enfoque disociativo al aglomerativo, en el que los grupos más grandes se alcanzan solo tras un gran número de uniones de grupos más pequeños. A pesar de esto, son métodos mucho menos conocidos que los aglomerativos, de ahí que no haya tanta bibliografía acerca de ellos.

Un aspecto fundamental en su aplicación es determinar el momento adecuado para parar la subdivisión de un cluster y empezar a fragmentar otro. Esto lo resuelve la variante del método propuesta por MacNaughton-Smith en el año 1964, diseñada especialmente para medidas de asociación positivas, y que presentamos formalmente a continuación:

Sea C un cluster inicial. Definimos un subconjunto $S \subset C$ como el grupo fragmentado, y su complemento $R = C \setminus S$ como el resto del cluster. El proceso de división consistirá en evaluar, para cada elemento $x \in R$, la diferencia entre su disimilaridad promedio con los elementos de R y su disimilaridad promedio con los elementos de S .

La distancia promedio de un elemento x respecto a un conjunto A viene dada por:

$$d(x, A) = \frac{1}{n_A} \sum_{y \in A} d(x, y)$$

donde $d(x, y)$ representa la medida de disimilaridad entre los elementos x e y .

Para cada $x \in R$, calculamos:

$$\Delta(x) = d(x, R) - d(x, S)$$

Si existe un $x' \in R$ tal que $\max_{x \in R} \Delta(x) > 0$, entonces x' se traslada a S . En caso contrario, el procedimiento se detiene y la división se completa.

Nota 3. Podemos inicializar el conjunto fragmentado S con el elemento $x_0 \in C$ que tenga la mayor disimilaridad promedio respecto a los demás elementos del cluster.

3.2.5 Métodos no jerárquicos

La información recogida en esta subsección y en las sucesivas subsubsecciones se ha obtenido principalmente de las fuentes bibliográficas [26, 32, 37].

Los *métodos no jerárquicos* para el análisis cluster, también conocidos como métodos de partición, simplemente dividen los datos en un número predeterminado k de clusters, donde no existe una relación jerárquica entre la solución con k clusters y la de $(k + 1)$, es decir, la solución con k clusters no es el paso previo para la solución con $(k + 1)$. Esta es la principal diferencia con respecto a los jerárquicos, donde el número de clusters era desconocido a priori.

En definitiva, dado un número k , buscamos particionar los datos en k clusters de modo que los elementos dentro de cada cluster sean similares entre sí, mientras que los elementos de clusters distintos sean bastante diferentes.

Se caracterizan por tener una estructura plana, es decir, no implican la construcción de una estructura jerárquica en forma de árbol, a diferencia de los métodos jerárquicos. En su lugar, los elementos se asignan a los clusters directamente, teniendo en cuenta el número predefinido de clusters, k . No se sigue un proceso progresivo de combinación o separación de grupos, como ocurre en los jerárquicos.

Nota 4. Los algoritmos no jerárquicos son mucho más eficientes que los jerárquicos en términos computacionales.

Método *k-means*

El algoritmo *k-means*, propuesto por MacQueen en el año 1967, es el método más popular entre los no jerárquicos. Debido a su extrema eficiencia, se usa con frecuencia en proyectos de grandes dimensiones. Se denomina *k-means* (k-medias) porque asigna cada individuo al cluster cuyo centroide esté más próximo, de entre k clusters prefijados. Dicho centroide se calcula a partir de los elementos que conforman el cluster después de cada asignación, en lugar de hacerlo al final de cada ciclo, como ocurre en otros métodos, lo cual es un aspecto clave de este algoritmo.

Cabe resaltar que este algoritmo necesita acceso a los datos originales y permite mover los elementos de un cluster a otro, una reasignación que no está disponible en los métodos jerárquicos.

Veamos cómo funciona el algoritmo:

Sea $\mathcal{L} = \{x_i / i = 1, \dots, n\}$ el conjunto de puntos que conforman nuestro conjunto de datos y sea k el número de clusters. Elegimos k elementos que nos servirán de

semillas, las cuales se reemplazarán por los centroides de los clusters en el transcurso del algoritmo. Hay varias formas de elegir dichas semillas:

- (1) Seleccionando k elementos al azar, aunque quizás separados por una mínima distancia.
- (2) Eligiendo los k primeros elementos del conjunto de datos, sujetos a una distancia mínima.
- (3) Seleccionando los k puntos más alejados entre sí en base a una medida de disimilaridad.

entre otras opciones.

Una vez elegidas las semillas, cada elemento de nuestro conjunto de datos se asignará al cluster que tenga la semilla más cercana, en términos de distancia euclídea. Una vez que un cluster pase a tener más de un elemento, se reemplaza la semilla por su centroide. Cuando todos los elementos hayan sido asignados, entonces nos planteamos si cada elemento está más cerca del centroide del cluster donde está asignado o del de otro cluster. Esto es, calculamos el Error Cuadrático Medio (ECM) de cada observación con respecto al centroide de su clúster actual y el resto de centroides:

$$ECM = \sum_{j=1}^k \sum_{i \in C_j} (x_{i_{C_j}} - \bar{x}_j)' (x_{i_{C_j}} - \bar{x}_j)$$

donde \bar{x}_j es el centroide del j -ésimo cluster y C_j es el cluster que contiene a $x_{i_{C_j}}$.

Si está más cerca del centroide de otro cluster, dicho elemento se traslada a dicho cluster y se recalculan los centroides de ambos. El proceso continúa hasta que no se puedan hacer más mejoras en términos de error cuadrático medio (se persigue minimizar el ECM).

Cabe destacar que este método es bastante sensible a la elección de las semillas al comienzo del mismo. Se aconseja probar con distintas semillas y analizar los resultados: si con diferentes elecciones de semillas se dan resultados muy diferentes, o se requieren demasiadas iteraciones para estabilizarse (convergencia lenta), podemos sospechar que los datos no forman grupos bien definidos de forma natural.

También puede usarse como mejora de los métodos jerárquicos: se agrupan primero los elementos mediante un método jerárquico y luego, utilizando como semillas los centroides de estos grupos aplicamos el algoritmo. De esta forma, podremos reasignar los puntos de un cluster a otro, en caso de necesitarlo. Suplimos así el handicap de los aglomerativos.

Ofrecemos a continuación un ejemplo con el que ilustraremos el funcionamiento del método. Este ejemplo es de elaboración propia, aunque basado en el ejemplo ilustrativo del método k -means recogido en la fuente bibliográfica [40].

Ejemplo 3.18. Supongamos que tenemos 8 genotipos, G_1, G_2, \dots, G_8 , sobre los que observamos la expresión de dos genes, A y B :

Genotipo	A	B
G_1	2.5	2.7
G_2	5.1	4.9
G_3	2.8	3.2
G_4	3.0	2.4
G_5	8.3	3.3
G_6	4.6	6.1
G_7	9.3	4.6
G_8	5.3	3.8

Table 3.7: Ejemplo método k -means

Queremos aplicar el método no jerárquico k -means para $k = 3$.

Comenzamos eligiendo por semillas 3 elementos aleatorios, $g_i = (a_i, b_i)$, $i = 1, \dots, 8$, de entre nuestro conjunto de datos. Estos serán los primeros centroides:

$$\bar{g}_{C_1} = (3.0, 2.4) \quad \bar{g}_{C_2} = (8.3, 3.3) \quad \bar{g}_{C_3} = (4.6, 6.1)$$

Calculamos las distancias de todos los puntos de nuestro dataset a estos centroides y obtenemos:

g_i	C_1	C_2	C_3	Cluster más cercano
g_1	0.5830952	5.830952	3.996248	C_1
g_2	3.2649655	3.577709	1.300000	C_3
g_3	0.8246211	5.500909	3.413210	C_1
g_4	0.0000000	5.375872	4.031129	C_1
g_5	5.3758720	0.000000	4.640043	C_2
G_6	4.0311289	4.640043	0.000000	C_3
G_7	6.6730802	1.640122	4.933559	C_2
G_8	2.6925824	3.041381	2.404163	C_3

Table 3.8: Cálculo distancias respecto centroides

A continuación, como todos los clusters han pasado de tener más de un elemento, $C_1 = \{g_1, g_3, g_4\}$, $C_2 = \{g_5, g_7\}$, $C_3 = \{g_2, g_6, g_8\}$, recalculamos los centroides:

$$\bar{g}_{C_1} = (2.767, 2.767) \quad \bar{g}_{C_2} = (8.80, 3.95) \quad \bar{g}_{C_3} = (5.0, 4.9333)$$

Calculamos de nuevo las distancias de cada punto a los nuevos centroides para ver si hay que hacer alguna reasignación de algún punto a otro cluster:

g_i	C_1	C_2	C_3	Cluster más cercano
g_1	0.2748737	6.422811	3.3522573	C_1
g_2	3.1615749	3.820013	0.1053987	C_3
g_3	0.4346136	6.046693	2.8007729	C_1
g_4	0.4346134	6.003541	3.2276321	C_1
g_5	5.5589767	0.820061	3.6820740	C_2
G_6	3.8042375	4.718315	1.2333649	C_3
G_7	6.7856876	0.820061	4.3128980	C_2
G_8	2.7359744	3.503213	1.1723348	C_3

Table 3.9: Cálculo distancias respecto centroides

Observamos que la asignación de puntos a clusters se mantiene constante con respecto al paso previo. Esto quiere decir que no es necesaria ninguna reasignación y, por tanto, que las asignaciones son definitivas. La agrupación la podemos ver en la figura 3.4.

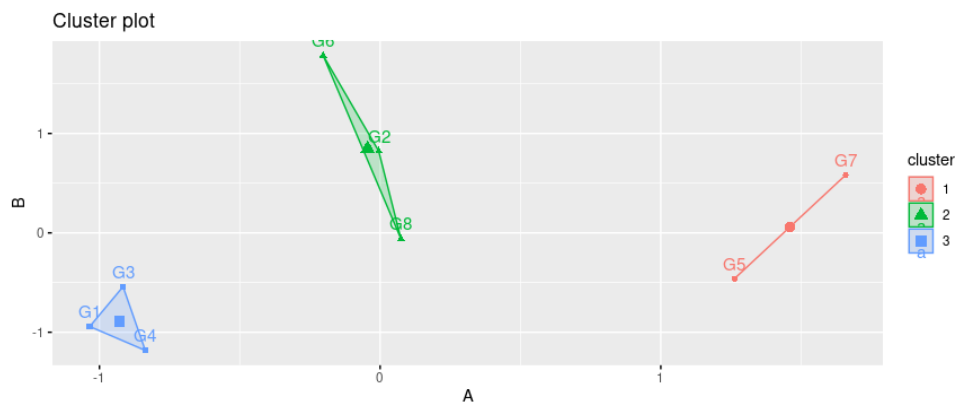


Figure 3.4: Gráfico resultado final k-means. (Fuente: elaboración propia).

Por último, supongamos que hay elementos en nuestro conjunto de datos que podríamos catalogar como *atípicos*, es decir, supongamos que tenemos *outliers*. Estos valores, al ser muy dispares respecto a los demás, van a tener una gran influencia en el cálculo de los centroides, lo cual va a condicionar la agrupación. Esto lo podemos solucionar con el método *k-medoids*, que presentamos a continuación.

Método *k-medoids*

El algoritmo *k-medoids* se propone como una variante del método *k-means* que es mucho más robusta a ruidos y outliers. En lugar de usar el centroide como centro de un grupo, este algoritmo usa un punto real del grupo para representarlo; el *medoide*, que es el objeto ubicado más centralmente en el grupo, y que minimiza la suma de disimilaridades respecto a los demás puntos, en lugar de la suma de los cuadrados de las distancias Euclideas [41]. Aquí es donde se refleja la robustez del método ante

datos anómalos como los outliers. Pese a las ventajas evidentes que tiene el método, se ha visto que para conjuntos de datos muy grandes, la eficiencia del método se ve disminuida.

El algoritmo comienza con la matriz de proximidad $D = (d_{ij})$, donde $d_{ij} = d(x_i, x_j)$, con d una medida de disimilaridad, que nos la pueden dar de antemano o bien la podemos calcular a partir del conjunto de datos, y una configuración inicial de los elementos en k clusters. Usando la matriz D , podemos encontrar el elemento de cada cluster que minimiza la disimilaridad total con respecto a todos los elementos de su cluster (el medoide). Esto es, dado un cluster C_m , $m \in \{1, \dots, k\}$, se busca encontrar un elemento \bar{x}_m que haga mínima la suma $\sum_{i \in C_m} d(x_{i_{C_m}}, \bar{x}_m)$.

Una vez hallados los medoides de los k clusters, observamos la distancia de cada punto a los distintos medoides. Si existe un cluster j de entre los k que había prefijados que tiene algún elemento x_j que está más cerca de otro medoide, entonces este elemento se reasigna al cluster de dicho medoide. Esta reasignación favorece a la minimización de la función objetivo:

$$d_{medoid} = \sum_{j=1}^k \sum_{i \in C_j} d(x_{i_{C_j}}, \bar{x}_j)$$

A continuación se relocalizan medoides y se vuelven a reasignar los elementos. Se repite el proceso hasta que no haya reasignaciones que reduzcan el valor de la función objetivo.

