TÉCNICAS MULTIVARIANTES PARA EL ANÁLISIS DE DATOS GENÓMICOS

DAVID SUÁREZ GONZÁLEZ

Facultad de Ciencias y E.T.S. de Ingenierías Infomática y Telecomunicación,
Universidad de Granada.

JOSÉ LUIS ROMERO BÉJAR

Departamento de Estadística e Investigación Operativa,
Facultad de Ciencias, Universidad de Granada.

Mayo de 2024

Técnicas Multivariantes para el Análisis de Datos Genómicos

David Suárez González

José Luis Romero Béjar

Técnicas Multivariantes para el Análisis de Datos Genómicos

Resumen

En un mundo donde la cantidad de datos genómicos disponibles crece exponencialmente, el análisis de esta información se ha convertido en un campo de investigación esencial para comprender la complejidad de la genética y la biología molecular. La genómica ha revolucionado la comprensión de la biología y la medicina, proporcionando una gran cantidad de datos que requieren un enfoque multidisciplinar para su análisis, donde confluyen tres campos como son la Informática, las Matemáticas y la Biología. El presente texto tiene como objetivo abordar esta creciente demanda de herramientas analíticas precisas y efectivas en el ámbito de la informática y las matemáticas.

En este trabajo se presenta un estudio sobre las bases formales matemáticas de las técnicas multivariantes, profundizando en el análisis de componentes principales (*PCA*) y en el análisis clúster (*AC*), destacando los supuestos necesarios y los desarrollos formales para la obtención de parámetros cuando corresponda. Este enfoque permitirá al lector adquirir el conocimiento necesario para abordar la complejidad de los datos genómicos y desempeñar un papel significativo en la investigación y la toma de decisiones en este campo en constante evolución.

Desde el punto de vista de la bioinformática, este trabajo se enfoca en capacitar al lector en el manejo de las complejas estructuras de datos genómicos, así como en la implementación de diversas técnicas multivariantes, con el objetivo de analizar estos datos. Se introducirá al lector al repositorio de datos genómicos *Gene Expression Omnibus* y a paquetes de *R/Bioconductor* con el fin de prepararlo en el marco informático para que el análisis final sea satisfactorio.

Finalmente, se ilustrará un ejemplo práctico en el que se realizará un análisis exploratorio de datos genómicos, donde confluirán todos los conocimientos adquiridos en este trabajo, con el fin de cargar satisfactoriamente un conjunto de datos genómicos, preprocesarlos para su posterior análisis y aplicarle a este conjunto técnicas de aprendizaje automático como la reducción de dimensiones y el *clustering* con el propósito de extraer conlusiones sobre el experimento estudiado.

Palabras Clave

Análisis Clúster, Análisis de Componentes Principales, Bioconductor, Datos Genómicos, RNA-Seq

Multivariate Methods Applied to Genomic Data Analysis

Abstract

In a world where the volume of available genomic data is growing exponentially, the analysis of this information has become a crucial field of research to understand the complexity of genetics and molecular biology. Genomics has revolutionized the understanding of biology and medicine, providing a wealth of data that requires a multidisciplinary approach for analysis, where computer science, mathematics, and biology converge. The aim of this thesis is to address the increasing demand for precise and effective analytical tools in the field of computer science and mathematics.

This work presents a study on the formal mathematical foundations of multivariate techniques, delving into principal component analysis (*PCA*) and cluster analysis (*CA*), highlighting the necessary assumptions and formal developments for parameter determination when appropriate. This approach will allow readers to gain the knowledge required to tackle the complexity of genomic data and play a significant role in research and decision-making in this rapidly evolving field.

From a computer science perspective, this work focuses on training the reader to handle complex genomic data structures and implement various multivariate techniques to analyze this data. The reader will be introduced to the genomic data repository *Gene Expression Omnibus* and *R/Bioconductor* packages to prepare them in the computational framework for successful analysis.

Finally, an exploratory analysis of genomic data will be conducted, where all the knowledge acquired in this work will come together with the aim of successfully uploading a genomic data set, preprocessing it for further analysis, and applying machine learning techniques such as dimensionality reduction and clustering to draw conclusions about the studied experiment.

Keywords

Bioconductor, Cluster Analysis, Genomic Data, Principal Component Analysis, RNA-Seq

ÍNDICE GENERAL

1. ESTADO DEL ARTE 9
2. OBJETIVOS 12

I. DATOS GENÓMICOS 13
3. INTRODUCCIÓN 14
4. DATOS ÓMICOS 17
   4.1. Estructura de los datos . . . 17
   4.2. Problema estadístico . . . 19

II. FUNDAMENTOS MATEMÁTICOS 21
5. TÉCNICAS MULTIVARIANTES 22
   5.1. Introducción . . . 22
      5.1.1. Problema de la reducción de la dimensión . . . 24
   5.2. Análisis de Componentes Principales (PCA) . . . 25
      5.2.1. Introducción . . . 26
      5.2.2. Aspectos formales . . . 27
      5.2.3. Criterios de elección de componentes principales . . . 32
      5.2.4. Análisis *PCA* . . . 34
   5.3. Análisis Clúster (AC) . . . 43
      5.3.1. Introducción . . . 43
      5.3.2. Medidas de proximidad . . . 44
      5.3.3. Métodos jerárquicos . . . 50
      5.3.4. Métodos no jerárquicos . . . 63

III. FUNDAMENTOS INFORMÁTICOS 68
6. PROYECTO GEO (GENE EXPRESSION OMNIBUS) 69
   6.1. Finalidad y alcance de Gene Expression Omnibus (GEO) . . . 70
   6.2. Estructura . . . 71
      6.2.1. Datos proporcionados por investigadores . . . 71
      6.2.2. DataSets construidos por GEO . . . 72
   6.3. Envío de datos . . . 73
   6.4. Búsqueda y descarga de datos . . . 76
7. ESTRUCTURAS DE ALMACENAMIENTO 77
   7.1. Microarrays . . . 77
   7.2. RNA-Seq . . . 79

7.2.1. Formatos . . . 79
7.3. *sc*RNA-Seq (Single-Cell RNA-Seq) . . . 80
8. R / BIOCONDUCTOR 82
8.1. Software R . . . 82
8.1.1. Importancia de R en el Análisis de Datos Genómicos . . . 82
8.2. Bioconductor . . . 83
8.2.1. Importancia General de Bioconductor . . . 84
8.3. Paquetes . . . 84
8.3.1. Paquetes de *CRAN* . . . 85
8.3.2. Paquetes de Bioconductor . . . 93

IV. APLICACIÓN 103
9. ANÁLISIS MULTIVARIANTE APLICADO AL TRATAMIENTO DE DATOS GENÓMICOS MEDIANTE RNA-SEQ 104
9.1. Datos utilizados . . . 104
9.2. Cuantificación de los datos de entrada para DESeq2 . . . 105
9.2.1. Lectura de los datos con *tximeta* . . . 106
9.2.2. *SummarizedExperiment* . . . 108
9.3. Objeto DES*eq*DataSet . . . 111
9.4. Análisis exploratorio y PCA . . . 113
9.4.1. *Variance stabilizing transformation* (VST) & Regularized-Logarithm Transformation (rlog) . . . 113
9.4.2. Distancias entre muestras . . . 116
9.4.3. Análisis de Componentes Principales (PCA) . . . 119
9.5. Análisis de Expresión diferencial . . . 122
9.6. Gene Clustering . . . 124
10. CONCLUSIONES Y TRABAJO FUTURO 127
10.1. Repaso de los objetivos del trabajo . . . 127
10.2. Trabajo futuro . . . 129

1

ESTADO DEL ARTE

A lo largo de la historia de la ciencia, el estudio de los organismos vivos ha sido una búsqueda constante para comprender las complejidades de la vida en sus niveles más fundamentales. Desde la invención del microscopio por *Antonie van Leeuwenhoek* en el siglo XVII, que permitió la observación de microorganismos por primera vez, hasta la era contemporánea de la genómica y las ciencias ómicas, se ha recorrido un fascinante viaje de descubrimiento. En este contexto, las ciencias ómicas han emergido como un pilar fundamental en la exploración de la biología y la genética moderna.

HISTORIA DE LAS CIENCIAS ÓMICAS

Antes de continuar, y para reforzar la comprensión a lo largo del trabajo, *ómicas* es como se conocen de manera informal a las distintas ramas de la ciencia que comprenden varias disciplinas de la biología cuyos nombres comparten el sufijo *-ómica*. Estas disciplinas incluyen la genómica, proteómica, metabolómica, metagenómica, fenómica y transcriptómica [1].

Durante el siglo XX, los investigadores se enfrentaron a la escasez de datos en sus esfuerzos por avanzar en estas áreas. En consecuencia, gran parte de estos esfuerzos en investigación se centraron en la generación de datos, cuyo volumen resultante era aún manejable. El análisis de estos datos se realizaba de manera manual o con herramientas informáticas y técnicas estadísticas básicas. Sin embargo, en las últimas dos décadas, se han producido avances tecnológicos significativos que han desencadenado una revolución en estos campos. La tecnología actual permite la observación y medición de sistemas biológicos con una precisión sin precedentes, y esto se logra a un costo asequible que continúa disminuyendo [2].

Importancia de las Ciencias Ómicas

Las ciencias ómicas trabajan como un detective molecular que nos ayuda a entender cómo funcionan nuestros cuerpos y cómo nos adaptamos a diferentes situaciones. Nos permiten descubrir los secretos detrás de la salud, la enfermedad y cómo evolucionamos. Los datos generados por estas técnicas no solo han transformado nuestra comprensión de la biología, sino que también tienen un impacto significativo en campos como la medicina, la biotecnología y la conservación de la biodiversidad.

A lo largo de los años, numerosos trabajos y proyectos han demostrado el potencial de las ciencias ómicas. Uno de los hitos más destacados fue el *Proyecto del Genoma Humano*. Este supuso un punto de inflexión, al poner a disposición de la comunidad científica la primera secuencia completa de un genoma humano en 2003, hito alcanzado tras un esfuerzo internacional de más de 300 millones de dólares (hoy día, secuenciar un genoma humano cuesta en torno a los 1,000 dólares). Este acontecimiento dio inicio a la denominada era post-genómica [3]. Otro ejemplo es el *Proyecto 1000 Genomas*, que catalogó las variaciones genéticas en poblaciones humanas a nivel global [4].

En este trabajo confluyen tres campos diferentes como son la matemática, la informática y la biología. Tiene especial interés y dificultad añadida puesto que el lenguaje inherente a cada disciplina debe ser integrado con las otras.

ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

Dentro de este vasto panorama de las ciencias ómicas, dos técnicas fundamentales han desempeñado un papel esencial en la exploración y el análisis de datos. La primera de estas es el Análisis de Componentes Principales (*PCA*), desarrollado por Karl Pearson en el inicio del siglo XX, publicando "On Lines and Planes of Closest Fit to Systems of Points in Space" [5] en 1901. El *PCA* ha permitido la reducción de la dimensionalidad de datos complejos, preservando información esencial y simplificando la interpretación.

Más tarde, en 1933, Harold Hotelling publicó "Analysis of a Complex of Statistical Variables into Principal Components" [6]. Hotelling extendió el trabajo de Pearson y formalizó el *PCA* tal como lo conocemos hoy en día, estableciendo sus fundamentos matemáticos.

A día de hoy, existen multitud de trabajos que tratan y profundizan sobre el *PCA*, destacando entre ellos los utilizados en esta sección del trabajo, "Applied Multivariate Analysis" [7] de Neil H. Timm; "Methods of Multivariate Analysis" [8] de Alvin C. Rencher y William F. Christensen; y "Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning" [9] de Alan J. Izenman.

ANÁLISIS CLÚSTER (AC)

El análisis clúster (*AC*) tiene sus raíces en la taxonomía –clasificación ordenada y jerárquica– biológica y la clasificación de organismos en grupos basados en similitudes morfológicas o características observables, y ha evolucionado hasta las técnicas modernas que identifican patrones de similitud y agrupaciones en datos biológicos.

Un trabajo de mucha relevancia histórica en el *AC* es "Some Methods for Classification and Analysis of Multivariate Observations" [10] de John MacQueen, donde introdujo el término *k-means* y describió este algoritmo ampliamente utilizado en análsisis clúster.

Además, relacionado con el tema principal de este trabajo, estudios como el *Proyecto Microbioma Humano* –que han revelado la riqueza de microorganismos que coexisten con nosotros y su influencia en la salud y la enfermedad– han aplicado exitosamente técnicas de *clustering* para identificar grupos microbianos asociados a diferentes condiciones de salud.

De entre la enorme cantidad de trabajos sobre *AC*, destacamos los utilizados para la redacción de esta memoria "Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning" [9] de Alan J. Izenman y "Applied Multivariate Statistical Analysis" [11] de W. Karl Hardle y Léopold Simar.

2

OBJETIVOS

Nuestro trabajo se inscribe en el contexto de una línea de investigación que se enfoca en explorar las posibilidades que ofrecen distintas técnicas de análisis multivariante en el análisis de datos genómicos, como se sugiere en el título del trabajo.

Los objetivos que hemos definido están alineados con esta temática, con la intención de concluir el proyecto con un sólido conocimiento de las ciencias ómicas incluidas en la informática, y qué técnicas informáticas y modelos matemáticos podemos aplicar sobre los datos que estas manejan. Con este enfoque en mente, hemos identificado los siguientes objetivos que guiarán el desarrollo del proyecto:

OBJ-1. Identificar patrones en los datos y la elaboración de predicciones basadas en modelos probabilísticos.

OBJ-2. Estudiar alguna/s de las diferentes técnicas de aprendizaje automático habituales en este contexto, como son la regresión lineal, la regresión logística, los árboles de decisión, las redes neuronales, reducción de la dimensión, *clustering*, entre otros, y se evaluará su eficacia en la clasificación de datos genómicos.

OBJ-3. Repaso de las bases teóricas de las técnicas multivariantes y los fundamentos matemáticos asociados a ellas.

OBJ-4. Aprender el manejo de estructuras de datos genómicos, comprender la implementación de distintas técnicas multivariantes (preferiblemente en *Python* o Lenguaje *R*), y aplicarlas a un conjunto de datos.

Al llegar al cierre de este trabajo, en su sección de conclusiones, se llevará a cabo un examen sobre si se lograron o no los cuatro objetivos previamente mencionados.

Parte I

DATOS GENÓMICOS

Se introduce el tipo de datos sobre el que versará el trabajo y su estructura, así como distintos problemas que se abordan desde el punto de vista del análisis de grupos de genes y expresiones diferenciales.

3

INTRODUCCIÓN

La Bioinformática se originó de la necesidad de desentrañar el código genético, y su surgimiento está ligado a la creciente cantidad de datos generados por la Biología Molecular. En sus inicios, esta última disciplina fue pionera en la producción de datos que exigían un enfoque innovador para su interpretación. El análisis de secuencias genéticas representó el primer y fundamental desafío que condujo al desarrollo de la Bioinformática, impulsado por la creciente demanda de capacidades analíticas avanzadas.

A lo largo de los años, el interés en el estudio de las células ha ido en constante aumento. Esto se debe a que las células desempeñan un papel crucial en el proceso de transmisión de la información hereditaria de una generación a otra. En la actualidad, se comprende que los genes[1] son los responsables de esta transmisión de información genética. Sin embargo, es importante destacar que este proceso no se trata de una acción aislada, sino del resultado de una compleja serie de fases llevadas a cabo por las dos principales moléculas en el ser humano: *DNA*[2] y *RNA*[3].

A medida que la investigación avanzaba, se lograban mayores avances y se desvelaban nuevas características relacionadas con el proceso en cuestión. Sin embargo, no fue hasta después de 1950 cuando los científicos finalmente descubrieron la estructura del *DNA*. Rosalind Franklin, una destacada química, fue una de las pioneras que contribuyó significativamente a este descubrimiento. Trabajando en colaboración con otros investigadores notables como Watson y Crick, lograron establecer la forma icónica del *DNA* de doble hélice (ver Figura 1) [12].

1 Que no son otra cosa de segmentos de *DNA*.

2 *Deoxyribonucleic Acid*. En español, Ácido Desoxirribonucleico, más conocido como *ADN*. A lo largo de este texto me referiré a él como *DNA*.

3 *Ribonucleic Acid*. En español, Ácido Ribonucleico, más conocido como *ARN*. A lo largo de este texto me referiré a él como *RNA*.

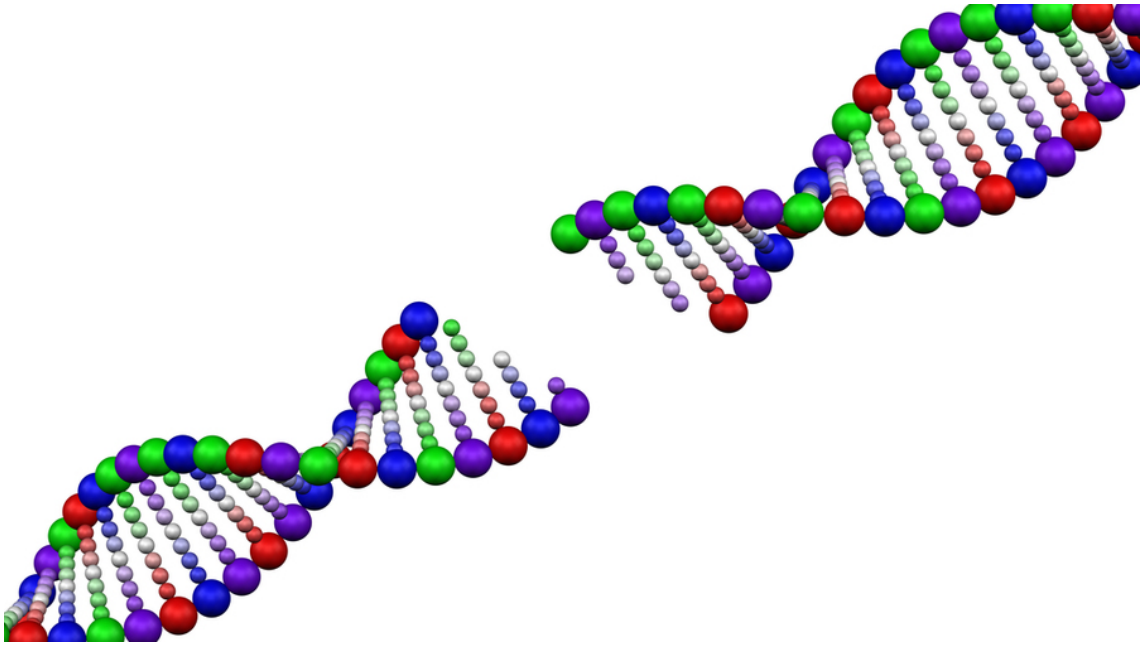

Figura 1: Figura de la estructura del *DNA* extraída de [13]

Este descubrimiento también amplió nuestra comprensión de la transmisión genética. El *DNA*, como se reveló, está compuesto por cuatro tipos de nucleótidos[4]: Adenina (*A*), Timina (*T*), Guanina (*G*) y Citosina (*C*). Su unión en la estructura de hélice mencionada anteriormente es complementaria, donde *A* se combina con *T* y *G* con *C* a través de puentes de hidrógeno. De esta forma se define la característica estructural del *DNA*.

Este conocimiento del emparejamiento de nucleótidos permite descifrar la secuencia de una hebra de *DNA* al conocer su cadena complementaria. Este proceso, conocido como hibridación, sienta las bases para diversas técnicas que se utilizan para analizar secuencias específicas de *DNA* y *RNA* [15].

El *RNA*, a diferencia del *DNA*, consiste en una sola cadena de nucleótidos. Dado que las secuencias de *DNA* son complementarias, poseer una de ellas implica tener la otra. Por lo tanto, el *RNA* está compuesto únicamente por una de las hebras[5]. Su función principal es transcribir la información genética contenida en el *DNA*. Los nucleótidos en el *RNA* son los mismos que en el *DNA*, con la excepción de que la Timina (*T*) en el *DNA* es reemplazada por el Uracilo (*U*) en el *RNA*.

4 Unidades y productos químicos que se unen para formar los ácidos nucleicos, principalmente *RNA* y *DNA* [14].

5 Cabe destacar, aunque no sea objeto de este trabajo estudiarlo más a fondo, que las secuencias son unidireccionales, cosa que en primera instancia, puede resultar, al menos, curioso.

Así pues, el *DNA* cumple un rol crucial al utilizar el proceso de transcripción para generar *RNA*, y en este proceso codifica las instrucciones necesarias para la síntesis de proteínas, asignando códigos específicos en sus segmentos para cada una de ellas. Estos segmentos, previamente mencionados como genes, en su conjunto conforman el genoma humano. La exploración del genoma humano ha generado una colaboración interdisciplinaria que ha llevado al análisis de las secuencias de *RNA*, conocido como transcriptoma, y esto ha permitido descubrir cómo los genes se activan y se expresan en el interior de una célula [16].

Ahora nos preguntamos, ¿qué es un transcriptoma? El genoma humano se compone de *DNA*, una molécula larga y enrollada que contiene las instrucciones necesarias para la creación y el mantenimiento de las células. Estas instrucciones se detallan en forma de pares de bases, que, como hemos explicado, están compuestos por cuatro sustancias químicas diferentes y complementarias, y organizadas en 20.000 a 25.000 genes. Para que estas instrucciones se pongan en práctica, el *DNA* debe ser *leído* y transcrito, es decir, copiado de alguna forma para crear *RNA*. Estas *lecturas* de los genes se conocen como transcritos, y un transcriptoma es una recopilación de todas las lecturas de genes presentes en una célula.

Una secuencia de *RNA* refleja la secuencia de *DNA* del cual se transcribió. Por lo tanto, al analizar la colección completa de secuencias de *RNA* en una célula (el transcriptoma), los investigadores pueden determinar cuándo y dónde se activa o desactiva cada gen en las células y tejidos de un organismo. Según la técnica empleada, es posible contar el número de transcritos para evaluar la cantidad de actividad de los genes, lo que se conoce como expresión génica, en un tipo específico de célula o tejido [17].

La transcriptómica, en su esencia, se centra en la evaluación de los niveles de actividad genética en los tejidos. Su principal objetivo radica en la comprensión de los motivos subyacentes a las variaciones genéticas en diversos tipos de células y su repercusión en el desarrollo de afecciones, como el cáncer [18].

Las ciencias ómicas han abierto la puerta a la generación masiva de datos. Acompañadas por notables avances en matemáticas computacionales, han allanado el camino para el uso de estrategias de Aprendizaje Automático (*Machine Learning*) y Big Data en el análisis y estudio de sistemas biológicos. Estos avances han logrado resultados que hace apenas dos décadas hubieran parecido inalcanzables. Además, han ampliado la influencia de la Bioinformática hacia otras áreas de investigación.

4

DATOS ÓMICOS

En el ámbito de la Biología, se ha acuñado el término *ómicas* para referirse a diversos procedimientos de obtención de información. Los datos ómicos representan un tipo específico de datos de alta dimensión, generados a través de tecnologías de alto rendimiento en varias disciplinas científicas ómicas. Este trabajo se centrará en la aplicación de métodos estadísticos para el análisis de datos de alto rendimiento.

4.1 ESTRUCTURA DE LOS DATOS

En este enfoque, se analizará un conjunto reducido de muestras y dentro de cada una se observarán un gran número de características, lo que implica que se está trabajando con muestras de alta dimensionalidad. Las características que se observan abarcan distintos tipos de datos, como la fluorescencia en el caso de *microarrays* de *DNA* o el número de lecturas alineadas en el contexto de procedimientos de secuenciación[1]. Estas características pueden estar asociadas a una muestra individual o a un grupo de muestras en un *microarray*. Alternativamente, la información puede estar relacionada con un gen específico o un exón[2].

El número de características observadas, que se representa como m, es considerablemente grande, alcanzando cifras en el rango de miles. Estas características serán observadas en un número reducido de muestras, que se denotan como n. El número de muestras es relativamente pequeño, en el mejor de los casos, llegando apenas a decenas.

1 En primera instancia muchos términos pueden no resultar familiares para algunos lectores, pero se irán esclareciendo a lo largo del texto.

2 Un exón es una región del genoma que finaliza con una molécula de *RNA*. Algunos exones son codificantes, es decir que contienen información para producir una proteína, mientras que otros no son codificantes. Los genes del genoma consisten en exones e intrones [19]. Para codificar el *RNA*, se utilizan solo los exones.

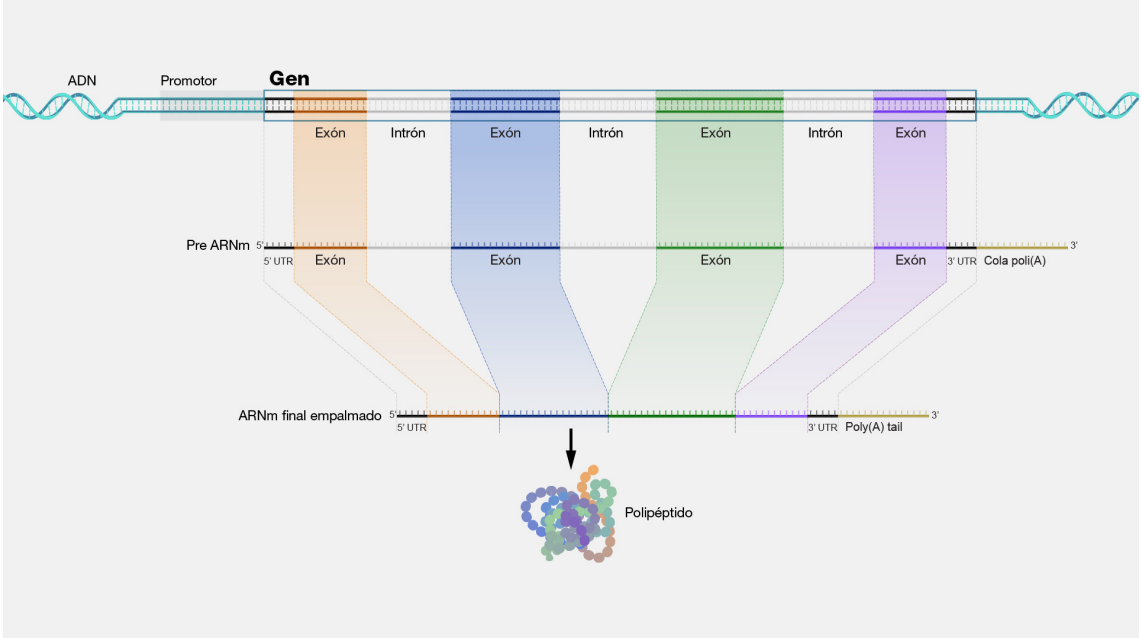

Figura 2: Figura de la transcripción de *DNA* a *RNA* extraída de [20]

De esta forma, el problema se sitúa en el contexto de la Estadística de Alta Dimensión, en la que se profundizará más adelante. En esta disciplina, m supera significativamente a n ($n \ll m$), lo que plantea un escenario novedoso en comparación con los procedimientos estadísticos tradicionales, donde, por norma general, sucede lo contrario ($n > m$). Si bien esta disparidad limita las posibilidades, también abre un nuevo y prometedor campo de investigación.

Las características se organizan en lo que se denomina una matriz de expresión, designada por

$$\boldsymbol{x} = \left[x_{ij}\right]_{i=1,\ldots,m;j=1,\ldots,n}$$

En esta matriz, el valor x_{ij} proporciona una cuantificación de la característica i en la muestra j. En el caso de que x_{ij} se refiera a un *microarray* de *DNA*, representa un nivel de fluorescencia y, por lo general, toma valores positivos. Es importante mencionar que en algunos procesos previos de manipulación de datos, como la normalización, es posible obtener expresiones negativas en lugar de valores exclusivamente positivos.

Cuando se manejan datos generados a través de la técnica de secuenciación *RNA*-Seq, aparecen recuentos, que se refieren al número de lecturas cortas que se alinean en un gen o en un exón específico. En este contexto, un mayor número de lecturas se traduce en una mayor expresión de esa característica. Los valores observados en una misma fila, que representan la misma característica en todas las muestras, suelen denominarse *perfil* en el ámbito de la transcriptómica.

En la matriz $\boldsymbol{x}$, los valores observados en diferentes muestras son independientes, aunque es posible que se hayan obtenido bajo diferentes condiciones. No se trata de réplicas de una misma condición experimental, pero cada observación es independiente de las demás en el contexto condicional. Sin embargo, es importante señalar que las filas de $\boldsymbol{x}$ son instantáneas de vectores que están correlacionados. Por ejemplo, en una matriz de expresión, los valores de expresión en las diferentes filas no son independientes, ya que los genes suelen operar de manera coordinada.

Habitualmente los datos de las columnas de la matriz $\boldsymbol{x}$ no son directamente comparables. Hay muchos artefactos técnicos, así como ruido, en la observación de la característica de interés. Se han desarrollado técnicas para corregirlo llamadas técnicas de preprocesado. Los datos después de la normalización siguen considerándose independientes por columnas (muestras) y dependientes por filas.

La información o variables que caracterizan a las muestras se conocen como metadatos o variables fenotípicas, donde un fenotipo se refiere a cualquier característica o rasgo observable de un organismo. Por lo general, intervienen varias variables fenotípicas. Se utiliza la notación $y = (y_1, \ldots, y_n)$ para representar los valores observados de una variable en las n muestras. Un escenario común de variable fenotípica se presenta cuando se trabaja con dos grupos de muestras, como casos y controles[3]. Cuando se presenta este escenario, se asigna el valor $y_i = 1$ a las muestras del grupo de casos y $y_i = 0$ a las muestras del grupo de controles. En situaciones donde hay más de dos grupos para comparar, como k grupos, se utiliza $y_i \in \{1, \ldots, k\}$ para $i = 1, ..., n$. Es importante destacar que la elección de los valores 0 y 1 es arbitraria, y se podría seleccionar cualquier otro par de valores para representar los grupos de interés.

4.2 PROBLEMA ESTADÍSTICO

El término *datos de alto rendimiento* lleva consigo un aire de enigma. Se refiere a datos que desafían lo que tradicionalmente se consideraba un requisito fundamental en estadística. Muchos procedimientos estadísticos inician con la premisa de que el número de observaciones, n, debe superar el número de variables por observación, m. No obstante, en la actualidad, es más común que los datos no sean recopilados manualmente por un investigador con lápiz y papel, sino que

3 En estudios genéticos, es común comparar dos grupos de individuos: los *casos* y los *controles*. Los *casos* son individuos que presentan una característica o condición específica que se está estudiando, como una enfermedad. Los *controles* son individuos que no tienen esa característica o condición y se utilizan como grupo de comparación para evaluar diferencias.

sean generados por dispositivos electrónicos conectados a computadoras. Esto da como resultado dimensiones de datos que pueden resultar abrumadoras: miles de variables en comparación con solo unas pocas decenas, o en el mejor de los casos, un número ligeramente superior a cien, de muestras. Ante este desafío, el análisis de datos de alto rendimiento se convierte en todo un reto y en una especie de *hacer lo que se pueda*, ya que las técnicas convencionales pueden no ser adecuadas.

En este punto se entra de lleno en la siguiente cuestión, ¿por qué los datos de alta dimensión son un problema? Cuando el número de características en un conjunto de datos supera al número de muestras disponibles, enfrentamos un escenario en el que una solución determinista es inalcanzable. En otras palabras, se vuelve imposible encontrar un modelo que pueda describir de manera precisa la relación entre las variables predictoras y la variable de respuesta, dado que no se cuenta con suficientes observaciones para entrenar dicho modelo.

Los estudios de genómica –en particular– generan datos de alta dimensión que involucran miles de genes. Estos datos suelen contener variables altamente correlacionadas, y, en general, se dispone de un número limitado de muestras para lograr una clasificación o predicción precisa. Los métodos estadísticos convencionales, como el análisis discriminante y la regresión logística, entre otros, a menudo no resultan eficaces en este contexto debido a problemas relacionados con el tamaño de la muestra y el riesgo de sobreajuste [21].

En este contexto, las técnicas estadísticas que se aplican a menudo son adaptaciones de procedimientos diseñados originalmente para situaciones donde se dispone de más muestras que variables observadas. Uno de los desafíos fundamentales que se abordarán en este trabajo es lo que se conoce como *expresión diferencial*. Aquí, nos enfocamos en una variable fenotípica y una característica en particular. La pregunta clave es si existe una asociación entre el perfil de expresión y la variable fenotípica. Este tipo de análisis se denomina *análisis de expresión diferencial marginal*. El objetivo es investigar posibles asociaciones entre grupos de características, es decir, grupos de filas en la matriz de expresión, y la variable fenotípica de interés. Esto se conoce como análisis de grupos de genes o *gene set analysis*.

