Análisis comparativo de diferentes métodos de agrupación para el tratamiento de datos de expresión genética.

Juan Alberto Gómez Sánchez
Máster en Bioinformática y Bioestadística
Computación e inteligencia artificial en problemas biológicos y clínicos

Romina Rebrij
David Merino Arranz

06/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	Análisis comparativo de diferentes métodos de agrupación para el tratamiento de datos de expresión genética.
Nombre del autor:	*Juan Alberto Gómez Sánchez*
Nombre del consultor/a:	*Romina Rebrij*
Nombre del PRA:	*David Merino Arranz*
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	Computación e inteligencia artificial en problemas biológicos y clínicos.
Idioma del trabajo:	*Castellano*
Palabras clave	*Agrupamiento, aprendizaje automático, expresión génica.*

Resumen del Trabajo (máximo 250 palabras):

En el presente proyecto se ha buscado obtener un método que permita comparar el funcionamiento de varios de los algoritmos de agrupamiento mas comúnmente utilizados al ser aplicados sobre una matriz de datos de expresión génica. La necesidad de poder seleccionar los algoritmos más eficientes surge del aumento progresivo en el volumen de este tipo de datos y los grandes requerimientos de análisis necesarios. Para conseguir los objetivos se han desarrollado los diferentes algoritmos elegidos utilizando un set de datos común con datos de muestras de tejido mamario, las cuales se dividían en clases según su tipo celular (lobular y ductal) y según su estado (normal o tumoral), y se han comparado sus resultados en función de diversos criterios como la internalidad, la estabilidad o la variación biológica. Se han obtenido resultados tanto de cómo agrupan los datos cada uno de los algoritmos, como de comparación entre todos ellos. Estos resultados apuntan a un mejor manejo de estos datos por parte de los algoritmos de tipo jerárquico, en especial el algoritmo DIANA. Finalmente se concluye que los algoritmos de agrupamiento no trabajan especialmente bien con este tipo de datos, ya que a la hora de dividir los datos en grupos no se ha obtenido un buen reparto de forma general, siendo la división entre células tumorales y normales la única agrupación que se puede valorar como positiva.

Abstract (in English, 250 words or less):

The aim of this project is to obtain a method that allow us to compare how some of the most common clustering algoritms works when they are applied over a genetic expression matrix. The desire to select the most efficient algorithms arises from the progressive increase of this data type volume and the huge requirements that it needs. To achieve the objetives it has been developed all the diferents chosen algoritms using it over a common data set which consist in samples of mammary tissue that are divided by his celular type (lobular and ductal) and by his state (normal and tumoral), and its results has been compared in function of diverse criteria like internality, stability and biological variation. Results of how this algorithms group the data has been obtained along with a comparative about the process. The results point to a better control of this data by the hierarchical algorithms, especially the DIANA algorithm. Finally, it is concluded that this type of algorithms doesn't works well with this type of data, because it didn´t get a good divide on the data set generally, being the division between tumoral and normal cells the only what can be rating as positive.

Indice de contenidos

Lista de figuras

1. Introducción

1.1. Contexto y justificación del Trabajo.

En los últimos años se han producido grandes avances en el ámbito de los estudios biomédicos, principalmente en el campo de la genómica, donde los estudios sobre expresión génica han derivado en el aumento en el tamaño de los grupos de datos de forma masiva, por lo que ha surgido la necesidad de implementar nuevas técnicas de análisis y revisión [1].

El aprendizaje automático (o machine learning) se ha impuesto como la solución a muchas de estas nuevas necesidades, permitiendo llevar a cabo análisis de grandes muestras con un coste muy reducido.

Uno de los grupos de métodos más utilizados son los denominados métodos de agrupamiento (o clustering), cuya importancia reside en la capacidad de agrupar datos, que aparentemente no tienen relación, en la misma clase, por lo que se pueden utilizar en multitud de campos donde se quieran clasificar grandes grupos muestrales [2].