Determinación del número de clusters óptimo

Los métodos no jerárquicos agrupan los datos en un número prefijado de clusters, k . Hay varios métodos para determinar dicho número clusters de forma óptima. A continuación describimos tres de los métodos más usados: el *método de Elbow* y el *método de Silhouette*.

Método de Elbow

Teniendo en cuenta que el objetivo que se persigue a la hora de agrupar los datos en k clusters es obtener dichos cluster de forma que se minimice la varianza total intra-clusters (wss), es decir, la suma de los errores cuadráticos para cada k , podemos determinar el número de clusters óptimo de la siguiente forma:

Para distintos valores de k ($k = 1, 2, \dots, 10, \dots, 15, \dots$) aplicamos un algoritmo de clustering. Para cada número de clusters con el que hemos probado el algoritmo, calculamos la varianza total intra-clusters. Obtenemos así una serie de puntos (k_i, wss_{k_i}) , cluster-wss. Graficando dichos puntos, obtenemos una curva de la wss en función del número de clusters, k .

El punto en el que la curva presente un ‘codo’, se considera indicador del número de clusters óptimo. Este punto es justo aquel en el que las varianzas intra-clusters pasan de disminuir rápidamente a empezar a disminuir de forma lineal conforme aumenta el número de clusters.

Ejemplo 3.19. Si aplicamos este método al ejemplo de los genotipos usado para ilustrar el método k -means, obtenemos la siguiente curva:

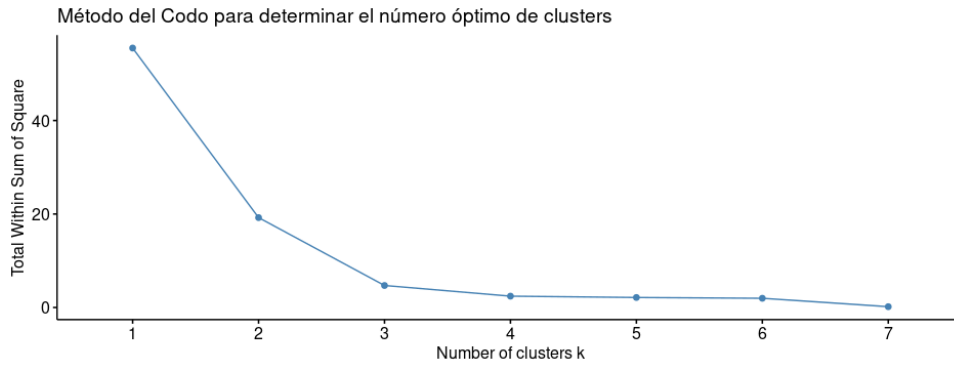


Figure 3.5: Método de Elbow aplicado al ejemplo del método k -means (Fuente: elaboración propia).

Observamos que el codo está en $k = 3$, justo el número de clusters que habíamos prefijado para aplicar el método k -means. Por tanto, fuimos bastante acertados.

Método de la Silueta

Este enfoque para la determinación del número óptimo de clusters, también conocido como *silueta promedio*, mide la calidad de un cluster, es decir, cómo de adecuado es un objeto dentro de un cluster o, equivalentemente, cómo de bien están separados los clusters. De acuerdo con este método, el número óptimo de clusters, k , vendrá dado por el valor k que maximice la silueta promedio, la cual se calcula de la forma que presentamos a continuación:

Supongamos que nos dan una partición de nuestro conjunto de datos en k clusters, \mathcal{C}_{\parallel} . Sea C_i el cluster que contiene al i -ésimo elemento. Sea a_i la distancia promedio del i -ésimo elemento al resto de objetos del mismo cluster C_i . Sea también C_j otro cluster, con $j \neq i$, y consideremos la distancia promedio del i -ésimo elemento a todos los de C_j , $d(i, C_j)$. Calculamos a continuación $d(i, C_j)$, para todos los clusters distintos de C_i . Sea entonces el mínimo de dichas distancias, $b_i = \min_{C_j \neq C_i} d(i, C_j)$. El i -ésimo valor de silueta promedio viene dado por:

$$s_i(\mathcal{C}_k) = s_{ik} = \frac{b_i - a_i}{\max(a_i, b_i)}$$

de manera que $s_{ik} \in [-1, 1]$.

Haciendo el promedio de las siluetas promedio para cada punto del conjunto de datos, obtenemos la *silueta promedio*, \bar{s}_k . Calculando esta silueta promedio para distintos valor de k , obtenemos una serie de puntos (k, \bar{s}_k) que podremos representar. El valor de k para el cual \bar{s}_k sea máximo, determina el número óptimo de clusters.

Ejemplo 3.20. Si aplicamos este método al ejemplo de los genotipos usado para ilustrar el método k -means, obtenemos la siguiente gráfica de puntos (k, \bar{s}_k) :

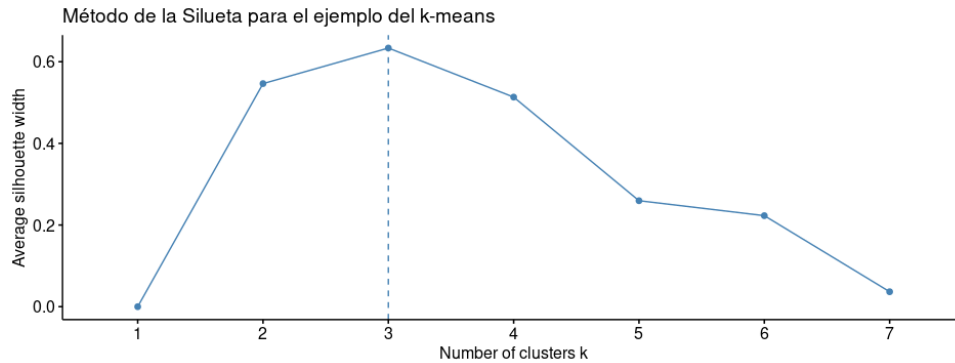


Figure 3.6: Método de la Silueta aplicado al ejemplo del método k -means (Fuente: elaboración propia).

Observamos que el máximo de la silueta promedio se alcanza en $k = 3$, como era de esperar.

Parte III

FUNDAMENTOS INFORMÁTICOS

4 Proyecto GEO (Gene Expression Omnibus)

La información recogida en este capítulo dedicado a la base de datos GEO ha sido extraída principalmente de las fuentes bibliográficas [42, 43, 44, 45, 46].

Gene Expression Omnibus (GEO) es una base de datos pública e internacional gestionada por el Centro Nacional de Información Biotecnológica (NCBI) de la Biblioteca Nacional de Medicina (NLM) de EE.UU. Está diseñada para almacenar y distribuir de manera libre conjuntos de datos relacionados con la expresión génica y otros estudios genómicos funcionales. Acepta tanto datos crudos como procesados, siempre con una descripción detallada del diseño experimental, las características de las muestras y las metodologías utilizadas en los estudios de genómica de alto rendimiento.

Fue lanzada en el año 2000 debido al rápido crecimiento de los datos de expresión génica generados por tecnologías como los microarrays y la secuenciación de ADN. GEO ofrece una estructura abierta y flexible, que permite enviar, almacenar y acceder a conjuntos de datos muy variados, procedentes de técnicas como RNA-seq, single-cell RNA-seq (scRNA-seq), arrays de proteínas o tejidos, RT-PCR, estudios de metilación del genoma, análisis de variaciones en el número de copias (CNV) y más. Aunque aproximadamente el 90% de los datos en GEO corresponden a estudios de expresión génica, su alcance se ha expandido con el tiempo. Actualmente, recibe datos de alrededor de 72 países, consolidándose como un recurso global, gratuito y de acceso público.

Aunque no tiene como objetivo reemplazar bases de datos internas especializadas en expresión génica, GEO complementa estas fuentes actuando como un repositorio terciario¹ y centralizado. Su función principal es ofrecer una plataforma donde se difunden datos provenientes de múltiples estudios y fuentes, facilitando el acceso a información crucial para la investigación. Ofrece herramientas para la consulta, visualización y análisis de estos datos directamente desde su página web, lo que hace que sea fácil de usar incluso para investigadores sin experiencia en bioinformática o software especializado.

Además, GEO sigue las normas establecidas por la *Functional Genomics Data Society*, adhiriéndose a las directrices MIAME (*Minimum Information About a Microarray Experiment*) y MINSEQE (*Minimum Information about a high-throughput SEQuencing Experiment*), lo que asegura que los datos sean interpretables y reproducibles por la comunidad científica. Estas directrices definen los requisitos mínimos de información que debe acompañar a cada experimento, permitiendo su correcta interpretación, re-

¹Un repositorio terciario se refiere a un sistema que integra y organiza datos de diversas fuentes primarias y secundarias, sirviendo como un centro de distribución más amplio y accesible.

utilización y comparación con otros estudios.

En lo que sigue, iremos detallando los tres objetivos principales que persigue GEO y que son:

1. Proporcionar una base de datos robusta y verátil para almacenar de manera eficiente datos genómicos funcionales de alto rendimiento (véase Organización de los datos).
2. Ofrecer prodemientos y formatos de envío sencillos que permitan a la comunidad científica depositar datos completos y bien anotados (véase Envío de datos).
3. Facilitar mecanismos intuitivos que permitan a los usuarios consultar, localizar, revisar y descargar estudios y perfiles de expresión génica de interés (véase Navegación, descarga y consulta).

4.1 Organización de los datos

La base de datos GEO almacena una amplia variedad de experimentos genómicos a gran escala, los cuales generan datos en muchos formatos, tipos de archivos y contenido. Esto presenta un gran desafío en cuanto a manejo y consulta de la información.

Es por ello por lo que ha sido diseñada con una estructura que permite adaptarse a esta diversidad de datos. Cabe señalar que los datos que vienen en tablas no se almacenan de forma complementamente estructurada dentro de la base de datos central, sino que se guardan como tablas de texto plano, delimitadas por tabulaciones, sin límites en cantidad de filas y columnas.

Los registros que hay almacenados en GEO, todos ellos proporcionados por los investigadores (*submitters*) se organizan de la siguiente forma:

- *Platforms*: son registros compuestos por:
 - Una descripción general del array o del secuenciador.
 - En el caso de las basadas en arrays, incluye además una tabla de datos que define la plantilla del array.

Cada registro Platform recibe un número de acceso único y estable en GEO, con el prefijo *GPLxxx* (por ejemplo, *GPL570*).

Puede estar referenciada por muchas muestras (*Samples*), las cuales pueden haber sido enviadas por múltiples remitentes (investigadores).

- *Samples*: una muestra describe:
 - Las condiciones bajo las cuales se manejó una muestra individual.

- Las manipulaciones a las que fue sometida.
- Un archivo original de datos en crudo o archivo de datos de secuenciación, procesados.

Al igual que antes, cada muestra tiene un Id asociado que comienza por *GSMxxx*.

Una muestra debe referenciar a una y solo una Platform, que debe estar previamente definida, pero puede estar incluida en múltiples Series.

- *Series*: un registro en serie:
 - Vincula un grupo de muestras relacionadas (Samples).
 - Sirve como punto central y descriptivo del estudio completo, ofreciendo una descripción en texto del experimento en su conjunto.
 - Incluye un archivo comprimido (.tar) que contiene los archivos originales de datos en crudo o los archivos de datos de secuenciación procesados.

Cada serie recibe un identificador único que empieza por *GSExxx*.

Como la estructura de la base de datos se organiza en torno a entidades bien definidas, como Platform, Sample y Serie, que poseen atributos específicos y mantienen relaciones entre sí con cardinalidades reconocibles, es lógico pensar que el diseño conceptual de GEO está basado en un modelo entidad-relación y, por ende, el lógico basado en el modelo relacional, que se obtiene a partir del conceptual mediante paso a tablas.

Cabe señalar que los archivos de datos en crudo que acompañan a los registros se almacenan en un servidor FTP¹ y están enlazados desde cada entrada.

En base a toda esta información acerca de las entidades involucradas y sus relaciones, podemos construir un posible diagrama E/R para esta base de datos:

Por otra parte, los datos de *Platform*, *Sample* y *Serie* que envían los investigadores a GEO son bastante diversos en cuanto a formato, contenido y nivel de detalle con el que se describen los experimentos. Sin embargo, a pesar de estas diferencias, todas las presentaciones de estudios de expresión génica basados en arrays comparten una serie de elementos comunes:

- Información para identificar la secuencia de cada elemento presente en la *Platform*. Es decir, cada sonda o fragmento de secuencia está asociado a una secuencia de nucleótidos específica, lo que permite localizarla de forma única y vincularla a un gen o región concreta del genoma.
- Mediciones de expresión normalizadas incluidas en las tablas de *Sample*.

¹FTP (*File Transfer Protocol*) es un servicio utilizado para el envío y obtención de archivos entre dos equipos remotos [47].