Además, se abordarán problemas de reducción de dimensión, con un enfoque particular en una técnica ampliamente utilizada: el análisis de componentes principales. También se utilizará clasificación tanto de características como de muestras como una herramienta exploratoria, lo que se conoce como análisis clúster.

Parte II

FUNDAMENTOS MATEMÁTICOS

Se exponen los fundamentos matemáticos con el propósito de su comprensión teórica y su utilización posterior en la implementación práctica.

5

TÉCNICAS MULTIVARIANTES

En el presente cápitulo dedicado a la disciplina matemática, se abordarán algunas ideas fundamentales relacionadas con la estadística multivariante, además de tratar ciertos aspectos referentes a la notación. El objetivo es facilitar la comprensión del contenido que vendrá a continuación, promoviendo una lectura más fluida y amena.

5.1 INTRODUCCIÓN

Esta breve introducción ha sido escrita tomando como base las referencias bibliográficas [7] y [8].

El análisis multivariante comprende un conjunto de enfoques utilizados cuando se obtienen múltiples mediciones de cada individuo en una o más muestras. En este contexto, nos referimos a estas mediciones como variables y a los individuos u objetos como unidades (que pueden ser unidades de investigación, unidades de muestreo u observaciones). Aunque es común encontrar conjuntos de datos multivariantes en la práctica, no siempre se analizan de esta manera. Sin embargo, ya no se justifica el uso exclusivo de procedimientos univariantes con estos datos, dado que disponemos de técnicas multivariantes y de potencia informática asequible para llevar a cabo dichos análisis.

Históricamente, las técnicas multivariantes se han empleado principalmente en las ciencias biológicas y del comportamiento. No obstante, el interés por estos métodos se ha expandido hacia numerosos otros campos de investigación. Esto incluye áreas como educación, química, ciencias ambientales, física, geología, medicina, ingeniería, derecho, negocios, literatura, religión, biología, psicología y muchos otros campos.

Estas técnicas resultan de gran utilidad cuando se recopilan observaciones para un grupo de sujetos en relación con un conjunto de variables de interés, conocidas como variables dependientes, y se busca establecer relaciones entre estas variables y otro conjunto de variables, denominadas variables independientes. Los datos obtenidos suelen organizarse en una matriz en la que las filas representan las observaciones y las columnas representan las variables.

Cuando se trata de respuestas multivariantes que son muestras de una o más poblaciones, se suele asumir inicialmente que la muestra proviene de una distribución de probabilidad multivariante. En este contexto, la distribución de probabilidad multivariante más comúnmente empleada es la distribución normal multivariante (*DNM*). Los modelos simples suelen involucrar una o más medias μ_i y matrices de covarianza Σ_i.

Además de las técnicas numéricas, también es posible emplear métodos gráficos para visualizar las relaciones entre los datos. Algunas representaciones básicas incluyen gráficos de dispersión o una matriz de gráficos de dispersión que muestra simultáneamente dos o tres variables. También se pueden utilizar gráficos de perfiles, estrellas, *biplots*, *sunburst*, contornos, caras de *Chernoff* y gráficos de *Fourier* de *Andrews* para representar datos multivariantes. Dado que resulta complicado detectar y describir relaciones entre variables en espacios de alta dimensionalidad, se han desarrollado diversas técnicas multivariantes para reducir la dimensionalidad de los datos. Dos de las técnicas de reducción de datos más comúnmente utilizadas son el análisis de componentes principales y el análisis factorial.

Otro desafío frecuente en el análisis multivariante de datos es la clasificación o agrupación de objetos. Para abordar este tipo de análisis, se recurre a técnicas multivariantes que incluyen el análisis clúster, los árboles de clasificación y regresión (*CART - Classification and regresion trees*) y las redes neuronales, entre otras.

5.1.1 *Problema de la reducción de la dimensión*

El problema de la reducción de la dimensión en el ámbito de la estadística multivariante se refiere a la necesidad de reducir el número de variables o dimensiones en un conjunto de datos multivariante, mientras se conserva la mayor cantidad posible de información relevante.

La reducción de dimensiones se emplea con frecuencia como una fase de preprocesamiento en el entrenamiento de sistemas. Dado que cada característica que se incluye en el análisis puede aumentar tanto el costo como el tiempo de procesamiento de los sistemas, existe una sólida motivación para diseñar e implementar sistemas que utilicen conjuntos de características más reducidos. Al mismo tiempo, existe una necesidad contrapuesta de incorporar un conjunto suficientemente extenso de características para no perder mucha información del sistema y lograr un rendimiento óptimo.

El problema radica en encontrar el número óptimo de características[1] para expresar en mayor medida la muestra inicial pero siendo un número suficientemente pequeño para que la reducción sea significativa. Esto trae consigo una serie de ventajas muy importantes. En conjuntos de datos con muchas variables, el espacio de características puede volverse demasiado disperso, lo que dificulta la detección de patrones significativos; además, en caso de que querer hacer cálculos computacionales, el tiempo de procesamiento necesario para analizar los datos disminuyen significativamente, lo que puede ser fundamental en aplicaciones en tiempo real o con recursos limitados [22].

Esta situación ha impulsado el desarrollo de diversas técnicas que buscan encontrar este subconjunto óptimo a partir de un conjunto inicial de características, como el Análisis de Componentes Principales (*PCA*) y el Análisis Factorial (*AF*), entre muchos otros. El desarrollo de esta sección tendrá por objetivo presentar, explicar y desarrollar el primero de los dos nombrados.

1 En la gran mayoría de situaciones, no existe dicho número óptimo, la elección vendrá motivada por el contexto del problema tratado en cada ocasión.

5.2 ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

El análisis de componentes principales (*PCA*) es una técnica aplicada a una sola muestra de datos, sin agrupaciones entre las observaciones, y sin dividir las variables en subconjuntos de variables dependientes y variables independientes, cuyo objetivo es maximizar la varianza de una combinación lineal de las variables. Es importante destacar el papel que desempeñan las componentes principales (*PCs - Principal Components*), que no son otra cosa que dichas combinaciones lineales de variables del conjunto inicial.

La primera componente principal, como acabamos de mencionar, es una combinación lineal que busca maximizar la varianza, esencialmente identificando una dimensión en la que las observaciones se encuentren separadas o dispersas al máximo. La segunda componente principal es una combinación lineal que busca maximizar la varianza en una dirección *ortogonal*[2] al primer componente principal, y así sucesivamente. En términos generales, las componentes principales definen dimensiones diferentes de las que se definen mediante funciones discriminantes o variantes canónicas [8].

Por tanto, resumiendo, las componentes principales son combinaciones lineales de las variables originales que tienen la máxima varianza y son mutuamente perpendiculares. Demostraremos que los coeficientes de estas combinaciones lineales corresponden a los vectores propios de la matriz de covarianzas, y que sus respectivas varianzas se relacionan con los valores propios asociados a estos vectores propios. En consecuencia, el análisis de componentes principales (*PCA*) permite identificar las direcciones en las que la dispersión es más amplia, es decir, donde la varianza es mayor [23].

En este capítulo se realizará en la sección 5.2.1 una introducción del *PCA*. En la sección 5.2.2 se tratarán los aspectos formales del *PCA*. En la sección 5.2.3 se discutirán sobre los distintos criterios para la elección del número adecuado de componentes principales. Por último, en la sección 5.2.4 se realizará un análisis *PCA* muy simple con el *software R* para facilitar al lector una comprensión visual de todo el marco teórico.

2 En un espacio vectorial V con producto interior, dos vectores $x, y \in V$ son ortogonales si el producto escalar $\langle x, y \rangle$ es cero. Esta situación se denota por $x \perp y$.

5.2.1 *Introducción*

La referencia principal de esta sección introductoria ha sido [9]. El análisis de componentes principales (*PCA*) se introdujo como técnica para derivar un conjunto reducido de proyecciones lineales ortogonales de una única colección de variables correladas, $\mathbf{X} = (X_1, ..., X_p)$, donde las proyecciones se ordenan por varianzas decrecientes. La varianza es un momento de segundo orden de una variable aleatoria y es una medida importante que explica la cantidad de información contenida en esa variable.

El (*PCA*) tiene como objetivo esencial condensar la información proveniente de múltiples variables en un conjunto reducido de variables o combinaciones lineales que maximizan la variabilidad[3]. Principalmente, el *PCA* se emplea como una técnica para reducir la complejidad de los datos, y su mayor *utilidad* radica en su aplicación preliminar antes de emplear otros métodos estadísticos, como la regresión o el análisis clúster (*AC*) –que trataremos más adelante–, entre otros.

El análisis de componentes principales se clasifica como un método de *aprendizaje no supervisado*, lo que significa que se aplica a datos sin etiquetar ni clasificar previamente. Su objetivo principal no es predecir una variable de salida específica, como se hace en el *aprendizaje supervisado*, si no más bien descubrir patrones, estructuras y relaciones inherentes en los datos. Una desventaja importante de esta técnica reside en la complejidad que implica validar los resultados, dado que no existe una variable de respuesta para realizar comparaciones directas.

Antes de aplicar el análisis de componentes principales, es fundamental asegurarse de que se cumplan ciertos requisitos:

R-1. Los datos no deben ser incorrelados.

R-2. Se debe llevar a cabo un análisis preliminar con el propósito de detectar valores atípicos en los datos. Estos valores inusuales requieren una atención especial, ya que el análisis de componentes principales es particularmente sensible a valores extremos.

3 En el contexto del análisis de componentes principales, la *variabilidad* se refiere a la dispersión, la diferencia o la variación que existe entre los datos originales en múltiples variables. El *PCA* busca reducir la dimensionalidad de los datos al transformarlos en un conjunto más pequeño de componentes principales, de manera que estos componentes retengan la mayor parte de la información o variabilidad contenida en las variables originales.

R-3. Es aconsejable estandarizar todas las variables previamente (ajustarlas para que tengan una media de 0 y una desviación estándar de 1). Esto se hace para evitar que las variables con escalas más amplias dominen sobre las demás en el análisis.

5.2.1.1 *Sobre las componentes principales*

La primera componente principal se alinea con la dirección en la cual las mediciones tienen la mayor variabilidad. La segunda componente principal se ajusta a la dirección con la mayor dispersión en los datos y que no está relacionada con la primera –estas direcciones son mutuamente perpendiculares, como hemos explicado anteriormente–. Este proceso se repite con la tercera y las subsiguientes componentes principales.

Ahora bien, es esencial comprender cuánta información de los datos originales se pierde al proyectar las mediciones en un espacio de menor dimensión, es decir, cuánta información capturan del sistema original las diversas componentes principales obtenidas. Esta evaluación se refleja en la proporción de variabilidad explicada por cada componente principal, así como en la proporción acumulativa de variabilidad explicada.

Estos valores son cruciales al determinar cuántas componentes principales debemos considerar en nuestro análisis. Indican la cantidad de información que se retiene al reducir la dimensionalidad de los datos y, por lo tanto, son fundamentales para tomar decisiones informadas sobre qué componentes principales incluir en el análisis y cuánta información se está dispuesto a sacrificar en favor de una representación más concisa de los datos.

5.2.2 *Aspectos formales*

En esta sección, se presenta una justificación formal de cómo las componentes principales se relacionan con los vectores propios y sus respectivas varianzas, que son consideradas como los valores propios correspondientes a esos vectores. Las bibliografías principales consultadas son [7], [8] y [9].

Sean $X_1, \ldots, X_p$, un total de p variables aleatorias correladas. Se denota por $\mathbf{X} = \left(X_1, \ldots, X_p\right)^t$ al vector aleatorio resultante. Sin pérdida de generalidad, se asumirá que $\mathbf{X}$ es centrado –esto es $E[X]$[4] $= 0$– y se denotará por $\Sigma = E\left[XX^t\right]$ a su matriz de covarianzas.

4 E[X] es como se denota la esperanza matemática de un vector aleatorio.

Se están considerando variables de la forma $U_1 = a_1^t X, \ldots, U_q = a_q^t X$, donde $a_1, \ldots, a_q \in \mathbb{R}^p$ son los vectores que se están buscando. El objetivo aquí es encontrar los vectores $a_1, \ldots, a_q$ adecuados. Estos vectores se utilizan para realizar una transformación lineal de las variables originales de $\mathbf{X}$ para obtener las nuevas variables $U_1, \ldots, U_q$. Cada uno de estos vectores a_i define una combinación lineal particular de las variables originales.

Se deben considerar unos requerimientos previos a la búsqueda de estas componentes principales U_i:

- Es esencial que las variables $U_1 = a_1^t X, \ldots, U_q = a_q^t X$ sean incorreladas, ya que este requisito garantiza la eliminación de información redundante en el análisis.
- Es importante destacar que la varianza de cada U_i debe ser máxima. Esta condición asegura que las nuevas variables proporcionen la máxima cantidad de información significativa posible. En otras palabras, se busca que cada U_i capture la mayor cantidad de variabilidad presente en los datos originales de $\mathbf{X}$.

Problema de la obtención de componentes principales (PCs)

Bajo las condiciones previamente mencionadas, el objetivo consiste en encontrar las variables $U_1 = a_1^t X, \ldots, U_q = a_q^t X$ de tal manera que sean mutuamente incorreladas, y cada U_i tenga la máxima varianza posible entre todas las combinaciones lineales de $\mathbf{X}$ que sean incorreladas con las demás, esto es, con $U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_q$.

Estas variables $U_1 = a_1^t X, \ldots, U_q = a_q^t X$, que son la solución de este problema, reciben el nombre de *componentes principales*.

Resolución del problema de la obtención de componentes principales (PCs)

La resolución de este problema sigue un enfoque secuencial e inductivo. En primer lugar, se obtiene U_1 al imponer que tiene la máxima varianza. Luego, se procede a obtener U_2 asegurando que sea la que tenga la mayor varianza entre todas las combinaciones lineales incorreladas (ortogonales) a U_1, y este proceso continúa de manera iterativa. A continuación, se proporcionará una justificación sobre cómo los coeficientes de las componentes principales están relacionados con los vectores propios correspondientes a los valores propios de mayor módulo en la matriz de covarianza en cada paso.

Paso 1

En el primer paso, se muestra el razonamiento matemático obtenido para obtener la primera componente principal, U_1, teniendo en cuenta el requisito de maximizar su varianza. Para asegurar que existe este máximo, es necesario imponer condiciones de acotación sobre el vector de pesos, en este caso, que a_1 sea un vector unitario.

$$\begin{aligned} &\max \operatorname{Var}[U_1] \\ \text{s.a. } &\|a_1\| = a_1^t a_1 = 1 \end{aligned}$$

Como se ha indicado anteriormente, el vector aleatorio X es centrado, esto es, $E[X] = 0$, por tanto, $E\left[a_1^t X\right] = 0$, y esto implica que $\operatorname{Var}[U_1] = E\left[U_1^2\right] = E\left[a_1^t X a_1^t X\right] = E\left[a_1^t X X^t a_1\right] = a_1^t E\left[XX^t\right] a_1 = a_1^t \Sigma a_1$ y, por tanto, el problema queda como sigue

$$\begin{aligned} &\max_{a_1} a_1^t \Sigma a_1 \\ \text{s.a. } &a_1^t a_1 = 1 \end{aligned}$$

Como último paso de simplificación al problema, aplicando el *Teorema de los multiplicadores de Lagrange* para la obtención de extremos condicionados, el problema se reduce a

$$\max_{a_1} \left\{ a_1^t \Sigma a_1 - \lambda \left(a_1^t a_1 - 1 \right) \right\}$$

Tras derivar –matricialmente– esta última expresión con respecto a a_1, sabiendo que Σ es simétrica y tras igualar la expresión resultante a cero, resulta

$$2\Sigma a_1 - 2\lambda a_1 = 0 \tag{5.1}$$

Es evidente que a_1 es el vector propio asociado al valor propio λ de la matriz de covarianza Σ, ya que la expresión anterior se puede reescribir como

$$(\Sigma - \lambda I) a_1 = 0,$$

lo que define el subespacio propio asociado a λ_1. Finalmente, se deduce que λ es la varianza de U_1, puesto que

$$\operatorname{Var}[U_1] = a_1^t \Sigma a_1 = \lambda a_1^t a_1 = \lambda,$$

esto se obtiene multiplicando 5.1 por la izquierda por a_1^t, y teniendo en cuenta que a_1 es unitario.

Se concluye que la primera componente principal es $U_1 = a_1^t X$, donde a_1 es el vector propio asociado al valor propio de Σ de mayor módulo.

Paso 2

Como segundo paso, se obtendrá la segunda componente principal, U_2, la cual debe ser incorrelada con la primera componente principal calculada previamente, a parte de mantener el requisito de varianza máxima. Para asegurar la existencia de este máximo, también es necesario imponer condiciones de acotación sobre el vector de pesos, en este caso, que a_2 también sea un vector unitario.

$$\begin{aligned} &\text{máx}\, \text{Var}\left[U_2\right] \\ \text{s.a. } &\|a_2\| = a_2^t a_2 = 1 \\ &\text{Cov}\left(U_1, U_2\right) = 0 \end{aligned}$$

Como se ha indicado anteriormente, el vector aleatorio X es centrado, esto es, $E[X] = 0$, por tanto, $E\left[a_2^t X\right] = 0$, y esto implica que, de manera análoga a la anterior, $\text{Var}\left[U_2\right] = a_2^t \Sigma a_2$. Del mismo modo $\text{cov}\left(U_1, U_2\right) = E\left[a_1^t X a_2^t X\right] = E\left[a_1^t X X^t a_2\right] = a_1^t E\left[XX^t\right] a_2 = a_1^t \Sigma a_2$ y, por tanto, el problema queda como sigue

$$\begin{aligned} &\text{máx}_{a_2}\, a_2^t \Sigma a_2 \\ \text{s.a. } &a_2^t a_2 = 1 \\ &a_1^t \Sigma a_2 = 0 \end{aligned}$$

Como último paso de simplificación al problema, aplicando el *Teorema de los multiplicadores de Lagrange* para la obtención de extremos condicionados, el problema se reduce a

$$\max_{a_2} \left\{ a_2^t \Sigma a_2 - \lambda \left(a_2^t a_2 - 1 \right) - \mu a_1^t \Sigma a_2 \right\}$$

Tras derivar –matricialmente– esta última expresión con respecto a a_2, sabiendo que Σ es simétrica y tras igualar la expresión resultante a cero, resulta

$$2\Sigma a_2 - 2\lambda a_2 - \mu \Sigma a_1 = 0 \tag{5.2}$$

Después de multiplicar 5.2 por a_1^t a la izquierda obtenemos,

$$2a_1^t \Sigma a_2 - 2\lambda a_1^t a_2 - \mu a_1^t \Sigma a_1 = 0$$

Teniendo en consideración que la segunda restricción impuesta es $a_1^t \Sigma a_2 = 0$, tenemos que $a_1^t a_2 = 0$ (ortogonalidad) y que $a_1^t \Sigma a_1 \neq 0$. Entonces la expresión resulta $\mu a_1^t \Sigma a_1 = 0$, donde deducimos que $\mu = 0$. Por lo tanto, la ecuación que se debe resolver es

$$2\Sigma a_2 - 2\lambda a_2 = 0 \rightarrow (\Sigma - \lambda I) a_2 = 0, \tag{5.3}$$

de esta forma, se puede concluir una vez más que a_2 es el vector propio vinculado al valor propio λ de la matriz Σ. Del mismo modo, se puede observar que $\text{Var}[U_2] = a_2^t \Sigma a_2 = \lambda a_2^t a_2 = \lambda$. Esto se obtiene al multiplicar por la izquierda por a_2^t 5.3 y teniendo en cuenta que a_2 es un vector unitario.

En resumen, la segunda componente principal se expresa como $U_2 = a_2^t X$, donde a_2 es el vector propio asociado al segundo valor propio de mayor magnitud en la matriz de covarianza Σ.

Paso 3

En este tercer paso se obtiene la tercera componente principal, U_3, incorrelada con la primera y segunda componentes principales calculadas anteriormente, maximizando su varianza. Para garantizar la existencia de este máximo también han de imponerse condiciones de acotación sobre el vector de pesos, en este caso que a_3 es también un vector unitario.

$$\begin{aligned} &\text{máx}\, \text{Var}\,[U_3] \\ \text{s.a. } &\|a_3\| = a_3^t a_3 = 1 \\ &\text{Cov}\,(U_1, U_3) = 0 \\ &\text{Cov}\,(U_2, U_3) = 0 \end{aligned}$$

Se debe justificar que a_3 es el vector propio asociado al tercer valor propio de mayor módulo de la matriz Σ.

Como se ha indicado anteriormente, el vector aleatorio X es centrado, esto es, $E[X] = 0$, entonces $\text{Var}\,[U_3] = E[U_3^2] = E[a_3^t X a_3^t X] = a_3^t E[XX^t] a_3 = a_3^t \Sigma a_3$ Además, vemos que $\text{Cov}\,(U_1, U_3) = E[a_1^t X a_3^t X] = a_1^t E[XX^t] a_3 = a_1^t \Sigma a_3$. Análogamente $\text{Cov}\,(U_2, U_3) = a_2^t \Sigma a_3$. Entonces las condiciones del problema para este paso resultan.

$$\begin{aligned} &\text{máx}\, a_3^t \Sigma a_3 \\ \text{s.a. } &\|a_3\| = a_3^t a_3 = 1 \\ &a_1^t \Sigma a_3 = 0 \\ &a_2^t \Sigma a_3 = 0 \end{aligned}$$

Aplicando el *Teorema de los multiplicadores de Lagrange* se obtiene,

$$\max_{a_3} \left\{ a_3^t \Sigma a_3 - \lambda \left(a_3^t a_3 - 1 \right) - \mu a_1^t \Sigma a_3 - \gamma a_2^t \Sigma a_3 \right\}$$

Derivando con respecto a a_3 resulta,

$$2\Sigma a_3 - 2\lambda a_3 - \mu \Sigma a_1 - \gamma \Sigma a_2 = 0 \tag{5.4}$$

Multiplicando 5.4 por a_1^t a la izquierda y resulta,

$$2a_1^t\Sigma a_3 - 2\lambda a_1^t a_3 - \mu a_1^t\Sigma a_1 - \gamma a_1^t\Sigma a_2 = \mu a_1^t\Sigma a_1 = 0$$

Se concluye que $\mu = 0$. Multiplicamos ahora 5.4 esta vez por a_2^t a la izquierda y obtenemos,

$$2a_2^t\Sigma a_3 - 2\lambda a_2^t a_3 - \mu a_2^t\Sigma a_1 - \gamma a_2^t\Sigma a_2 = \gamma a_2^t\Sigma a_2 = 0$$

Se obtiene que $\gamma = 0$. Por tanto, la expresión derivada resulta,

$$2\Sigma a_3 - 2\lambda a_3 = 0 \Leftrightarrow (\Sigma - \lambda I)a_3 = 0$$

Donde a_3 es el vector propio asociado al valor propio λ de Σ. Multiplicando ahora 5.4 pero por a_3^t a la izquierda, se tiene que,

$$2a_3^t\Sigma a_3 - 2\lambda a_3^t a_3 - \mu a_3^t\Sigma a_1 - \gamma a_3^t\Sigma a_2 = a_3^t\Sigma a_3 - \lambda = 0 \Rightarrow a_3^t\Sigma a_3 = \lambda$$

Entonces, $\text{Var}[U_3] = a_3^t\Sigma a_3 = \lambda$. Por tanto, la tercera componente principal $U_3 = a_3^t X$, con a_3 el vector propio asociado al tercer valor propio de Σ con mayor módulo.

5.2.3 *Criterios de elección de componentes principales*

No hay un único método o enfoque definitivo para determinar la cantidad óptima de componentes principales que se deben utilizar. En cambio, la pregunta clave se relaciona con las magnitudes de los valores propios de la matriz de covarianza Σ. ¿Hasta qué punto puede ser pequeño un valor propio sin que la componente principal correspondiente pierda su relevancia o significado? Las principales referencias bibliográficas consultadas para esta sección son [7] y [8].

En cada aplicación en la que implementemos un *PCA* debe decidirse cuántas componentes principales deben conservarse para resumir eficazmente los datos. Algunos métodos propuestos son los siguientes:

M-1. Seleccionar un número de componentes suficiente para explicar un porcentaje determinado de la varianza total.

M-2. Seleccionar las componentes cuyos valores propios sean mayores que la media de los valores propios, $\sum_{i=1}^{p} \lambda_i / p$. Para una matriz de correlaciones, esta media es 1.

M-3. Utilizar los distintos gráficos conocidos como gráficos del codo (*scree plots* en inglés, que son gráficos que muestran λ_i frente a i, y buscan una ruptura natural entre los valores propios *grandes* y los valores propios *pequeños* de una manera visual para el encargado de realizar el análisis.

A continuación se analizan los tres criterios anteriores para elegir los componentes que deben conservarse. Debe tenerse en cuenta que los componentes más pequeños pueden contener información valiosa que no debe ignorarse de forma rutinaria, debe valorarse en cada caso.

En el método M-1., el reto consiste en seleccionar un porcentaje de umbral adecuado. Si se apunta demasiado alto, se corre el riesgo de incluir componentes específicos de la muestra o de la variable. Por *específico de la muestra* se entiende que un componente puede no generalizarse a la población o a otras muestras. Un componente específico de una variable está dominado por una única variable y no representa un resumen compuesto de varias variables.

Definiendo la varianza total –univariante– como la traza de Σ y recordando que la suma de las raíces de $|\Sigma - \lambda \mathbf{I}| = 0$ es igual a la $\text{tr}(\Sigma)$, tenemos que[5]

$$\text{tr}(\Sigma) = \lambda_1 + \lambda_2 + \ldots + \lambda_p \tag{5.5}$$

es maximizada por las *PCs*. Dado que la varianza generalizada de p variables es $|\Sigma| = \prod_{i=1}^{p} \lambda_i$, resulta que la media geométrica

$$\bar{\lambda}_g = |\Sigma|^{1/p} = \left(\prod_{i=1}^{p} \lambda_i \right)^{1/p}$$

es maximizada por las *PCs*.

Dado que la $\text{tr}(\Sigma) - \sum_{j=1}^{k} \lambda_j = \sum_{j=k+1}^{p} \lambda_j$, la proporción de la varianza –univariante– total explicada por k *PCs* es

$$\rho_k^2 = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j} = \frac{\sum_{j=1}^{k} \lambda_j}{\text{tr}(\Sigma)}$$

que puede utilizarse como criterio para seleccionar un subconjunto de k componentes de p. Usualmente se suele tomar como objetivo obtener un valor de entre el 70 % y el 80 % de la varianza total explicada con unas pocas *PCs*.

5 Para cualquier matriz cuadrada A con valores propios $\lambda_1, \ldots, \lambda_n$ tenemos que $tr(A) = \sum_{i=1}^{n} \lambda_i$.

El método M-2. se utiliza ampliamente y es el predeterminado en muchos paquetes de software. Se sabe que $tr(\Sigma) = \sum_i \lambda_i$ por la igualdad 5.5, y el valor propio medio es también la varianza media de las variables individualmente. Así pues, este método retiene los componentes que representan más varianza que la varianza media de las variables. En los casos en que los datos pueden resumirse con éxito en un número relativamente pequeño de dimensiones, suele haber una gran diferencia entre los dos valores propios que caen a ambos lados de la media, como se puede ver en el ejemplo *12.2.2* de [8].

Los *gráficos del codo* de los que se habla en el método M-3. reciben su nombre por su similitud en apariencia con un codo, como indica su nombre[6]. En la práctica, el punto de inflexión entre la curva empinada y la línea recta puede no ser tan claro o puede haber más de una curva perceptible. En tales casos, este enfoque no es tan concluyente.

Usualmente, lo ideal no es utilizar solamente un método para estimar el número de componentes principales, si no, utilizar diversos y contrastarlos para sacar una conclusión sobre el número de *PCs* con el que quedarse. Recuérdase que generalmente no existe un número óptimo de *PCs* que considerar. En cualquier caso, siempre va a depender de la naturaleza de los datos y del propio experimentador.

5.2.4 *Análisis PCA*

En esta sección se realiza un breve análisis *PCA*, sin mayor interés en los datos en sí, si no con el objetivo de servir de apoyo gráfico al marco teórico de esta técnica. Si bien es cierto que en el Capítulo 9 se realiza un *PCA* sobre los datos del estudio, al ser datos de alto rendimiento muy específicos, muchas fases intermedias son omitidas o, simplemente, no pueden ser aplicadas. Por tanto, se considera que ilustrar un ejemplo de un análisis *PCA* sencillo puede facilitar al lector la comprensión de esta técnica, hasta ahora solo expuesta teóricamente. Comentar que el código se mostrará en cuadros de fondo azul, y las salidas correspondientes en cuadros de fondo naranja.

Para realizar este análisis se han empleado las siguientes versiones de software: *R* 4.3.1 (2023-06-16) y *RStudio* 2023.06.0+421. En caso de querer replicar este análisis es recomendable escribir el código manualmente, copiar código directamente de este documento puede dar lugar a errores.

6 Como curiosidad, *scree plot* puede traducirse como *acantilado* o *barranco*, dándose ese nombre debido a su similitud en apariencia con un acantilado con escombros rocosos en su parte inferior.

Lo primero que se tiene que hacer es cargar los datos, usaremos unos datos ya recopilados y que se encuentran en un paquete de *R*. Este paquete es *archdata*, que cuenta con información relativa a la concentración de 11 elementos químicos en un conjunto de 105 muestras de residuos de vidrio recolectadas tanto en *Mancetter* como en *Leicester*. En particular se cargarán los datos *RBGlass1* en un *data.frame*, que contiene concentraciones de estos 11 elementos químicos en las 105 muestras de residuos de vidrio romano-británicos procedentes de dos hornos diferentes, como se ha indicado antes, ubicados en *Leicester* y *Mancetter*.

```
library(archdata)
data("RBGlass1")
df<-RBGlass1
head(df, 5)
```

Output:

```
      Site   Al   Fe   Mg   Ca    Na   K   Ti   P   Mn   Sb   Pb
1 Mancetter 2.51 0.53 0.56 6.98 17.44 0.73 0.09 0.15 0.58 0.12 0.03
2 Mancetter 2.36 0.49 0.53 6.71 17.69 0.68 0.09 0.13 0.40 0.23 0.04
3 Mancetter 2.30 0.36 0.49 8.10 15.94 0.68 0.07 0.13 0.77 0.00 0.01
4 Mancetter 2.42 0.52 0.56 6.93 17.59 0.72 0.09 0.14 0.47 0.18 0.02
5 Mancetter 2.32 0.37 0.51 7.51 16.27 0.69 0.07 0.13 0.21 0.00 0.02
```

Se observa que la primera columna representa la ubicación del resto de vidrio –*Leicester* o *Mancetter*–. Esta información no aporta nada al *PCA*, así que eliminamos esta columna del *data.frame* y hacemos una copia, ya que se trabajará sobre estos datos, pero luego se querrá recuperar los originales. Advertir que se usa *df* de *data.frame*.

```
df_pca<-df[,-1]
df_original<-df_pca
```

Lo primero que hay que preguntarse es si tiene sentido realizar un *PCA*. Como se vio antes en el marco teórico, deben cumplirse ciertos requisitos. El primer punto R-1. decía que los datos deben ser correlados. Para esto se utiliza la función *cor* del paquete base de *R*, que proporciona la matriz de correlaciones.

```
cor(df_pca)
```

La salida es muy extensa como para incluirla, pero en ella se observa una correlación importante entre algunos pares de variables, como entre sodio (NA) y antimonio (Sb)

```
cor(df_pca$Na,df_pca$Sb)
```

Output:

```
0.8291989
```

o entre titanio (Ti) y hierro (Fe)

```
cor(df_pca$Ti,df_pca$Fe)
```

Output:

```
0.7734593
```

Otra forma de verificar que los datos no son incorrelados, es utilizar el contraste de esfericidad de *Bartlett*. Este permite verificar de un modo sencillo si las correlaciones entre variables son significativamente diferentes a 0. La hipótesis nula de este test es que el determinante de la matriz de correlaciones es 1, así que buscamos rechazarla.