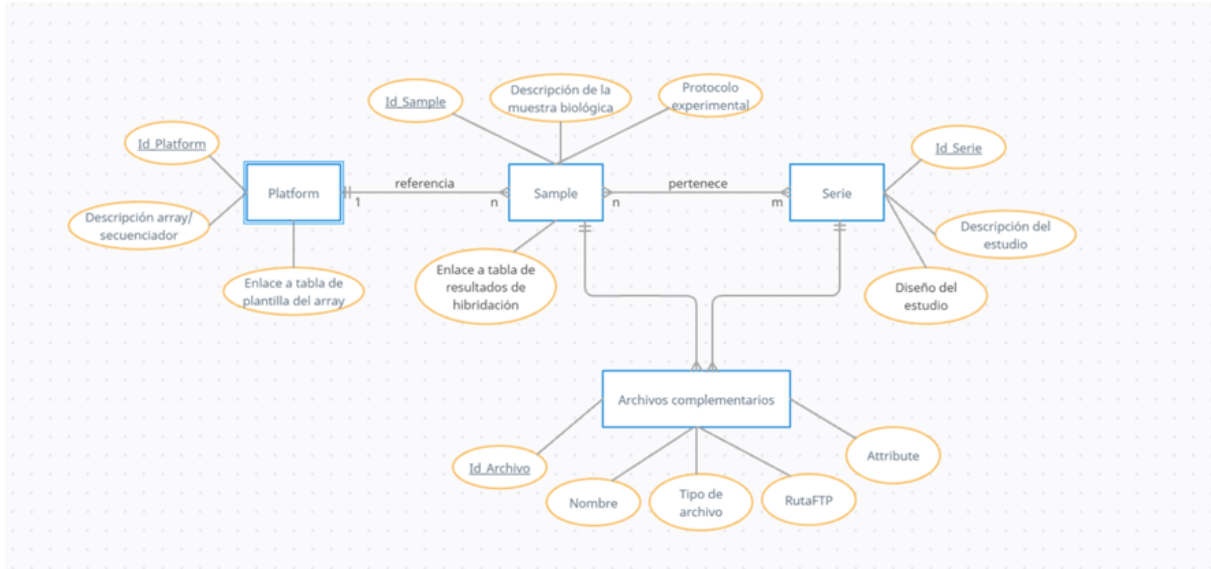


Figure 4.1: Diagrama Entidad-Relación (Fuente: elaboración propia).

- Una descripción textual que recoge de dónde procede la muestra biológica y cuál es el objetivo del estudio.

Para organizar toda esta información y facilitar su consulta y análisis, se lleva a cabo un proceso que combina extracción automatizada de datos con una revisión manual, conocida como *curación de datos*². Gracias a este procedimiento, los datos enviados por los investigadores se reorganizan en registros más estructurados y accesibles denominados *DataSets*.

Un *DataSet* agrupa un conjunto de muestras biológicas que son comparables entre sí desde un punto de vista biológico y estadístico, y que pertenecen a una misma *Platform*. Esto significa que todas comparten el mismo conjunto de sondas o elementos en el array, y que las mediciones de expresión se han procesado aplicando criterios homogéneos de normalización y ajuste de fondo. De esta forma, se asegura que los valores se puedan comparar de forma adecuada.

Además, a partir de estos *DataSets* se generan los denominados *Profiles*. Un *Profile* consiste en las mediciones de expresión de un gen concreto a lo largo de todas las muestras incluidas en un *DataSet*. Estos perfiles permiten estudiar cómo varía la expresión de un gen específico en diferentes condiciones experimentales o tipos de muestra, y se pueden consultar mediante la herramienta *GEO Profiles*, que ofrece opciones de búsqueda y visualización específicas para este tipo de datos. Todos ellos se indexan en la base de datos *Entrez GEO Profiles*.

²La curación manual consiste en una revisión realizada por personal especializado que garantiza la calidad, consistencia y coherencia de los datos recopilados, corrigiendo posibles errores y asegurando que se ajustan a los estándares establecidos [48].

Es importante señalar que no todos los registros *Series* que se envían a GEO acaban formando parte de un *DataSet*. Este trabajo de selección y reorganización lo realiza el personal encargado de la curación y mantenimiento de la base de datos, que se encarga de elegir qué estudios reúnen las condiciones necesarias para integrarse en un *DataSet*. Además, los *DataSets* y *Profiles* son la base para muchas de las herramientas de visualización y análisis que ofrece GEO, como los gráficos de perfiles de expresión génica o los clústeres de agrupación de muestras. Por último, cada *DataSet* recibe un identificador único y estable con el prefijo *GDS*, y se pueden consultar desde la interfaz *Entrez GEO DataSets*.

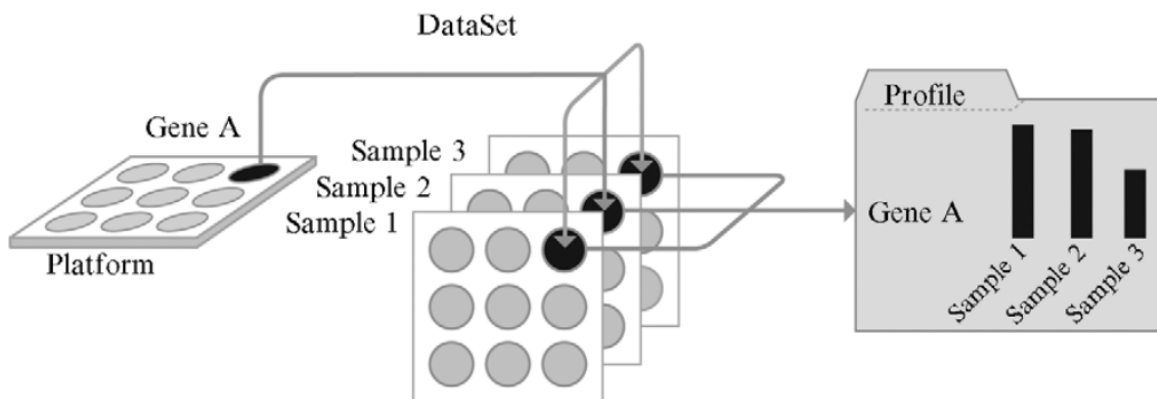


Figure 4.2: Relación entre Platforms, Samples, DataSets y Profiles (Fuente: [45]).

4.2 Envío de datos

Como se adelantó en la introducción, la base de datos GEO se adhiere a las reglas establecidas en MIAME y, por consiguiente, el envío de los procedimientos promueven su cumplimiento. Sin embargo, es el propio investigador el que tiene la responsabilidad, en última instancia, de asegurar que los datos estén suficientemente anotados y que se cumple la normativa para investigación con sujetos humanos. A continuación se va a hacer un desglose de los pasos de los que se compone el flujo de datos que siguen los investigadores al enviar datos a GEO:

1. *Inicio del depósito de datos:* Los investigadores envían sus datos a GEO antes de enviar su manuscrito a una revista. Esto lo hacen a través de su cuenta en de MyNCBI.
2. *Formatos de envío:* GEO permite varios formatos de envío, como hojas de cálculo o XML, según las instrucciones de envío.
3. *Validación y curación:* Los envíos pasan primero por una validación sintáctica automática. Luego, un curador de GEO revisa que los datos estén bien organizados

y contengan suficiente información. Si hay algún problema, el curador colabora con los autores hasta resolverlos.

4. *Asignación de identificadores*: Cuando todo está correcto, se asignan números de acceso estables que se pueden citar en el manuscrito.
5. *Privacidad previa a la publicación*: Los datos se mantienen privados hasta que el artículo es publicado. Durante ese tiempo, los autores pueden generar una URL que dé acceso confidencial a editores y revisores.
6. *Disponibilidad pública e indexación*: Tras su publicación, los datos de Platform, Sample y Serie se indexan en la base de datos *Entrez GEO DataSets*, donde los usuarios pueden consultar y descargar los datos, o realizar un análisis de expresión génica mediante la herramienta de comparación *GEO2R*.
7. *Interoperabilidad con otras bases de datos*: Algunas partes del envío se transfieren a otras bases de datos como *SRA* (para secuencias) o *BioProject* (para descripciones de estudios), con enlaces recíprocos hacia GEO.
8. *Curación mensual adicional*: Cada mes, algunas Series se seleccionan para una curación más profunda, creando *GEO DataSets* y *GEO Profiles*, derivados de estos.

4.3 Navegación, descarga y consulta

4.3.1 Navegación

El navegador de repositorios GEO cuenta con pestañas que contienen tablas donde se listan los registros de Series, Samples, Platforms y DataSets. Las tablas incluyen información que se puede buscar y filtrar, así como enlaces a registros relacionados y descargas de archivos complementarios. Estas tablas se pueden exportar e incluyen información adicional que no se muestra en el navegador, como identificadores de *PubMed* y accesiones relacionadas en *SRA*.

4.3.2 Descarga de datos

Todos los datos almacenados en GEO pueden descargarse en una gran variedad de formatos usando varios mecanismos que se presentan a continuación.

Para descargar registros GEO originales podemos hacerlo de cualquiera de las siguientes formas:

- *Enlaces a registros de Series*: Al pie de cada registro Serie de GEO, figuran enlaces a descargas de familias de experimentos en diversos formatos y archivos complementarios. Estos archivos están comprimidos con gzip (extension .gz o .tgz). Para descomprimir y leer estos archivos, se ha de utilizar la herramienta *WinZip* o *7-Zip*.

4 Proyecto GEO (Gene Expression Omnibus)

Accession	Title	Series type(s)	Organism(s)	Samples	GDS	Supplementary	Contact	Release date
GSE131863	CARM1 is an integral component of the NuRD chromatin remodeling complex	Genome binding/occupancy profiling by high throughput sequencing	Homo sapiens	7		BIGWIG SRA Run Selector	Wen Liu	May 19, 2025
GSE131864	CARM1 is an integral component of the NuRD chromatin remodeling complex (RNA-Seq)	Expression profiling by high throughput sequencing	Homo sapiens	8		BIGWIG SRA Run Selector	Wen Liu	May 19, 2025
GSE131865	CARM1 is an integral component of the NuRD chromatin remodeling complex	Expression profiling by high throughput sequencing Genome binding/occupancy profiling by high throughput sequencing	Homo sapiens	15		BIGWIG	Wen Liu	May 19, 2025

Figure 4.3: Navegador GEO (Fuente: [49]).

- *Descarga FTP*: Todos los registros GEO y los archivos de datos en crudo pueden descargarse gratuitamente desde el sitio FTP de GEO. Sin embargo, GEO contiene a día de hoy tal cantidad de envíos que ya no se puede acceder a algunos directorios principales mediante navegadores web debido a errores de latencia. En tales casos, es necesario pasar por alto el directorio principal e ir directamente al directorio de destino. Por ejemplo, para la serie *GSE1000*:

ftp : // ftp.ncbi.nlm.nih.gov/geo/series/GSE1nnn/GSE1000/matrix/

La mayoría de los archivos del servidor FTP están comprimidos con gzip (extensión .gz y .tgz). Para descomprimirlos y leerlos, si se usa Windows, se puede hacer con WinZip o 7-Zip. Si se dispone de UNIX, se deberán de seguir los comandos tar y gunzip para extraer los archivos, por ejemplo:

```
$ tar -xf GSExxxx_RAW.tar
$ gunzip *gz
```

- *Barra de visualización por identificadores (Accessions)*: Se encuentra en la parte superior de cada registro GEO y puede utilizarse para descargar o visualizar registros completos o parciales, Platforms, Samples y Series relacionados. La función *Scope* (el alcance) permite visualizar un único número de acceso (*Self*), cualquiera (*Platform*, *Sample* o *Serie*), o todos (*Family*) los registros relacionados con el acceso especificado (el identificador que hayamos puesto). *Amount* determina la cantidad de datos mostrados con opciones que incluyen sólo metadatos (*Brief*), metadatos y las 20 primeras filas de la tabla de datos (*Quick*), sólo tabla de datos (*Data*) o registros completos de metadatos/tabla de datos (*Full*). *Format* controla si los registros se muestran en formato HTML, SOFT (texto sin formato) o MINiML (XML).
- *Construcción de una URL*: Una forma alternativa de usar la barra de visualización

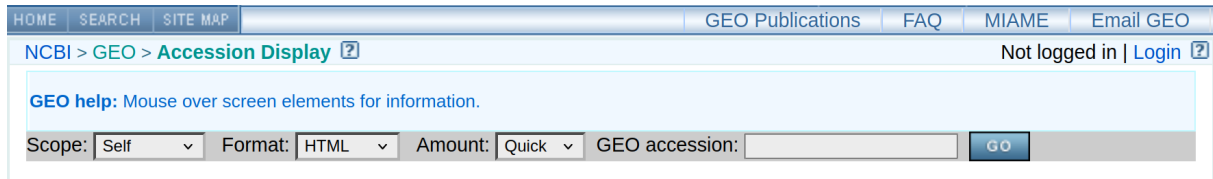


Figure 4.4: Barra de navegación de accesiones (Fuente: [50]).

de accesos anteriormente descrita, es construir una URL para devolver los datos que nos interesen. Las URLs tienen el siguiente formato:

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=gpl96&targ=self&view=brief&form=text>

Esta URL, bajo el esquema `http` y la autoridad `www.ncbi.nlm.nih.gov`, accederá a `acc.cgi` a través de la ruta `geo/query/acc.cgi` y realizará una consulta para devolver un fichero de texto que contiene una breve vista de la accesión `GPL96`.

Los posibles valores para las componentes de la consulta son:

- *acc* = un identificador GEO válido, según el formato *glpxxx*, *gsmxxx* o *gsesxxx*.
 - *targ* = *self*, *gsm*, *gpl*, *gse* o *all*.
 - *view* = *brief*, *quick*, *data* o *full*.
 - *form* = *text*, *html* o *xml*.
- *Descargas de consultas en Entrez GEO DataSets*: Todos los registros originales pueden buscarse y descargarse a través de la interfaz *Entrez GEO DataSets*. Los resultados pueden exportarse configurando la barra de herramientas de la parte superior de la página como '*Enviar a: Archivo*'.

Para descargar *DataSets* y *Profiles* curados, podemos usar:

- *Enlaces en DataSets*: Los enlaces a los archivos *SOFT* de *DataSet* están disponibles en el botón «download» de cada registro de *DataSet*. Estos archivos están comprimidos con *gzip*.
- *Descarga FTP*: Todos los registros *GEO DataSet* están disponibles, como hemos dicho antes, en el servidor FTP de GEO.
- *Valores de Perfil*: Para descargar datos de perfil, usamos el botón *Download profile data* situado en la parte superior de las páginas de recuperación de *Entrez GEO Profiles* para descargar los valores de expresión de los genes encontrados en la consulta.
- *Descargas de consultas a Entrez GEO DataSets y Entrez GEO Profiles*: Es posible exportar resúmenes de documentos de *Entrez GEO DataSets* y *Entrez GEO Profiles* configurando la barra de herramientas de la cabecera de la página como '*Enviar a: Archivo*'.

4.3.3 Consulta de datos

NCBI dispone de un potente sistema de búsqueda y recuperación llamado *Entrez*, que permite consultar el contenido de su red de bases de datos integradas. Los datos de GEO están disponibles en dos bases de datos independientes a las que ya hemos hecho referencia en numerosas ocasiones: *Entrez GEO DataSets* y *Entrez GEO Profiles*.

El flujo de trabajo habitual consiste en que el usuario primero identifique estudios de interés mediante la búsqueda en *Entrez GEO DataSets*, y posteriormente utilice *GEO2R* o *GEO Profiles* para localizar genes específicos o patrones de expresión génica dentro de ese estudio. También es posible consultar directamente *Entrez GEO Profiles*.

Además, *Entrez* genera numerosos enlaces que conectan datos relacionados: enlaces entre bases de datos que vinculan GEO con otros recursos de NCBI como *PubMed*, *GenBank* o *Gene*; y enlaces dentro de la propia base de datos, que conectan genes relacionados por patrón de expresión, posición cromosómica o secuencia.

Ambas bases de datos permiten refinar búsquedas mediante filtros por campos específicos, búsqueda por facetas, que sería la forma más sencilla de consulta, y consultas avanzadas combinadas.

Veamos cómo se puede consultar la base de datos *Entrez GEO DataSets* mediante consultas avanzadas. Como ya se ha mencionado anteriormente, esta base de datos almacena las descripciones originales de los registros de Platform, Sample y Series aportados por los autores, así como los *DataSets* curados. Puede buscarse utilizando distintos atributos como palabras clave, organismo, tipo de estudio o autores.

GEO DataSets GEO DataSets ▾ **Search** [Help](#)

Advanced

GEO DataSets Advanced Search Builder

Use the builder below to create your search

[Edit](#) [Clear](#)

Builder

All Fields ▾ - [Show index list](#)

AND ▾ All Fields ▾ - + [Show index list](#)

Search or [Add to history](#)

Figure 4.5: Construcción de consulta avanzada en *Entrez GEO DataSets* (Fuente: [51]).

Algunos ejemplos de consultas serían:

- Recuperar estudios que investigan el efecto del tabaco o la dieta en mamíferos no humanos:

```
(smok* OR diet) AND (mammals[organism] NOT human[organism])
```

- Buscar estudios que analicen expresión génica mediante secuenciación de nueva generación:

```
"expression profiling by high throughput sequencing"[DataSet Type]
```

- Recuperar envíos que incluyan archivos Affymetrix CEL:

```
cel[Supplementary Files]
```

- Localizar DataSets curados que incluyan 'edad' como variable experimental:

```
age[Subset Variable Type]
```

- Consultar estudios que contengan entre 100 y 500 muestras:

```
100:500[Number of Samples]
```

- Buscar estudios en los que aparezca 'Smith, A.' como autor:

```
smith a[Author]
```

Por su parte, como ya se mencionó anteriormente, la base de datos *Entrez GEO Profiles*, almacena perfiles de expresión génica derivados de *DataSets* curados de GEO. Cada perfil se muestra como un gráfico que representa el nivel de expresión de un gen a lo largo de todas las muestras de un *DataSet*. En la parte inferior del gráfico hay unas barras que indican el contexto experimental, lo que permite ver fácilmente si un gen está diferencialmente expresado en distintas condiciones.

Se puede consultar usando diferentes atributos, como palabras clave, símbolos o nombres de genes, identificadores de *GenBank* o perfiles que estén marcados como diferencialmente expresados. Algunos ejemplos de consultas incluyen:

- Recuperar todos los perfiles de expresión génica para CYP1A1:

```
CYP1A1[Gene Symbol]
```

- Recuperar perfiles de expresión génica de CYP1A1 o ME1 en DataSets que investiguen los efectos del tabaco o la dieta:

```
(CYP1A1[Gene Symbol] OR ME1[Gene Symbol]) AND (smok* OR diet)
```

- Recuperar perfiles para todas las quinasas en el DataSet con número de accesión GDS182:

```
kinase[Gene Description] AND GDS182
```

- Recuperar perfiles para genes que tengan el término Gene Ontology (GO) 'apoptosis' en el DataSet con accesión GDS182:

4 Proyecto GEO (Gene Expression Omnibus)

Profile GDS4515 / 202859_x_at
Title Microsatellite-unstable colorectal cancer
Organism Homo sapiens

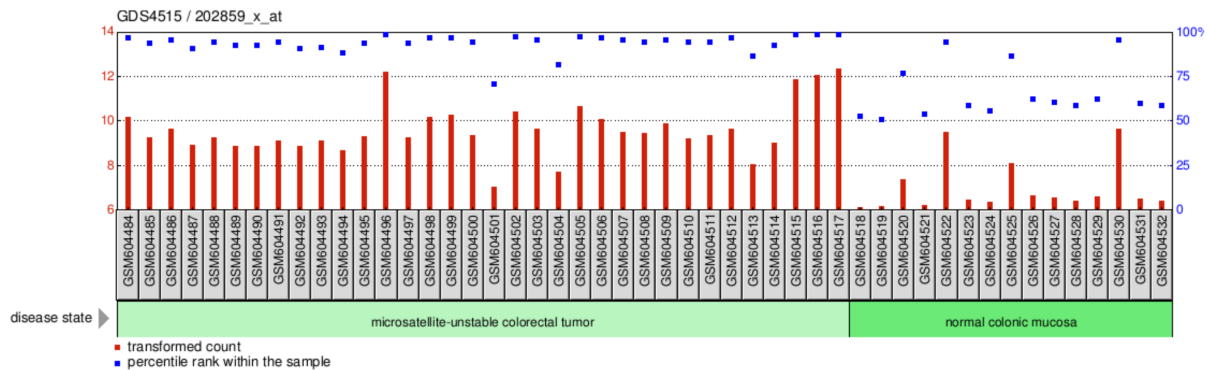


Figure 4.6: Ejemplo de perfil (Fuente: [52]).

apoptosis[Gene Ontology] AND GDS182

- Recuperar perfiles de genes que se encuentren dentro del rango 10000:3000000 en el cromosoma 8 de ratón:
`(8[Chromosome] AND 10000:3000000[Base Position]) AND mouse[organism]`
- Recuperar genes que muestran expresión diferencial en DataSets que examinan el efecto de un agente:
`agent[Flag Information] AND "value subset effect"[Flag Type]`

Tras haber explorado en detalle la base de datos GEO, la organización de los datos y las herramientas de descarga y consulta, en la siguiente sección nos centraremos en *Bioconductor*, un conjunto de paquetes de software en R diseñado para el análisis e interpretación de datos genómicos. Veremos cómo Bioconductor permite aprovechar al máximo la información obtenida de GEO mediante potentes técnicas estadísticas y bioinformáticas.

5 Bioconductor

La información recogida en este capítulo ha sido extraída principalmente de las fuentes bibliográficas [8, 53, 54, 55].

Bioconductor es un proyecto de código abierto y un repositorio de paquetes para R que se usa mucho en bioinformática y biología computacional. Básicamente, es una plataforma que ofrece muchas herramientas para analizar datos biológicos, especialmente en el campo de las ómicas. La idea principal del proyecto es crear y compartir software libre que ayude a realizar análisis de datos biológicos de forma rigurosa y reproducible.

5.1 Objetivos del proyecto

Desde que empezó en 2001, Bioconductor ha ido adaptándose a nuevas tecnologías, desde los microarrays hasta la transcriptómica espacial, que es una técnica más reciente. Sus objetivos son varios, pero entre los principales están:

- Dar acceso a métodos estadísticos y gráficos avanzados para analizar datos genómicos.
- Facilitar la incorporación de metadatos biológicos en estos análisis.
- Ofrecer una plataforma común que permita desarrollar software que sea fácil de ampliar, escalable y que funcione bien con otras herramientas.
- Promover la creación de documentación clara y la reproducibilidad en los estudios.
- Formar a investigadores para que puedan usar técnicas computacionales y estadísticas en el análisis de datos genómicos.

5.2 Integración de R en Bioconductor

Para conseguir esto, Bioconductor se apoya en R, que es un lenguaje interpretado y de alto nivel muy usado en estadística y ciencia de datos. R permite crear de forma rápida nuevas formas de analizar datos, tiene un sistema para empaquetar el software junto con la documentación y ofrece estructuras orientadas a objetos que ayudan a manejar la complejidad de los problemas en biología computacional. Además, da acceso a datos en línea y tiene herramientas para hacer simulaciones, modelado y visualizar resultados.

Bioconductor aprovecha todo esto y añade sus propias estructuras y métodos para trabajar con datos genómicos a gran escala, como secuenciación de ADN, ARN o microarrays. También facilita crear flujos de trabajo donde se combinan diferentes tipos de datos y métodos estadísticos, regresión, análisis de redes, aprendizaje automático y visualización.

5.3 Paquetes

Los paquetes de Bioconductor se dividen en cuatro grupos principales:

- *Software*: algunos paquetes proporcionan las bases para almacenar y acceder a los datos, y otros ofrecen herramientas para analizar esos datos. Esta separación hace que sea más fácil reutilizar estructuras y probar distintas formas de análisis sin aprender cosas nuevas cada vez.
- *Datos de anotación*: contienen bases de datos con información genómica como identificadores de genes o rutas biológicas.
- *Datos experimentales*: son conjuntos de datos estándar que se usan para mostrar ejemplos en los que se aplican los paquetes.
- *Workflows*: son colecciones de documentos que explican cómo usar varios paquetes juntos para hacer un análisis completo, pero no tienen código nuevo.

5.4 GEO y Bioconductor

Dado que *GEO* es una de las bases de datos públicas más completas y utilizadas para almacenar datos de expresión génica, resulta muy interesante poder integrarla con *Bioconductor*. Así, tendríamos una fuente bastante extensa de datos sobre los que aplicar todo el potencial de los paquetes que ofrece Bioconductor. Para hacer posible esta conexión se desarrolló el paquete *GEOquery*, que actúa como puente entre ambos, facilitando la descarga, consulta y manipulación de los datos desde R y su uso dentro del ecosistema de Bioconductor.

5.4.1 GEOquery

El objetivo principal de *GEOquery* es descargar y leer los archivos en formato SOFT que usa GEO, manteniendo toda la información de los registros originales. Lo bueno de *GEOquery* es que hace muy fácil acceder a los datos de GEO desde R, ya que básicamente todo se puede hacer con un único comando: `getGEO`.

Por ejemplo, para descargar un dataset completo (en este caso, el *GDS1*), primero se carga la librería:

```
library(GEOquery)
```

Luego se usa el comando:

```
gds <- getGEO('GDS1')
```

Esto descarga el archivo comprimido en una carpeta temporal, lo descomprime, lo interpreta y lo guarda en una variable de R llamada `gds`. Esa variable contiene ya todos los datos de `GDS1` organizados en una estructura de tipo `GDS` que se puede utilizar dentro de R y Bioconductor. También se puede hacer lo mismo con una muestra individual, cambiando el identificador del dataset por el de la muestra (*GSMxxx*).