Este test en *R* se realiza fácilmente con la función *cortest.bartlett* que está incluida en el paquete *pysch*. Es importante destacar que esta función requiere que la matriz de correlaciones pasada como *input* esté normalizda, así que se inluye un paso intermedio para cumplir esto.

```
library(psych)
df_normalizado<-scale(df_pca)
cortest.bartlett(cor(df_normalizado))
```

Output:

```
$chisq
[1] 789.2636

$p.value
[1] 1.328886e-130

$df
[1] 55
```

El estadístico *chi*-cuadrado (*chisq*) es una medida de cuánto las varianzas observadas difieren de las varianzas esperadas si las varianzas fueran iguales en todos los grupos. Valores más grandes indican una mayor discrepancia entre las varianzas observadas y esperadas. El *p-value* es la probabilidad de que el valor estadístico calculado sea posible dada la hipótesis nula como cierta (igualdad de varianzas). En este caso, el valor extremadamente pequeño del p-*value* sugiere que hay evidencia significativa en contra de la hipótesis nula, lo que significa que las

varianzas entre los grupos son probablemente muy diferentes. El valor *df* representa los grados de libertad de la prueba de *Bartlett*. Estos grados de libertad están relacionados con el número de grupos o muestras en los datos y se utilizan en el cálculo del estadístico de *chi*-cuadrado y el *p-value*.

En resumen, el resultado de la prueba de *Bartlett* muestra que hay una diferencia significativa en las varianzas entre los grupos en el conjunto de datos normalizado. Esto significa que las varianzas no son iguales entre los grupos.

Como segundo requisito, se debía asegurar realizar un análisis exploratorio preliminar de los datos con el propósito de detectar *outliers* (valores atípicos) R-2., puesto que el *PCA* es muy sensible a estos. Un *boxplot* es una buena herramienta para iniciar este análisis.

```
boxplot(df_pca,main="Analisis exploratorio de datos",
        xlab="Elementos quimicos",
        ylab="% de concentracion",
        col=c(1:11))
```

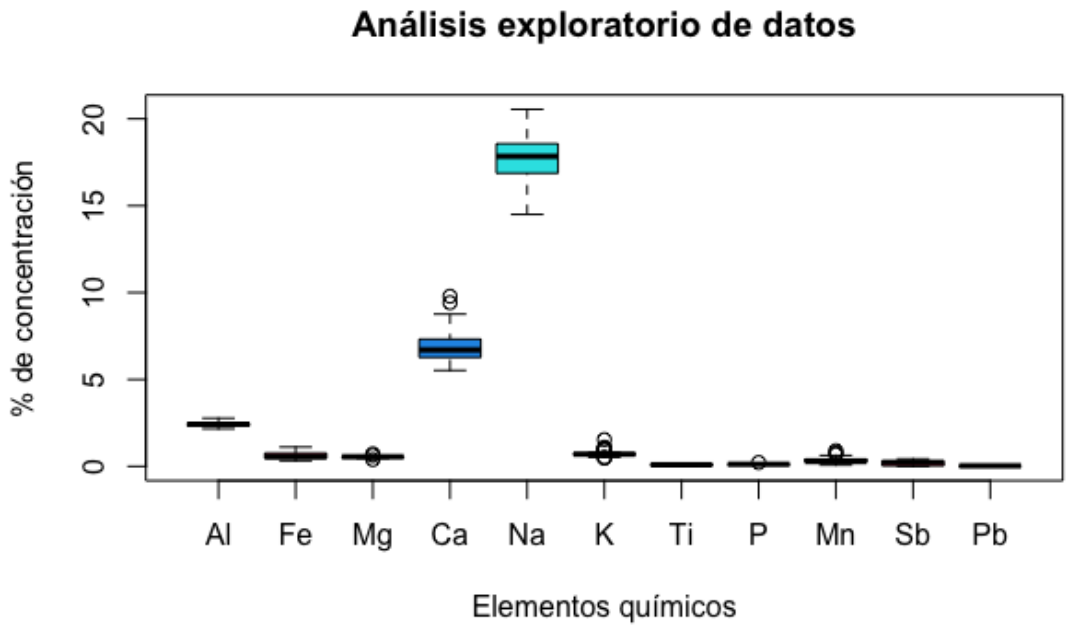

Figura 3: Figura de un *boxplot* exploratorio para detectar *outliers*

En la Figura 3 se observa que tanto el Magnesio (Mg), el Calcio (Ca), el Potasio (K), el Fósforo (P) como el Manganeso (Mn) presentan *outliers*. No podemos

continuar con el *PCA* hasta que *se limpien* estos *outliers* del conjunto de datos. Se crea una función que eliminará estos valores y los sustituirá por el promedio del resto de valores restantes. Posteriormente se aplica a cada uno de los elementos que presentaban *outliers*.

```
erase_outliers<-function(data,na.rm=T){
  H<-0.5*IQR(data)
  data[data<quantile(data,0.25,na.rm = T)-H]<-NA
  data[data>quantile(data,0.75, na.rm = T)+H]<-NA
  data[is.na(data)]<-mean(data,na.rm=T)
  data
}

df_pca$Mg<-erase_outliers(df_pca$Mg)
df_pca$Ca<-erase_outliers(df_pca$Ca)
df_pca$K<-erase_outliers(df_pca$K)
df_pca$P<-erase_outliers(df_pca$P)
df_pca$Mn<-erase_outliers(df_pca$Mn)
```

Ahora se muestra una comparación entre los datos originales y los datos resultantes tras el procesamiento.

```
par(mfrow=c(1,2))

boxplot(df_original,main="Datos originales",
        xlab="Elementos quimicos",
        ylab="% de concentracion",
        col=c(1:11))

boxplot(df_pca,main="Datos sin outliers",
        xlab="Elementos quimicos",
        ylab="% de concentracion",
        col=c(1:11))
```

Llegado este punto, se han creado las circunstancias adecuadas para llevar a cabo este *PCA*. Los datos muestran correlaciones significativas entre sí y, además, no se hallan valores atípicos en los datos procesados, como se observa en el gráfico de la derecha en la Figura 4.

Ahora, se está en disposición de realizar el *PCA*. Esto se hará mediante el uso de la función *prcomp* del paquete base de *R*. Para terminar de cumplir todos los requisitos, faltaría el requisito R-3., por ello se pasan como parámetros *scale* y *center* como *TRUE*.

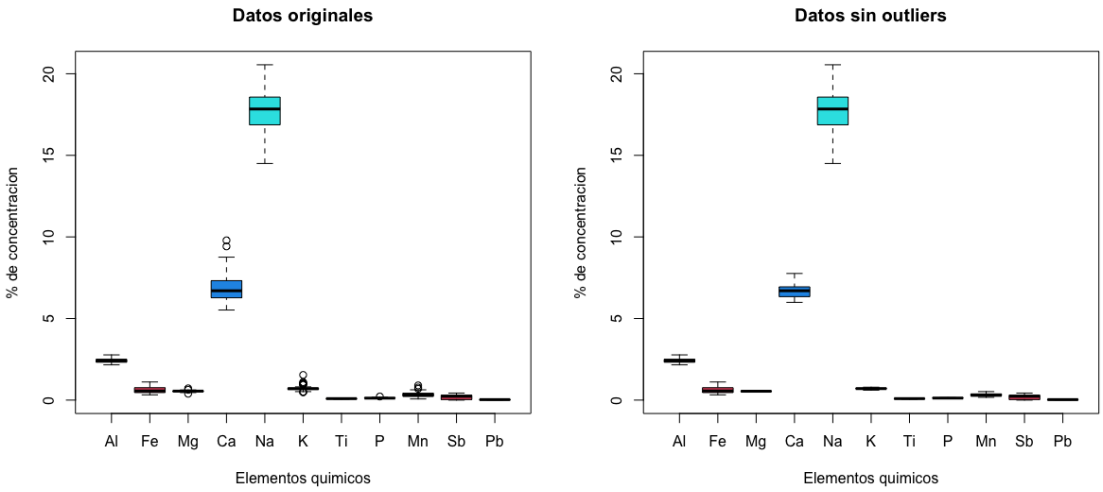

Figura 4: Figura de la comparativa entre los datos originales y los procesados

```
PCA<-prcomp(df_pca, scale=T, center = T)
```

Este objeto *PCA* contiene la información del análisis y sus componentes principales. Al ejecutar la siguiente función se obtiene información relevante de las desviaciones típicas de cada componente principal, la proporción de varianza explicada por cada una y la proporción de varianza explicada acumulada.

```
summary(PCA)
```

Output:

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6
Standard deviation     2.3013  1.2366  1.0556  0.85768 0.80471 0.69456
Proportion of Variance 0.4815  0.1390  0.1013  0.06687 0.05887 0.04386
Cumulative Proportion  0.4815  0.6205  0.7218  0.78865 0.84752 0.89138
                          PC7     PC8     PC9     PC10    PC11
Standard deviation     0.65347 0.55114 0.47699 0.42368 0.23887
Proportion of Variance 0.03882 0.02761 0.02068 0.01632 0.00519
Cumulative Proportion  0.93020 0.95781 0.97849 0.99481 1.00000
```

Una vez tenemos hecho el *PCA*, el siguiente paso es concluir el número de *PCs* será con el que quedarse. Como se comentó anteriormente, existen varios métodos. En primera instancia se utilizarán distintos gráficos conocidos como métodos del codo. Se realizará un primer gráfico donde se mostrará la proporción de varianza explicada por cada *PC*, y otro segundo donde se plasmará la proporción de varianza explicada acumulada. Para realizar estos gráficos se ha utilizado el paquete *ggplot2*.

```
library(ggplot2)

# Proporcion de varianza explicada individual
varianza_explicada <- PCA$sdev^2 / sum(PCA$sdev^2)
ggplot(data = data.frame(varianza_explicada, pc = 1:11),
       aes(x = pc, y = varianza_explicada, fill=varianza_explicada )) +
  geom_col(width = 0.3) +
  scale_y_continuous(limits = c(0,0.6)) + theme_bw() +
  labs(x = "Componente principal", y= " Proporcion de varianza explicada
     ")

# Proporcion de varianza explicada acumulada
varianza_acum<-cumsum(varianza_explicada)
ggplot( data = data.frame(varianza_acum, pc = 1:11),
        aes(x = pc, y = varianza_acum ,fill=varianza_acum )) +
  geom_col(width = 0.5) +
  scale_y_continuous(limits = c(0,1)) +
  theme_bw() +
  labs(x = "Componente principal",
       y = "Proporcion varianza acumulada")
```

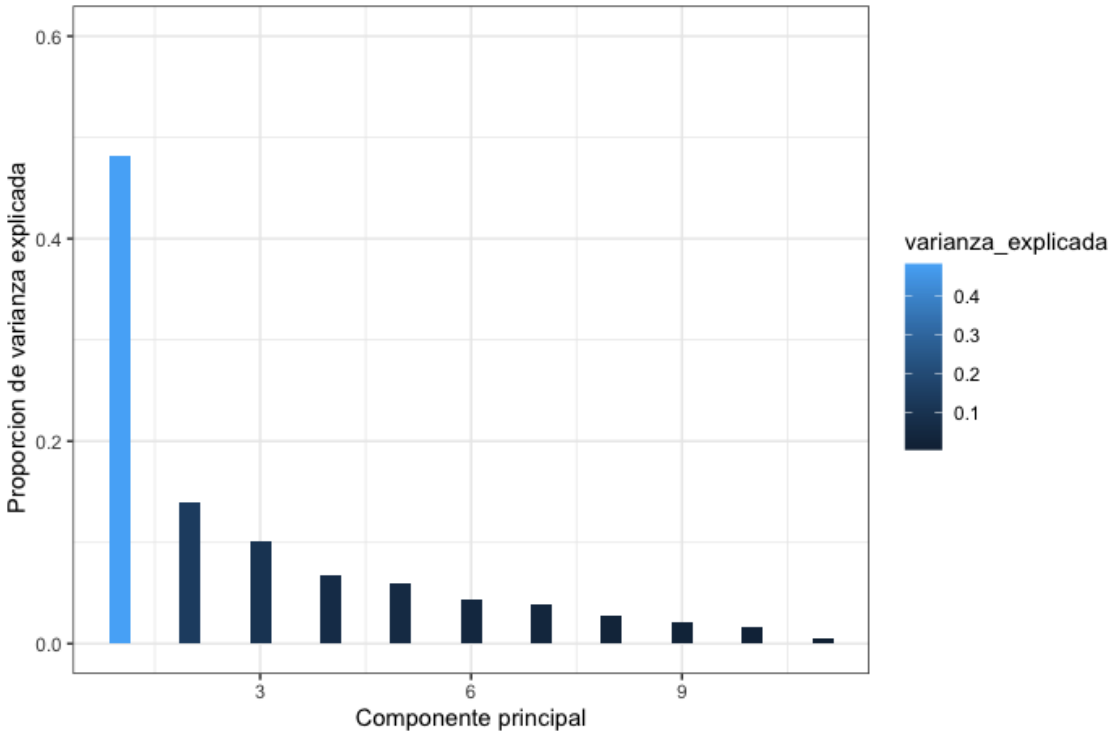

Figura 5: Gráfico del método del codo para la varianza explicada individual

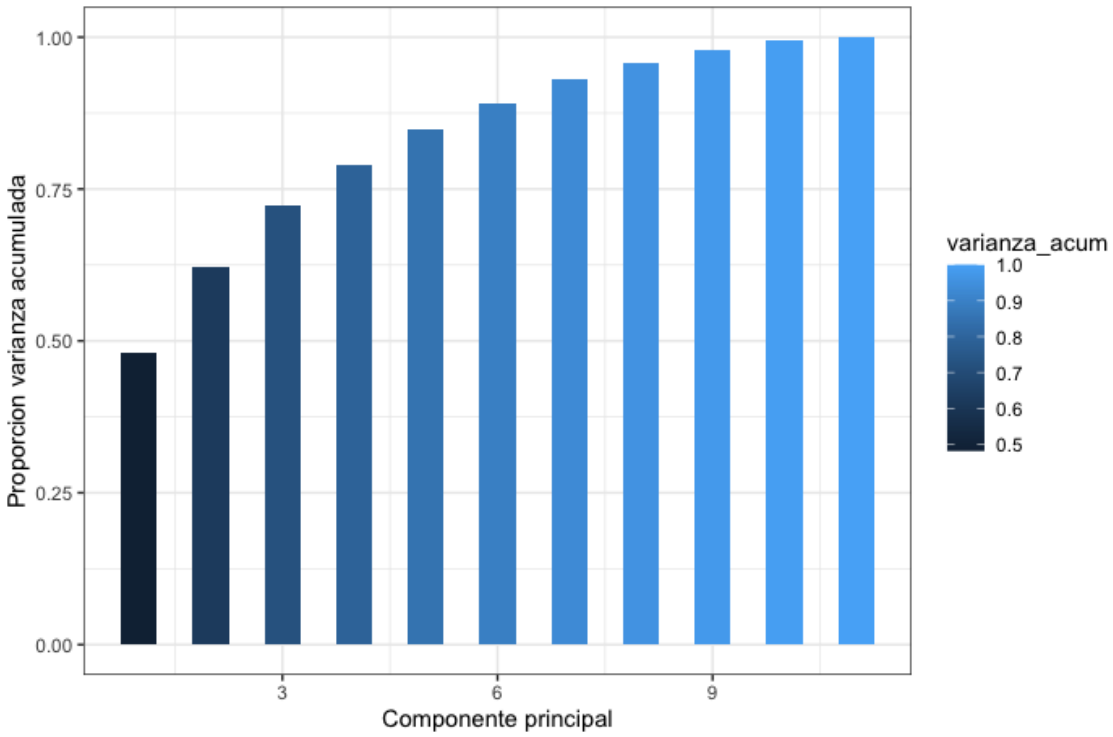

Figura 6: Gráfico del método del codo para la varianza explicada acumulada

Viendo estos dos gráficos (ver Figura 5) y (ver Figura 6) se puede intuir que el número de *PCs* a elegir estará entre 3 y 4, pero no queda muy claro. Es por esto que es usual, y, sobre todo, recomendable, utilizar varios métodos a la hora de elegir el número de componentes que quedarse. Otro de los métodos descritos fue promediar las varianzas explicadas por las *PCs* y se seleccionar aquellas cuya proporción de varianza explicada supera la media. Esto se ve ejecutando lo siguiente.

```
mean(PCA$sdev^2)
```

Output:

```
[1] 1
```

Se observa en la salida anterior que la media de las varianzas explicadas de las componentes es 1. A continuación se identifican cuáles son estas varianzas.

```
PCA$sdev^2
```

Output:

```
[1] 5.29618997 1.52916207 1.11422634 0.73560892 0.64755228 0.48241121
    0.42701875 0.30375371 0.22751706 0.17950284 0.05705686
```

Se obtiene que las *PCs* que superan la media son la 1, la 2 y la 3. Este método combinado con los gráficos anteriores puede dar la fiabilidad necesaria como para escoger las tres primeras componentes principales sin mucho temor a dejar mucha información relevante sin explicar. Volver a recalcar que existe un amplio abanico de métodos para elegir el número adecuado de *PCs*, y que la utilización de varios es casi obligatorio si se quiere realizar un análisis consistente.

5.3 ANÁLISIS CLÚSTER (AC)

5.3.1 *Introducción*

La referencia principal de esta sección introductoria ha sido [11]. En esta sección 5.3.1 se lleva a cabo una introducción del *AC*. En la sección 5.3.2 se tratan las medidad de proximidad del *AC*. En la sección 5.3.3 se discuten los distintos métodos jerárquicos para el *AC*. Por último, en la sección 5.3.4 se introducen los distintos métodos no jerárquicos para el *AC*.

Cuando se trata de analizar conjuntos de elementos en un conjunto de datos multivariantes, se pueden presentar dos situaciones distintas. En un escenario, se plantea un conjunto de datos que contiene mediciones sobre individuos, y pretende determinar si existen agrupaciones o categorías naturales de individuos. En otro contexto, se desea asignar los individuos a categorías preexistentes. El análisis clúster ofrece herramientas y métodos para abordar el primer escenario, es decir, cuando se tiene una matriz de datos con medidas multivariantes sobre un gran número de individuos u objetos, su propósito es crear subgrupos naturales o agrupamientos de individuos.

Para lograrlo, se agrupan individuos que exhiben similitudes según un criterio adecuado. Una vez que se han formado los clústeres, resulta beneficioso describir cada grupo empleando herramientas descriptivas para comprender con mayor profundidad las diferencias que existen entre los grupos creados.

¿Qué es un clúster?

Para responder adecuadamente a esta pregunta se han consultado las referencias [9] y [24]. Hasta ahora, los términos clúster y grupo se han utilizado de forma totalmente intuitiva, sin ningún intento de definición formal. Se trata de una pregunta difícil de responder, principalmente porque no existe una definición universalmente aceptada de lo que constituye exactamente un clúster. En consecuencia, los distintos métodos de *clustering* no suelen producir soluciones idénticas, ni siquiera similares. Por ejemplo, en [25] se ha sugerido que el criterio último para evaluar el significado de tales términos es el juicio de valor del usuario. Si el uso de un término como *clúster* produce una respuesta de valor para el investigador, eso es todo lo que se necesita.

En general, se considera que un cluster es un grupo de elementos –objetos, puntos– en el que cada elemento está *cerca* (en algún sentido apropiado) de un

elemento central de un clúster y que los miembros de diferentes clústeres están *lejos* unos de otros. En cierto sentido, los clústeres pueden considerarse *regiones de alta densidad* de un espacio multidimensional [26]. A primera vista, esta noción parece correcta si los clústeres se consideran regiones elípticas convexas.

Sin embargo, no es difícil concebir situaciones en las que las agrupaciones naturales de elementos no sigan este patrón. Cuando la dimensión de un espacio es lo suficientemente grande, estos elementos multidimensionales, representados como puntos en ese espacio, pueden congregarse en conglomerados que se curvan y retuercen unos alrededor de otros; incluso si los diversos enjambres de puntos no se solapan (lo cual es poco probable), las configuraciones de puntos con formas extrañas pueden ser casi imposibles de detectar e identificar mediante las técnicas actuales.

5.3.2 *Medidas de proximidad*

Dado que el análisis clúster intenta identificar los vectores de observación que son similares y agruparlos en clústeres, muchas técnicas utilizan un índice de similaridad o proximidad entre cada par de observaciones. Una medida conveniente de la proximidad es la distancia entre dos observaciones. Dado que la distancia aumenta a medida que dos unidades se alejan, la distancia es en realidad una medida de disimilaridad –término acuñado del inglés–. A continuación se introducen las posibles funciones que pueden elegirse para medir la similitud entre los grupos que sucesivamente se van formando, distinguiendo primeramente entre disimilaridad –distancias métricas– y similaridades [8].

5.3.2.1 *Disimilaridad*

Para el desarrollo de esta sección se ha consultado, en su mayor parte, las referencias [9] y [24].

Cuando todas las variables registradas son continuas, las proximidades entre individuos suelen cuantificarse mediante medidas de disimilaridad o medidas de distancia. Debido a esto, la definición de la disimilaridad se asemeja a la de la distancia. La forma en que se define la distancia en una aplicación específica a menudo depende de la perspectiva subjetiva.

Definición 10.1. Sean x,y dos puntos cualesquiera de $\mathbb{R}^n$. La disimilaridad es una función $d : \Omega \times \Omega \longrightarrow \mathbb{R}$, tal que $x, y \in \Omega \subseteq \mathbb{R}^n$ satisface las tres propiedades siguientes:

1. $d_{xy} \geq 0$;

2. $d_{xy} = 0 \iff x = y$;

3. $d_{xy} = d_{yx}$ (Propiedad simétrica).

Estas medidas de disimilaridad son denominadas métricas o ultramétricas según cumplan con una cuarta propiedad. Una medida de disimilaridad es considerada métrica si cumple con la siguiente propiedad

4a. $d_{xy} \leq d_{xz} + d_{zy}, \quad \forall z \in \Omega,$

y una disimilaridad es considerada ultramétrica si cumple con esta otra

5. $d_{xy} \leq \text{máx}\{d_{xz}, d_{yz}\}$.

A continuación, se exploran algunas de las medidas de disimilitud que se suelen emplear con frecuencia en aplicaciones prácticas.

En términos generales, se considera un grupo de n elementos para los cuales se han registrado p características diferentes, identificadas como $X_1, \ldots, X_p$. Esto proporciona un conjunto de datos que contiene un total de $n \times p$ observaciones, las cuales se organizan en una matriz con dimensiones de $n \times p$.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{pmatrix}$$

Dentro de la matriz $\boldsymbol{X}$, cada fila correspondiente a i contiene los valores de todas las características para el sujeto i. En contraste, cada columna j exhibe los valores de la característica j a través de todos los sujetos en la muestra.

Es fundamental resaltar que es habitual establecer una diferenciación entre las medidas de asociación aplicadas a los sujetos y aquellas aplicadas a las características. A pesar de que es aceptable utilizar estas medidas tanto para los sujetos como para las características –considerándolas simplemente en un espacio p-dimensional o n-dimensional, respectivamente, lo que sería equivalente a transponer la matriz–.

La selección entre las *medidas de asociación entre individuos* y las *medidas de asociación entre variables* en el análisis clúster depende en gran medida de la naturale-

za de los datos y de los objetivos del análisis. Ambos tipos de medidas resultan útiles y se aplican en diferentes contextos.

En primer lugar, las medidas de asociación entre individuos se utilizan comúnmente cuando se busca agrupar individuos con características similares en clústeres. Este enfoque es particularmente relevante en situaciones como la segmentación de clientes, donde el objetivo es reunir a clientes con perfiles similares para ofrecerles productos o servicios específicos. Las medidas de similitud entre individuos, como las distancias euclidianas o el coeficiente de correlación, son fundamentales en esta perspectiva.

Por otra parte, las medidas de asociación entre variables se emplean cuando se desea analizar las relaciones o dependencias entre las propias características. Esto es valioso en el análisis de datos para comprender cómo se relacionan las variables entre sí y puede ayudar a identificar variables redundantes o irrelevantes. Las medidas de similitud entre variables, como las correlaciones, son esenciales para este enfoque.

En muchos casos, se lleva a cabo un análisis de clústeres que incorpora ambas perspectivas. Esto implica, por ejemplo, realizar un análisis de componentes principales (*PCA*) o un análisis factorial (*AF*) para reducir la dimensionalidad y comprender las relaciones entre las variables. Luego, se aplican medidas de similitud entre individuos para agruparlos según sus características en el espacio de menor dimensión.

En resumen, no se trata de elegir una perspectiva sobre la otra, sino de utilizar las medidas apropiadas en función de los objetivos de tu análisis y la naturaleza de tus datos. El enfoque puede variar según el proyecto, y suele ser beneficioso considerar tanto las medidas de similitud entre individuos como las de similitud entre variables para obtener una comprensión completa de los datos y sus estructuras subyacentes.

Medidas de asociación entre variables

Para combinar variables de manera efectiva, se necesitan medidas numéricas que describan las relaciones entre ellas. Todas las técnicas de agrupación requieren que estas medidas de asociación sean comparables, lo que significa que podemos determinar la fuerza de la relación entre variables. Sin embargo, debemos elegir una medida adecuada según el problema específico.

Cabe destacar que estas medidas son utilizadas en menor medida que las descritas posteriormente, así que simplemente se mencionan algunas con alguna breve explicación, sin entrar en su marco teórico.

Coseno del ángulo entre vectores

La medida de asociación basada en el coseno del ángulo entre vectores es especialmente útil para evaluar la dirección en la que dos variables se relacionan en un espacio multidimensional. Cuando el coseno del ángulo entre dos vectores es cercano a 1, significa que los vectores apuntan en una dirección similar y, por lo tanto, las variables están altamente correlacionadas en esa dirección. Por otro lado, si el coseno del ángulo se acerca a 0, indica que los vectores son ortogonales, lo que sugiere una relación débil o nula entre las variables en esa dirección.

Esta medida permite identificar no solo la fuerza de la asociación entre las variables, sino también la dirección en la que se produce esa asociación.

Coeficiente de correlación

El coeficiente de correlación mide la fuerza y la dirección de la relación lineal entre dos variables. Proporciona una medida numérica que indica si las variables están positivamente correlacionadas (valores cercanos a 1), negativamente correlacionadas (valores cercanos a -1), o si no muestran correlación (valores cercanos a 0). Esta medida es esencial para comprender cómo las variables se relacionan y se utilizan para agrupar variables con patrones de correlación similares.

Medidas de asociación entre individuos

En general, estas son las más utilizadas, y dentro de estas tenemos varias formas de definir una disimilaridad, siendo las más populares la distancia euclídea y la distancia *Manhattan* o *City Block*.

Distancia euclídea

A continuación se seleccionan dos individuos seleccionados de la población, lo que implica elegir dos filas de la matriz de datos $\boldsymbol{X}$:

$$x_i = \left(x_{i1}, \ldots, x_{ip}\right)'$$
$$x_j = \left(x_{j1}, \ldots, x_{jp}\right)'$$

Esta métrica euclidiana (la más comúnmente utilizada), se extiende a más de dos dimensiones y es una generalización de la distancia entre dos puntos en un plano. Es la derivada de la norma $\mathbf{L}_2$ de un vector[7]:

$$\|x_i\|_2 = \sqrt{x_i' x_i} = \sqrt{\sum_{l=1}^{n} x_{il}^2}$$

A partir de esta métrica, se obtiene la distancia euclidiana

$$d_2\left(x_i, x_j\right) = \left\|x_i - x_j\right\|_2 = \sqrt{\left(x_i - x_j\right)'\left(x_i - x_j\right)} = \sqrt{\sum_{l=1}^{n}\left(x_{il} - x_{jl}\right)^2}$$

Esta métrica comparte la propiedad de invariancia ante transformaciones ortogonales con la norma $\mathbf{L}_2$. Estas transformaciones se expresan como $\widetilde{x}_i = \theta x_i$, donde θ es una matriz $n \times n$ que satisface: $\theta'\theta = \theta\theta' = I$. De hecho:

$$\|\theta x_i\|_2 = \sqrt{x_i'\theta'\theta x_i} = \sqrt{x_i' x_i} = \|x_i\|_2$$

de esta manera obtenemos

$$d_2\left(\theta x_i, \theta x_j\right) = d_2\left(x_i, x_j\right)$$

Adicionalmente, es importante destacar que estas transformaciones, junto con las traslaciones, son las únicas para las cuales la distancia d_2 mantiene su invariancia[8].

Distancia City Block

Se obtiene esta distancia de una manera análoga a la anterior, siendo esta derivada de la norma $\mathbf{L}_1$, resultando la distancia d_1 o distancia *Manhattan* o *City Block* $(p = 1)$

$$d_1\left(x_i, x_j\right) = \sum_{l=1}^{n}\left|x_{il} - x_{jl}\right|$$

7 Recordar que dado un espacio vectorial X sobre un cuerpo K, una norma es una aplicación $\|\cdot\| : X \longrightarrow K_0^+$ que verifica
1. $\|x\| = 0 \Leftrightarrow x = 0$
2. $\|\alpha x\| = |\alpha|\|x\| \quad \forall \alpha \in K \quad \forall x \in X$
3. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in X$

8 En efecto, si se considera $\widehat{x}_i = a + x_i$ y $\widehat{x}_j = a + x_j$, entonces se tiene:

$$d_2\left(\widehat{x}_i, \widehat{x}_j\right) = \left\|\widehat{x}_i - \widehat{x}_j\right\|_2 = \left\|(a + x_i) - (a + x_j)\right\|_2 = \left\|x_i - x_j\right\|_2 = d_2\left(x_i, x_j\right)$$

Por norma general, se puede obtener de esta forma todas las distancias derivadas de la norma $\mathbf{L}_p$ de un vector.

5.3.2.2 *Similaridad*

Para el desarrollo de esta sección se ha consultado, en su mayor parte, las referencias [7] y [24].

Mientras que las medidas de disimilitud son más apropiadas para datos continuos, cuando se trabaja con datos en los cuales todas las variables son categóricas, es más común utilizar medidas de similitud. Por lo general, estas medidas se escalan para que se encuentren en el intervalo [0, 1], aunque ocasionalmente también se expresan como porcentajes en el rango 0-100 %. En este contexto, dos sujetos i y j se consideran completamente similares, con un coeficiente de similitud s_{ij} igual a uno, si tienen valores idénticos para todas las variables. Por otro lado, un valor de similitud igual a cero indica que los dos sujetos difieren al máximo en todas las variables.

Dados dos objetos $\mathbf{x}_i$ y $\mathbf{x}_j$ en un espacio p-dimensional, una medida de similitud satisface las siguientes condiciones.

1. $0 \leq s_{ij} \leq 1$ para todos $\mathbf{x}_i$ y $\mathbf{x}_j$;
2. $s_{ij} = 1$ si, y solo si $\mathbf{x}_i = \mathbf{x}_j$;
3. $s_{ij} = s_{ij}$ (Propiedad simétrica).

La condición (3) vuelve a asegurar que la medida es simétrica, mientras que las condiciones (1) y (2) garantizan que la medida es siempre positiva y alcanza el valor de uno únicamente cuando los objetos i y j son idénticos.

Dada una medida de similitud, es posible crear siempre una medida de disimilitud utilizando la relación $d_{rs} = 1 - s_{rs}$ o alguna otra función decreciente. Sin embargo, es importante destacar que la nueva medida de disimilitud puede no cumplir con todas las propiedades de una métrica.

En el caso contrario, cuando se parte de una medida de disimilitud d_{rs} y se desea construir una medida de similitud como $s_{rs} = 1/(1 + d_{rs})$, es importante tener en cuenta que dado que d_{rs} no está acotada positivamente, se cumple que $0 < s_{rs} \leq 1$, lo que significa que s_{rs} nunca alcanza el valor de cero. Por lo tanto, no es posible obtener s_{rs} directamente a partir de d_{rs} utilizando esta relación inversa.

Por lo tanto, estas fórmulas nos permiten emplear todos los ejemplos analizados en la sección anterior como indicadores de similitud simplemente utilizando

la relación mencionada. Esto se relaciona con la idea de que la disimilitud y la similitud están en una relación inversamente proporcional, lo que significa que a medida que una aumenta, la otra disminuye de manera intuitiva. En la mayoría de los casos es preferible utilizar la medida de disimilitud en lugar de la medida de similitud derivada de ella. Como resultado, en general, el uso de medidas de disimilitud es más común.

Una medida de similitud ampliamente sugerida por algunos investigadores consiste en calcular la correlación de *Pearson* entre los objetos $\boldsymbol{y}_r$ e $\boldsymbol{y}_s$, donde r y s representan índices que van desde 1 hasta n. Esta correlación se define como:

$$q_{rs} = \frac{\sum_{j=1}^{p} \left(y_{rj} - \bar{y}_{r.}\right)\left(y_{sj} - \bar{y}_{s.}\right)}{\left[\sum_{j=1}^{p} \left(y_{rj} - \bar{y}_{r.}\right)^2 \sum_{j=1}^{p} \left(y_{sj} - \bar{y}_{s.}\right)^2\right]}$$

donde $\bar{y}_{r.} = \sum_j y_{rj}/p$ y $\bar{y}_{s.} = \sum_j y_{sj}/p$. La notación q_{rs} es usada debido a que estamos calculando la correlación entre las filas de la matriz de datos. Sin embargo, debido a que $-1 \leq q_{rs} \leq 1$, no satisface la condición (1). Para corregir esta situación, uno podría plantearse usar las cantidades $|q_{rs}|$ o $1 - q_{rs}^2$.

5.3.3 *Métodos jerárquicos*

Para desarrollar la sección de métodos jerárquicos la referencia principal ha sido [9].

El análisis clúster (*AC*) a través de métodos jerárquicos busca primordialmente la creación de agrupamientos al combinar o dividir clústeres con el fin de minimizar una distancia o disimilitud particular o maximizar una medida de similitud específica. Estos métodos se subdividen en dos categorías fundamentales: los métodos aglomerativos, que unen clústeres, y los métodos disociativos, que los separan, cada uno con diversas variantes.

- Los Métodos Aglomerativos, también conocidos como Métodos Ascendentes, inician el análisis con un número de grupos de clústeres equivalente a la cantidad de individuos en los datos. A medida que progresa el proceso, estos grupos se fusionan de forma gradual en un proceso ascendente, hasta que, al final, todos los casos se encuentren agrupados en un único clúster consolidado.

- Los Métodos Disociativos, también conocidos como Métodos Descendentes, comienzan con un clúster inicial que engloba todos los casos bajo análisis. A

medida que avanza el proceso, este clúster se divide repetidamente en grupos más pequeños, sucesivamente, hasta que, al concluir, se obtienen tantos clústeres como casos bajo estudio.

En síntesis, en el análisis clúster, los métodos jerárquicos permiten una formación o división gradual y jerárquica de clústeres, según se empleen métodos aglomerativos o disociativos. El propósito radica en descubrir una estructura de clústeres que sea congruente con las distancias o similitudes entre los datos. Estos métodos proporcionan flexibilidad para explorar diversos niveles de detalle en la agrupación de datos.

Dendrogramas

Un dendrograma es un diagrama de tipo arbóreo que constituye el producto final de todos los métodos de agrupación jerárquica. Su función principal radica en mostrar el proceso de formación de grupos en distintos niveles de detalle. Para simplificar, el dendrograma desvela cómo los grupos de una solución con k clústeres surgen de la fusión de algunos clústeres presentes en una solución con $k+1$ clústeres. Es relevante destacar que el dendrograma puede presentarse tanto en orientación horizontal como vertical, y ambas modalidades brindan la misma información sustancial.

En el contexto del dendrograma, la *altura* se convierte en una clave esencial para la interpretación de cómo los elementos y los clústeres se amalgaman para crear entidades más grandes. Esta altura representa un indicador crucial. Cuando elementos de naturaleza similar se fusionan, sucede a una menor altura en el dendrograma, mientras que la unión de elementos con menor semejanza se manifiesta a alturas superiores.

La disparidad de alturas en el dendrograma es, en última instancia, una representación de la similitud o distancia entre los elementos. Si se observa una amplia brecha entre las alturas donde se concretan estas fusiones de clústeres, se revela una estructura más definida en los datos. En resumen, cuanto mayor sea la diferencia de alturas en el dendrograma, mayor claridad aportará a la comprensión de la estructura subyacente de los datos.

En el proceso de obtener una partición de los datos en un número predeterminado de grupos, se emplea un enfoque de *corte* en el dendrograma. Esta técnica implica trazar una línea horizontal en el dendrograma a una altura específica. La cantidad de líneas verticales intersectadas por esta línea horizontal determina el número de grupos que se formarán.

Cada punto de intersección entre la línea horizontal y una de las líneas verticales representa un grupo individual, y los elementos ubicados debajo de dicha intersección se consideran miembros de ese grupo. Esta estrategia de corte proporciona un medio efectivo para segmentar los datos en clústeres coherentes según las necesidades del análisis.

Es importante destacar que en el dendrograma, las distancias horizontales entre elementos carecen de relevancia en la definición de la solución de clústeres, ya que son las distancias verticales las que adquieren importancia determinante.

5.3.3.1 *Métodos jerárquicos aglomerativos*

En el siguiente apartado, se examinan diversas estrategias para la fusión de clústeres en varias etapas de un proceso jerárquico. Es fundamental enfatizar que no hay una sola estrategia óptima que se aplique universalmente a todos los problemas, ya que la elección del método puede conducir a resultados variados. Por lo general, la selección del método se basa en el juicio y la experiencia del investigador, así como en el conocimiento específico del problema en cuestión.

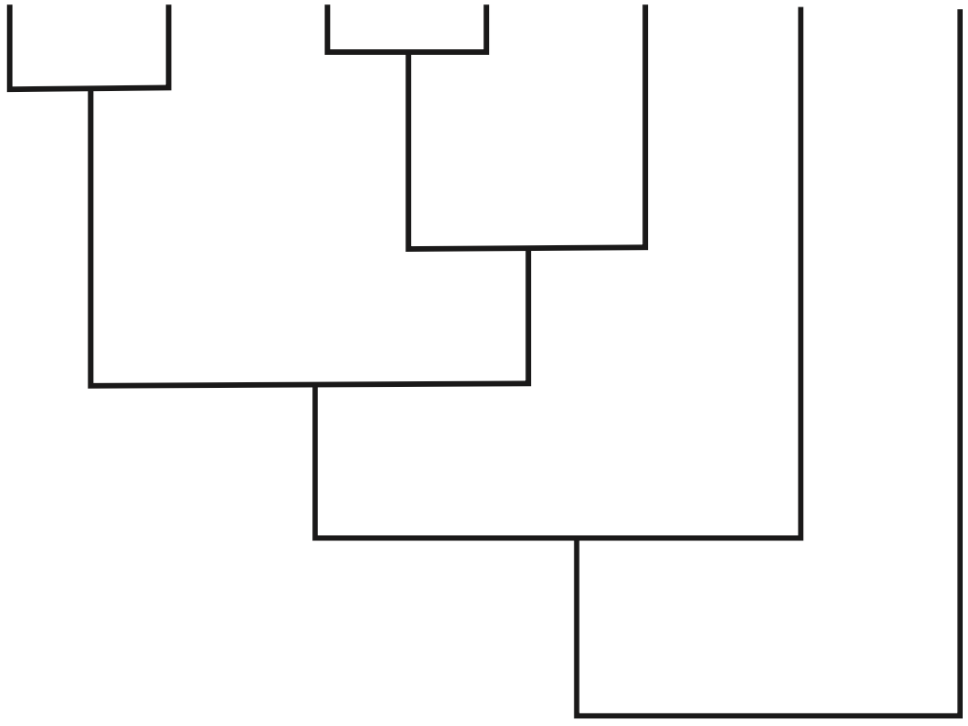

Figura 7: Figura de un dendrograma

Una estrategia sugerida implica la utilización de múltiples procedimientos distintos para luego comparar los resultados obtenidos. Este enfoque resulta beneficioso, ya sea porque los diferentes métodos arrojen resultados coincidentes o divergentes. La comparación de los resultados derivados de varios métodos confiere una perspectiva más sólida y contribuye a la obtención de conclusiones más resistentes. En última instancia, la elección del método para la fusión de clústeres se basará en la comprensión profunda del problema en cuestión y en el juicio informado del investigador.

Se ilustran cinco estrategias diferentes, presentando en detalle las cuatro primeras, mientras que la quinta estrategia se examinará de manera más extensa, con ejemplos y una comprensión más profunda.

Estrategia de la distancia mínima o similaridad máxima

En la literatura anglosajona, esta estrategia se conoce como encadenamiento simple (*single linkage*). En este enfoque, la distancia o similitud entre dos clústeres se basa en la mínima distancia –o máxima similitud– entre sus componentes.

Por lo tanto, al concluir la etapa k-ésima, con $n - k$ clústeres formados, la distancia entre los clústeres C_i (con n_i elementos) y C_j (con n_j elementos) se expresaría como:

$$d\left(C_i, C_j\right) = \operatorname{Min}_{\substack{x_l \in C_i \\ x_m \in C_j}} \left\{d\left(x_l, x_m\right)\right\} \quad l = 1, \ldots, n_i; m = 1, \ldots, n_j$$

mientras que si optáramos por utilizar una medida de similitud en lugar de distancia, la similitud entre los dos clústeres se calcularía como:

$$s\left(C_i, C_j\right) = \operatorname{Max}_{\substack{x_l \in C_i \\ x_m \in C_j}} \left\{s\left(x_l, x_m\right)\right\} \quad l = 1, \ldots, n_i; m = 1, \ldots, n_j$$

Tiene un enfoque metaheurístico de *vecino más próximo*.

Estrategia de la distancia máxima o similaridad mínima

En el método denominado procedimiento de encadenamiento completo (*complete linkage*), se adopta la perspectiva de medir la distancia o similitud entre dos clústeres en función de sus elementos más diferentes. En otras palabras, la distancia o similitud entre los clústeres se determina mediante la máxima distancia –o mínima similitud– entre sus componentes.

Entonces, de manera análoga a la estrategia previa, en la etapa k-ésima, con $n-k$ clústeres ya formados, las medidas de distancia y similitud entre los clústeres C_i y C_j (con n_i y n_j elementos, respectivamente) se calculan como sigue:

$$d\left(C_i, C_j\right) = \operatorname{Max}_{\substack{x_l \in C_i \\ x_m \in C_j}} \left\{d\left(x_l, x_m\right)\right\} \quad l = 1, \ldots, n_i; m = 1, \ldots, n_j$$

$$s\left(C_i, C_j\right) = \operatorname{Min}_{\substack{x_l \in C_i \\ x_m \in C_j}} \left\{s\left(x_l, x_m\right)\right\} \quad l = 1, \ldots, n_i; m = 1, \ldots, n_j$$

Tiene un enfoque metaheurístico de *vecino más lejano*.

Estrategia de la distancia, o similitud, promedio no ponderada

En esta estrategia, también conocida como *unweighted aritmethic average*, la distancia o similitud entre el clúster C_i y el cluster C_j se determina como el promedio aritmético de las distancias o similitudes entre sus componentes.

Por lo tanto, si el clúster C_i –con n_i elementos– está compuesto, a su vez, por dos subclústeres C_{i_1} y C_{i_2} –con n_{i_1} y n_{i_2} elementos, respectivamente–, y el clúster C_j consta de n_j elementos, la distancia o similitud entre ellos se calcula mediante la siguiente fórmula

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

Es importante notar que en este método no se toma en consideración el tamaño de ninguno de los clústeres involucrados en el cálculo. Esto implica que se otorga igual importancia a la distancia $d\left(C_{i_1}, C_j\right)$ que a la distancia $d\left(C_{i_2}, C_j\right)$.

Estrategia de la distancia, o similitud, promedio ponderada

En esta estrategia, también conocida como *weighted aritmethic average*, la distancia o similitud entre dos clústeres se define como el promedio ponderado de las distancias o similitudes entre los elementos de un clúster en relación a los del otro.

Consideremos dos clústeres, C_i y C_j. Supongamos que el clúster C_i se desglosa en dos subclústeres, C_{i_1} y C_{i_2}, con n_{i_1} y n_{i_2} elementos, respectivamente. El número total de elementos en C_i se representa como $n_i = n_{i_1} + n_{i_2}$, mientras que el número de elementos en C_j es n_j. Entonces, en lo que respecta a las distancias –y este

razonamiento se aplica igualmente a las similitudes–, la distancia promedio ponderada se calcula de la siguiente manera, donde $x_i \in C_i, x_{i_1} \in C_{i_1}, x_{i_2} \in C_{i_2}, x_j \in C_j$

$$d\left(C_i, C_j\right) = \frac{1}{\left(n_{i_1} + n_{i_2}\right) n_j} \sum_{i=1}^{n_{i_1}+n_{i_2}} \sum_{j=1}^{n_j} d\left(x_i, x_j\right) =$$

$$= \frac{1}{\left(n_{i_1} + n_{i_2}\right) n_j} \sum_{i_1=1}^{n_{i_1}} \sum_{j=1}^{n_j} d\left(x_{i_1}, x_j\right) + \frac{1}{\left(n_{i_1} + n_{i_2}\right) n_j} \sum_{i_2=1}^{n_{i_2}} \sum_{j=1}^{n_j} d\left(x_{i_2}, x_j\right) =$$

$$= \frac{n_{i_1}}{\left(n_{i_1} + n_{i_2}\right) n_{i_1} n_j} \sum_{i_1=1}^{n_{i_1}} \sum_{j=1}^{n_j} d\left(x_{i_1}, x_j\right) + \frac{n_{i_2}}{\left(n_{i_1} + n_{i_2}\right) n_{i_2} n_j} \sum_{i_2=1}^{n_{i_2}} \sum_{j=1}^{n_j} d\left(x_{i_2}, x_j\right) =$$

$$= \frac{n_{i_1}}{n_{i_1} + n_{i_2}} d\left(C_{i_1}, C_j\right) + \frac{n_{i_2}}{n_{i_1} + n_{i_2}} d\left(C_{i_2}, C_j\right) =$$

$$= \frac{n_{i_1} d\left(C_{i_1}, C_j\right) + n_{i_2} d\left(C_{i_2}, C_j\right)}{n_{i_1} + n_{i_2}}$$

de esta forma, la distancia $d\left(C_i, C_j\right)$ se obtiene como el promedio ponderado de las distancias entre cada uno de los dos clústeres previos, C_{i_1} y C_{i_2}, en relación con el clúster C_j.

Método de Ward

El método de *Ward* se caracteriza por ser un enfoque jerárquico en el cual, en cada paso del proceso, se fusionan los dos clústeres que muestren el menor aumento en el valor total de la suma de los cuadrados de las diferencias de cada individuo con respecto al centroide[9] del clúster. Notemos por

- x_{ij}^k se refiere al valor de la j-ésima variable para el i-ésimo individuo dentro del clúster k, considerando que este clúster contiene n_k individuos.
- m^k representa el centroide del clúster k, cuyas componentes son m_j^k.
- E_k denota la suma de cuadrados de los errores del clúster k, es decir, la distancia euclidiana al cuadrado entre cada individuo en el clúster k y su centroide.

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

9 Este elemento no es más que el punto del espacio que minimiza la suma de los cuadrados de las distancias de las observaciones al centroide que se utiliza en cada etapa.

- E representa la suma total de los cuadrados de los errores que abarcan todos los clústeres. En otras palabras, si consideramos que existen "h" clústeres en total, resulta

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

El proceso comienza con m clústeres, cada uno de los cuales consta de un solo individuo, lo que significa que en esta etapa inicial, cada individuo coincide con el centro del clúster. Por lo tanto, en este primer paso, se tiene $E_k = 0$ para cada clúster, lo que implica que $E = 0$. El objetivo principal del método de *Ward* radica en encontrar, en cada etapa, los dos clústeres cuya unión genere el menor aumento en la suma total de errores, E.

En el supuesto que los clústeres C_p y C_q se fusionan para dar lugar a un nuevo clúster C_t. El incremento en el valor de E será

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

Ahora bien

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

donde

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

teniendo en cuenta que

$$2m_j^p m_j^q = \left(m_j^p\right)^2 + \left(m_j^q\right)^2 - \left(m_j^p - m_j^q\right)^2$$

resulta

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

puesto que $n_t = n_p + n_q$, tras dividir por n_t^2 se obtiene

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

por tanto, se puede expresar ΔE_{pq} como:

$$n_p \sum_{j=1}^{n}\left(m_j^p\right)^2 + n_q \sum_{j=1}^{n}\left(m_j^q\right)^2 - n_t \sum_{j=1}^{n}\left[\frac{n_p}{n_t}\left(m_j^p\right)^2 + \frac{n_q}{n_t}\left(m_j^q\right)^2 - \frac{n_p n_q}{n_t^2}\left(m_j^p - m_j^q\right)^2\right]$$

$$\Delta E_{pq} = n_p \sum_{j=1}^{n}\left(m_j^p\right)^2 + n_q \sum_{j=1}^{n}\left(m_j^q\right)^2 - m_p \sum_{j=1}^{n}\left(m_j^p\right)^2 - n_q \sum_{j=1}^{n}\left(m_j^q\right)^2 + \frac{n_p n_q}{n_t}\sum_{j=1}^{n}\left(m_j^p - m_j^q\right)^2$$

$$= \frac{n_p n_q}{n_t}\sum_{j=1}^{n}\left(m_j^p - m_j^q\right)^2$$

El menor incremento en los errores cuadráticos es directamente proporcional a la distancia euclidiana al cuadrado entre los centroides de los clústeres que se fusionan. Es importante destacar que la suma total E no disminuye a medida que avanzamos en el proceso, lo que implica que el método no presenta las limitaciones que se observan en los métodos de centroides previos.

Para concluir, se examina como se pueden calcular los diversos incrementos a partir de los cálculos previos.

Se supone que C_t es el resultado de la unión de C_p y C_q, y además, se tiene otro clúster C_r que es diferente de los dos mencionados. En este caso, el aumento potencial en el valor de E debido a la unión de C_r y C_t se calcula de la siguiente manera

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

Sabiendo que

$$m_j^t = \frac{n_p m_j^p + n_q m_j^q}{n_t}$$

$$n_t = n_p + n_q$$

y

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

se deduce

$$\left(m_j^r - m_j^t\right)^2 = \left(m_j^r\right)^2 + \left(m_j^t\right)^2 - 2m_j^r m_j^t =$$

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

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

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

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

por tanto

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

$$= \frac{n_r n_t}{n_r + n_t}\sum_{j=1}^{n}\left[\frac{n_p}{n_t}\left(m_j^r - m_j^p\right)^2 + \frac{n_q}{n_t}\left(m_j^r - m_j^q\right)^2 - \frac{n_p n_q}{n_t^2}\left(m_j^p - m_j^q\right)^2\right] =$$

$$= \tfrac{n_r n_p}{n_r+n_t}\textstyle\sum_{j=1}^{n}\left(m_j^r - m_j^p\right)^2 + \tfrac{n_q n_r}{n_r+n_t}\sum_{j=1}^{n}\left(m_j^r - m_j^q\right)^2 - \tfrac{n_r n_p n_q}{n_t(n_r+n_t)}\sum_{j=1}^{n}\left(m_j^p - m_j^q\right)^2 =$$

$$= \frac{1}{n_r + n_t}\sum_{j=1}^{n}\left[n_r n_p\left(m_j^r - m_j^p\right)^2 + n_r n_q\left(m_j^r - m_j^q\right)^2 - \frac{n_r n_p n_q}{n_p + n_q}\left(m_j^p - m_j^q\right)^2\right] =$$

$$= \frac{1}{n_r + n_t}\left[\left(n_r + n_p\right)\Delta E_{rp} + \left(n_r + n_q\right)\Delta E_{rq} - n_r \Delta E_{pq}\right]$$

Por último, aunque no se demuestra en este trabajo, se menciona –como curiosidad para el lector– que la relación anterior sigue siendo válida cuando se emplea una distancia derivada de una norma que proviene de un producto escalar o cumple con la ley del paralelogramo.

A continuación, se ilustra un ejemplo de este método.

Ejemplo 1 En este ejemplo –Ejemplo 8.7 de la bibliografía mencionada, si se quiere acceder al estudio original– se verá cómo se aplica este procedimiento en un ejemplo con 5 individuos en los que se registran dos variables. A continuación, se presentan los datos:

Individuo	X_1	X_2
A	10	5
B	20	20
C	30	10
D	30	15
E	5	10

Nivel 1

En primer lugar, se calculan $\begin{pmatrix} 5 \\ 2 \end{pmatrix} = 10$ posibles combinaciones.

Partición	Centroides	E_k	E	ΔE
$(A,B),C,D,E$	$C_{AB} = (15,12,5)$	$E_{AB} = 162'5$ $E_C = E_D = E_E = 0$	162'5	162'5
$(A,C),B,D,E$	$C_{AC} = (20,7,5)$	$E_{AC} = 212'5$ $E_B = E_D = E_E = 0$	212'5	212'5
$(A,D),B,C,E$	$C_{AD} = (20,10)$	$E_{AD} = 250$ $E_B = E_C = E_E = 0$	250	250
$(A,E),B,C,D$	$C_{AE} = (7'5,7'5)$	$E_{AE} = 25$ $E_B = E_C = E_D = 0$	25	25
$(B,C),A,D,E$	$C_{BC} = (25,15)$	$E_{BC} = 100$ $E_A = E_D = E_E = 0$	100	100
$(B,D),A,C,E$	$C_{BD} = (25,17'5)$	$E_{BD} = 62'5$ $E_A = E_C = E_E = 0$	62'5	62'5
$(B,E),A,C,D$	$C_{BE} = (12'5,15)$	$E_{BE} = 162'5$ $E_A = E_C = E_D = 0$	162'5	162'5
$(C,D),A,B,E$	$C_{CD} = (30,12'5)$	$E_{CD} = 12'5$ $E_A = E_B = E_E = 0$	12'5	12'5
$(C,E),A,B,D$	$C_{CE} = (17'5,10)$	$E_{CE} = 312'5$ $E_A = E_B = E_D = 0$	312'5	312'5
$(D,E),A,B,C$	$C_{DE} = (17'5,12'5)$	$E_{DE} = 325$ $E_A = E_B = E_C = 0$	325	325

A partir de estos datos, se puede deducir que en esta etapa se fusionan los elementos C y D. La configuración actual es la siguiente: $(C,D),A,B,E$.

Nivel 2

Con la configuración actual, se toman las $\begin{pmatrix} 4 \\ 2 \end{pmatrix} = 6$ combinaciones posibles.

Partición	Centroides	E_k	E	ΔE
$(A,C,D),B,E$	$C_{ACD} = (23'33, 10)$	$E_{ACD} = 316'6$ $E_B = E_E = 0$	316'66	304'16
$(B,C,D),A,E$	$C_{BCD} = (26'66, 15)$	$E_{BCD} = 116'66$ $E_A = E_E = 0$	116'66	104'16
$(C,D,E),A,B$	$C_{CDE} = (21'66, 11'66)$	$E_{CDE} = 433'33$ $E_A = E_B = 0$	433'33	420'83
$(A,B),(C,D),E$	$C_{AB} = (15, 12'5)$ $C_{CD} = (30, 12'5)$	$E_{AB} = 162'5$ $E_{CD} = 12'5$ $E_E = 0$	175	162'5
$(A,E),(C,D),B$	$C_{AE} = (7'5, 7'5)$ $C_{CD} = (30, 12'5)$	$E_{AE} = 25$ $E_{CD} = 12'5$ $E_B = 0$	37'5	25
$(B,E),(C,D),A$	$C_{BE} = (12'5, 15)$ $C_{CD} = (30, 12'5)$	$E_{BE} = 162'5$ $E_{CD} = 12'5$ $E_A = 0$	175	162'5

Se puede inferir a partir de esto que en esta etapa se unen los elementos A y E. La configuración actual es la siguiente: $(A,E),(C,D),B$.

Nivel 3

Con la configuración actual, se toman las $\begin{pmatrix} 3 \\ 2 \end{pmatrix} = 3$ combinaciones posibles.

Partición	Centroides	E_k	E	ΔE
$(A,C,D,E),B$	$C_{ACDE} = (18'75, 10)$	$E_{ACDE} = 568'75$ $E_B = 0$	568'75	531'25
$(A,B,E),(C,D)$	$C_{ABE} = (11'66, 11'66)$ $C_{CD} = (30, 12'5)$	$E_{ABE} = 233'33$ $E_{CD} = 12'5$	245'8	208'3
$(A,E),(B,C,D)$	$C_{AE} = (7'5, 7'5)$ $C_{BCD} = (26'66, 15)$	$E_{AE} = 25$ $E_{BCD} = 116'66$	141'66	104'16

Se concluye que en esta etapa unimos los clústeres B y (C, D). La configuración actual es $(A, E), (B, C, D)$.

Nivel 4

Es evidente que en este paso se fusionarán los dos clústeres existentes. A continuación, se presentan los valores del centroide y los incrementos en las distancias

Partición	Centroide	E	ΔE
(A, B, C, D, E)	$C_{ABCDE} = (19, 12)$	650	508'34

El dendrograma correspondiente se muestra en la siguiente figura

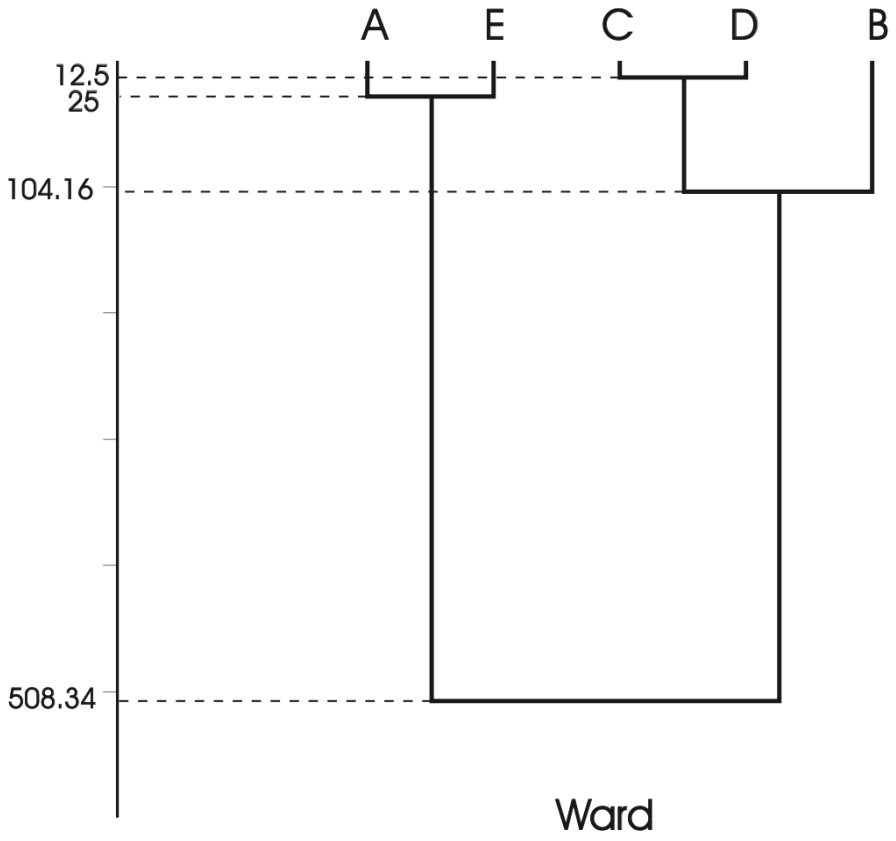

Figura 8: Figura del dendrograma resultante del **Ejemplo 1**

5.3.3.2 *Métodos jerárquicos disociativos*

Como se explicó al inicio de este capítulo, los métodos disociativos operan en contraposición a los métodos aglomerativos. Inician con un único grupo que abarca todos los casos y, a medida que avanzan, dividen progresivamente este grupo en unidades más pequeñas. Al concluir el proceso, se obtienen tantos grupos como casos hayan sido procesados.

Respecto al cálculo de la distancia entre grupos en estos métodos, se sigue una filosofía similar a la de los métodos aglomerativos. No obstante, dado que se inicia con un único grupo que se debe subdividir, la estrategia se enfoca en maximizar las distancias o minimizar las similitudes. El objetivo aquí es separar los individuos menos similares del clúster principal.

Los métodos disociativos se dividen en dos categorías principales:

- Monotéticos: Estos métodos se basan en la división de datos utilizando un solo atributo. Son especialmente útiles cuando los datos tienen una naturaleza binaria.

- Politéticos: Contrariamente, los métodos politéticos realizan divisiones teniendo en cuenta los valores de todas las variables en conjunto.

Es relevante notar que esta categoría de métodos es menos común en comparación con los métodos aglomerativos, lo que se traduce en una disponibilidad limitada de literatura y documentación. Un tema fundamental que puede surgir al aplicar estos métodos es la determinación del momento adecuado para detener la subdivisión de un clúster específico antes de proceder a la división de otro clúster diferente. Este interrogante puede ser abordado a través de una variante propuesta por MacNaughton-Smith en 1964 [27], especialmente diseñada para medidas de asociación que son de naturaleza positiva.

La premisa fundamental de este enfoque consiste en dividir los elementos en dos grupos en cada iteración: uno denominado *clúster A* y el otro identificado como *el resto* o *clúster B*. El proceso se inicia seleccionando el elemento que, en promedio, es el más diferente de todos los demás elementos en el conjunto de datos y se asigna a clúster A. Posteriormente, una vez que los datos han sido divididos en A y B, se realiza el siguiente cálculo para cada elemento que pertenece al clúster B:

Se realizan dos cálculos esenciales:

- En primer lugar, se calcula la diferencia promedio entre ese elemento y todos los demás elementos que componen el clúster B.

- Luego, se efectúa el cálculo de la diferencia promedio entre ese elemento y todos los elementos que se encuentran en el clúster A.

Seguidamente, se resta la segunda diferencia de la primera para cada elemento que pertenece a B. Si todas las diferencias resultan ser negativas, esto indica que se ha alcanzado un punto de detención en el algoritmo. No obstante, si se encuentra alguna diferencia positiva (lo que significa que el elemento en B, en promedio, está más cerca del clúster A que de los otros elementos en B), se selecciona el elemento en B que exhibe la mayor diferencia positiva. Acto seguido, se traslada al clúster A y se repite el proceso.

Este algoritmo posibilita la división de los datos en dos clústeres, denominados A y B, siguiendo un enfoque binario. Además, cabe destacar que este mismo procedimiento puede aplicarse de manera independiente para obtener divisiones binarias dentro de los clústeres A y B, lo que posibilita subdivisiones adicionales en cada subgrupo.

5.3.4 *Métodos no jerárquicos*

Para desarrollar la sección de métodos no jerárquicos las referencias principales han sido [9] y [28]. Cabe mencionar que, aunque esta sección se encuentre en el capítulo dedicado a la disciplina matemática, bien podría encontrarse perfectamente dentro del capítulo dedicado a la disciplina informática, dado a su naturaleza puramente experimental y algorítmica.

En contraste con los métodos jerárquicos en el análisis clúster, los métodos no jerárquicos adoptan un enfoque completamente diferente. Su objetivo primordial se centra en la clasificación de observaciones en un número predefinido de clústeres, que previamente ha sido denotado como k.

Aquí se destacan algunas características fundamentales de los métodos no jerárquicos:

- Estructura sin jerarquía: En contraposición a los métodos jerárquicos, los métodos no jerárquicos no involucran la construcción de una estructura de árbol. Se centran en la asignación directa de objetos a clústeres sin crear una jerarquía de fusiones o divisiones.

- Asignación directa: En estos métodos, la asignación de objetos a clústeres se produce una vez que se ha determinado previamente el número de clústeres que se formarán. A diferencia de los métodos jerárquicos, no se establece una jerarquía que implique fusiones o divisiones de clústeres.

- Método destacado: Dentro de los métodos no jerárquicos, el método de las *k*-medias (*k-means*) es uno de los más destacados. Este algoritmo, propuesto por MacQueen en 1967, goza de una reputación sólida como uno de los métodos más efectivos y ampliamente aplicados en la clasificación por clústeres.

- El algoritmo *k-means* parte de la premisa inicial de que existen *k* clústeres predefinidos. En iteraciones sucesivas, clasifica las observaciones en estos *k* clústeres. Durante estas iteraciones, el algoritmo modifica la posición de los centros de clúster con el objetivo de minimizar la distancia entre las observaciones y el centro del clúster al que están asignadas.

En resumen, los métodos no jerárquicos, como *k-means*, son especialmente adecuados cuando se dispone de información previa sobre el número deseado de clústeres y no se busca la creación de una estructura jerárquica. Estas técnicas se emplean extensamente en el ámbito de la Minería de Datos como herramientas de aprendizaje no supervisado para la clasificación de datos en grupos predefinidos.

5.3.4.1 *Método k-means*

El ampliamente utilizado algoritmo *k-means* [10] destaca por su alta eficiencia, lo que lo convierte en una elección frecuente para proyectos de agrupación a gran escala. Es importante señalar que el algoritmo *k-means* requiere acceso a los datos originales.

Este algoritmo inicia su proceso de una de dos maneras:

- Asigna inicialmente los elementos (observaciones) a uno de los *k* clústeres predefinidos y posteriormente calcula los centroides[10] de estos clústeres.

- Se especifican previamente los *k* centroides de los clústeres[11], que pueden ser elementos seleccionados al azar o pueden obtenerse al cortar un dendrograma a una altura apropiada.

El algoritmo continúa con un proceso iterativo en el que busca minimizar el Error Cuadrático Medio[12] (*ECM*). Este proceso implica reasignar elementos a los clústeres en cada iteración, deteniéndose cuando ninguna reasignación adicional reduce el valor del *ECM*.

10 Se recuerda que este elemento no es más que el punto del espacio que minimiza la suma de los cuadrados de las distancias de las observaciones al centroide que se utiliza en cada etapa.

11 Puntos semilla.

12 Error cuadrático medio. Es una medida de cuánta variación o dispersión existe dentro de los clústeres

Es importante tener en cuenta que la solución final, que representa una configuración de los elementos en k agrupaciones, generalmente no es única. El algoritmo *k-means* tiende a converger hacia un mínimo local del Error Cuadrático Medio. Por lo tanto, se recomienda ejecutar el algoritmo utilizando diferentes asignaciones iniciales aleatorias de los elementos a los k clústeres o seleccionando aleatoriamente k centroides iniciales. De esta manera, se aumenta la probabilidad de encontrar el mínimo más bajo del *ECM* y, en consecuencia, obtener la mejor solución de agrupación basada en k clústeres.

Algorithm 1 Algoritmo del método *clustering K-means* [9].

1. Input: Observaciones $\mathcal{L} = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, $K =$ número de clústeres.

2. Hacer una de las siguientes:
   - Realizar una asignación aleatoria inicial de los elementos –observaciones– en K clústeres y, para el clúster k, calcular su centroide actual, $\overline{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
   - Especificamos previamente los centroides de los K clústeres, $\overline{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.

3. Calculamos el Error Cuadrático Medio de cada observación con respecto al centroide de su clúster actual:

$$\text{ECM} = \sum_{k=1}^{K} \sum_{c(i)=k} \left(\mathbf{x}_i - \overline{\mathbf{x}}_k\right)^{\tau} \left(\mathbf{x}_i - \overline{\mathbf{x}}_k\right),$$

donde $\overline{\mathbf{x}}_k$ es el centroide del clúster k-ésimo y $c(i)$ es el clúster que contiene a $\mathbf{x}_i$.

4. Reasignar cada elemento al clúster cuyo centroide se encuentre más cerca de dicho elemento, de esta manera el *ECM* se verá reducido en gran magnitud. Debemos actualizar los centroides de cada clúster despues de esta reasignación.

5. Repetir los pasos 3 y 4 hasta que ninguna reasignación mejore –reduzca– el valor del *ECM*.

Método k-means – Inconvenientes

Aunque pueda parecer, de primeras, que el método *k-means* es óptimo para casi cualquier situación, y bajo cualquier circunstancia, esto no es así.

El algoritmo de *clustering k-means* es altamente sensible a la elección del conjunto inicial de centroides, comúnmente conocidos como *puntos semilla*. Esto plantea el desafío de determinar los puntos semilla que generen una estructura de clúster óptima. Un enfoque destacado para la selección de puntos semilla, propuesto por *Forgy* en 1965 [29], implica crear inicialmente k particiones mutuamente excluyentes de los datos, lo que permite calcular centroides distintos para cada partición. Estos centroides se utilizan como los centroides iniciales.

Otro aspecto importante es que se requiere conocer de antemano el número de clústeres, lo cual puede ser un desafío si no se dispone de información sólida sobre los datos. A pesar de esto, existen estrategias que abordan la determinación del número óptimo de clústeres.

Sin embargo, el algoritmo *k-means* puede resultar sensible a los *outliers*[13], ya que una media se ve fácilmente influida por los valores extremos, lo que puede afectar la calidad de la agrupación. Una solución a este problema implica la exclusión de los valores atípicos o la consideración de métodos más robustos como *k-medoids*.

5.3.4.2 *Método k-menoids*

El método *clustering k-medoids* es una variante de *k-means* más robusta frente a ruidos y *outliers*. En lugar de utilizar el punto medio como centro de un clúster, *k-medoids* utiliza un punto real del clúster para representarlo. El *medoide*.

Pero este método sigue sin ser el método definitivo para cualquier situación. El método *k-medoids*, si bien es efectivo en conjuntos de datos pequeños, presenta una desventaja importante cuando se aplica a grandes conjuntos de datos; su eficiencia es limitada. En este algoritmo, los centroides utilizados en los pasos 2, 3 y 4 del algoritmo *k-means* se reemplazan por *medoides*, y la función objetivo, que en *k-means* es el Error Cuadrático Medio (ECM), se sustituye por el Error Cuadrático Medio de Medoides (ECM_{med}).

13 Valores atípicos.

Algorithm 2 Algoritmo del método *clustering K-medoids* [9].

1. Input: Matriz de proximidad relativa $\mathbf{D} = \left(d_{ij}\right)$; $K =$ número de clústeres.

2. Realizar la asignación inicial de los elementos –observaciones– en K clústeres.

3. Localizar el *medoide* de cada clúster. El medoide del clúster k-ésimo se define como el elemento en dicho cluster que minimiza la disimilitud respecto a todos los demás elementos del mismo, con $k = 1, 2, \ldots, K$.

4. Para el k-ésimo clúster, reasignar el elemento i_k al cluster cuyo *medoide* se más cercano a él, de esta forma, la función objetivo,

$$\mathrm{ECM}_{\mathrm{med}} = \sum_{k=1}^{K} \sum_{c(i)=k} d_{ii_k}$$

se reduce en gran magnitud, donde $c(i)$ es el clúster que contiene al elemento i-ésimo.

5. Repetir los pasos 3 y 4 hasta que ninguna reasignación mejore –reduzca– el valor del ECM_{med}.

Parte III

FUNDAMENTOS INFORMÁTICOS

Conceptos fundamentales de la informática necesaria para comprender la aplicación práctica.

6

PROYECTO GEO (GENE EXPRESSION OMNIBUS)

A la hora de redactar este capítulo, las referencias bibliográficas principales consultadas son [30] y [31].

El proyecto *Gene Expression Omnibus* (*GEO*) del *National Center for Biotechnology Information* se inició en respuesta a la creciente demanda de un repositorio público de datos de expresión génica de alto rendimiento. *GEO* es un repositorio que archiva y distribuye gratuitamente una gran cantidad de datos moleculares de alto rendimiento, predominantemente datos de expresión génica generados por la tecnología de *microarrays* de *DNA*.

La base de datos tiene un diseño flexible que puede manejar diversos estilos de datos, tanto procesados como sin procesar, en una infraestructura de apoyo a la Información Mínima sobre un Experimento de *Microarrays*[1] que promueve la presentación de datos totalmente anotados. *GEO* almacena actualmente alrededor de mil millones de medidas individuales de expresión génica, derivadas de más de 100 organismos, enviadas por más de 1500 laboratorios, que abordan una amplia gama de fenómenos biológicos. Para maximizar la utilidad de estos datos, se han implementado varias interfaces y aplicaciones web fáciles de usar que permiten una exploración, consulta y visualización efectivas de estos datos a nivel de genes individuales o de estudios completos.

Este capítulo describe cómo se almacenan los datos, los procedimientos de envío y los mecanismos de recuperación y consulta de datos. *GEO* es de acceso público en `http://www.ncbi.nlm.nih.gov/projects/geo/`.

1 *GEO* está diseñado para recopilar y presentar datos que cumplen con los estándares de Información Mínima sobre un Experimento de *Microarrays* (*MIAME*), que es un conjunto de pautas para garantizar que los datos de los experimentos de *microarrays* sean completos y rastreables [32].

6.1 FINALIDAD Y ALCANCE DE GENE EXPRESSION OMNIBUS (GEO)

La era postgenómica –a la que ya se ha referido anteriormente– ha dado lugar a una multitud de metodologías de alto rendimiento que generan volúmenes masivos de datos de expresión génica. El repositorio *GEO* fue creado por el *NCBI* en el año 2000 para albergar y distribuir estos datos al público sin restricciones ni requisitos de acceso. La función principal de *GEO* es archivar los datos y funcionar como centro de depósito y recuperación de datos. *ArrayExpress*[2] cumple una función similar.

GEO es actualmente la mayor fuente de datos de expresión génica totalmente pública. En el momento de la escritura del artículo referenciado, la base de datos contenía más de 80.000 muestras, que comprenden aproximadamente mil millones de mediciones de expresión individual, 13 millones de perfiles de expresión génica, para más de 100 organismos, enviados por casi 1.500 laboratorios. Estos datos abordan una gran diversidad de temas biológicos, como enfermedades, desarrollo, evolución, metabolismo, toxicología, inmunidad, ecología y transgénesis. La mayoría de los datos son proporcionados por la comunidad investigadora en cumplimiento de las cláusulas de subvenciones o revistas que exigen que los datos de *microarrays* estén disponibles en un repositorio público, con el objetivo de facilitar la evaluación independiente de los resultados, el reanálisis y el acceso completo a todas las partes del estudio.

Aunque se trata principalmente de un servicio de almacenamiento y recuperación de datos, desde el principio quedó claro que el recurso también debía permitir la búsqueda eficaz y la minería de datos para identificar entradas de interés. En consecuencia, se han desarrollado varias herramientas web de consulta fáciles de usar para ayudar incluso a quienes no están familiarizados con la tecnología de *microarrays* a explorar y analizar eficazmente los datos de *GEO*.

Sin embargo, es importante tener en cuenta que *GEO* no está pensado para ser utilizado como un sistema de gestión de información de laboratorio o un entorno de pre-análisis, ya que los datos enviados a *GEO* son generalmente datos procesados que forman la base para la discusión en los manuscritos que los acompañan.

2 Almacena datos de experimentos genómicos de alto rendimiento y ofrece datos a la comunidad investigadora para su utilización. Las lecturas de datos sin procesar de los estudios de secuenciación de alto rendimiento se envían al Archivo Europeo de Nucleótidos (*ENA*), y se proporcionan enlaces para descargar dichas lecturas desde el *ENA*.

6.2 ESTRUCTURA

La arquitectura de la base de datos *GEO* está diseñada para la captura, almacenamiento y recuperación eficientes de conjuntos heterogéneos de datos moleculares de alto rendimiento. La estructura es lo suficientemente flexible como para adaptarse a la evolución de las tecnologías más avanzadas. Existen muchas variedades diferentes de tecnología de *microarrays*, y los investigadores utilizan una amplia variedad de paquetes de hardware y software para generar y procesar los datos. En consecuencia, los datos tienen muchos estilos diferentes y comprenden contenidos diversos. Y lo que es más importante, los datos de expresión génica carecen de valor a menos que se complementen con su contexto biológico y con las metodologías de análisis de datos con las que se generaron.

Los extensos detalles técnicos sobre el diseño de la base de datos y el flujo de datos están fuera del alcance de este trabajo, pero una cuestión importante que ayuda a entender mejor esta base de datos es que los datos y los metadatos se almacenan por separado dentro de la base de datos.

La versatilidad de *GEO* se atribuye en gran medida al hecho de que los datos tabulares[3] no están completamente granulados en la base de datos central, sino que se tratan como *blobs*, es decir, tablas de texto comprimido, delimitadas por tabulaciones que pueden contener cualquier número de filas o columnas. Los datos de las columnas seleccionadas se extraen a una base de datos secundaria y se utilizan en posteriores aplicaciones de indexación y consulta. Los metadatos descriptivos o informativos se normalizan completamente en el esquema según sea necesario.

6.2.1 *Datos proporcionados por investigadores*

Los datos suministrados por los investigadores se almacenan en tres entidades principales en una base de datos relacional de Microsoft SQL Server.

- *Platform* (Plataforma): Incluye una descripción resumida de la matriz y una tabla de datos que define la plantilla de la matriz. Cada fila de la tabla corresponde a un único elemento e incluye la anotación de la secuencia y la información de seguimiento facilitada por el remitente.

3 Los datos tabulares son aquellos que pueden organizarse en un formato de marco de datos bidimensional. En esta estructura, cada registro se representa en una fila, y cada fila contiene una o más columnas. Dentro de estas columnas, los datos pueden ser de distintos tipos, como numéricos, categóricos o de texto, y se almacenan en las celdas del marco de datos.

- *Sample* (Muestra): Incluye una descripción de la fuente biológica y los protocolos experimentales a los que se ha sometido, así como una tabla de datos que contiene las mediciones de hibridación de cada elemento en la plataforma correspondiente.

- *Series* (Serie): Define un conjunto de muestras relacionadas que se consideran parte de un estudio y describe el objetivo y el diseño general del estudio.

A cada uno de estos tres objetos se le asigna un número de acceso que puede utilizarse para citar y recuperar los registros. Además de las tablas de datos de muestras y la información descriptiva, se aceptan archivos complementarios adjuntos, como imágenes originales de escaneado de *microarrays*, que se almacenan en un sitio *FTP*[4] con enlaces a bases de datos.

6.2.2 *DataSets construidos por GEO*

A pesar de la variabilidad en el estilo y el contenido de los datos, siempre se proporciona la siguiente información:

- Información de seguimiento de la secuencia para cada característica del *array*.

- Medidas de hibridación normalizadas.

- Una descripción de la fuente biológica utilizada en cada hibridación.

Utilizando una combinación de extracción de datos automatizada y tratamiento manual, esta información se presenta en una unidad de nivel superior denominada *GEO DataSet* (ver Figura 9). Un *DataSet* representa una colección de hibridaciones de muestras procesadas de forma similar y relacionadas experimentalmente, y proporciona una sinopsis de un estudio. Las muestras dentro de un *DataSet* se categorizan, además, según variables experimentales, por ejemplo, se organizan por género o por el estado de la enfermedad.

Un *DataSet* proporciona dos perspectivas distintas de los datos.

- Una representación centrada en el experimento que encapsula todo el estudio. Esta información se presenta como un *DataSet record*. Los *DataSet records*

4 Servidor de protocolo de transferencia de archivos. *FTP* es un protocolo de red que se utiliza para transferir archivos entre un cliente y un servidor a través de una conexión de red. El uso de un servidor *FTP* permite una distribución eficiente de archivos grandes y diversos relacionados con los datos del proyecto o experimento.

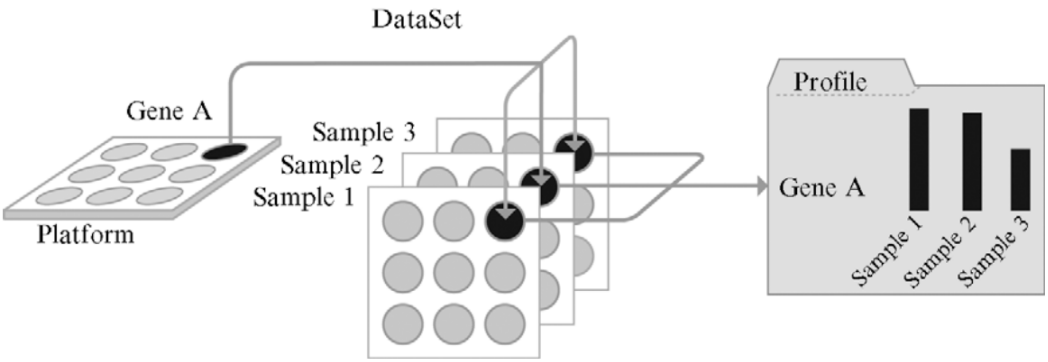

Figura 9: Diagrama de las relaciones entre *platforms*, *samples*, *DataSets* y perfiles *GEO*, extraída de [31]

comprenden una sinopsis del experimento, un desglose de las variables fenotípicas, varias herramientas de visualización y análisis de datos y opciones de descarga (ver Figura 10).

- Una representación centrada en el gen que presenta mediciones cuantitativas de la expresión génica para un gen a través de un *DataSet*. Esta información se presenta como un *Perfil GEO*. Un perfil *GEO* incluye la anotación de la identidad del gen, el título del conjunto de datos, enlaces a información auxiliar y un gráfico que muestra el nivel de expresión de ese gen en cada muestra del conjunto de datos (ver Figura 11).

Los *DataSets* permiten la transformación de diversos estilos de datos enviados para que sean fácilmente accesibles en un formato uniforme sobre el que basar las herramientas de análisis de datos posteriores.

6.3 ENVÍO DE DATOS

La base de datos *GEO* es una infraestructura de apoyo a *MIAME*. Aunque los procedimientos de envío promueven el cumplimiento de *MIAME*, en última instancia es responsabilidad de los investigadores remitentes asegurarse de que sus datos están suficientemente bien anotados. Existen varias formas de depositar datos en *GEO*. La decisión de qué método utilizar depende de la cantidad de datos que se vayan a enviar, del formato en el que se encuentren y del nivel de conocimientos informáticos del remitente. Independientemente del método de envío, los registros finales de *GEO* tienen el mismo aspecto y contienen información equivalente.

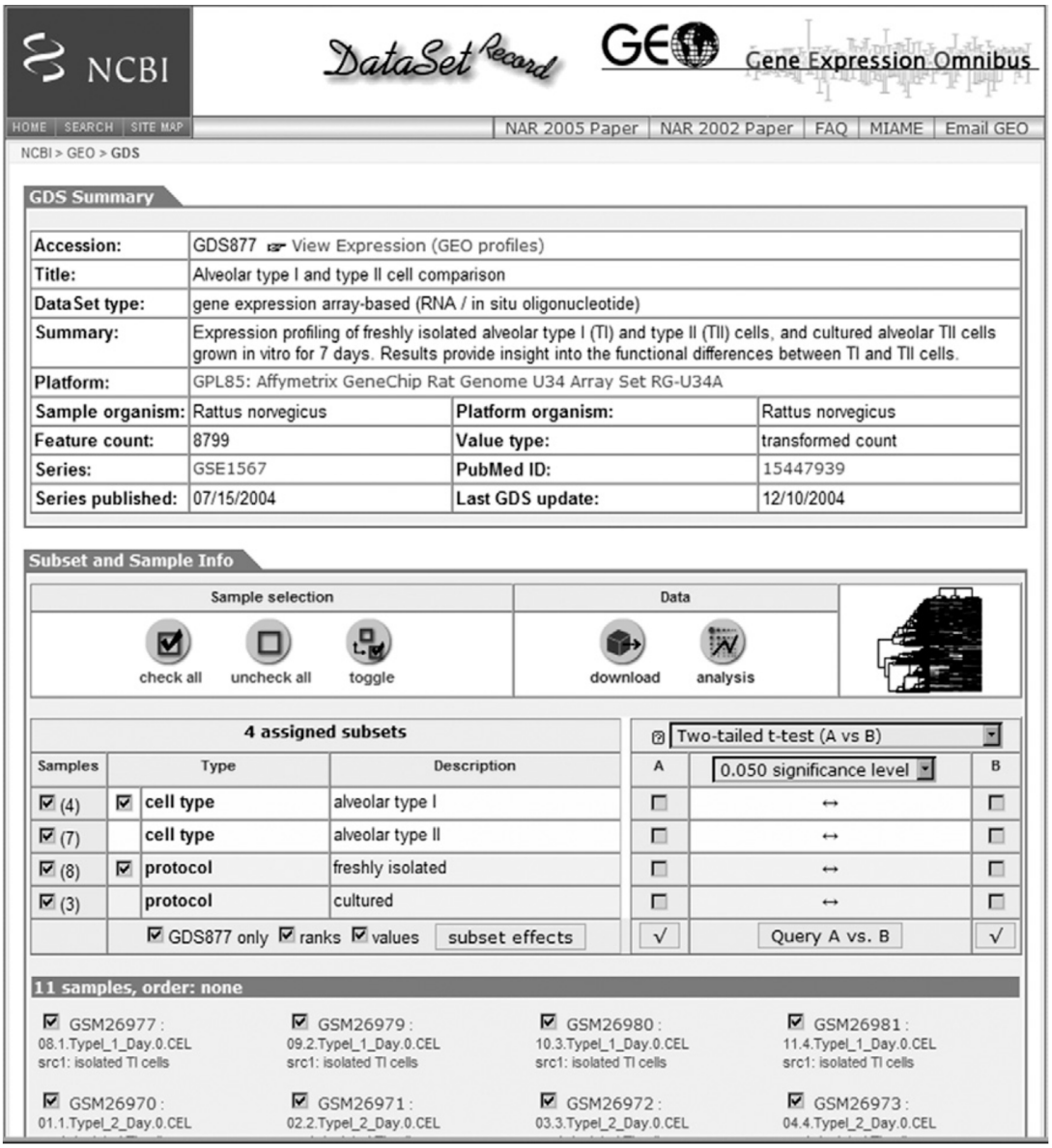

Figura 10: *Screenshot* de un *DataSet* típico *GDS877*, extraída de [33]

- Envío web: Está diseñado para que sea un envío rápido y sencillo de registros individuales por parte de remitentes ocasionales. Esta vía comprende un conjunto de formularios web interactivos que ofrecen un sencillo procedimiento paso a paso para el depósito de tablas de datos e información descriptiva adjunta.

- Envío directo por lotes (*batch*) utilizando el formato *Simple Omnibus Format in Text* (*SOFT*): *SOFT* es un formato simple diseñado para el envío (y recuperación) rápido de datos por lotes. Un único archivo *SOFT* puede contener tanto tablas de datos como información descriptiva adjunta para múltiples

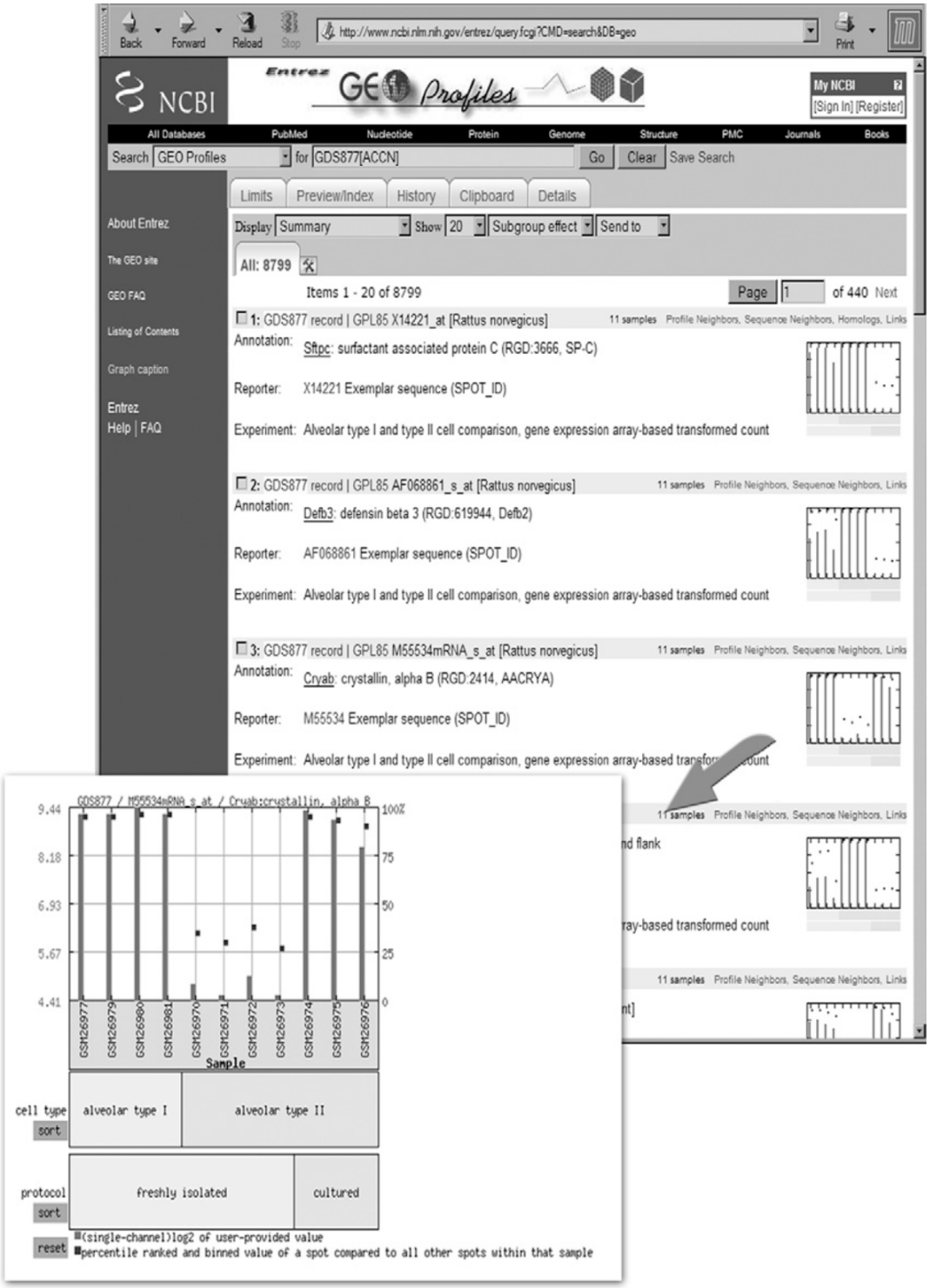

Figura 11: *Screenshot* de los resultados de la recuperación de perfiles *GEO* de *Entrez*; cada entidad incluye un identificador de secuencia e información del *DataSet* y una imagen en miniatura del perfil, extraída de [33]

plataformas, muestras y/o registros de series. Los archivos *SOFT* pueden generarse fácilmente a partir de aplicaciones comunes de bases de datos y hojas de cálculo, y pueden cargarse directamente en la base de datos.

- Envío mediante *FTP*: Si los datos ya están en un formato matricial, se recomienda el envío a través de una hoja de cálculo con formato *SOFT*. Estos tipos de archivos se transfieren a *GEO* a través de *FTP*.

Todos los envíos son revisados y comprobados por un revisor designado por *GEO*, que se asegura de que los registros contienen información significativa y están organizados correctamente. Si no se identifican problemas estructurales o de contenido, los envíos se aprueban y se les asignan números de acceso *GEO*. Los números de acceso *GEO* son únicos y estables y pueden ser citados en artículos y trabajos.

6.4 BÚSQUEDA Y DESCARGA DE DATOS

En el momento de la redacción de [30], era posible recuperar *platforms*, *samples* y *series* completas únicamente por número de acceso. Se ha llevado a cabo una amplia indexación y vinculación de los datos de *GEO*, que pueden consultarse a través de una base de datos Entrez[5], denominada *Entrez ProbeSet*. Las coincidencias están vinculadas a la entrada *GEO* completa, así como a otras bases de datos *Entrez* y a entradas *Entrez ProbeSet* relacionadas.

Existen varias opciones de descarga.

- Cada *platform*, *sample* y *series* tiene un mecanismo en la cabecera de la página que permite la descarga (formato *SOFT*) o la visualización (*HTML*) de ese registro y/o de los registros relacionados, con la opción de restringir sólo los datos descriptivos o los datos tabulares.

- Los *DataSets* incluyen un enlace para la descarga de una matriz de texto delimitada por tabuladores y la anotación genética del elemento asociado.

- Todas las *platforms*, *samples*, *series*, *DataSets* y datos adicionales están disponibles para su descarga masiva a través de *FTP* en `ftp://ftp.ncbi.nih.gov/pub/geo/`.

5 *Entrez* es un sistema de bases de datos de biología molecular que proporciona acceso a datos de secuencias de nucleótidos y proteínas, información centrada en genes y cartografía genómica, datos de estructuras tridimensionales, *PubMed MEDLINE* y mucho más. El sistema está producido por el *National Center for Biotechnology Information* (*NCBI*) y está disponible a través de Internet [34].

7

ESTRUCTURAS DE ALMACENAMIENTO

Las tecnologías de *microarrays*, *RNA*-Seq (Secuenciación de *RNA*) y *scRNA*-Seq (Secuenciación de *RNA* a nivel de célula única) han desempeñado un papel fundamental en la revolución de la genómica. Estas técnicas permiten a los científicos investigar y comprender a un nivel sin precedentes la expresión genética y la regulación de los genes en una variedad de organismos y contextos biológicos.

Las estructuras de almacenamiento desempeñan un papel crucial en la gestión y el análisis de los datos generados por estas tecnologías de secuenciación. Estas estructuras se han convertido en una parte esencial de la investigación en genómica, ya que la cantidad de datos producidos puede ser abrumadora y su análisis requiere sistemas de almacenamiento eficientes y organizados.

A continuación, se ofrecerá una breve introducción a las estructuras de almacenamiento específicas para cada una de estas tecnologías:

7.1 MICROARRAYS

Los *microarrays* son una tecnología que permite medir la expresión de miles de genes simultáneamente. Los datos de *microarrays* se almacenan típicamente en formato tabular, donde las filas representan genes y las columnas representan muestras. Cada celda de la tabla contiene información sobre la expresión del gen en una muestra específica. El almacenamiento eficiente de estos datos es esencial para llevar a cabo análisis estadísticos y visualizaciones que ayuden a descubrir patrones de expresión génica.

La expresión de un gen se refiere al proceso mediante el cual se convierte la información genética en una serie de copias de *RNA* mensajero (*mRNA*). Los *microarrays*, en este contexto, son herramientas que se utilizan para cuantificar la cantidad de *mRNA* presente para cada gen en una muestra. La medida de

expresión de un gen se representa como un valor de luminiscencia (mayor luminiscencia a mayor presencia), que está directamente relacionado con la cantidad de *mRNA* específico de ese gen presente en el chip de *DNA* (utilizado para coger la muestra).

Cuando se leen datos generados a través de un chip de *DNA*[1], se está analizando información obtenida de chips de oligonucleótidos. Estos chips contienen múltiples sondas (pequeñas secuencias de *DNA*) relacionadas con un gen en particular. Cada sonda consiste en pequeñas celdas que contienen oligonucleótidos con aproximadamente 25 pares de bases, y cada oligonucleótido es una parte específica de un gen en particular. En teoría, solo el *RNA* derivado de ese gen específico debería unirse a la sonda correspondiente, esto es lo que se conoce como hibridación. Por tanto, cuanto más coincidencia haya, mayor luminiscencia (es referido así porque se utiliza un material fluorescente que se ilumina).

El objetivo principal de este trabajo será el estudio analítico exploratorio mediante la técnica *RNA*-Seq, debido a que es la más utilizada a día de hoy, y es más efectiva que los *microarrays*. Sin embargo, el análisis de expresión génica con *microarrays* ha sido históricamente muy importante y tiene una gran presencia a lo largo de la bibliografía de ciencias ómicas. Un breve análisis de expresión génica utilizando *microarrays* en español se puede encontrar en [35]. Es muy recomendable su visualización en caso de querer iniciarse en el empleo de esta técnica.

1 Por ejemplo, mediante un escáner *Affymetrix GeneChip*.

7.2 RNA-SEQ

Hasta hace unos años, los *microarrays* eran la técnica predominante para inferir perfiles de transcripción. Sin embargo, estas herramientas tenían limitaciones notables, como la necesidad de conocimiento previo de los genes y su incapacidad para identificar variantes genéticas. Esto impulsó la búsqueda de nuevos enfoques que superaran estas restricciones.

En la actualidad, *RNA*-Seq se ha convertido en una de las tecnologías más solicitadas debido a su capacidad para descubrir nuevos genes, su aplicabilidad en una amplia gama de cuestiones científicas y su alta sensibilidad. La secuenciación de *RNA*, o *RNA*-Seq, es una técnica que permite cuantificar y caracterizar la expresión génica y las variantes de *RNA*. Los datos de RNA-Seq suelen generarse en forma de archivos de lecturas (reads) que representan fragmentos de *RNA*. Estos archivos de lecturas pueden ser masivos y requieren sistemas de almacenamiento y bases de datos especializadas para su gestión. Además, los datos de *RNA*-Seq pueden ser alineados y procesados para cuantificar la abundancia de transcriptomas y realizar un análisis diferencial de expresión [36]. Es por esto, que el desarrollo práctico de este tema será el basado en esta técnica, que se encuentra más adelante, en el Capítulo 9.

7.2.1 *Formatos*

Se ha introducido ya en qué formatos se almacenan y manejan los *microarrays*. La técnica *RNA*-Seq requiere unos formatos específicos para su tratamiento, que se exponen a continuación. En particular, son dos los más utilizados: *FASTA* y *FASTQ*.

7.2.1.1 *FASTA*

El formato *FASTA* es un formato de texto ampliamente utilizado para representar secuencias de nucleótidos o aminoácidos. En este formato, tanto los nucleótidos como los aminoácidos se representan mediante una sola letra, y se incluyen símbolos para denotar espacios –*gaps*– o posiciones desconocidas en la secuencia. Su estructura es sencilla y consta de dos líneas. La primera línea comienza con el símbolo ">" seguido de una descripción de la secuencia. En la siguiente línea, se presenta la secuencia de bases o aminoácidos. Es recomendable que no se superen

las 80 columnas de texto en una sola línea, y se pueden tener tantas líneas como sea necesario para representar la secuencia completa.

7.2.1.2 *FASTQ*

El formato *FASTQ*, ampliamente utilizado para datos de secuencias, se compone de cuatro líneas por lectura, y cada una proporciona información específica:

- La primera línea, que inicia con el carácter "@", contiene el nombre de la secuencia y puede incluir una descripción opcional.
- La segunda línea contiene la secuencia real de letras, dependiendo del tipo de secuencia (nucleótidos o aminoácidos).
- La tercera línea, que comienza con el carácter "+", suele contener información adicional sobre la secuencia, aunque es opcional.
- La cuarta línea proporciona una medida de la calidad o confiabilidad de cada base en la secuencia de la segunda línea. Esta calidad se cuantifica utilizando el índice *Phred*[2] y su correspondiente codificación.

A continuación se presenta un ejemplo de un formato *FASTQ*:

```
@SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
ACAGGGACGCCATCGAATCCGGATCNTNNNNNNNNNNNNANNNNNNNNNN
+SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
dee\edYcdc`bbY`S]bb_]]Ua^BBBBBBBBBBBBBBBBBBBBBBBBB
```

7.3 scRNA-seq (single-cell rna-seq)

La técnica conocida como *scRNA*-Seq se ha consolidado como la opción principal para investigar la diversidad celular, ya que permite analizar cada célula de manera independiente dentro de una misma muestra. Sin embargo, se verá que no es oro todo lo que reluce. A pesar de ser también una técnica de secuenciación, es una variante concreta de *RNA*-Seq que cuantifica el transcriptoma en células individuales. Esta técnica ha permitido el estudio de la heterogeneidad celular y la caracterización de las distintas poblaciones celulares presentes en una misma muestra [37].

2 Cuantifica la confianza o precisión o calidad de la cada una de las bases que tenemos en la secuencia

La secuenciación célula a célula ha proporcionado una visión más precisa de los procesos moleculares, eliminando la suposición de la no variabilidad intracelular. Sin embargo, el primer paso en la tecnología *scRNA*-Seq involucra el desafiante aislamiento de las células, que a menudo requiere marcadores específicos para distinguir poblaciones celulares. Además, se enfrenta a obstáculos relacionados con el tiempo y el presupuesto, lo que puede dificultar la realización de investigaciones.

En el ámbito biomolecular, es crucial conocer los tipos celulares que componen una muestra, por ejemplo, por medio de la realización de una biopsia, y caracterizarlos. Por lo tanto, es necesario desarrollar algoritmos que descompongan la mezcla de expresión génica obtenida, ya sea a través de *microarrays* o *RNA*-Seq, con el fin de determinar el número y la contribución de las diversas subpoblaciones celulares presentes en la muestra [38].

La secuenciación de *RNA* de célula única es una técnica de vanguardia en la actualidad, perteneciente a las tecnologías de secuenciación de nueva generación (*NGS – Next-Generation-Sequencing*). Esta técnica ha revolucionado el estudio de muestras heterogéneas al considerar la variabilidad de célula a célula y permitir la identificación de tipos celulares en cada muestra [39]. A pesar de sus numerosas ventajas, como la capacidad de definir los tipos celulares presentes, el *scRNA*-Seq enfrenta limitaciones significativas en términos de coste, tiempo y la dificultad de disociar tejidos *fibrosos* y *pequeños*, como en el caso de aneurismas. Además, se ven afectados por los *efectos por lote* que pueden generar factores de confusión en los resultados. Como respuesta a estos desafíos, se ha propuesto el desarrollo de un método computacional que permita identificar los tipos celulares sin requerir el aislamiento manual de las células en el laboratorio.

8

R / BIOCONDUCTOR

Las principales referencias bibliográficas de este capítulo han sido [40], [41] y [42].

R y *Bioconductor* han sido desarrollados por un esfuerzo colaborativo que involucra a numerosos individuos, algunos de los cuales tienen raíces en el ámbito académico, mientras que en otras ocasiones provienen de empresas que encuentran beneficio en la disponibilidad de software compatible con su hardware.

8.1 SOFTWARE R

El software *R* es un lenguaje de programación y entorno de software de código abierto, que se ha establecido como una herramienta esencial en el ámbito de la estadística –en general– y del análisis de datos genómicos –en particular–. *R* no solo ha demostrado ser eficaz en el procesamiento y análisis de datos estadísticos, sino que también ha evolucionado para satisfacer las demandas específicas de la genómica y la biología computacional.

8.1.1 *Importancia de R en el Análisis de Datos Genómicos*

R ha emergido como un componente integral en la investigación genómica y ofrece una serie de ventajas significativas para los investigadores. En primer lugar cuenta con una amplia variedad de paquetes y herramientas estadísticas. *R* posee una extensa biblioteca de paquetes desarrollados por la comunidad que abarcan desde análisis estadísticos tradicionales hasta técnicas más avanzadas específicas para la genómica. El proyecto *Bioconductor*, una plataforma dedicada al análisis de datos biológicos, proporciona una serie de paquetes y recursos especializados. Luego, la capacidad de *R* para crear gráficos de alta calidad y visualizaciones es

fundamental para la representación de datos complejos, en especial, enfocado al ámbito de este trabajo, destacando perfiles de expresión génica, estructuras de *DNA*, o análisis de metilación. Los usuarios de *R* pueden escribir sus propias funciones y scripts, lo que les brinda un control total sobre el análisis y la capacidad de personalizar los procedimientos de acuerdo con sus necesidades y objetivos específicos de investigación.

Por último, pero no menos importante, la generación de informes y documentos reproducibles a través de herramientas como *R Markdown* promueve la transparencia y la reproducibilidad en la investigación estadística y genómica, mejorando la comunicación científica y la colaboración entre investigadores. Profundizando más en este aspecto, *R* ofrece herramientas para lograr su documentación y reproducibilidad.

- ***R Markdown*** es una herramienta que permite crear documentos que combinan código *R*, resultados y texto descriptivo en un solo archivo. Esto facilita la creación de informes reproducibles que documentan cada paso del análisis.

- Utilizar sistemas de control de versiones como ***Git*** junto con *R* facilita la colaboración y el seguimiento de cambios en el código y los datos.

- ***RStudio*** proporciona una interfaz amigable para la organización de proyectos, lo que ayuda a mantener ordenados los scripts, los datos y la documentación relacionada con un proyecto genómico.

8.2 BIOCONDUCTOR

El proyecto *Bioconductor* es una plataforma de código abierto y un repositorio de software diseñado específicamente para el análisis y la visualización de datos genómicos y biológicos de alto rendimiento. Las herramientas contenidas en *Bioconductor* representan muchos métodos de vanguardia para el análisis de datos de *microarrays* y genómica. Fue creado en 2001 y se ha convertido en una de las herramientas más esenciales en la comunidad de genómica y biología computacional. Cabe mencionar que *Bioconductor* está basado en el lenguaje de programación *R*.

8.2.1 *Importancia General de Bioconductor*

Bioconductor desempeña un papel fundamental en la investigación en genómica y biología computacional por varias razones. En primer lugar, permite un acceso a herramientas especializadas. Este proporciona una amplia gama de paquetes y herramientas específicas para el análisis de datos genómicos, como la secuenciación de próxima generación (*NGS*), donde se encuentra la ténica *scRNA-Seq* que trataremos en el próximo capítulo, *microarrays*, proteómica y más. Esto facilita la aplicación de análisis avanzados en investigación biológica. Además, dispone de un mantenimiento y actualizaciones constantes, ya que la comunidad de *Bioconductor* trabaja activamente en el desarrollo y mantenimiento de paquetes, lo que garantiza que los investigadores tengan acceso a las últimas técnicas y enfoques en genómica y análisis biológico. Por último, permite la reproducibilidad de la investigación, puesto que *Bioconductor* proporciona herramientas y promueve prácticas que permiten a otros investigadores replicar y verificar los resultados.

Además, dentro del análisis de datos genómicos, *Bioconductor* es fundamental en el preprocesamiento de datos, en el análisis diferencial de expresión génica, en el análisis de variantes genéticas y en la visualización de datos biológicos.

8.3 PAQUETES

Se ha llevado a cabo la instalación de todos los paquetes utilizados en la aplicación posterior asegurando que sean compatibles con versiones de *R* iguales o superiores a 4.3.1. Esta precaución ha sido adoptada para evitar posibles conflictos que pudieran surgir entre los distintos paquetes. En el marco de este trabajo, se han empleado las siguientes versiones de software: *R* 4.3.1 (2023-06-16), *BiocManager* 1.30.22 y *RStudio* 2023.06.0+421. Adicionalmente, se ha tomado en consideración la incorporación de paquetes de R disponibles en el repositorio *Comprehensive R Archive Network* (*CRAN*) `https://cran.r-project.org` y paquetes que no se han utilizado en la versión final, pero que son útiles para trabajos relacionados.

8.3.1 *Paquetes de CRAN*

8.3.1.1 *ggplot2*

El paquete *ggplot2* es un popular paquete de visualización de datos en *R* que proporciona una forma elegante y flexible de crear gráficos de alta calidad, incluyendo gráficos de dispersión, barras, líneas y mucho más.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("ggplot2")
```

Detalles

En esta sección, se mostrarán los detalles sobre este paquete. Se comentarán y explicarán cada una de las secciones que aquí aparecen, con el fin de quedar explicado para este paquete, y así, para los próximos de esta sección.

Version	3.4.4
Depends	R ($\geq$ 3.3)
Imports	cli, glue, grDevices, grid, gtable ($\geq$ 0.1.1), isoband, lifecycle ($>$ 1.0.1), MASS, mgcv, rlang ($\geq$ 1.1.0), scales ($\geq$ 1.2.0), stats, tibble, vctrs ($\geq$ 0.5.0), withr ($\geq$ 2.5.0)
Suggests	covr, dplyr, ggplot2movies, hexbin, Hmisc, knitr, lattice, mapproj, maps, multcomp, munsell, nlme, profvis, quantreg, ragg, RColorBrewer, rmarkdown, rpart, sf ($\geq$ 0.7-3), svglite ($\geq$ 1.2.0.9001), testthat ($\geq$ 3.1.2), vdiffr ($\geq$ 1.0.0), xml2
Enhances	sp
Published	2023-10-12
Author	Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington
Maintainer	Thomas Lin Pedersen <thomas.pedersen@posit.co>
BugReports	https://github.com/tidyverse/ggplot2/issues
License	MIT
URL	https://github.com/tidyverse/ggplot2
NeedsCompilation	no

A continuación, se procede con la explicación de cada sección.

- **Version**: Indica la versión actual del paquete.
- **Depends**: Lista los paquetes de *R* de los que este paquete depende para funcionar correctamente.
- **Imports**: Esta sección muestra cualquier otro paquete de *R* que se importa para utilizar funciones específicas, pero que no son estrictamente necesarios para el funcionamiento básico del paquete.
- **Suggests**: Lista paquetes que no son necesarios para el funcionamiento básico del paquete, pero que se sugieren para una funcionalidad adicional o mejoras en la experiencia del usuario.
- **Enhances**: Si el paquete mejora o amplía la funcionalidad de otros paquetes se mencionaría aquí. Esta sección puede indicar relaciones entre diferentes paquetes.
- **Published**: Muestra la fecha de publicación de la versión actual del paquete.
- **Author**: Enumera a los autores del paquete.
- **Maintainer**: Proporciona la información de contacto del mantenedor actual del paquete.
- **BugReports**: Indica la ubicación en la que los usuarios pueden informar sobre errores o problemas relacionados con el paquete.
- **License**: Especifica la licencia bajo la cual se distribuye el paquete.
- **URL**: Proporciona un enlace web donde los usuarios pueden encontrar más información sobre el paquete.
- **NeedsCompilation**: Indica si el paquete requiere compilación de código para su instalación.

8.3.1.2 *BiocManager*

BiocManager es un paquete en *R* que simplifica la instalación y gestión de paquetes del proyecto *Bioconductor*.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("BiocManager", repos = "https://cloud.r-project.org")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	1.30.22
Depends	
Imports	utils
Suggests	BiocVersion, remotes, rmarkdown, testthat, withr, curl, knitr
Enhances	
Published	2023-08-08
Author	Martin Morgan, Marcel Ramos
Maintainer	Marcel Ramos <marcel.ramos@roswellpark.org>
BugReports	https://github.com/Bioconductor/BiocManager/issues
License	Artistic-2.0
URL	https://Bioconductor.github.io/BiocManager/
NeedsCompilation	no

8.3.1.3 *glmpca*

Implementa una versión generalizada del análisis de componentes principales (*GLM-PCA*) para la reducción dimensional de datos distribuidos de forma no normal, como recuentos o matrices binarias.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("glmpca")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	0.2.0
Depends	R ($\geq$ 3.5)
Imports	MASS, methods, stats, utils
Suggests	covr, ggplot2, knitr, logisticPCA, markdown, Matrix, testthat
Enhances	
Published	2020-07-18
Author	F. William Townes, Kelly Street, Jake Yeung
Maintainer	F. William Townes <will.townes@gmail.com>
BugReports	https://github.com/willtownes/glmpca/issues
License	LGPL ($\geq$ 3)
URL	https://github.com/willtownes/glmpca
NeedsCompilation	no

8.3.1.4 *pheatmap*

El paquete *pheatmap* en *R* es una herramienta ampliamente utilizada para crear mapas de calor a partir de datos matriciales, lo que lo convierte en una poderosa herramienta de visualización en bioinformática y análisis de datos. Pheatmap permite a los usuarios personalizar la apariencia de los mapas de calor de manera flexible, ajustando colores, tamaños de letra y otros aspectos estéticos.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("pheatmap")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	1.0.12
Depends	R ($\geq$ 2.0)
Imports	grid, RColorBrewer, scales, gtable, stats, grDevices, graphics
Suggests	
Enhances	
Published	2019-01-04
Author	Raivo Kolde
Maintainer	Raivo Kolde <rkolde@gmail.com>
BugReports	
License	GPL-2
URL	https://cran.r-project.org/web/packages/pheatmap/pheatmap.pdf
NeedsCompilation	no

8.3.1.5 *archdata*

El paquete *archdata* en *R* es una valiosa herramienta para la exploración y análisis de datos arqueológicos y de patrimonio histórico. Este paquete proporciona conjuntos de datos y funciones específicas para la gestión, análisis y visualización de datos arqueológicos, incluyendo información sobre hallazgos, excavaciones y sitios.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("archdata")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	1.2-1
Depends	
Imports	
Suggests	ca, circular, plotrix, MASS, spatstat
Enhances	
Published	2021-01-12
Author	David L. Carlson, Georg Roth
Maintainer	David L. Carlson <dcarlson@tamu.edu>
BugReports	
License	GPL-2 | GPL-3
URL	https://cran.r-project.org/web/packages/archdata/archdata.pdf
NeedsCompilation	no

8.3.1.6 *psych*

El paquete *psych* en *R* es una herramienta esencial para el análisis psicométrico y estadístico en psicología y ciencias sociales. Ofrece una amplia gama de funciones que permiten a los investigadores realizar análisis factoriales, análisis de escalamiento multidimensional, pruebas de confiabilidad, análisis de componentes principales, y una variedad de técnicas estadísticas relacionadas.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("psych")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	2.3.9
Depends	
Imports	mnormt, parallel, stats, graphics, grDevices, methods, lattice, nlme
Suggests	psychTools, GPArotation, lavaan, lme4, Rcsdp, graph, knitr, Rgraphviz
Enhances	
Published	2023-09-26
Author	William Revelle
Maintainer	William Revelle <revelle@northwestern.edu>
BugReports	
License	GPL-2 | GPL-3
URL	https://personality-project.org/r/psych/
NeedsCompilation	no

8.3.1.7 *RColorBrewer*

El paquete *RColorBrewer* en *R* es una valiosa herramienta para la creación de paletas de colores atractivas y efectivas en gráficos y visualizaciones. Este paquete proporciona una amplia selección de paletas de colores predefinidas que son especialmente útiles para representar datos categóricos o cualitativos.

Instalación

Para instalar este paquete se debe iniciar *R* y ejecutar

```
install.packages("RColorBrewer")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *CRAN*.

Version	1.1-3
Depends	R ($\geq$ 2.0.0)
Imports	MASS, methods, stats, utils
Suggests	covr, ggplot2, knitr, logisticPCA, markdown, Matrix, testthat
Enhances	
Published	2022-04-03
Author	Erich Neuwirth
Maintainer	Erich Neuwirth <erich.neuwirth@univie.ac.at>
BugReports	
License	Apache License 2.0
URL	https://github.com/willtownes/glmpca
NeedsCompilation	no

8.3.2 *Paquetes de Bioconductor*

8.3.2.1 *GEOquery*

Este paquete establece un puente entre *GEO* y *Bioconductor* al proporcionar herramientas para leer archivos directamente desde *GEO* en función del número de acceso. La función *getGEO*(), cuando se le proporciona un identificador *GSE*[1] como su único argumento, se encarga de descargar los archivos de matriz asociados a esa serie de *GEO* y los devuelve en forma de una lista de objetos *ExpressionSet*[2]. En la mayoría de situaciones, una serie de *GEO* corresponderá a un solo experimento.

Instalación

Para instalar este paquete se debe iniciar *R* en una versión 4.3 o superior y ejecutar

```
BiocManager::install("GEOquery")
```

Documentación

Para ver la documentación de la versión de este paquete instalada en el sistema, se inicia *R* y se ejecuta:

```
browseVignettes("GEOquery")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *Bioconductor*. Se explicarán cada una de las secciones que aquí aparecen y no han sido explicadas para los paquetes de *CRAN*, con el fin de quedar explicado para este paquete de *Bioconductor*, y así, el entendimiento de otros paquetes sea aún más sencillo.

1 *GEO Series* (listas de archivos *GSM* (*GEO Sample*) que juntos forman un solo experimento).
2 Un *ExpressionSet* es un objeto de datos utilizado en bioinformática y análisis de datos genómicos. Está especialmente diseñado para contener información de expresión génica, en especial datos de *microarrays*, junto con metadatos relacionados.

biocViews	DataImport, Microarray, OneChannel, SAGE, Software, TwoChannel
Version	2.70.0
In Bioconductor since	BioC 1.8 (R-2.3) (17.5 years)
License	MIT
Depends	ethods, Biobase
Imports	readr ($\geq$ 1.3.1), xml2, dplyr, data.table, tidyr, magrittr, limma, curl, R.utils
LinkingTo	
Suggests	knitr, rmarkdown, BiocGenerics, testthat, covr, markdown
SystemRequirements	
Enhances	
URL	https://github.com/seandavi/GEOquery
Depends On Me	DrugVsDisease, dyebiasexamples, GEOmetadb, GSE103322, GSE13015, GSE62944, SCAN.UPC
Imports Me	BeadArrayUseCases, bigmelon, BioPlex, ChIPXpress, crossmeta, DExMA, EGAD, GEOexplorer, GSE13015, healthyControlsPresenceChecker, minfi, Moonlight2R, MoonlightR, phantasus, recount, SRAdb
Suggests Me	airway, antiProfilesData, AUCell, autonomics, CBNplot, COTAN, ctsGE, dearseq, debCAM, diffcoexp, dyebias, EpiDISH, EpiMix, fgsea, FLAMES, GCSscore, GeneExpressionSignature, GenomicOZone, GeoTcgaData, methylclock, multiClust, MultiDataSet, muscData, omicsPrint, parathyroidSE, PCAtools, phantasusLite, prostateCancerCamcap, prostateCancerGrasso, prostateCancerStockholm, prostateCancerTaylor, prostateCancerVarambally, quantiseqr, RegEnrich, RegParallel, RGSEA, Rnits, runibic, skewr, spatialHeatmap, TargetScore, zFPKM
Links To Me	
Build Report	

A continuación, se procede con la explicación de cada sección no explicada anteriormente para los paquetes de *CRAN*.

- **biocViews**: Esta sección describe las categorías o vistas de *Bioconductor* a las que pertenece el paquete. Estas vistas son etiquetas que ayudan a los usuarios a entender el propósito y la utilidad del paquete.
- **Linking To**: Lista las bibliotecas o los recursos externos de los que el paquete depende para su funcionamiento.
- **System Requirements**: Requisitos de hardware o sistema operativo específicos requeridos por el paqute para su funcionamiento.
- **Depends On Me**: Esta sección enumera otros paquetes que dependen del paquete actual para su funcionamiento.
- **Imports Me**: Esta sección enumera los paquetes que importa el paquete actual. Los paquetes que importan otro paquete suelen utilizar sus funciones o datos en su propio código.
- **Suggests Me**: Esta sección enumera los paquetes que sugieren el uso del paquete actual. Los paquetes que sugieren otro paquete lo hacen porque pueden ser útiles en combinación con él, pero no son necesarios para su funcionamiento básico.
- **Links To Me**: Esta sección enumera otros paquetes o recursos que están relacionados con el paquete actual. A menudo, se utiliza para indicar vínculos o relaciones con otros paquetes o recursos en el ecosistema de *Bioconductor*.
- **Build Report**: Este campo puede proporcionar un enlace o información relacionada con un informe de compilación o construcción del paquete. Es útil para los desarrolladores o usuarios que deseen obtener más detalles técnicos sobre la construcción y compilación del paquete.

8.3.2.2 *DESeq2*

Se encarga de la estimación de la dependencia entre la varianza y la media en datos de recuento de ensayos de secuenciación de alto rendimiento y también se hace cargo de obtener la expresión diferencial basada en un modelo que utiliza la distribución binomial negativa.

Instalación

Para instalar este paquete se debe iniciar *R* en una versión 4.3 o superior y ejecutar

```
BiocManager::install("DESeq2")
```

Documentación

Para ver la documentación de la versión de este paquete instalada en el sistema, se inicia *R* y se ejecuta:

```
browseVignettes("DESeq2")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *Bioconductor*.

biocViews	Bayesian, ChIPSeq, Clustering, DifferentialExpression, GeneExpression, ImmunoOncology, Normalization, PrincipalComponent, RNASeq, Regression, Sequencing, Software, Transcription
Version	1.40.2
In Bioconductor since	BioC 2.12 (R-3.0) (10.5 years)
License	LGPL ($\geq$ 3)
Depends	S4Vectors ($\geq$ 0.23.18), IRanges, GenomicRanges, SummarizedExperiment($\geq$ 1.1.6)
Imports	BiocGenerics ($\geq$ 0.7.5), Biobase, BiocParallel, matrixStats, methods, stats4, locfit, ggplot2, Rcpp ($\geq$ 0.11.0)
LinkingTo	Rcpp, RcppArmadillo
Suggests	testthat, knitr, rmarkdown, vsn, pheatmap, RColorBrewer, apeglm, ashr, tximport, tximeta, tximportData, readr, pbapply, airway, pasilla($\geq$ 0.2.10), glmGamPoi, BiocManager

SystemRequirements	
Enhances	
URL	https://github.com/mikelove/DESeq2
Depends On Me	DEWSeq, DEXSeq, metaseqR2, octad, rgsepd, rnaseqDTU, rnaseqGene, SeqGSEA, TCC, tRanslatome
Imports Me	Anaquin, animalcules, anota2seq, APAlyzer, benchdamic, BloodCancerMultiOmics2017, BRGenomics, CeTF, circRNAprofiler, consensusDE, coseq, countsimQC, DaMiRseq, debrowser, DEFormats, DEGreport, DELocal, deltaCaptureC, DEsubs, DiffBind, easier, EBSEA, eegc, ERSSA, exomePeak2, ExpHunterSuite, FieldEffectCrc, GDCRNATools, GeneTonic, Glimma, GRaNIE, hermes, HTSFilter, icetea, ideal, IHWpaper, INSPEcT, IntEREst, isomiRs, kissDE, magpie, microbiomeExplorer, microbiomeMarker, MLSeq, multiSight, muscat, NBAMSeq, NetActivity, ORFik, OUTRIDER, pairedGSEA, PathoStat, pcaExplorer, phantasus, POMA, proActiv, recountWorkflow, RegEnrich, regionReport, ReportingTools, RiboDiPA, Rmmquant, scBFA, scGPS, SEtools, singleCellTK, SNPhood, srnadiff, systemPipeTools, TBSignatureProfiler, TEKRABber, UMI4Cats, vidger, vulcan
Suggests Me	aggregateBioVar, apeglm, bambu, biobroom, BiocGenerics, BioCor, BiocSet, BioNERO, CAGEr, CAGEWorkflow, CBNplot, compcodeR, curatedAdipoChIP, curatedAdipoRNA, dearseq, derfinder, dittoSeq, EDASeq, EnhancedVolcano, EnrichmentBrowser, EWCE, fishpond, fluentGenomics, gage, GenomicAlignments, GenomicRanges, GeoTcgaData, glmGamPoi, GSE62944, HiCDCPlus, IHW, InteractiveComplexHeatmap, miRmine, NxtIRFcore, OPWeight, PCAtools, phyloseq, progeny, PROPER, recount, RegParallel, RUVSeq, scran, seqpac, Single.mTEC.Transcriptomes, sparrow, spatialHeatmap, SpliceWiz, subSeq, SummarizedBenchmark, systemPipeR, systemPipeShiny, TFEA.ChIP, tidybulk, topconfects, tximeta, tximport, variancePartition, Wrench, zinbwave
Links To Me	
Build Report	

8.3.2.3 *genefilter*

Algunas funciones básicas para filtrar datos de genes.

Instalación

Para instalar este paquete se debe iniciar *R* en una versión 4.3 o superior y ejecutar

```
BiocManager::install("genefilter")
```

Documentación

Para ver la documentación de la versión de este paquete instalada en el sistema, se inicia *R* y se ejecuta:

```
browseVignettes("genefilter")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *Bioconductor*.

biocViews	Microarray, Software
Version	1.82.1
In Bioconductor since	BioC 1.6 (R-2.1) or earlier (> 18.5 years)
License	Artistic-2.0
Depends	
Imports	MatrixGenerics(≥ 1.11.1), AnnotationDbi, annotate, Biobase, graphics, methods, stats, survival, grDevices
LinkingTo	
Suggests	class, hgu95av2.db, tkWidgets, ALL, ROC, RColorBrewer, BiocStyle, knitr
Enhances	
URL	https://github.com/seandavi/GEOquery
Depends On Me	cellHTS2, CNTools, GeneMeta, Hiiragi2013, maEndToEnd, rnaseqGene, SISPA, sva

Imports Me	a4Base, annmap, arrayQualityMetrics, Category, cbaf, ClassifyR, countsimQC, covRNA, DEXSeq, ENmix, FlowSorted.Blood.EPIC, GISPA, GSRI, IHWpaper, metaseqR2, methylCC, methylumi, minfi, MLInterfaces, mogsa, NBAMSeq, NxtIRFcore, pcaExplorer, PECA, phenoTest, protGear, pwrEWAS, Ringo, RNAinteractMAPK, SGCP, spatialHeatmap, SpliceWiz, tilingArray, XDE, zinbwave
Suggests Me	annotate, BioNet, BloodCancerMultiOmics2017, categoryCompare, clusterStab, codelink, cola, compcodeR, curatedBladderData, curatedCRCData, curatedOvarianData, DelayedArray, EnrichedHeatmap, factDesign, ffpe, ffpeExampleData, gageData, GenomicFiles, GOstats, GSAR, GSEAlm, GSVA, logicFS, lumi, MAQCsubset, MMUPHin, npGSEA, oligo, phyloseq, pvac, qpgraph, RforProteomics, rheumaticConditionWOLLBOLD, rtracklayer, siggenes, simplifyEnrichment, Single.mTEC.Transcriptomes, TCGAbiolinks, topGO
Links To Me	
Build Report	

8.3.2.4 *tximeta*

Este paquete se encarga de realizar tareas de anotación y recopilación de metadatos durante la importación de la cuantificación de los transcriptomas de *Salmon*[3] a *R/Bioconductor*. Los metadatos se añaden a los datos automáticamente, facilitando así el proceso de análisis genómico y ayudando a la reproducibilidad del experimento.

Instalación

Para instalar este paquete se debe iniciar *R* en una versión 4.3 o superior y ejecutar

```
BiocManager::install("tximeta")
```

Documentación

Para ver la documentación de la versión de este paquete instalada en el sistema, se inicia *R* y se ejecuta:

```
browseVignettes("tximeta")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *Bioconductor*.

biocViews	Annotation, DataImport, FunctionalGenomics, GeneExpression, GenomeAnnotation, ImmunoOncology, Preprocessing, RNASeq, ReportWriting, ReproducibleResearch, SingleCell, Software, Transcription, Transcriptomics
Version	1.18.3
In Bioconductor since	BioC 3.8 (R-3.5) (5 years)
License	GPL-2
Depends	

3 *Salmon* es una herramienta utilizada para cuantificar la expresión de transcritomas a partir de datos *RNA*-seq. Emplea algoritmos complejos y novedosos para proporcionar estimaciones precisas de la expresión de forma muy rápida y utilizando poca memoria. Realiza esta inferencia teniendo en cuenta atributos experimentales y sesgos comúnmente observados en datos *RNA*-seq reales previamente observados.

Imports	SummarizedExperiment, tximport, jsonlite, S4Vectors, IRanges, GenomicRanges, AnnotationDbi, GenomicFeatures, ensembldb, BiocFileCache, AnnotationHub, Biostrings, tibble, GenomeInfoDb, tools, utils, methods, Matrix
LinkingTo	
Suggests	knitr, rmarkdown, testthat, tximportData, org.Dm.eg.db, DESeq2, fishpond, edgeR, limma, devtools
SystemRequirements	
Enhances	
URL	https://github.com/mikelove/tximeta
Depends On Me	rnaseqGene
Imports Me	IsoformSwitchAnalyzeR
Suggests Me	DESeq2, fishpond, fluentGenomics
Links To Me	
Build Report	

8.3.2.5 *airway*

Este paquete contiene un objeto *RangedSummarizedExperiment* de recuentos de lecturas de genes sobre un experimento de *RNA*-Seq en cuatro líneas celulares del músculo liso bronquial que se trataron con dexametasona.

Instalación

Para instalar este paquete se debe iniciar *R* en una versión 4.3 o superior y ejecutar

```
BiocManager::install("airway")
```

Documentación

Para ver la documentación de la versión de este paquete instalada en el sistema, se inicia *R* y se ejecuta:

```
browseVignettes("airway")
```

Detalles

En esta sección, se muestran los detalles sobre este paquete de *Bioconductor*.

biocViews	ExperimentData, GEO, RNASeqData, SequencingData
Version	1.20.0
License	LGPL
Depends	R ($\geq$ 3.5.0), SummarizedExperiment
Imports	
LinkingTo	
Suggests	knitr, GEOquery, markdown
SystemRequirements	
Depends On Me	rnaseqGene
Imports Me	consensusDE, NetActivity
Suggests Me	apeglm, awst, biobroom, BioCor, BiocSet, CeTF, DelayedArray, DESeq2, EnhancedVolcano, EnrichmentBrowser, ExperimentSubset, ideal, IHW, IHWpaper, InteractiveComplexHeatmap, iSEEu, OPWeight, pathwayPCA, pcaExplorer, PCAtools, progeny, RegParallel, runibic, SummarizedExperiment, TTMap, weitrix
Links To Me	

Parte IV

APLICACIÓN

Realización de un análisis exploratorio a nivel de genes y expresión diferencial utilizando la técnica de *RNA*-Seq

9

ANÁLISIS MULTIVARIANTE APLICADO AL TRATAMIENTO DE DATOS GENÓMICOS MEDIANTE RNA-SEQ

A lo largo de este capítulo se realizará un análisis de expresión diferencial de *RNA*-Seq a nivel de gen utilizando paquetes de *Bioconductor*. Este análisis se ha basado en el ya realizado en el artículo [43]. Se comenzará a partir de unos archivos *FASTQ*, se mostrarán cómo se cuantificaron los transcritos utilizados y se prepararán conjuntos de datos a nivel de genes para el análisis posterior. Se realizará un análisis exploratorio de datos para evaluar la calidad y explorar la relación entre las muestras, se llevará a cabo un análisis de expresión génica diferencial y se explorarán visualmente los resultados.

En caso de querer replicar este análisis es recomendable escribir el código manualmente, copiar código directamente de este documento puede dar lugar a errores.

9.1 DATOS UTILIZADOS

Los datos utilizados se almacenan en el paquete *airway* [44] que resume un experimento de *RNA*-Seq en el que las células del músculo liso bronquial se trataron con dexametasona[1]. Los glucocorticoides son utilizados, por ejemplo, por las personas con asma para reducir la inflamación de las vías respiratorias.

En el experimento, cuatro líneas celulares[2] humanas del músculo liso bronquial fueron tratadas con dexametasona durante 18 horas. Para cada una de las cuatro líneas celulares, se tiene una muestra tratada y otra sin tratar.

1 La dexametasona es un glucocorticoide, es decir, es similar a una hormona natural producida por las glándulas suprarrenales, con efectos antiinflamatorios. Por lo general, se usa para reemplazar este producto químico cuando su cuerpo no fabrica suficiente.

2 Entiéndase como línea celular, en este caso, a una población de células derivadas del músculo liso bronquial.

9.2 CUANTIFICACIÓN DE LOS DATOS DE ENTRADA PARA DESEQ2

Como entrada, los métodos estadísticos basados en recuentos, como *DESeq2*[3], esperan datos de entrada, por ejemplo, de *RNA*-Seq u otro experimento de secuenciación de alto rendimiento, en forma de matriz de recuentos no normalizada. El valor de la i-ésima fila y la j-ésima columna de la matriz indica cuántas lecturas se pueden asignar al gen i en la muestra j.

Los valores de la matriz deben ser recuentos o recuentos estimados de lecturas de secuenciación. Esto es importante para el modelo estadístico de *DESeq2*, ya que sólo los recuentos permiten evaluar correctamente la precisión de la medición. Es importante no proporcionar nunca recuentos que hayan sido prenormalizados para la profundidad de secuenciación[4], ya que el modelo estadístico es más potente cuando se aplica a recuentos no normalizados, y está diseñado para tener en cuenta internamente las diferencias de tamaño.

Para realizar esta cuantificación se deben alinear las lecturas con el genoma y, a continuación, contar el número de lecturas que coinciden con los modelos de genes. Un método –previemente introducido– de cuantificación de transcriptomas popular e increíblemente rápido para la alineación del genoma y el recuento de lecturas es *Salmon*[5], que realiza el mapeo o la alineación de lecturas con transcriptomas de referencia, generando recuentos estimados por transcriptoma. Después de ejecutar una de estas herramientas, se suele utilizar el paquete *tximeta* [46] para ensamblar matrices estimadas de recuento para su posterior uso con paquetes de expresión génica diferencial de *Bioconductor*.

3 Como se mencionó en el anterior capítulo, es un método estadístico basado en recuentos que estima la dependencia entre la varianza y la media en los datos de recuento de ensayos de secuenciación de alto rendimiento y comprueba la expresión diferencial basándose en un modelo que utiliza la distribución binomial negativa.

4 No es más que el número total de lecturas o fragmentos de secuenciación generados para una muestra determinada en un experimento de secuenciación de alto rendimiento

5 Este suele ser un proceso costoso y tedioso, el paquete *airway* contiene dos directorios de cuantificación generados por *Salmon*, así que este paso no será mostrado. Una guía sobre cómo utilizar *Salmon* se puede encontrar en [45]. Destacar que *Salmon* utiliza archivos *.fastq*, mencionados previamente en el trabajo.

9.2.1 *Lectura de los datos con tximeta*

En primer lugar, se debe cargar el paquete del experimento [44]. Destacar que este proceso se realizará únicamente para las dos muestras –caso y control– de una única línea celular, debido a la alta carga computacional de este proceso, de modo que este proceso quede plasmado para información y conocimiento del lector. El resultado de este proceso para las cuatro líneas celulares ya se encuentra dentro del paquete, y será cargado posteriormente.