Estructuras de datos en *GEOquery*

Una de las ventajas que tiene *GEOquery* es que organiza la información descargada en estructuras de datos específicas para facilitar su manejo. Estas estructuras se agrupan de la misma forma en la que GEO organiza sus datos:

- *GDS*, *GPL* y *GSM*: son las clases básicas. Cada una contiene una cabecera con metadatos (extraídos tal cual de GEO) y una tabla llamada *GEODataTable*. Esta tabla tiene dos partes: los nombres de las columnas y una tabla de datos 2D. Por ejemplo, para consultar los metadatos de una muestra:

```
Meta(gsm)[1:3]
```

Y para ver parte de la tabla de datos:

```
Table(gsm)[1:3, c(2, 6, 7)]
```

O las descripciones de las columnas:

```
Columns(gsm)[c(2, 6, 7), ]
```

La clase *GPL* funciona igual, pero en vez de datos de expresión contiene información sobre las características de los arrays. Por otro lado, *GDS* incluye además anotaciones asociadas a cada muestra.

- **GSE**: es una clase compuesta que representa un conjunto de muestras (*GSM*) y plataformas (*GPL*). Tiene su propia sección de metadatos, pero en lugar de una tabla de datos, contiene dos listas: una con objetos *GPL* y otra con objetos *GSM*. Gracias a los métodos de acceso y a las capacidades de manipulación de datos de R, se pueden consultar y reorganizar estos datos de forma flexible para los análisis que se quieran hacer.

Conversión a estructuras de Bioconductor

Los datos de *GEO* tienen una estructura parecida a la de objetos usados en otros paquetes de *Bioconductor*, como *MAList* (de *limma*) o *ExpressionSet* (de *Biobase*). Por eso, existen funciones como *GDS2MA* y *GDS2eSet* que permiten convertir un objeto *GDS* a estos formatos.

Por ejemplo, para pasar el objeto *gds* que descargamos antes a un *ExpressionSet*:

```
eset <- GDS2eSet(gds, do.log2 = TRUE)
```

Así se obtiene un *ExpressionSet* que contiene los datos de *GEO* junto con la información de las muestras, los datos transformados en logaritmo (si se especifica) y las anotaciones de las sondas de la plataforma asociada.

Además, existen funciones parecidas para convertir a otros tipos de objetos de *Bioconductor*, lo que permite integrar estos datos directamente en los análisis de expresión génica, clasificación, visualización o análisis de redes que se hacen con otros paquetes del entorno.

En resumen, *GEOquery* sirve de puente entre las herramientas de *Bioconductor* y los datos públicos disponibles en *GEO*. Este puente es el que usaremos para desarrollar la aplicación con la que pondremos fin a este trabajo y cuyo diseño, desarrollo y documentación veremos en profundidad en el siguiente capítulo.

Parte IV

APLICACIÓN

Diseño y desarrollo de una aplicación para análisis de datos RNA-Seq

6 Aplicación para el análisis multivariante de datos transcriptómicos (RNA-Seq): diseño y desarrollo

6.1 Introducción

Del mismo modo que dos ríos confluyen para formar una corriente más caudalosa y poderosa, los contenidos matemáticos e informáticos desarrollados a lo largo del presente trabajo encuentran su punto de unión en el diseño y desarrollo de una aplicación interactiva. Las técnicas multivariantes, con especial énfasis en el análisis clúster y el análisis de componentes principales (PCA), constituyen el eje matemático de este trabajo, mientras que la exploración de recursos bioinformáticos como la base de datos GEO, R como lenguaje de programación y el proyecto Bioconductor, proporcionan el soporte informático necesario.

Esta aplicación, desarrollada con Shiny en R, representa la culminación práctica del trabajo, al permitir al usuario cargar datos transcriptómicos, aplicar análisis estadísticos complejos y visualizar de forma clara los resultados. Así, se materializa la fusión de ambos mundos —matemático e informático— en una herramienta útil para la exploración e identificación de patrones biológicos en datos RNA-Seq.

Como toda pieza de software, su desarrollo no puede abordarse de forma improvisada, sino que debe seguir una metodología que garantice su funcionalidad, usabilidad y calidad. Por ello, en este capítulo se expone el proceso de desarrollo de la aplicación, siguiendo las fases clásicas del desarrollo del software. En primer lugar, se realiza un análisis y especificación de requisitos que debe cumplir la herramienta. A continuación, se describe el diseño de la aplicación, tanto a nivel funcional como de interfaz. Posteriormente se detalla la fase de implementación, abordando las decisiones técnicas adoptadas y tecnologías utilizadas [56]. Finalmente, se ha incluido un manual de usuario sencillo y exhaustivo para que cualquier persona interesada pueda comprender y usar fácilmente la herramienta.

6.2 Análisis y especificación de requisitos

La información necesaria para llevar a cabo esta sección ha sido extraída de la fuente bibliográfica [57].

En esta sección se ha llevado a cabo un proceso de análisis y especificación de los requisitos, de datos, de información, de almacenamiento, funcionales y no funcionales

necesarios.

- *Requisitos de datos:* son los datos de insumo para que la aplicación pueda operar.
 - Fichero en formato CSV o TSV que contenga una matriz de expresión génica, donde las filas representen genes o transcritos y las columnas correspondan a las muestras. El archivo debe contener al menos dos muestras para que se pueda hacer algún análisis multivariante.
- *Requisitos de información:* se refieren a la información que se puede generar a través del sistema.
 - Gráfico con matriz de correlación.
 - Tabla con los valores de las componentes principales por muestra.
 - Gráficos de PCA.
 - Gráficos de varianza explicada y acumulada por las CPs.
 - Dendrograma para análisis cluster jerárquico.
 - Gráficos para determinación del número óptimo de clusters.
 - Gráfico de análisis cluster no jerárquico (kMeans).
 - Tabla de expresión diferencial.
 - Mensajes de error claros en caso de problema con los datos.
- *Requisitos de almacenamiento:* están asociados a lo que se requiere almacenar a nivel local o en bases de datos para que el sistema pueda suministrar la información o salidas requeridas. En nuestro caso, aunque la aplicación web se ejecuta de forma local sin base de datos persistente, existen algunos elementos que deben mantenerse en memoria durante la sesión para permitir el flujo de trabajo.
 - Dataset creado a partir de los datos cargados por el usuario.
 - Resultados intermedios del análisis.
- *Requisitos funcionales:* definen lo que la aplicación debe ser capaz de hacer desde el punto de vista del usuario y la lógica de funcionamiento.
 - Permitir al usuario cargar un archivo CSV o TSV con datos de expresión génica.
 - Validar el formato del archivo de entrada.
 - Notificar al usuario de los errores cometido durante la subida de los datos.
 - Permitir al usuario ejecutar el análisis de componentes principales (PCA) sobre los datos válidos.
 - Permitir elegir al usuario qué tipo de análisis clúster quiere aplicar sobre los datos.
 - Ejecutar el análisis cluster con el método elegido por el usuario.

- Ejercutar expresión diferencial.
 - Permitir al usuario visualizar los resultados en forma de gráficos interactivos y tablas.
 - Resetear todo el proceso para hacerlo de nuevo con nuevos datos sin necesidad de reiniciar la aplicación.
 - Acceder al manual de usuario de la aplicación.
 - Descargar el código fuente de la aplicación.
 - Consultar la información de contacto del desarrollador.
- *Requisitos no funcionales*: son condiciones que debe cubrir el sistema que se desarrollará con respecto a aspectos de calidad, rendimiento y facilidad de uso.
 - La interfaz debe ser clara, intuitiva y visualmente atractiva para usuarios con conocimientos básicos de bioinformática.
 - La aplicación debe ser capaz de procesar archivos de gran tamaño sin tiempos de espera demasiado prolongados.
 - Debe poder ejecutarse en cualquier sistema que tenga R y Shiny instalados, así como ser accesible desde un navegador cualquiera a través de un enlace.
 - Debe gestionar bien los errores en los datos de entrada sin bloquear la aplicación.

6.3 Diseño de la aplicación

6.3.1 Arquitectura general

La estructura general de la aplicación se divide en dos partes principales:

- *Interfaz de usuario (UI)*: que define qué va a ver el usuario, la estructura de navegación y los elementos interactivos que la componen.
- *Servidor (server)*: que contiene toda la lógica que hay por debajo en cuanto a procesamiento de datos y generación de resultados.

6.3.2 Diseño de la Interfaz de Usuario

La interfaz gráfica se organiza de la siguiente manera:

- *Cabecera superior* que ocupa el 25% del alto y el 100% del ancho de la página. Consta de:
 - Imagen de fondo.
 - Nombre de la aplicación sobre la imagen de fondo.
 - Logo de la Universidad de Granada situado en la esquina superior derecha.

- *Barra Lateral* en el lado izquierdo de la pantalla (25% el ancho), que consta de:
 - Selector de archivo (.csv / .tsv).
 - Opciones para seleccionar el delimitador: punto, punto y coma o tabulador.
 - Opciones para seleccionar el símbolo de decimal: coma o punto.
- *Panel principal* en el que aparecen todos los resultados. Justo debajo de la cabecera se encuentra un menú principal en línea con tres secciones en forma de botones: PCA, Análisis Cluster y Expresión diferencial. La sección de Análisis cluster consta de dos botones, *Jerárquico* y *No Jerárquico* que a su vez consta de dos subopciones: *Método del Codo* y *Método de la Silueta*, una barra deslizante para elegir números mayores o iguales que dos y un botón *Ejecutar K-means*.
- *Pie de página* en el que se ofrecen enlaces al manual de usuario, a la información de contacto del desarrollador y un enlace de descarga del código de la aplicación para quien quiera ejecutarlo de forma local en su máquina.
- La página de *Contacto* contiene simplemente datos para comunicación con el desarrollador de la aplicación.

6.3.3 Diseño del flujo de datos y procesos

El flujo general de la aplicación sigue esta lógica:

- *Carga del fichero de datos*: el usuario selecciona el archivo y se leen los datos utilizando los parámetros seleccionados por el mismo.
- *PCA*: se calcula la matriz de componentes principales y se generan dos gráficos: un scatterplot de las dos primeras componentes principales y otro con la varianza explicada por las distintas CPs.
- *Análisis cluster*: se despliegan dos opciones al clicar sobre este botón:
 - *Jerárquico*: se genera un dendrograma.
 - *No jerárquico (K-means)*: se abren dos opciones para elegir un método para determinar el número óptimo de clusters: el método del Codo o el de la silueta. Se visualiza el gráfico correspondiente en función del método elegido. Con la barra deslizante se escoge el número de clusters, k con el que aplicar el método k-means y se pulsa el botón *Ejecutar k-means* tras lo cual se genera el gráfico correspondiente con los k clusters agrupando las muestras.
- (Opcional y recomendable al principio) *Consulta manual de usuario*: el usuario pulsa el enlace al manual de usuario para saber cómo funciona la aplicación.
- *Descarga del código fuente y Contacto*: el usuario quiere descargar el código fuente y/o contactar con el desarrollador. Pulsa en el primer enlace para la descarga y pulsa sobre el de contacto para acceder a la página con la información de contacto del desarrollador.

6.4 Implementación

Teniendo claros todos los requisitos que, a priori¹, ha de cumplir nuestra aplicación y habiendo hecho el diseño de la misma, procedemos a construir el código fuente siguiendo el diseño previamente definido y cumpliendo con los requisitos especificados.

En nuestro caso, se ha utilizado el lenguaje de programación R junto con el framework de desarrollo de aplicaciones web, *Shiny*.

Toda la información para el desarrollo de la aplicación se ha extraído de las siguientes fuentes bibliográficas [58, 59, 60, 61, 62, 63].

6.4.1 Estructura del código

El código de la aplicación se encuentra estructurado en un único archivo `app.R`, donde se definen claramente las dos partes fundamentales de toda aplicación Shiny y que ya hemos comentado anteriormente:

- *UI (Interfaz de usuario)*: se ha implementado usando `fluidPage()` y está estructurada en secciones claramente definidas (cabecera, barra lateral, panel principal y pie de página).
- *Server (Servidor)*: contiene la funcionalidad asociada a la lectura de datos, validación, análisis multivariante (PCA y clustering), expresión diferencial, generación de gráficos y gestión de errores.

Ambas partes están conectadas mediante el comando `shinyApp(ui = ui, server = server)` que lanza la aplicación.

6.4.2 Paquetes utilizados

Para el desarrollo de la aplicación Shiny y la realización del análisis de datos, se han empleado los siguientes paquetes de R:

- *Interfaz y componentes interactivos*: `shiny`, `shinydashboard`, `shinyWidgets`, DT.
- *Visualización*: `ggplot2`, `ggpubr`, `corrplot`, `RColorBrewer`, `ggdendro`, `scatterplot3d`, `pheatmap`, `factoextra`.
- *Análisis estadístico y clustering*: `cluster`.
- *Análisis de expresión génica*: `DESeq2`, `edgeR`.

¹Los requisitos no son una parte cerrada en el proceso de desarrollo del software, si no que son dinámicos. Pueden surgir necesidades durante la fase de desarrollo del mismo o incluso podemos ver que algunos son prescindibles y se prefieran quitar de la lista.

6.4.3 Lectura y validación de datos

Se ha implementado un mecanismo de validación que garantiza que el archivo subido por el usuario tenga la estructura esperada. Esto incluye:

- Validación del formato del archivo (extensiones .csv y .tsv).
- Verificación del número mínimo de columnas (al menos dos muestras).
- Comprobación de que todos los valores, excepto la primera columna, sean numéricos.
- Gestión de separadores (coma, punto y coma, tabulador) y notación decimal (punto o coma).

6.4.4 Implementación de PCA

La funcionalidad de PCA ha sido implementada usando funciones de `prcomp()` y visualizada con `fviz_pca_ind()` para mostrar la distribución de las muestras, y `fviz_eig()` para representar la varianza explicada por cada componente. Se ha garantizado que el cálculo solo se ejecute una vez que se hayan validado y cargado correctamente los datos.

6.4.5 Implementación del análisis cluster

El análisis de agrupamiento se ha dividido en:

- *Jerárquico*: utilizando `hclust()` y representado mediante dendrogramas con `fviz_dend()`.
- *No jerárquico (K-means)*:
 - Métodos para determinar el número óptimo de clusters: `fviz_nbclust()` con los métodos del Codo (wss) y de la Silueta (silhouette).
 - Agrupamiento con `kmeans()` y representación gráfica con `fviz_cluster()`.

6.4.6 Análisis de expresión diferencial

El análisis de expresión diferencial se ha implementado utilizando los paquetes `DESeq2` y `edgeR`, herramientas ampliamente utilizadas en el análisis de datos transcriptómicos.

El proceso incluye los siguientes pasos:

- Construcción de un objeto `DESeqDataSet` a partir de la matriz de conteos y la información de las condiciones experimentales de cada muestra.
- Estimación de la dispersión y ajuste del modelo negativo binomial utilizando la función `DESeq()`.

- Extracción de resultados usando la función `results()`, incluyendo el cálculo de valores de p ajustados (FDR).
- Filtrado de genes significativamente expresados ($p\text{-adj} < 0.1$) y presentación de los resultados en una tabla.

6.4.7 Interacción del usuario y control de flujo

Para facilitar el uso, la aplicación cuenta con botones interactivos que permiten al usuario controlar cuándo cargar los datos (`actionButton("cargar")`) y cuándo reiniciar la interfaz desde cero (`actionButton("reset")`). Así es todo más intuitivo y evitamos que todos los análisis se tengan que volver a ejecutar de nuevo (evitamos recargar la página).

El flujo de trabajo dentro de la aplicación está gestionado mediante funciones como `reactiveVal()` y `observeEvent()`, que aseguran que los análisis y gráficos solo se actualicen cuando los datos han sido correctamente cargados y validados. Así evitamos cálculos innecesarios o errores.

6.4.8 Diseño visual y experiencia de usuario

El diseño visual de la app se ha personalizado usando código CSS directamente en la propia aplicación, a través de funciones de Shiny que permiten usar código HTML y CSS tanto para estructurar la interfaz como para aplicar estilos. Tiene sentido poder usar HTML y CSS, al ser Shiny un framework de desarrollo web.

6.4.9 Gestión de errores y robustez

Para garantizar un uso seguro y controlado de la aplicación, se han implementado diferentes mecanismos de gestión de errores que minimizan la probabilidad de fallos y mejoran la experiencia del usuario:

- Uso sistemático de la función `req()` en los módulos de análisis para asegurarse de que los datos hayan sido cargados antes de ejecutar cualquier cálculo o mostrar resultados. Esto previene errores derivados de trabajar con objetos vacíos o inexistentes.
- Eliminación automática de genes cuya suma total de conteos sea cero mediante la instrucción `df <- df[rowSums(df) != 0,]`, evitando problemas en los análisis estadísticos y de clustering.
- Verificación previa del separador de columnas mediante una lectura parcial del fichero y recuento del número de columnas en la primera línea, mostrando una alerta con `sendSweetAlert()` si se detecta que el separador indicado por el usuario no coincide con el contenido del archivo.

- Control de inputs obligatorios (archivo, separador y símbolo decimal) antes de proceder a cargar los datos, mostrando avisos emergentes amigables cuando falta alguno de ellos.
- Comprobación previa de valores ausentes en los resultados de análisis de expresión diferencial. Antes de seleccionar genes significativos, se eliminan aquellos con valores NA en padj, evitando errores en las ordenaciones y filtrados.
- Inclusión de un botón de reinicio que permite al usuario restaurar el estado inicial de la aplicación, limpiando los datos cargados y reseteando los valores de los controles de entrada, mejorando así la robustez operativa durante sesiones prolongadas.
- Validación previa del número de columnas para detectar potenciales errores de formato en el fichero cargado, mediante la instrucción:

```
n_col_elegido <- length(strsplit(lineas[1], sep)[[1]])
```

con su correspondiente alerta en caso de detectarse solo una columna.

Gracias a estas medidas, la aplicación se comporta de forma robusta y segura ante errores de usuario, problemas de formato o datos anómalos, minimizando así la posibilidad de fallos durante su uso.

6.5 Manual de usuario

En esta sección presentamos un manual de usuario sencillo y exhaustivo en el que se explica con sumo detalle en qué consiste la aplicación y cómo usarla. Se ha redactado de tal forma que cualquier usuario pueda hacer uso de ella, quedando bajo su responsabilidad e interés el saber interpretar los resultados gráficos que se obtienen al ejecutar ciertas funcionalidades de la misma.

6.5.1 ¿Qué es esta aplicación?

Esta aplicación permite analizar datos de expresión génica (es decir, los niveles de actividad de diferentes genes en distintas muestras biológicas). Usted no necesita saber de matemáticas ni de informática, incluso ni de biología para poder usarla; la propia app le guía paso a paso y muestra gráficos que resumen resultados obtenidos. Si lo que le interesa es interpretar correctamente los resultados, es recomendable que conozca de antemano el estudio del que provienen los datos, como mínimo, y tenga algunos conocimientos básicos, o bien se apoye de alguien que los tenga.

6.5.2 ¿Qué necesita para usarla?

Tan solo necesita:

- Un archivo en formato CSV (.csv) o TSV (.tsv) que contenga los datos.
- Un ordenador con conexión a Internet, si ejecuta la aplicación online, y tener R o Shiny instalados si se hace de forma local.

El archivo debe tener los nombres de los genes en la primera columna (los identificadores, generalmente), los nombres de las muestras en la primera fila y solo números en las celdas, que serán los niveles de expresión. Por ejemplo:

Gen	Muestra1	Muestra2	Muestra3
A	4.5	3.9	4.1
B	5.0	6.0	2.9

Table 6.1: Ejemplo del contenido de un fichero con datos de expresión

6.5.3 ¿Cómo iniciar la aplicación?

Puede iniciar la aplicación de dos formas distintas:

- En línea, simplemente haciendo click en el enlace proporcionado.
- En local, para lo cual tendrá que abrir el archivo *app.R* en *Rstudio* y pulsar el botón *Run App* en la parte superior.

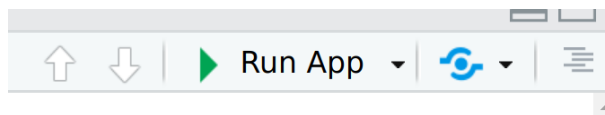


Figure 6.1: Botón Runapp RStudio (Fuente: elaboración propia).

6.5.4 Estructura de la pantalla

Una vez iniciada la aplicación, se abrirá la página que puede ver en la figura 6.2:

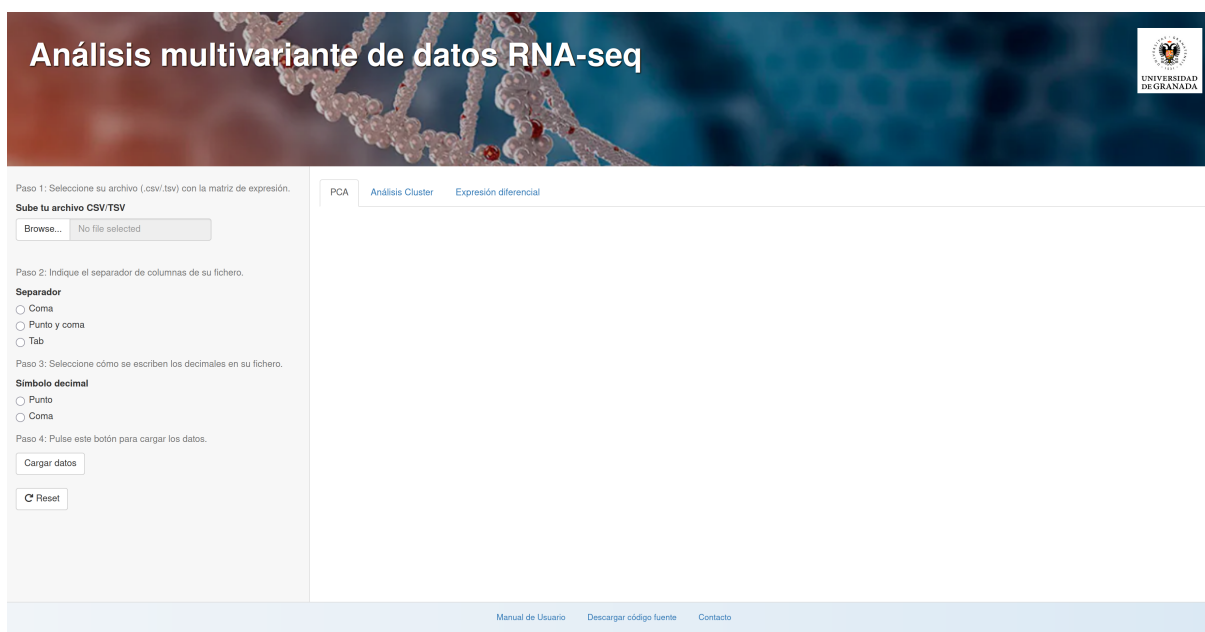


Figure 6.2: Interfaz de usuario (Fuente: elaboración propia).

En ella, verá:

- Una cabecera con el título de la aplicación y el logo de la Universidad de Granada.
- A la izquierda, por debajo de la cabecera, una barra lateral donde podrá:
 - Subir su fichero con los datos.
 - Seleccionar el tipo de separador (coma, punto y coma, tabulador).
 - Indicar si los números decimales están escritos con coma o punto.
- En la parte principal, a la derecha de la barra lateral, verá:
 - Un menú con tres botones: PCA, Análisis Cluster, Expresión Diferencial.
 - Gráficos y resultados que se actualizan según los botones que pulse.

- En la panel lateral, tiene a su disposición un botón *Reset* con el que puede deshacer todas las acciones y volverá todo al estado justamente posterior a la carga de datos.
- A pie de página verá tres enlaces:
 - Un enlace al manual sobre el manejo de la aplicación, que está usted leyendo en estos momentos.
 - Un enlace de descarga del código fuente desde un repositorio público de *Google Drive*.
 - Un enlace a una página en la que se muestra información de contacto del desarrollador de la aplicación, por si usted tuviera necesidad de comunicarse con el mismo.

6.5.5 ¿Cómo subir los datos?

Para subir los datos y poder hacer análisis multivariante con ellos, deberá seguir los siguientes pasos:

- Haga click en el botón *Subir archivo* de la barra lateral.
- Se abrirá el explorador de archivos de su ordenador, donde deberá buscar el archivo que contiene los datos.
- Seleccione el separador correcto:
 - Coma (,) si los valores están separados por comas en el fichero.
 - Punto y coma (;) si vienen de Excel.
 - Tabulador si están separados por espacios grandes.
- Seleccione si los decimales estarán escritos con punto (.) o coma (,).
- Pulse el botón *Cargar datos* para cargar los datos del fichero.
- Una vez cargado correctamente, podrá ir a las secciones de análisis.

Estos mismos pasos se indican en la propia barra lateral de la interfaz, para guiarlo en todo el proceso, que puede resultar confuso.

Al pulsar el botón *Cargar datos*, necesario pulsarlo para ver los resultados, pueden surgir algunos errores por diferentes causas que se detallan a continuación:

- Si no ha seleccionado ningún fichero de su gestor de archivos, obtendrá el mensaje de error de la figura 6.3.



¡Falta el archivo!

Por favor, seleccione un fichero CSV o TSV antes de cargar.



Figure 6.3: Error ningún fichero seleccionado (Fuente: elaboración propia).

- Si ha seleccionado mal el separador de los distintos registros de su fichero, obtendrá el aviso de la figura 6.4.



Separador sospechoso

Parece que el separador que has indicado (',') no está presente en el archivo. ¿Seguro que es el correcto?