```
library(airway)
```

La función *system.file* en *R* se suele utilizar para averiguar en qué parte del ordenador se han instalado los archivos de un paquete. Aquí se pide la ruta completa al directorio *extdata*, donde los paquetes de *R* almacenan datos externos. En este directorio, se encuentran una serie de archivos. Aparecen dos directorios que se encuentran en el directorio *quants*, que contienen dos archivos, que son la salida de la cuantificación hecha con *Salmon*.

```
dir <- system.file("extdata", package="airway", mustWork=TRUE)
list.files(file.path(dir, "quants"))
```

Output:

```
## [1] "SRR1039508" "SRR1039509"
```

Normalmente, se tiene una tabla con información detallada para cada una de las muestras que enlaza las muestras con los directorios *FASTQ* y *Salmon* asociados. Se carga dicho archivo *CSV* con *read.csv*.

```
csvfile <- file.path(dir, "sample_table.csv")
coldata <- read.csv(csvfile, row.names=1, stringsAsFactors=FALSE)
```

Para terminar la carga de datos de cuantificación de *Salmon* en *R*, se trabaja con las dos muestras del paquete *airway* que se acaba de mencionar. Son de interés las dos primeras filas, que son los dos archivos generados por *Salmon*. Se renombra la columna *Run* a una columna llamada *names* –para ser más descriptivos– y se crea una columna llamada *archivos*, para llevar control de versiones.

```
coldata <- coldata[1:2,]
coldata$names <- coldata$Run
coldata$files <- file.path(dir, "quants", coldata$names, "quant.sf.gz")
```

Ahora ya es momento de cargar *tximeta* y ejecutar su función principal, ya descrita en el capítulo anterior.

```
library(tximeta)
se <- tximeta(coldata)
```

Output:

```
importing quantifications
reading in files with read_tsv
1 2
found matching transcriptome:
[ GENCODE - Homo sapiens - release 29 ]
loading existing TxDb created: 2023-10-13 11:23:39
Loading required package: GenomicFeatures
Loading required package: AnnotationDbi

Attaching package: 'AnnotationDbi'

The following object is masked from 'package:dplyr':

    select

loading existing transcript ranges created: 2023-10-13 11:23:40
```

Si la comprobación con el transcriptoma de referencia fue exitosa y reconocida por *tximeta*, y si se tiene una conexión a Internet que funcione, *tximeta* localizará y descargará los datos de anotación –metadatos– relevantes de varias fuentes. Los datos de anotación solo se descargan y analizan una vez. Se debe tener en cuenta que *tximeta* importa datos a nivel de transcriptoma.

Como este trabajo trata del análisis a nivel de gen, se procede a transformar las cuantificaciones del nivel de transcriptoma al nivel de gen. La tabla que se encarga de la asignación transcriptoma-gen se crea automáticamente utilizando los metadatos descargados y almacenados en el objeto ***se*** –haciendo referencia a *SummarizedExperiment*–. Esto se hace con la función *summarizeToGene()* y el resultado se guarda en un objeto llamado ***gse*** –haciendo referencia a *gene SummarizedExperiment*–.

```
gse <- summarizeToGene(se)
```

Output:

```
loading existing TxDb created: 2023-10-13 11:23:39
obtaining transcript-to-gene mapping from database
loading existing gene ranges created: 2023-10-13 11:24:15
summarizing abundance
summarizing counts
summarizing length
```

A continuación se describe la clase del objeto creado por *tximeta* que se guardó anteriormente como ***se*** y ***gse***, y se ilustra cómo crear un objeto *DESeqDataSet* a partir de él para utilizarlo con *DESeq2*.

9.2.2 *SummarizedExperiment*

Se mantienen los nombres en inglés durante esta sección. La traducción al español suena rara, además, vagamente se usa incluso entre los hispanohablantes, así que no existe motivo alguno para traducir.

Este objeto es una estructura de datos utilizada para trabajar con datos *RNA-Seq*. Su objetivo principal es proporcionar una forma eficiente y organizada de almacenar datos experimentales, como la expresión génica o la información de secuenciación, junto con metadatos relevantes. Observando la Figura 12, el bloque *assay* –bloque rosa– contiene la matriz de recuentos, el bloque *rowRanges* –bloque azul– contiene información sobre los genes y el bloque *colData* –bloque verde– contiene información sobre las muestras. La línea resaltada en un color más oscuro en cada bloque representa la primera fila de cada objeto. Cabe destacar que la primera fila de *rowRanges* y la primera fila de *assay* se alinean.

Lo que se ha hecho con *tximeta* es crear un objeto ***gse*** de *SummarizedExperiment*. De esta forma queda explicado cómo realizar este proceso. En este caso, se utiliza el objeto ***gse*** que ya se encuentra procesado dentro del paquete *airway* y contiene las 4 líneas celulares, y no solo una. Para ello se ejecuta

```
data(gse)
```

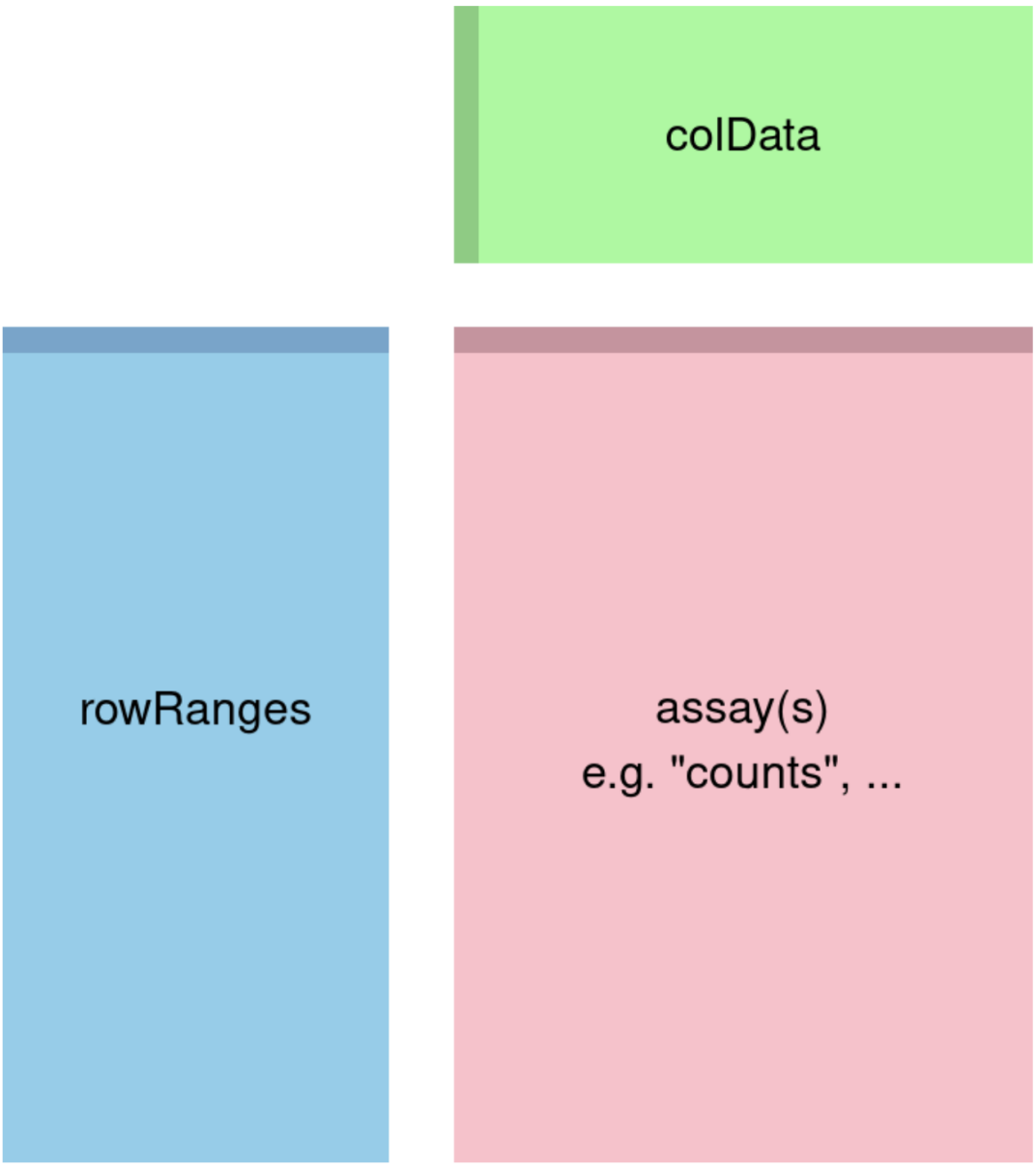

Figura 12: Componentes de un objeto *SummarizedExperiment*, extraída de [43]

Los nombres de los *assays* pueden examinarse con *assayNames*, y los propios *assays* se almacenan como una lista de matrices. Este objeto ***gse***, contiene tres *assays*:

- *counts* (recuentos): los recuentos estimados para cada gen y muestra.
- *abundance* (abundancia): las abundancias de los transcriptomas estimadas en *TPM*[6].
- *length* (longitud): las longitudes efectivas de los genes.

6 *Transcripts Per Million*.

Para acceder a los distintos componentes del objeto *SummarizedExperiment* se utilizan funciones homónimas de *R*: *assay* (o *assays*), *rowRanges* y *colData*.

Se comprueba que los *assays* de nuestro objeto ***gse*** son los que se han mencionado.

```
assayNames(gse)
```

Output:

```
[1] "counts"    "abundance" "length"
```

Se muestra qué contienen estos *assays*.

```
head(assay(gse), 3)
```

Output:

```
                  SRR1039508 SRR1039509 SRR1039512 SRR1039513
ENSG00000000003.14    708.164    467.962    900.992    424.368
ENSG00000000005.5       0.000      0.000      0.000      0.000
ENSG00000000419.12    455.000    510.000    604.000    352.000
                  SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003.14   1188.295   1090.668    805.929    599.337
ENSG00000000005.5       0.000      0.000      0.000      0.000
ENSG00000000419.12    583.000    773.999    409.999    499.000
```

El bloque *rowRanges*, cuando se le hace una llamada, muestra los rangos de los cinco primeros y los cinco últimos genes –no se muestra su salida debido a su gran tamaño, queda desvirtuada al incluirlo en el trabajo–. Este bloque también contiene metadatos sobre las secuencias –en este experimento son cromosomas–, que son accedidos mediante *seqinfo()*.

El bloque *colData* para nuestro objeto *SummarizedExperiment* refleja el *data.frame* que fue proporcionado a la función *tximeta* para importar los datos de cuantificación. Aquí se puede ver que hay columnas que indican los nombres de las muestras, así como el *ID* del donante –*donor*–, y la condición de tratamiento –tratado con dexametasona o no tratado–.

```
colData(gse)
```

Output:

```
DataFrame with 8 rows and 3 columns
                names       donor     condition
            <factor> <factor>      <factor>
SRR1039508 SRR1039508   N61311   Untreated
SRR1039509 SRR1039509   N61311   Dexamethasone
SRR1039512 SRR1039512   N052611  Untreated
SRR1039513 SRR1039513   N052611  Dexamethasone
SRR1039516 SRR1039516   N080611  Untreated
SRR1039517 SRR1039517   N080611  Dexamethasone
SRR1039520 SRR1039520   N061011  Untreated
SRR1039521 SRR1039521   N061011  Dexamethasone
```

Una vez se ha preparado y se conoce el objeto *SummarizedExperiment*, se está listo para comenzar el análisis. Cabe mencionar que, muchísimos analistas se saltan el paso de procesado –simplemente se encargan del proceso analítico en sí una vez los datos se encuentran ya procesados–, ya que es un procedimiento rudo y tedioso que queda a disposición de ingenieros de datos especializados.

9.3 OBJETO DESeqDATASET

Los paquetes de software de *Bioconductor* a menudo definen y utilizan una clase de objetos personalizada para el almacenamiento de datos que se asegura de que todas las especificaciones y características de los datos que son necesarias se proporcionan de forma coherente y cumplen los requisitos particulares de cada paquete. Además, *Bioconductor* tiene clases de datos generales –como *SummarizedExperiment*– que pueden usarse para mover datos entre paquetes, debido a que cumplen las especificaciones de muchos de estos.

En *DESeq2*, su clase personalizada se llama *DESeqDataSet*. Está construida sobre la clase *SummarizedExperiment*, y es fácil convertir objetos *SummarizedExperiment* en objetos *DESeqDataSet*. La principal diferencia es que se accede al bloque *assay* utilizando la función de acceso *counts*. A parte, la clase *DESeqDataSet* obliga a que los valores de esta matriz sean enteros no negativos.

En primer lugar, se ven los valores de las columnas de *colData*, primero para los donantes.

```
gse$donor
```

Output:

```
[1] N61311  N61311  No52611 No52611 No80611 No80611 No61011 No61011
Levels: No52611 No61011 No80611 N61311
```

Y ahora las condiciones de las muestras.

```
gse$condition
```

Output:

```
[1] Untreated     Dexamethasone Untreated     Dexamethasone Untreated
[6] Dexamethasone Untreated     Dexamethasone
Levels: Untreated Dexamethasone
```

Se observa que como condiciones no se tienen etiquetas muy descriptivas, se procede a cambiarlas. Es importante mencionar que cuando se renombran las etiquetas, se debe mantener el orden original, en este caso viene primero la condición *Untreated*, y luego la condición *Dexamethasone*.

```
levels(gse$condition) <- c("untreated", "treated")
```

Proceden a renombrarse como *untreated* y *treated*. Si se vuelve al comienzo del documento, es lo que se conoce también como caso y control.

La fórmula de diseño[7] más sencilla para la expresión diferencial sería *condition*, donde *condition* es una columna en *colData()* que especifica a cuál de dos grupos pertenecen las muestras. Para este análisis, se especifica la fórmula para la expresión diferencial como $\sim$ *donor* + *condition*, lo que significa que se quiere probar el efecto en cada línea celular de cada donante.

Una vez que se tiene al objeto *SummarizedExperiment* con todos sus metadatos, anotaciones y cambios correspondientes, se puede construir un objeto *DESeqDataSet* a partir de él, que será el punto de partida del análisis. Se añade la fórmula de diseño elegida para el análisis.

```
library(DESeq2)
dds <- DESeqDataSet(gse, design = ~ donor + condition)
```

Y ya se tendría el objeto *DESeqDataSet* llamado ***dds*** listo para comenzar el análisis.

7 Una fórmula de diseño de expresión diferencial es un plan estratégico que se utiliza para llevar a cabo un experimento de manera que se puedan identificar de manera fiable los genes cuya expresión cambia entre las condiciones de estudio especificadas en la fórmula.

9.4 ANÁLISIS EXPLORATORIO Y PCA

La matriz de recuentos del objeto *DESeqDataSet* contiene muchas filas con solo ceros, y, además, muchas filas con sólo unos pocos recuentos en total. Para reducir el tamaño del objeto, y para aumentar la velocidad de computación, se pueden eliminar las filas que no tienen ninguna o casi ninguna información sobre la cantidad de expresión génica. Aquí se aplica la regla de filtrado menos estricta: eliminar las filas del *DESeqDataSet* que no tienen recuentos, o solo un único recuento en todas las muestras. Cabe mencionar que para algunos *datasets*, podría tener sentido hacer un filtrado más estricto, aunque esto depende siempre de la naturaleza de los datos y del problema. Para este análisis, con el filtrado primero es suficiente.

Se observa que se pasa de

```
nrow(dds)
```

Output:

```
[1] 58294
```

a

```
keep <- rowSums(counts(dds)) > 1
dds <- dds[keep,]
nrow(dds)
```

Output:

```
[1] 31604
```

Que es una reducción de filas considerable cuando se habla de datos genómicos.

9.4.1 *Variance stabilizing transformation (VST) & Regularized-Logarithm Transformation (rlog)*

Muchos métodos estadísticos comunes para el análisis exploratorio de datos multidimensionales, como son el análisis cluster (*AC*) y el análisis de componentes principales (*PCA*), funcionan mejor para datos que generalmente tienen el mismo rango de varianza respecto a diferentes medias, dicho de otra forma, los datos tienen un rango de varianza similar en diferentes medias. Cuando la cantidad

esperada de varianza es aproximadamente la misma en diferentes medias, se dice que los datos son *homoscedásticos*.

Para los recuentos de *RNA*-Seq, sin embargo, la varianza esperada crece con la media. Por ejemplo, si se realiza un *PCA* directamente sobre una matriz de recuentos, el gráfico resultante suele depender principalmente de los genes con recuentos más altos porque muestran las mayores diferencias absolutas entre las muestras. Una estrategia sencilla, y a menudo utilizada para evitar esto, es tomar el logaritmo de los valores de recuento normalizados más un pseudoconteo de 1 –así se evita el valor 0–; sin embargo, dependiendo de la elección del pseudoconteo, los genes con los recuentos más bajos contribuirán con una gran cantidad de ruido al gráfico resultante, porque tomar el logaritmo de los recuentos pequeños en realidad infla su varianza.

Como solución a esto, *DESeq2* ofrece dos transformaciones para datos de recuento que estabilizan la varianza a través de la media: la *variance stabilizing transformation* (*VST*) para datos binomiales negativos con una tendencia de dispersión de su media, implementada en la función *vst*, y la transformación logarítmica regularizada o *rlog*. Los datos transformados mediante *VST* o *rlog* se vuelven entonces aproximadamente homoscedásticos y pueden utilizarse directamente para calcular distancias entre muestras, hacer un *PCA*, o hacer un *AC*.

Llegados a este punto cabría hacerse la pregunta sobre qué transformación elegir. La *VST* es mucho más rápida de calcular y es menos sensible a los *outliers* que la *rlog*. El *rlog* tiende a funcionar bien en conjuntos de datos pequeños ($n \leq 30$). Por lo tanto, se recomienda utilizar el *VST* para conjuntos de datos medianos y grandes ($n > 30$); aunque lo ideal es realizar ambas transformaciones y comparar los resultados del estudio diferencial y del *PCA*.

Tanto *VST* como *rlog* devuelven un objeto *DESeqTransform* basado en la clase *SummarizedExperiment*. Los valores transformados ya no son recuentos y se almacenan en el bloque *assay*. El bloque *colData* que se encuentra en ***dds*** sigue siendo accesible desde el nuevo objeto.

Se procede a realizar las transformaciones. Primero la *VST*.

```
library(vsn)
vst_transf <- vst(dds, blind = FALSE)
head(assay(vst_transf), 3)
```

Output:

```
                   SRR1039508 SRR1039509 SRR1039512 SRR1039513
ENSG00000000003.14  10.105781   9.852029  10.169726   9.991545
ENSG00000000419.12   9.692244   9.923647   9.801921   9.798653
ENSG00000000457.13   9.449592   9.312186   9.362754   9.459168
                   SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003.14  10.424865  10.194490  10.315814  10.002177
ENSG00000000419.12   9.763455   9.874703   9.683211   9.845507
ENSG00000000457.13   9.281415   9.395937   9.477971   9.477027
```

y

```
colData(vst_transf)
```

Output:

```
## DataFrame with 8 rows and 5 columns
##                    names     donor condition
##                <factor> <factor>  <factor>
## SRR1039508 SRR1039508  N61311  untreated
## SRR1039509 SRR1039509  N61311  treated
## SRR1039512 SRR1039512  N052611 untreated
## SRR1039513 SRR1039513  N052611 treated
## SRR1039516 SRR1039516  N080611 untreated
## SRR1039517 SRR1039517  N080611 treated
## SRR1039520 SRR1039520  N061011 untreated
## SRR1039521 SRR1039521  N061011 treated
```

Ahora se realiza la transformación *rlog*.

```
rl_transf <- rlog(dds, blind = FALSE)
head(assay(rl_transf), 3)
```

Output:

```
                   SRR1039508 SRR1039509 SRR1039512 SRR1039513
ENSG00000000003.14   9.482613   9.172197   9.558383   9.346001
ENSG00000000419.12   8.860186   9.150196   9.000042   8.995902
ENSG00000000457.13   8.354790   8.166700   8.236582   8.366693
                   SRR1039516 SRR1039517 SRR1039520 SRR1039521
ENSG00000000003.14   9.851349   9.587602   9.727248   9.357876
ENSG00000000419.12   8.951327   9.091075   8.848782   9.054384
ENSG00000000457.13   8.121781   8.282307   8.392384   8.391023
```

En las llamadas anteriores, se ha especificado *blind = FALSE*, lo que significa que las diferencias entre las líneas celulares de donantes y el tratamiento no contribuirán a la tendencia varianza-media de la que se estaba hablando. El diseño experimental no se utiliza directamente en la transformación.

Para mostrar el efecto de las distintas transformaciones, se procede a comparar la primera muestra con la segunda, primero utilizando simplemente una tranformación dada por función *log2* –después de añadir un pseudoconteo de 1, para evitar tomar el valor de cero–, y después utilizando los valores transformados *VST* y *rlog*.

```
library(dplyr)
library(ggplot2)

dds <- estimateSizeFactors(dds)

df <- bind_rows(
  as_data_frame(log2(counts(dds, normalized=TRUE)[, 1:2]+1)) %>%
         mutate(transformation = "log2(x␣+␣1)"),
  as_data_frame(assay(vst_transf)[, 1:2]) %>% mutate(transformation = "
     vst"),
  as_data_frame(assay(rl_transf)[, 1:2]) %>% mutate(transformation = "
     rlog"))

colnames(df)[1:2] <- c("x", "y")

lvls <- c("log2(x␣+␣1)", "vst", "rlog")
df$transformation <- factor(df$transformation, levels=lvls)

ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) +
  coord_fixed() + facet_grid( . ~ transformation)
```

Según la Figura 13 se observa que, mientras el *rlog* tiene aproximadamente la misma escala que los datos normalizados en primera instancia, el *VST* tiene un desplazamiento hacia arriba para los valores más pequeños. Entonces, se deduce que serán las diferencias entre muestras las que contribuirán a los cálculos de distancia y al *PCA*.

Se puede ver cómo los genes con recuentos bajos parecen ser excesivamente variables en la escala logarítmica ordinaria *log2*, mientras que el *VST* y el *rlog* comprimen las diferencias para los genes con recuentos bajos y evitan excesivo ruido en los datos.

9.4.2 *Distancias entre muestras*

Un primer paso que suele resultar útil en un análisis de *RNA*-Seq suele ser evaluar la similitud general entre las muestras. Se utiliza la función *dist* de *R* para calcular la distancia euclídea entre muestras. Para asegurarse de que se tiene una

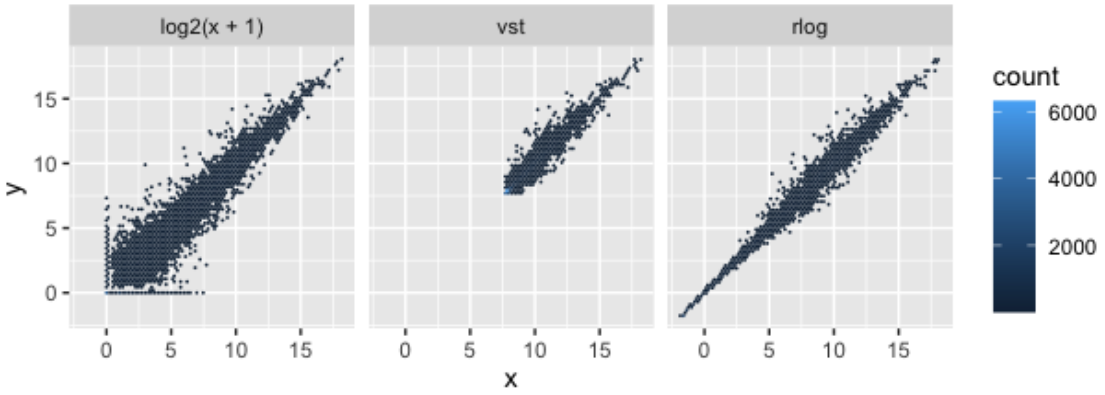

Figura 13: Comparativa gráfica de la dispersión de dos muestras utilizando datos normalizados sin transformar (*log2*), la transformación *VST* y la transformación *rlog*

contribución aproximadamente igual de todos los genes, se utiliza sobre los datos *VST*. Es necesario trasponer la matriz de valores usando *t*, porque la función *dist* espera que las diferentes muestras sean filas, y las diferentes dimensiones –en este estudio, y en el marco de aplicación de este trabajo, genes– sean columnas. Se utiliza un mapa de calor (*heatmap*) para visualizar estas distancias, utilizando el paquete *pheatmap* de *R*.

A parte, para poder mostrar la matriz de distancias entre muestras con las filas/columnas ordenadas por distancias, se debe escribir *sampleDists* al argumento *clustering_distance* de la función *pheatmap*. De lo contrario, la función *pheatmap* asumiría que la matriz contiene los valores de los datos en sí, y calcularía las distancias entre las filas/columnas de la matriz de distancias, que no es lo que se busca. También se especifica manualmente una paleta de colores azules utilizando la función *colorRampPalette* del paquete *RColorBrewer*. La Figura 14 en la página siguiente muestra la salida obtenida.

```
library(pheatmap)
library(RColorBrewer)

sampleDists <- dist(t(assay(vsd)))

sampleDistMatrix <- as.matrix( sampleDists )

rownames(sampleDistMatrix) <- paste( vsd$dex, vsd$cell , sep = " - " )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)

pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```

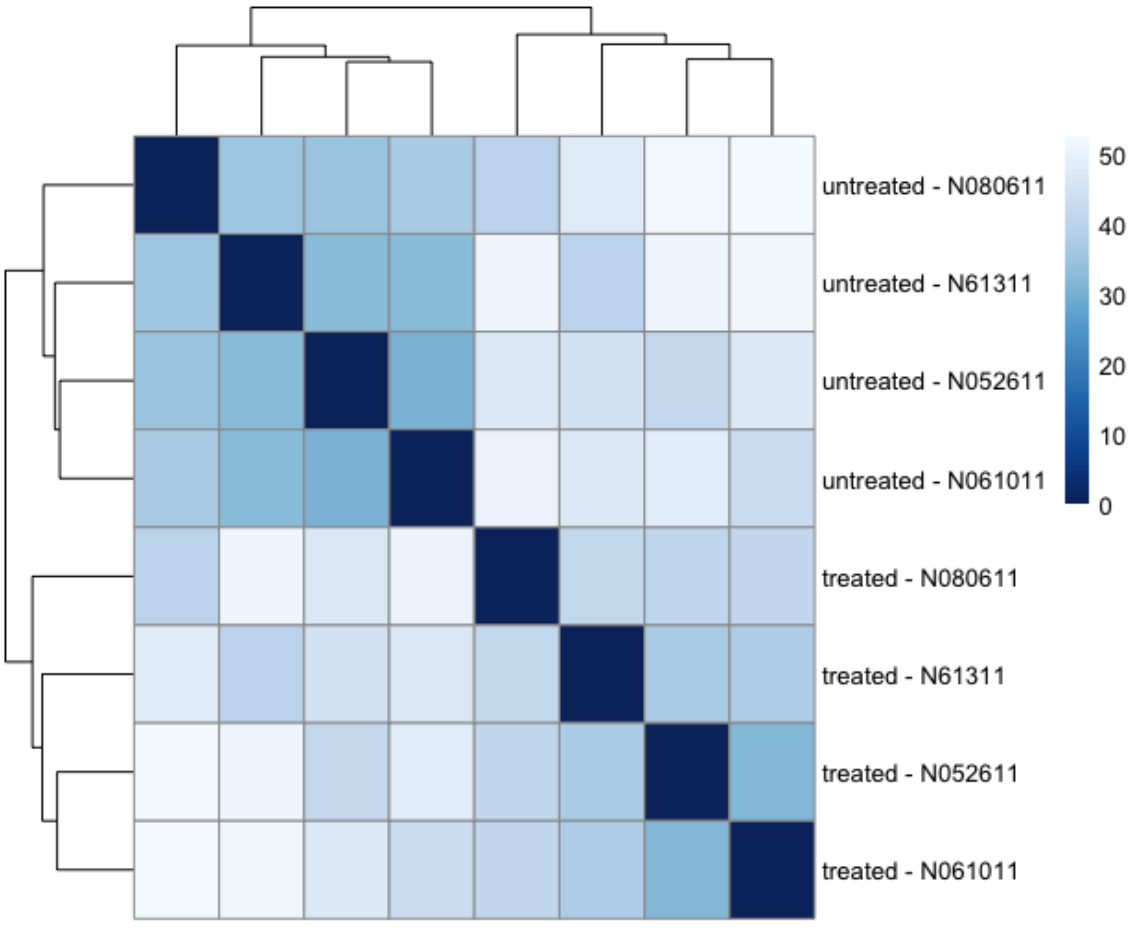

Figura 14: *Heatmap* de las distancias entre muestras utilizando los valores *VST*

Se debe mencionar que se han cambiado los nombres de las filas de la matriz de distancias para que contengan el tipo de tratamiento y el número de paciente en lugar del *ID* de la muestra, de modo que se tenga toda esta información a la vista al observar el gráfico.

9.4.3 *Análisis de Componentes Principales (PCA)*

Una de las mejores y más intuitivas maneras de visualizar las distancias entre muestras es el análisis de componentes principales (*PCA*). En este método, las muestras se proyectan en un plano 2D de forma que se extiendan en las dos direcciones que explican la mayoría de las diferencias –recordamos, que se refiere a las dos componentes principales con mayor varianza explicada.