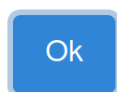


Figure 6.4: Separador posiblemente erróneo (Fuente: elaboración propia).

- Si no ha seleccionado ningún separador, verá el aviso de la figura 6.5.



Figure 6.5: Ningún separador seleccionado (Fuente: elaboración propia).

- Si no ha seleccionado ningún símbolo decimal, el mensaje que obtendrá será el de la figura 6.6.



Figure 6.6: Ningún símbolo decimal seleccionado (Fuente: elaboración propia).

6.5.6 PCA (Análisis de Componentes Principales)

Si pulsa sobre el botón *PCA* del menú en línea de la parte principal, se hará todo el proceso correspondiente al Análisis de Componentes Principales con los datos que haya subido. El PCA sirve para reducir la complejidad de los datos y poder

deshacernos de variables que no aportan mucho al estudio, o lo que es lo mismo, que explican muy poca varianza. Además, nos dará pistas acerca de una posible agrupación de los mismos.

¿Qué verá?

Tras unos segundos de espera mientras se hacen todos los cálculos, podrá ver un gráfico con puntos de colores que representan las muestras de su conjunto de datos. Cada eje (PC1, PC2) representa una "combinación" de genes. Verá también otro gráfico que mostrará cuánta información contiene cada eje (varianza explicada por las distintas componentes principales).

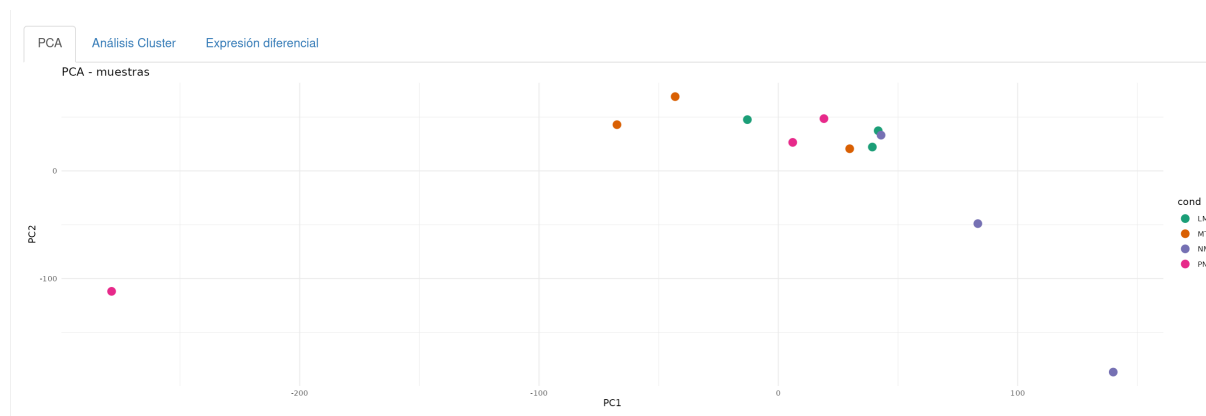


Figure 6.7: Análisis de Componentes Principales (PCA) (Fuente: elaboración propia).

¿Qué tiene que hacer usted?

Nada, solamente tiene que limitarse a cargar los datos. El análisis lo hace automáticamente por debajo. Tras él, con el apoyo de los gráficos, podrá sacar sus propias conclusiones en función de sus intereses.

Una interpretación básica puede ser la siguiente:

- Si ve puntos juntos: esas muestras se parecen.
- Si ve puntos alejados: son diferentes.
- Si en el gráfico de la varianza observa que hay barras muy altas en comparación con otras, significa que estas explican mucha más información que otras y por tanto, podríamos deshacernos de ellas de cara al análisis.

6.5.7 Análisis Cluster

Si pulsa sobre el botón *Análisis Cluster*, es porque quiere realizar clustering sobre los datos cargados. El análisis cluster sirve para agrupar las muestras en grupos disjuntos

que se llaman *clusters*. Al hacer click, se abren dos opciones:

- *Jerárquico*: se muestra un dendrograma (un árbol cuyas hojas son las muestras y que representa la agrupación jerárquica de las muestras del conjunto de datos, según similitud. Cada bifurcación une dos grupos o elementos similares, y la altura de la unión indica cómo de diferentes son. Se puede cortar a una determinada altura para definir el número de clusters).

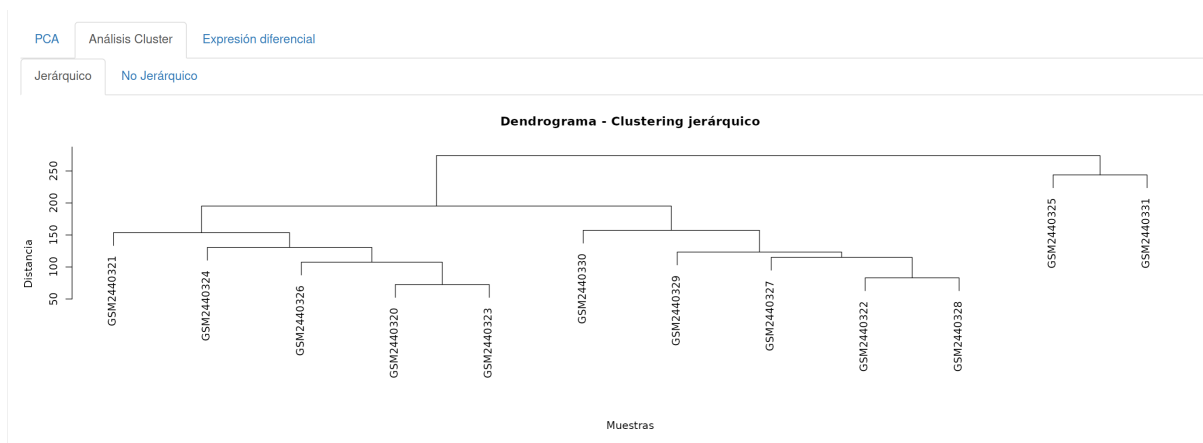


Figure 6.8: Análisis cluster jerárquico - Dendrograma (Fuente: elaboración propia).

- *No jerárquico*:
 - Primero puede determinar cuál podría ser el mejor número de clusters en los que puede agrupar las muestras. Aparecerán dos opciones:
 - * *Método del codo*: al hacer click, se mostrará un gráfico que tiene una "curva". En el eje X, se disponen distintos valores para el número de clusters (k). En el punto de la gráfica en el que se aprecie un "codo", se encuentra el mejor número de clusters.

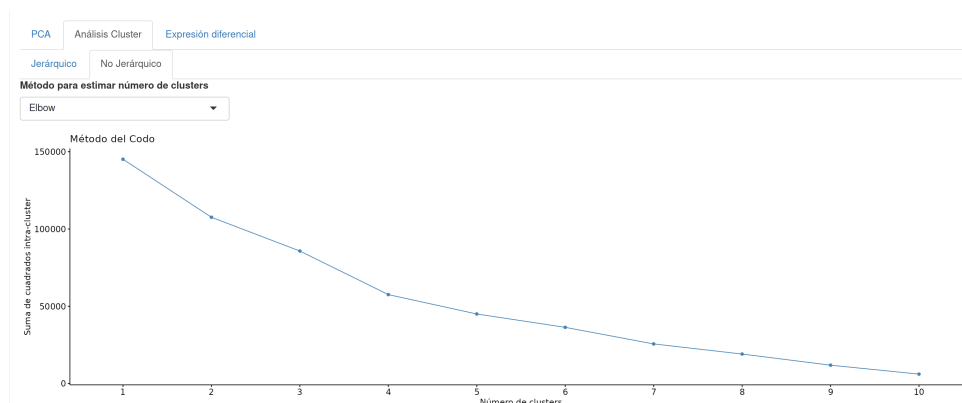


Figure 6.9: Método del codo (Fuente: elaboración propia).

- * *Método de la silueta*: muestra también una gráfica de forma que el punto en el que la gráfica alcance su máximo, ese será el candidato a número óptimo de clusters.

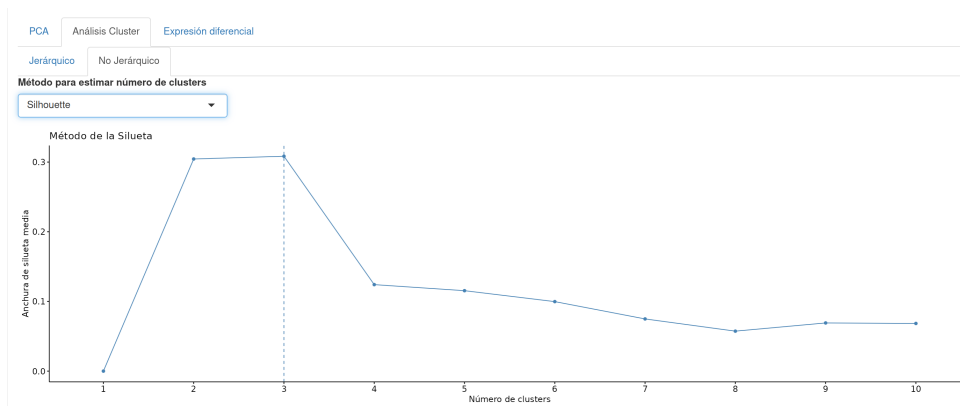


Figure 6.10: Método de la silueta (Fuente: elaboración propia).

- Luego, verá una barra horizontal deslizante con la que podrá elegir el número de clusters con el que aplicar el algoritmo k-means.
- Pulsando finalmente sobre el botón *Ejecutar K-means* tras haber escogido el número óptimo de clusters, verá un gráfico con los distintos grupos que se habrán formado, en colores diferentes para distinguirlos.

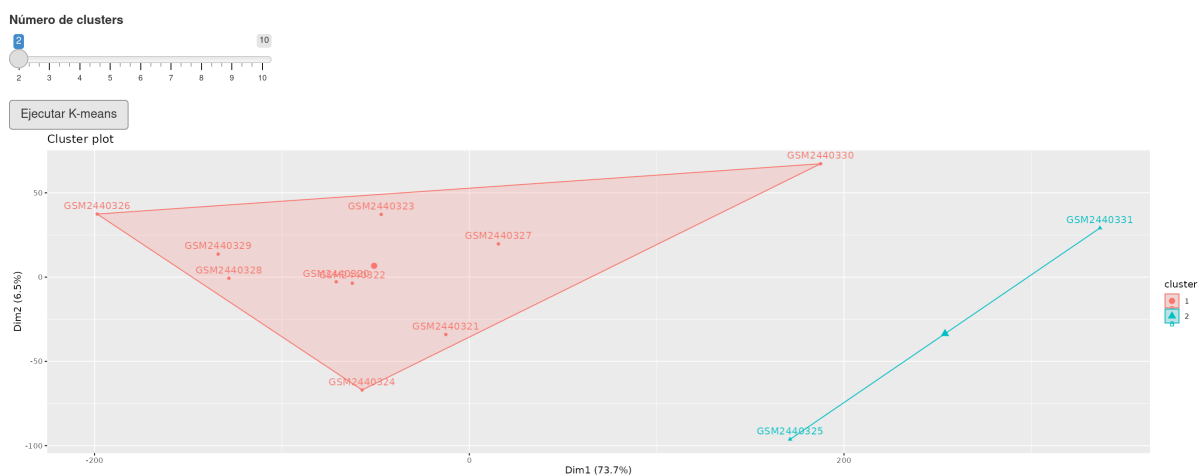


Figure 6.11: Método K-means (Fuente: elaboración propia).

6.5.8 Análisis de Expresión Diferencial

Al pulsar el botón *Expresión diferencial*, la aplicación realiza un análisis estadístico para identificar genes cuya expresión varía significativamente entre condiciones.

La tabla resultante muestra las siguientes columnas principales:

6 Aplicación para el análisis multivariante de datos transcriptómicos (RNA-Seq): diseño y desarrollo

- *baseMean*: expresión promedio del gen en todas las muestras.
- *log2FoldChange*: cambio en la expresión entre condiciones (en escala logarítmica base 2).
- *lfcSE*: error estándar del cambio estimado.
- *stat*: estadístico del test de comparación.
- *pvalue*: probabilidad de que el cambio sea por azar.
- *padj*: p-valor ajustado para controlar falsos positivos; valores menores a 0.1 indican significancia.

Esta información le permitirá identificar genes con cambios significativos en la expresión entre los grupos analizados.