Como se observa en la Figura 15, el eje x es la dirección que más separa a las muestras –mayor dispersión en la varianza, como se explicó en la parte de fundamentos matemáticos–. Los valores de las muestras en este eje se denominan, en este caso, PC_1 –primera componente principal–. El eje y es una dirección que debe ser ortogonal a la primera. Los valores de las muestras en este eje se denominan PC_2 –segunda componente principal–. El porcentaje de la varianza total que está contenida en cada componente principal (*PCs*) se imprime junto con su *label* o etiqueta en la Figura 15. Se observa que estos porcentajes no suman 100 %, esto es debido a que hay más *PCs* que contienen la varianza restante –aunque cada una de estas explicará menos varianza que la inmediatamente inferior–.

Aún así, estas dos componentes principales suman un total de un 73 % de varianza acumulada explicada, que es un resultado bastante bueno en este tipo de problemas. A continuación, se muestra cómo se crea este gráfico *PCA*.

```
percentVar <- round(100 * attr(pcaData, "percentVar"))
pcaData <- plotPCA(vst_transform, intgroup = c( "condition", "donor"),
   returnData = TRUE)
pcaData
```

Output:

```
                PC1        PC2             group condition    donor
SRR1039508 -14.311369 -2.6000421 untreated:N61311 untreated  N61311
SRR1039509   8.058574 -0.7500532   treated:N61311   treated  N61311
SRR1039512  -9.404122 -4.3920761 untreated:N052611 untreated N052611
SRR1039513  14.497842 -4.1323833   treated:N052611   treated N052611
SRR1039516 -12.365055 11.2109581 untreated:N080611 untreated N080611
SRR1039517   9.343946 14.9115160   treated:N080611   treated N080611
SRR1039520 -10.852633 -7.7618618 untreated:N061011 untreated N061011
SRR1039521  15.032816 -6.4860576   treated:N061011   treated N061011
                name
SRR1039508 SRR1039508
SRR1039509 SRR1039509
SRR1039512 SRR1039512
SRR1039513 SRR1039513
SRR1039516 SRR1039516
SRR1039517 SRR1039517
SRR1039520 SRR1039520
SRR1039521 SRR1039521
```

Esta salida ilustra las coordenadas del registro del gen en un sistema de coordenadas dado por las dos componentes principales –PC_1 y PC_2–.

Ahora ya se está en disposición de obtener el gráfico *PCA*.

```
ggplot(pcaData, aes(x = PC1, y = PC2, color = dex, shape = cell)) +
  geom_point(size =3) +
  xlab(pasteo("PC1: ", percentVar[1], "% variance")) +
  ylab(pasteo("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA with VST data")
```

A partir del gráfico *PCA* (ver Figura 15), se observa que las diferencias entre las células –las diferentes formas de trazado– son considerables, aunque no más fuertes que las diferencias debidas al tratamiento con dexametasona –color rojo frente a azul–. Esto demuestra por qué es importante tener este factor en cuenta en los análisis de expresión diferencial.

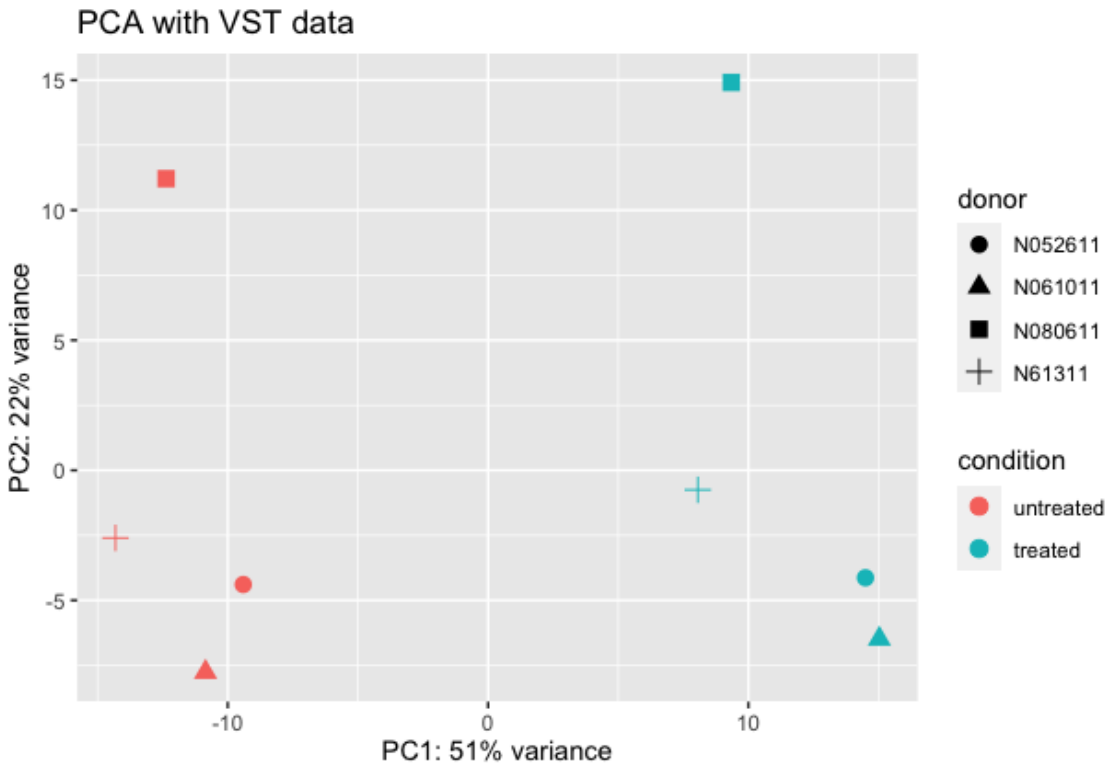

Figura 15: Gráfico *PCA* usando los valores transformados *VST* especificando la línea celular del donante (símbolo de trazado) y si ha seguido tratamiento o no (color)

9.5 ANÁLISIS DE EXPRESIÓN DIFERENCIAL

Como ya se realizó un diseño experimental cuando se creó el *DESeqDataSet*, se puede ejecutar el *pipeline*[8] de expresión diferencial en los datos sin procesar con tan solo una llamada a la función *DESeq*.

```
dds <- DESeq(dds)
```

Esta función imprimirá un mensaje con los distintos pasos que realiza. Brevemente, estos pasos son

- Estimación de los factores de tamaño, que controlan las diferencias en la profundidad de secuenciación de las muestras.
- Estimación de los valores de dispersión para cada gen.
- Ajuste de un modelo lineal generalizado.

Se devuelve un *DESeqDataSet* que contiene todos los parámetros ajustados. Ahora, el objetivo es extraer las tablas de resultados de interés de este objeto.

Si se llama a la función *results* sin ningún argumento, se extraerán del objeto los *p-values* estimados para la última variable de la fórmula de diseño –en este caso es conveniente recordar que era $\sim$ *donor* + *condition*–. Si hay más de 2 niveles para esta variable, *results* extraerá la tabla de resultados para una comparación del último nivel sobre el primero. Como solo se tienen dos, esto no es relevante.

Como *resultado* es un objeto *DataFrame*, lleva metadatos con información sobre el significado de las columnas.

```
resultado <- results(dds, contrast=c("condition","treated","untreated"))
mcols(resultado, use.names = TRUE)
```

Output:

```
DataFrame with 6 rows and 2 columns
                     type             description
                 <character>          <character>
baseMean       intermediate mean of normalized c..
log2FoldChange      results log2 fold change (ML..
lfcSE               results standard error: cond..
stat                results Wald statistic: cond..
pvalue              results Wald test p-value: c..
padj                results   BH adjusted p-values
```

8 Un *pipeline* en este contexto se refiere a una secuencia de pasos automatizados utilizados para el análisis de expresión génica diferencial.

La primera columna, *baseMean*, es simplemente la media de los valores de recuento normalizados, divididos por los factores de tamaño, tomados sobre todas las muestras del set *DESeqDataSet*. Las cuatro columnas restantes se refieren a la comparación del nivel *treated* sobre el nivel *untreated* para la variable *condition*.

La columna *log2FoldChange* dice cuánto parece haber cambiado la expresión del gen debido al tratamiento con dexametasona en comparación con las muestras no tratadas. Este valor se indica en una escala logarítmica de base 2. Por supuesto, esta estimación tiene una incertidumbre asociada, que está disponible en la columna *lfcSE*.

El propósito de una prueba de expresión diferencial es comprobar si los datos proporcionan pruebas suficientes para concluir si el efecto que se está midiendo en el experimento –en este caso, el efecto de tratar con dexametasona una enfermedad en particular– es un efecto real o sólo una fluctuación aleatoria. *DESeq2* realiza para cada gen una prueba de hipótesis para ver si la evidencia es suficiente para rechazar la hipótesis nula –esta afirma que hay un efecto nulo del tratamiento sobre el gen–. Lo que se quiere es descartar esta hipótesis nula, y rechazar la afirmación que la diferencia observada entre el tratamiento y el control es simplemente causada por la variabilidad experimental. Como es habitual en estadística, el resultado de esta prueba se presenta como un *p-value*[9], y se encuentra en la columna *p-value*.

También se pueden resumir los resultados con la siguiente línea de código, que proporciona alguna información adicional.

```
summary(resultado)
```

Output:

```
out of 31604 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 2373, 7.5%
LFC < 0 (down)     : 1949, 6.2%
outliers [1]       : 0, 0%
low counts [2]     : 14706, 47%
(mean count < 9)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

9 Un *p-value* se define como la probabilidad de que un valor estadístico calculado sea posible dada una hipótesis nula cierta.

Se observa que hay muchos genes con expresión diferencial debida al tratamiento con dexametasona al nivel *FDR*[10] aproximadamente del 10 % –esto es, *p-value* $<$ 0,1–. Esto tiene sentido, ya que es bien sabido que las células musculares lisas bronquiales reaccionan a los glucocorticoides. Sin embargo, hay otras maneras de ser más estricto sobre qué conjunto de genes se consideran significativos, por ejemplo, se puede bajar el umbral de la tasa *FDR* –el umbral que se encuentra en la columna *padj* en la tabla de resultados–. Si se baja el umbral de la tasa de falsos descubrimientos, también se debería informar de ello a la función *results*(), para que la función pueda utilizar este umbral para el filtrado que este realiza.

A veces, un subconjunto de los *p-values* en *resultado* será *NA*[11]. Esta es la forma que tiene *DESeq* de informar de que todos los recuentos para este gen eran cero y, por lo tanto, no se aplicó ninguna prueba. Además, a los *p-values* se les puede asignar *NA* si el gen fue excluido del análisis.

9.6 GENE CLUSTERING

En el *heatmap* de la Figura 14 realizado anteriormente, el dendrograma que aparece en el lateral izquierdo muestra un *clustering* jerárquico de las muestras. Pero se está realizando un análisis genético, así que es bueno saber que este tipo de *clustering* también puede realizarse para los genes. Dado que solo sería relevante realizar un *clustering* para los genes que realmente aportan información valiosa, normalmente solo se agrupa un subconjunto de los genes. En este caso, se seleccionarán los 20 genes con la mayor varianza entre muestras. Una vez más, se trabjará con los datos *VST*, debido a que dan mejor resultado.

```
library(genefilter)
topGen <- head(order(rowVars(assay(vst_transf)), decreasing = TRUE), 20)
```

Un *heatmap* se vuelve más informativo cuando no se enfoca en la intensidad de expresión absoluta[12], sino en la medida en que cada gen difiere en una muestra particular en comparación con la media de ese gen en todas las muestras.

10 False Discovery Rate, ayuda a los investigadores a evaluar la probabilidad de identificar erróneamente un resultado como significativo cuando, en realidad, no lo es.

11 *Not Available*, No Disponible en español.

12 Esto se refiere a la medida directa de cuánto se expresa un gen en una muestra sin considerar ninguna normalización o comparación con otras muestras.

Por lo tanto, se normalizan los valores de cada gen en todas las muestras y crea el *heatmap* de la Figura 16. Además, se proporciona un *data.frame* que le indica a la función *pheatmap* cómo etiquetar las columnas.

```
mat  <- assay(vsd)[ topGen, ]
mat  <- mat - rowMeans(mat)
anno <- as.data.frame(colData(vsd)[, c("donor","condition")])
pheatmap(mat, annotation_col = anno)
```

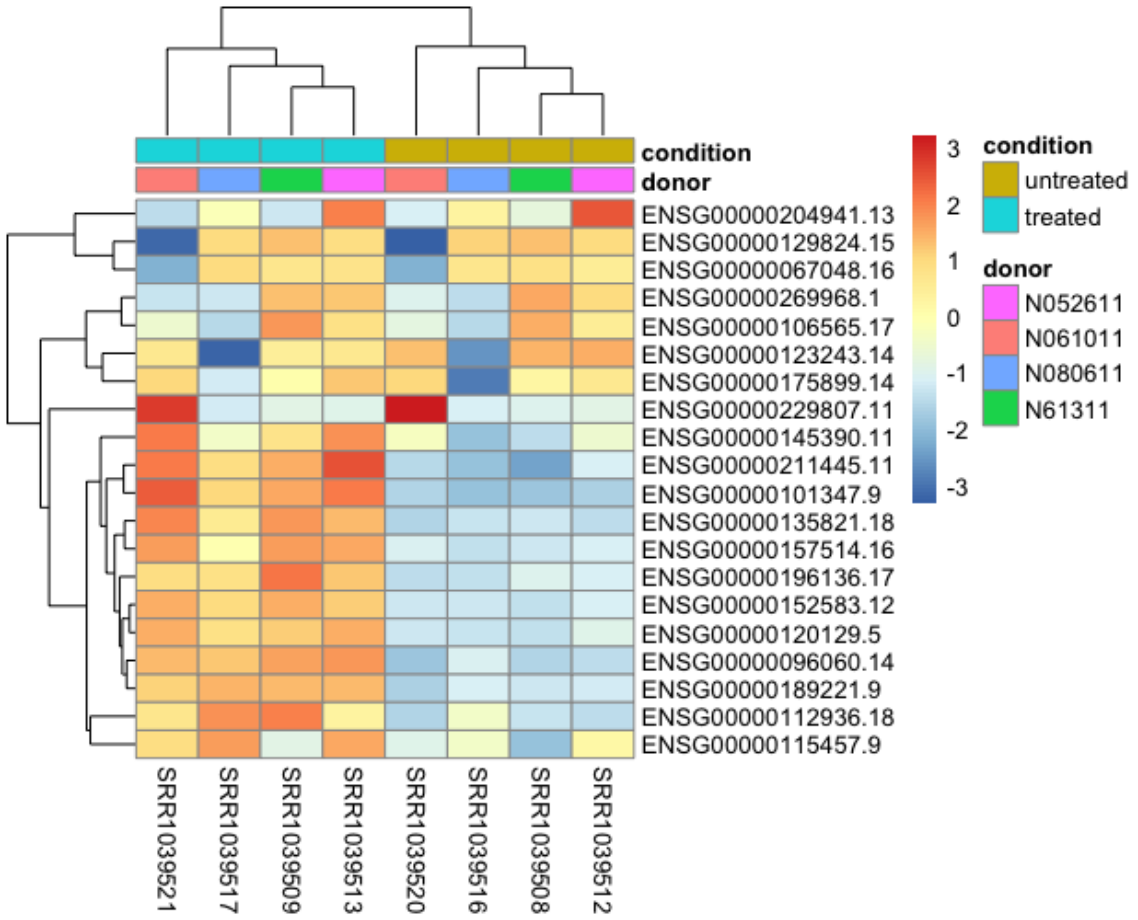

Figura 16: *Heatmap* de los valores relativos transformados *VST* entre las distintas muestras para los 20 genes con mayor varianza entre muestras

Es conveniente guiar al lector a través del mapa de calor con el fin de que sea capaz de entenderlo. En primer lugar, cada fila del *heatmap* representa un gen diferente. Estos genes son aquellos con la mayor varianza entre las muestras, lo que significa que son los genes que mostraron los cambios más significativos en su expresión entre las diferentes muestras y condiciones. Cada columna representa

una muestra diferente, las cuatro muestras de la izquierda representan aquellas que fueron tratadas con dexametasona, y las cuatro de la izquierda las muestras que se usaron como control.

Tanto arriba como a la izquierda de este *heatmap* se dibuja un dendrograma. Se recuerda que este es un gráfico de árbol jerárquico que funciona como representación visual de un *clustering* jerárquico. Cuanto más cerca estén dos genes o muestras en el dendrograma, más similares son en términos de expresión génica.

Se observa que para cada muestra y cada gen se le asocia un color, que va desde un azul intenso, desvaneciéndose hacia blanco y llegando a un color rojo intenso. Cuanto más cálido sea el color de la celda, mayor expresión génica se ha detectado de un gen para una muestra específica; y cuanto más frío, menor expresión génica.

Por último, el lector puede llegar a preguntarse, una vez tenemos esta información, cómo puede comenzar a interpretar el mapa de color. Como primera observación uno puede fijarse en áreas de colores cálidos para identificar genes que están altamente expresados en ciertas muestras y áreas de colores fríos para genes con baja expresión. Si, por ejemplo, se observa una fila que es predominantemente roja en las columnas que representan las células tratadas pero azul en las no tratadas, eso indica que el tratamiento aumentó la expresión de ese gen específico. Por otro lado, si se tiene una fila que es predominantemente azul en las células tratadas pero roja en las no tratadas, eso indica que el tratamiento disminuyó la expresión de ese gen.

Finalmente, si se aprecian patrones similares en las columnas que representan diferentes donantes, eso sugiere que el efecto es debido al tratamiento y no es específico de un donante en particular.

Ahora, ya para finalizar el análisis, se exponen las conclusiones que se deducen a partir de este mapa de calor. En la parte superior del mapa de calor, se utilizan barras de colores para representar el estado del tratamiento y la información relacionada con las líneas celulares de los donantes. Se observan claramente distintos clústeres de genes en la imagen. Es importante destacar que la línea celular del donante N061011 se diferencia claramente de las de los otros 3, ya sea tratado o no; así como existe otro grupo de genes en el que las muestras tratadas con dexametasona exhiben una expresión génica más elevada frente a las que no. A la hora de analizar las muestras, se encuentran dos clústeres bien diferenciados, que puede deberse a las condiciones de las extracciones de las muestras o a cualquier otro motivo que desconocemos.

10

CONCLUSIONES Y TRABAJO FUTURO

Este capítulo consta de dos secciones distintas. En la primera, se evalúan los objetivos iniciales del proyecto para determinar su grado de cumplimiento y éxito. Luego, se explorarán las oportunidades de trabajo futuro que queda abierto tras el desarrollo de este trabajo.

10.1 REPASO DE LOS OBJETIVOS DEL TRABAJO

LLegados a este punto del trabajo, es esencial destacar que, a lo largo de este proceso de investigación y análisis, se han explorado en profundidad los temas y cuestiones planteadas en esta investigación. Se han reunido datos, revisado literatura relevante, aplicado técnicas rigurosas y reflexionado sobre los resultados obtenidos. Las conclusiones que se presentan a continuación tienen el objetivo de discernir si los objetivos que fueron planteados en un primer momento han sido logrados y en qué medida.

En relación al (OBJ-1.), se ha logrado identificar patrones en los datos y elaborar predicciones basadas en modelos probabilísticos a lo largo de este trabajo. Este logro se ha evidenciado en varias secciones clave del documento. En la sección 5.2.4, se realizó un análisis *PCA* que ilustró de manera efectiva el marco teórico y ayudó a identificar patrones subyacentes en los datos. La sección 5.3.3 proporcionó un ejemplo completo de análisis clúster (*AC*) utilizando el método de *Ward*, lo que permitió identificar agrupaciones y patrones relevantes en los datos utilizados para el ejemplo. Además, en el Capítulo 9, se aplicaron con éxito estos enfoques en el análisis de datos genómicos mediante *RNA*-Seq, lo que resultó en la identificación de patrones genéticos y la elaboración de predicciones significativas.

Estos logros destacan la capacidad para aplicar métodos probabilísticos y modelos de datos de manera efectiva en la investigación genómica y subrayan la contribución valiosa de este trabajo a la comprensión y aplicación de modelos probabilísticos en el análisis de datos de alta dimensión.

Se puede concluir que el (OBJ-2.), que consistía en explorar y evaluar diversas técnicas de aprendizaje automático en el contexto de la clasificación de datos genómicos, ha sido alcanzado. En el Capítulo 5, se llevó a cabo un estudio exhaustivo de técnicas como la reducción de la dimensión y el clustering, proporcionando una base sólida para el análisis de datos genómicos. En el Capítulo 9, se aplicaron estas técnicas con éxito en la clasificación de datos genómicos, demostrando su eficacia en la resolución de problemas específicos en este campo. Este logro refleja la capacidad adquirida para aplicar de manera efectiva herramientas de aprendizaje automático en un contexto biológico y subraya la contribución significativa de este trabajo a la comprensión y análisis de datos genómicos.

En lo referente al (OBJ-3.), se ha logrado llevar a cabo un examen de las bases matemáticas relacionadas con las técnicas multivariantes, con especial hincapié en el análisis de componentes principales (*PCA*) y el análisis clúster (*AC*), abordando estos temas de manera integral en el Capítulo 5. Este análisis ha abarcado un amplio abanico de conceptos fundamentales en el ámbito de las matemáticas, la estadística y el procesamiento de datos multivariantes. Además, se ha profundizado en los aspectos teóricos que respaldan estas técnicas, permitiendo una comprensión sólida de su funcionamiento y aplicación en el contexto de este trabajo.

Por último, con relación al (OBJ-4.), se ha cumplido con éxito la finalidad de aprender el manejo de estructuras de datos genómicos, comprender la implementación de diversas técnicas multivariantes, y aplicarlas a un conjunto de datos específico. A lo largo de este trabajo, se abordó de manera integral la complejidad de las estructuras de datos genómicos, como se detalló en el Capítulo 4, resaltando los desafíos inherentes a su gestión. En el Capítulo 7, se exploraron las distintas estructuras de almacenamiento y las técnicas asociadas, incluyendo *microarrays*, *RNA*-Seq y *scRNA*-Seq, lo que proporcionó una base sólida para su aplicación posterior. En el Capítulo 6 se realtó la importancia del proyecto *Gene Expression Omnibus* como un repositorio público invaluable de datos genómicos, proporcionando un contexto clave para la investigación. En el Capítulo 8, se profundizó en *R* y *Bioconductor*, detallando las especificidades y paquetes utilizados en este contexto. Finalmente, en el Capítulo 9, se aplicaron de manera exitosa estos conocimientos a un conjunto de datos genómicos específicos, en este caso, células del músculo liso bronquial, llevando a cabo un análisis efectivo de datos genómicos.

Este logro es el cúlmen de este trabajo y plasma la capacidad para enfrentar desafíos técnicos y avanzar en la comprensión y aplicación de herramientas en el campo de la genómica.

10.2 TRABAJO FUTURO

El análisis de datos genómicos es un campo en constante evolución, y a medida que se avanza en la comprensión de los procesos biológicos y la genómica, se abren numerosas oportunidades para la investigación futura. A continuación, se presentan algunas posibles áreas de desarrollo que podrían ampliar y mejorar el trabajo realizado en este estudio.

La investigación futura podría enfocarse en el desarrollo de algoritmos de análisis más avanzados y específicos para datos genómicos, en particular, y para datos de alta dimensionalidad, en general. La estadística de alta dimenionalidad presenta problemas matemáticos y computacionales reales que no son nada fácil de solucionar. Esto podría incluir métodos de reducción de dimensionalidad, técnicas de *clustering* más sofisticadas o enfoques de aprendizaje automático adaptados a datos biológicos de alta dimensionalidad.

El análisis de datos de expresión génica a nivel de una sola célula, *scRNA*-Seq, que se mencionó anteriormente en el texto, es una área emergente y que puede ser de gran relevancia para comprender la heterogeneidad celular y los procesos biológicos a un nivel más detallado. Explorar técnicas de análisis multivariante para estos datos e invertir en ellla para reducir su coste tanto monetario como computacional es un desafío interesante al que ya se está enfrentando la comunidad científica.

Es de vital importancia promover la colaboración y el acceso a datos genómicos a través de plataformas como Gene Expression Omnibus. La implementación de políticas y tecnologías que faciliten la compartición y el acceso a datos entre investigadores conseguiría acelerar el avance en este campo.

Por último, a medida que la cantidad de datos genómicos aumenta exponencialmente, la optimización de las estrategias de gestión y almacenamiento de datos se vuelve crítica. La investigación focalizada en desarrollar nuevas técnicas de almacenamiento y recuperación de datos es un área importante para futuras investigaciones.

BIBLIOGRAFÍA

[1] O. A. Prabal Subedi, Simone Moertl, "Omics in radiation biology: Surprised but not disappointed," *Radiation*, vol. 2, pp. 124–129, 2022.

[2] A. M. Lesk, *Introduction to Bioinformatics*. Oford University Press, 5th ed., 2019.

[3] National Human Genome Research Institute (NHGRI). An official website of the United States government., "Proyecto genoma humano." https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost. Accessed: 2023-09-26.

[4] M. S. Zhong Wang, Mark Gerstein, "A global reference for human genetic variation," *The 1000 Genomes Project Consortium*, vol. 526, p. 68–74, 2015.

[5] Karl Pearson, "On lines and planes of closest fit to systems of points in space," 1901.

[6] H. Hotteling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24(6), p. 417–441, 1933.

[7] N. H. Timm, *Applied Multivariate Analysis*. Springer Texts in Statistics, 1st ed., 2002.

[8] W. F. C. Alvin C. Rencher, *Methods of Multivariate Analysis*. Wiley, 3rd ed., 2012.

[9] A. J. Izenman, *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Texts in Statistics, 2nd ed., 2008.

[10] J. MacQueen, *Some methods for classification and analysis of multivariate observations*. Berkeley Symp. on Math. Statist. and Prob., 1st ed., 1967.

[11] L. S. W. Karl Hardle, *Applied Multivariate Statistical Analysis*. Springer-Verlag Berlin Heidelberg, 4th ed., 2015.

[12] A. Fierro, "Breve historia del descubrimiento de la estructura del dna," *Revista Médica Clínica Las Condes*, vol. 12, pp. 71–75, 2001.

[13] Massachusetts Institute of Technology, "Memory-making involves extensive dna breaking." https://news.mit.edu/2021/memory-making-involves-extensive-dna-breaking-0714. Accessed: 2023-10-12.

[14] National Human Genome Research Institute (NHGRI). An official website of the United States government., "Nucleótidos." https://www.genome.gov/es/genetics-glossary/Nucleotido#:~:text=Los%20nucleÃştidos%20son%20las%20unidades,por%20un%20uracilo%20(U). Accessed: 2023-10-12.

[15] S. R. M.-D. P. R. y. C. S. Lleonart, M. E., "Técnicas de hibridación, clonación y secuenciación de ácidos nucleicos en el diagnóstico anatomopatológico," *Revista Española de Patología*, vol. 30(3), pp. 249–257, 1997.

[16] J. R. R. P. Pablo J. Patiño, "El dogma central de la biología molecular," *Fondo Editorial Biogénesis. Universidad de Antioquia*, 2006.

[17] National Cancer Institute (NCI), the U.S. government's principal agency for cancer research. An official website of the United States government., "Transcriptoma." https://www.genome.gov/es/about-genomics/fact-sheets/Transcriptoma. Accessed: 2023-10-13.

[18] National Cancer Institute (NCI), the U.S. government's principal agency for cancer research. An official website of the United States government., "Transcriptomics." https://www.cancer.gov/publications/dictionaries/cancer-terms/def/transcriptomics. Accessed: 2023-09-26.

[19] National Human Genome Research Institute (NHGRI). An official website of the United States government., "Exon." https://www.genome.gov/es/genetics-glossary/Exon. Accessed: 2023-10-12.

[20] National Human Genome Research Institute (NHGRI). An official website of the United States government., "Gen." https://www.genome.gov/es/genetics-glossary/Gen. Accessed: 2023-10-12.

[21] B. S. J. V. Rosario Madero, E. Pérez Fernández, *Clasificación y predicción con datos de alta dimensión: Aplicación del algoritmo BOOSTING en el desarrollo de biomarcadores*. XXXI Congreso Nacional de Estadística e Investigación Operativa; V Jornadas de Estadística Pública, 1st ed., 2009.

[22] J. R. P. G. C. Andrés J. Hernández, Edilson Delgado Trejos, "Reducción de dimensiones para clasificación de datos multidimensionales usando medidas de información," *Scientia et Technica*, vol. 32, 12 2006.

[23] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Wiley, 3rd ed., 2003.

[24] M. L. D. S. Brian S. Everitt, Sabine Landau, *Cluster Analysis*. Wiley, 5th ed., 2011.

[25] R. E. Bonner, "On some clustering techniques," *IBM Journal of Research and Development*, vol. 8, pp. 22–32, 1964.

[26] J. A. Hartigan, *Clustering Algorithms*. Wiley, 4th ed., 1975.

[27] M. B. D. L. G. M. P. MacNaughton, W. T. Williams, "Dissimilarity analysis: a new technique of hierarchical sub-division," *Nature*, vol. 202, pp. 1034–1035, 1964.

[28] H. J. Jin, X., *K-Medoids Clustering*. Springer Boston, MA., 4th ed., 2011.

[29] E. Forgy, "Cluster analysis of multivariate data: Efficiency vs interpretability of classifications," *Biometrics*, vol. 21, pp. 768–780, 1965.

[30] A. E. L. Ron Edgar, Michael Domrachev, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30(1), pp. 207–210, 2002.

[31] R. E. Tanya Barret, "Gene expression omnibus: Microarray data storage, submission, retrieval, and analysis," *Methods Enzymol*, vol. 411, pp. 352–369, 2006.

[32] J. Q. G. S. P. S. C. S. J. A. W. A. C. A. B. H. C. C. T. G. P. G. F. C. H. I. F. K. V. M. J. C. M. H. P. A. R. U. S. S. S.-K. J. S. R. T. J. V. M. V. A. Brazma, P. Hingamp, "Minimum information about a microarray experiment (miame)-toward standards for microarray data," *Nat Genet.*, vol. 29(4), pp. 365–371, 2001.

[33] C. G. L. A. Z. T. L. D. Robert Gonzalez, Yee Hwa Yang, "Freshly isolated rat alveolar type i cells, type ii cells, and cultured type ii cells have distinct molecular phenotypes," *Am J Physiol Lung Cell Mol Physiol.*, vol. 288(1), pp. 179–189, 2005.

[34] N. C. for Biotechnology Information. An official website of the United States government., "Entrez molecular sequence database system." https://www.ncbi.nlm.nih.gov/Web/Search/entrezfs.html. Accessed: 2023-10-14.

[35] S. L. J. M.-Z. U. A. X. E. Sabio., "Análisis de expresión génica con microarrays." https://hackmd.io/@laurichi13/rkCfeK6mi. Accessed: 2023-10-02.

[36] F.-L.-W.-P. B. A. N. K. L. X. Zhao, S., "Comparison of rna-seq and microarray in transcriptome profiling of activated t cells," *PloS one*, vol. 9(1), 2014.

[37] M. H. Tallulah S. Andrews, "Identifying cell populations with scrnaseq," *Molecular Aspects of Medicine*, vol. 59, pp. 114–122, 2018.

[38] T. L. Valentina Proserpio, "Single-cell technologies are revolutionizing the approach to rare cells," *Immunol Cell Biol.*, vol. 94(3), pp. 225–229, 2015.

[39] R. A. I. Stephanie C. Hicks, Mingxiang Teng, "On the widespread and critical impact of systematic bias and batch effects in single-cell rna-seq data," *bioRxiv*, 2015.

[40] P. S. M. Sean Davis, "Geoquery: a bridge between the gene expression omnibus (geo) and bioconductor," *Bioinformatics Applications Note*, vol. 23(14), pp. 1846–1847, 2007.

[41] T. R. P. for Statistical Computing., "R-project." https://www.r-project.org. Accessed: 2023-10-22.

[42] B. O. source software for bioinformatics., "Bioconductor." https://bioconductor.org. Accessed: 2023-10-22.

[43] V. K. W. H. Michael I. Love, Simon Anders, "Rna-seq workflow: gene-level exploratory analysis and differential expression [version 2; peer review: 2 approved]," *F1000Research*, vol. 4, 2016.

[44] M. Love, "Ranged summarized experiment for rna-seq in airway smooth muscle cells, by himes et al plos one 2014," *Bioconductor*, 2014.

[45] a. Salmon: Fast and bias-aware transcript quantification from RNA-seq data., "Salmon." https://combine-lab.github.io/salmon/getting_started/. Accessed: 2023-10-14.

[46] P. H.-R. P. Michael Love, Charlotte Soneson, "Transcript quantification import with automatic metadata," *Bioconductor*, 2020.