Bibliography

- [1] Instituto Roche. *Informe Anticipando Ciencias Ómicas*. Último acceso: 14 de enero de 2025. 2025. URL: https://www.institutoroche.es/static/archivos/Informes_anticipando_CIENCIAS_OMICAS.pdf.
- [2] Universidad Nacional Autónoma de México. *Revista UNAM Vol. 18 Núm. 7 Artículo 54*. Último acceso: 14 de enero de 2025. 2025. URL: <https://revista.unam.mx/vol.18/num7/art54/index.html?>.
- [3] Universidad Nacional Autónoma de México. *Artículo en Revista UNAM*. Último acceso: 15 de enero de 2025. 2017. URL: https://www.revista.unam.mx/vol.18/num7/art54/PDF_art54.pdf.
- [4] Adi L. Tarca et al. *Machine Learning and Its Applications to Biology*. © 2007 Public Library of Science. 2007. URL: <https://doi.org/10.1371/journal.pcbi.0030116>.
- [5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. © 2009 Springer Science+Business Media, LLC. New York, 2009. URL: <https://link.springer.com/book/10.1007/978-0-387-84858-7>.
- [6] Minoru Kanehisa et al. "KEGG: new perspectives on genomes, pathways, diseases and drugs". In: *Nucleic Acids Research* 45.Database issue (2017). Published online 29 November 2016, pp. D353–D361. DOI: [10.1093/nar/gkw1092](https://doi.org/10.1093/nar/gkw1092). URL: <https://doi.org/10.1093/nar/gkw1092>.
- [7] Hiromi W. L. Koh et al. *iOmicsPASS: network-based integration of multiomics data for predictive subnetwork discovery*. © 2021 Nature Publishing Group. 2021. URL: <https://www.nature.com/npjbsa>.
- [8] Wolfgang Huber et al. "Orchestrating high-throughput genomic analysis with Bioconductor". In: *Nature Methods* 12.2 (2015), pp. 115–121. DOI: [10.1038/nmeth.3252](https://doi.org/10.1038/nmeth.3252). URL: <https://doi.org/10.1038/nmeth.3252>.
- [9] Guillermo Ayala. *Bioinformática Estadística*. Edición digital en PDF. Valencia, España, 2023.
- [10] National Human Genome Research Institute. *Tecnología de microarrays (chips de ADN o ARN)*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.genome.gov/es/genetics-glossary/Tecnolog%C3%ADa-de-microarrays-chips-de-ADN-o-ARN>.
- [11] National Human Genome Research Institute. *Exón*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.genome.gov/genetics-glossary/Exon>.

Bibliography

- [12] National Human Genome Research Institute. *Intrón*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.genome.gov/es/genetics-glossary/Intron>.
- [13] National Human Genome Research Institute. *Expresión génica*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.genome.gov/es/genetics-glossary/Expresion-genica>.
- [14] National Human Genome Research Institute. *Fenotipo*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.genome.gov/es/genetics-glossary/Fenotipo>.
- [15] ZhiCheng Dong and Yan Chen. *Transcriptomics: Advances and Approaches*. Recibido el 14 de agosto de 2013; aceptado el 6 de septiembre de 2013. Guangzhou 510650, China, 2013.
- [16] Mark Gerstein Zhong Wang and Michael Snyder. *RNA-Seq: a revolutionary tool for transcriptomics*. Publicado en versión final editada en enero de 2009. Department of Molecular, Cellular et al., 2009. DOI: [10.1038/nrg2484](https://doi.org/10.1038/nrg2484).
- [17] Instituto Roche. *Contig*. Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.institutoroche.es/recursos/glosario/contig>.
- [18] Your Genome. *What is RNA Splicing?* Último acceso: 23 de febrero de 2025. 2025. URL: <https://www.yourgenome.org/theme/what-is-rna-splicing/>.
- [19] Ali Mortazavi et al. *Mapping and quantifying mammalian transcriptomes by RNA-Seq*. © 2008 Nature Publishing Group. 2008. URL: <http://www.nature.com/naturemethods>.
- [20] Mark D. Robinson Alicia Oshlack and Matthew D. Young. *From RNA-seq reads to differential expression results*. REVIEW. 2010. URL: <http://genomebiology.com/2010/11/12/220>.
- [21] Ashraful Haque et al. *A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications*. REVIEW, Open Access. 2017. DOI: [10.1186/s13073-017-0467-4](https://doi.org/10.1186/s13073-017-0467-4).
- [22] Dragomirka Jovic et al. *Single-cell RNA sequencing technologies and applications: A brief overview*. REVIEW, Recibido: 5 de agosto de 2021; Revisado: 9 de diciembre de 2021; Aceptado: 20 de diciembre de 2021. 2021. DOI: [10.1002/ctm2.694](https://doi.org/10.1002/ctm2.694).
- [23] Broad Institute. *Processing Single-Cell RNA-Seq Data*. Último acceso: 23 de febrero de 2025. 2019. URL: https://broadinstitute.github.io/2019_scWorkshop/processing-scrnaseq-data.html.
- [24] C. Radhakrishna Rao. *Multivariate Analysis: Some Reminiscences on Its Origin and Development*. 1983. URL: <https://www.jstor.org/stable/25052296>.
- [25] Mushtak A.K. Shiker. *Multivariate Statistical Analysis*. 2012. URL: <https://www.britishjournalofscience.com>.

Bibliography

- [26] Alvin C. Rencher and William F. Christensen. *Methods of Multivariate Analysis*. 3rd. Wiley Series in Probability and Statistics. Editors: David J. Balding, Noel A. C. Cressie, Garrett M. Fitzmaurice, Harvey Goldstein, Iain M. Johnstone, Geert Molenberghs, David W. Scott, Adrian F. M. Smith, Ruey S. Tsay, Sanford Weisberg. Provo, Utah: John Wiley Sons, Inc., 2012.
- [27] José Luis Romero Béjar and Carlos Francisco Salto Díaz. *Tema 3.- Análisis de componentes principales (ACP)*. Asignatura: Estadística Multivariante (Prácticas). Grados en: Física y Matemáticas; Ingeniería Informática y Matemáticas; Matemáticas (4º Curso - 1er semestre 2023-2024). 2023. URL: <https://digibug.ugr.es/bitstream/handle/10481/85857/ACP.pdf?sequence=1&isAllowed=y>.
- [28] José Luis Romero Béjar and Carlos Francisco Salto Díaz. *Tema 4.- Análisis factorial (AF)*. Asignatura: Estadística Multivariante (Prácticas). Grados en: Física y Matemáticas; Ingeniería Informática y Matemáticas; Matemáticas (4º Curso - 1er semestre 2023-2024). 2023. URL: <https://digibug.ugr.es/bitstream/handle/10481/85859/AF.pdf?sequence=1&isAllowed=y>.
- [29] Neil H. Timm. *Applied Multivariate Analysis*. Primera edición. 2002.
- [30] Wolfgang Karl Härdle and Léopold Simar. *Applied Multivariate Statistical Analysis*. Library of Congress Control Number: 2015933294. 2015. DOI: [10.1007/978-3-662-45171-7](https://doi.org/10.1007/978-3-662-45171-7).
- [31] Brian S. Everitt et al. *Cluster Analysis*. Quinta edición. 2011.
- [32] José Luis Romero Béjar and Carlos Francisco Salto Díaz. *Tema 6.- Análisis cluster (AC)*. Material protegido por la Licencia Creative Commons CC BY-NC-ND. 2023. URL: <https://digibug.ugr.es/bitstream/handle/10481/85861/AC.pdf?sequence=1&isAllowed=y>.
- [33] Usuario de Stack Overflow. *Adding labels to cluster*. Pregunta y respuestas en la plataforma Stack Overflow. 2018. URL: <https://stackoverflow.com/questions/50107157/adding-labels-to-cluster>.
- [34] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Primera edición. 1988.
- [35] Rafael Payá Albert. *Definición de norma*. Material de Análisis I. 2015. URL: <https://www.ugr.es/~rpaya/documentos/AnalisisI/2014-15/Normados.pdf>.
- [36] Universidad de Granada. *Técnicas de clustering y la métrica de Mahalanobis*. Material de estudio sobre clustering. 2025. URL: <https://www.ugr.es/~gallardo/pdf/cluster-2.pdf>.
- [37] Alan Julian Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Series Editors: G. Casella, S. Fienberg, I. Olkin. 2008.
- [38] José Gallardo. *Análisis de conglomerados (Cluster Analysis)*. Apuntes del Departamento de Estadística e Investigación Operativa, Universidad de Granada. 2006. URL: <https://www.ugr.es/~gallardo/pdf/cluster-3.pdf>.

Bibliography

- [39] José Luis Romero Béjar and Guillermo Arturo Cañadas De la Fuente. *Análisis Cluster - Práctica 3*. Apuntes de práctica, Universidad de Granada. 2024.
- [40] Patricia Quesada. *Análisis Multivariante de Datos Ómicos*. Trabajo Fin de Grado, Universidad de Granada. 2024. URL: file:///home/quintin/Descargas/0_20240609%20TFG_Quesada_Patricia_Final.pdf.
- [41] Shie Mannor. *k-Armed Bandit*. Israel Institute of Technology, Haifa, Israel. 2010.
- [42] Emily Clough and Tanya Barrett. *The Gene Expression Omnibus database*. Methods Mol Biol, 1418: 93–110. Disponible en PMC: 2016-07-14. doi:10.1007/978-1-4939-3578-9_5. National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health. July 2016.
- [43] Ron Edgar, Michael Domrachev, and Alex E. Lash. *Gene Expression Omnibus: NCBI gene expression and hybridization array data repository*. Nucleic Acids Research, 30(1): 207–210. National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health. Recibido: 15 de agosto de 2001; Aceptado: 10 de octubre de 2001. Jan. 2002.
- [44] Tanya Barrett. *Gene Expression Omnibus (GEO)*. En: The NCBI Handbook [Internet]. 2ª edición. Bethesda (MD): National Center for Biotechnology Information (US). Disponible en: <https://www.ncbi.nlm.nih.gov/books/>. May 2013.
- [45] Tanya Barrett and Ron Edgar. *Gene Expression Omnibus: Microarray Data Storage, Submission, Retrieval, and Analysis*. Methods in Enzymology, vol. 411, pp. 352–369. DOI: 10.1016/S0076-6879(06)11019-8. 2006.
- [46] National Center for Biotechnology Information. *Gene Expression Omnibus (GEO) Overview*. Disponible en: <https://www.ncbi.nlm.nih.gov/geo/info/overview.html>. May 2024.
- [47] InfraNetworking. *¿Qué es un servidor FTP y para qué sirve?* Disponible en: <https://blog.infranetworking.com/servidor-ftp/>. Jan. 2023.
- [48] Wikipedia contributors. *Curación de datos*. Disponible en: https://es.wikipedia.org/wiki/Curaci%C3%B3n_de_datos. May 2024.
- [49] National Center for Biotechnology Information. *GEO Browse*. Disponible en: <https://www.ncbi.nlm.nih.gov/geo/browse/>. May 2025.
- [50] National Center for Biotechnology Information. *GEO Query*. Disponible en: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi>. 2025.
- [51] National Center for Biotechnology Information. *Advanced Search - GEO DataSets*. Accedido: 19 de mayo de 2025. May 2025. URL: <https://www.ncbi.nlm.nih.gov/gds/advanced>.
- [52] National Center for Biotechnology Information. *Profile Graph - GDS4515:202859_x_at*. Disponible en: https://www.ncbi.nlm.nih.gov/geo/tools/profileGraph.cgi?ID=GDS4515:202859_x_at. May 2025.
- [53] Bioconductor. *About Bioconductor*. Disponible en: <https://bioconductor.org/about/>. May 2025.

Bibliography

- [54] National Cancer Institute Center for Cancer Research. *Lesson 8: Data Frames — Introduction to R*. Disponible en: https://bioinformatics.ccr.cancer.gov/docs/rintro/Lesson_8/. May 2025.
- [55] Sean Davis and Paul S. Meltzer. *GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor*. *Bioinformatics*, vol. 23, no. 14, pp. 1846–1847. DOI: <https://doi.org/10.1093/bioinformatics/btm254>. July 2007.
- [56] Mohammad Ikbal Hossain. *Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management*. *International Journal for Multidisciplinary Research (IJFMR)*, vol. 5, no. 5. E-ISSN: 2582-2160. Disponible en: <https://www.ijfmr.com/research-paper.php?id=6223>. Sept. 2023.
- [57] Brunil Romero. *Práctica 2: Analizar y diseñar el sistema de información de la empresa creada*. Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Granada. Curso 2023–2024. 2024.
- [58] PBC Posit. *Shiny - Lesson 1: Basic App Structure*. Disponible en: <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/>. May 2025.
- [59] PBC Posit. *Shiny Gallery - Explore Examples of Shiny Apps*. Disponible en: <https://shiny.posit.co/r/gallery/>. May 2025.
- [60] PBC Posit. *Using HTML Tags in Shiny*. Disponible en: <https://shiny.posit.co/r/articles/build/html-tags/>. May 2025.
- [61] Hadley Wickham. *Mastering Shiny - Layouts*. Disponible en: <https://mastering-shiny.org/action-layout.html>. May 2025.
- [62] The Programming Historian. *Creación de una aplicación web con Shiny en R*. Disponible en: <https://programminghistorian.org/es/lecciones/creacion-de-aplicacion-shiny>. May 2025.
- [63] PBC Posit. *Shiny Layout Guide*. Disponible en: <https://shiny.posit.co/r/articles/build/layout-guide/>. May 2025.