Estos métodos representarán el tema central de este trabajo, donde se aplicarán sobre un grupo de datos de expresión génica obtenidos de un estudio sobre cáncer de mama (código NCBI: GSE5764). En este grupo de datos se encuentran diferentes muestras de células pertenecientes a dos tipos de tejidos: ductal y lobular, además en cada tejido se diferencian células normales y tumorales. En todas ellas se ha analizado el nivel de expresión de un gran número de genes. El análisis del comportamiento de los diferentes métodos sobre estos datos se utilizará para medir la eficacia de cada uno y poder compararlos, así se conseguirá determinar cual funciona mejor con este tipo de datos.

El motivo principal para desarrollar este proyecto es que durante los últimos años se ha producido un aumento en la cantidad de datos que se extraen en

estudios sobre expresión génica, y por ello, es necesario desarrollar y perfeccionar herramientas que nos permitan analizar todos estos datos [3].

Una de las herramientas que mejores resultados está dando es el “Machine Learning”; por ello, en este proyecto, se han elegido unos de los principales métodos, como son los métodos de agrupamiento, para tratar los datos obtenidos en un estudio sobre expresión génica [4].

1.2. Objetivos del Trabajo.

Objetivos generales:

Definir un método para determinar cuál de los algoritmos de agrupamiento es más eficaz a la hora de trabajar con datos de expresión génica.

Objetivos específicos:

Agrupar los datos de expresión génica según su tipo celular (lobular o ductal) y su estado (tumoral o normal).

Obtener medidas comparativas entre los diferentes algoritmos usados**.**

1.3. Enfoque y método seguido.

A la hora de realizar un proyecto de este tipo, donde se busca determinar entre varios métodos cual es el mejor para nuestro interés, se pueden seguir dos vías. En primer lugar, se pueden utilizar procedimientos realizados en proyectos similares de forma que se aplicarían los mismos procesos pero utilizando nuestros datos. Por otro lado, se puede investigar cada método de forma individual para elegir cuáles se van a utilizar, y en función de esto, planificar de forma completa todo el procedimiento a seguir.

En mi caso, la estrategia a seguir es la segunda; de esta forma no se dejan posibles errores de planteamiento en manos de terceros.

1.4. Planificación del trabajo.

La planificación del trabajo se ha llevado a cabo utilizando un diagrama de Gantt en el cual se han indicado las tareas a realizar junto con los tiempos requeridos para cada una.

Durante el desarrollo del proyecto se han eliminado, modificado e incluido diferentes tareas, por ello se muestran tanto las tareas y el diagram Gantt originales y finales para poder comparar.

Tareas planificadas:

1.1.1.- Investigar diversos métodos de agrupamiento y decidir cuáles utilizar. (9 días)

1.1.2.- Organizar, analizar y tratar los datos de partida. (5 días)

1.1.3.- Elegir un método de división de datos. Implementación para separar un grupo de entrenamiento y uno de prueba. (4 días)

1.1.4.- Implementar los algoritmos elegidos sobre los datos.(18 días)

1.1.5.- Redactar el primer informe intermedio. (3 días)

1.2.1.- Utilizar métodos de análisis para comprobar la eficacia de cada algoritmo sobre los datos y determinar los factores que mejor describan la eficacia de cada uno.(10 días)

1.2.2.- Realizar una comparativa entre todos los algoritmos.(7 días)

1.2.3.- Determinar cuál es el más eficaz teniendo en cuenta todos los factores valorables.(7 días)

1.2.4.- Redactar el segundo informe intermedio.(4 días)

1.2.5.- Preparar los entregables: informe dinámico y script. (18 días)

1.3.- Redactar la memoria final del proyecto. (14 días)

1.4.- Preparar una presentación. (7 días)

Tareas llevadas a cabo:

1.1.1.- Investigar diversos métodos de agrupamiento y decidir cuáles utilizar. (9 días)

1.1.2.- Organizar, analizar y tratar los datos de partida. (5 días)

1.1.3.- Implementar los algoritmos elegidos sobre los datos.(18 días)

1.1.4.- Redactar el primer informe intermedio. (3 días)

1.2.1.- Reducir el volumen del set de datos original para que todos algoritmos funcionen correctamente con él. (1 día)

1.2.2.- Implementar un método para seleccionar el número de grupos óptimo para los algoritmos.(5 días)

1.2.3.- Utilizar métodos de análisis para comprobar la eficacia de cada algoritmo sobre los datos y determinar los factores que mejor describan la eficacia de cada uno.(10 días)

1.2.4.- Realizar una comparativa entre todos los algoritmos y determinar cuál es el más eficaz teniendo en cuenta todos los factores valorables. (7 días)

1.2.5.- Redactar el segundo informe intermedio.(4 días)

1.2.6.- Preparar los entregables: informe dinámico. (18 días)

1.3.- Redactar la memoria final del proyecto. (14 días)

1.4.- Preparar una presentación. (7 días)

Diagrama de Gantt

	Modo de	Nombre de tarea	Comienzo	Fin	Duración	Predecesoras	Comienzo previsto	Fin de línea base
1		Investigar diversos métodos de agrupamiento y decidir cuales utilizar	lun 19/03/18	mar 27/03/18	9 días		lun 19/03/18	mar 27/03/18
2		Organizar, analizar y tratar los datos de partida	mié 28/03/18	dom 01/04/18	5 días	1	mié 28/03/18	dom 01/04/18
3		~~Elegir un método de división de los datos.~~	~~mié 28/03/18~~	~~sáb 31/03/18~~	~~4 días~~	~~1~~	~~mié 28/03/18~~	~~sáb 31/03/18~~
4		Implementar los algoritmos elegidos sobre los datos	lun 02/04/18	jue 19/04/18	18 días	2	lun 02/04/18	jue 19/04/18
5		Redactar el primer informe intermedio	sáb 21/04/18	lun 23/04/18	3 días	4	sáb 21/04/18	lun 23/04/18
6		Reducir el set de datos para que todos los algoritmos funcionen	mar 24/04/18	mar 24/04/18	1 día	5	mar 24/04/18	mar 24/04/18
7		Elegir el valor de k (número óptimo de clusters) para aplicar en los algoritmos elegidos.	mié 25/04/18	dom 29/04/18	5 días	6	mié 25/04/18	dom 29/04/18
8		Utilizar métodos de análisis para comprobar la eficacia de cada algoritmo sobre los datos y determinar los factores que mejor describan la eficacia de cada uno	lun 30/04/18	mié 09/05/18	10 días	7	lun 23/04/18	mié 02/05/18
9		Realizar una comparativa entre todos los algoritmos y determinar cuál es el más eficaz teniendo en cuenta todos los factores valorables	jue 10/05/18	mié 16/05/18	7 días	8	jue 03/05/18	mié 16/05/18
10		Redactar el segundo informe intermedio	jue 17/05/18	dom 20/05/18	4 días	9	jue 17/05/18	dom 20/05/18
11		Preparar los entregables: informe dinámico y script	lun 30/04/18	jue 17/05/18	18 días	7	lun 30/04/18	jue 17/05/18
12		Comparativa de la salida de los mejores algoritmos con las clases originales	lun 21/05/18	vie 25/05/18	5 días	10	lun 21/05/18	vie 25/05/18
13		Redactar la memoria final del proyecto	lun 21/05/18	dom 03/06/18	14 días	10	lun 21/05/18	dom 03/06/18
14		Preparar una presentación	lun 04/06/18	dom 10/06/18	7 días	13	lun 04/06/18	dom 10/06/18

Ilustración 1: Diagrama de Gantt (parte 1)

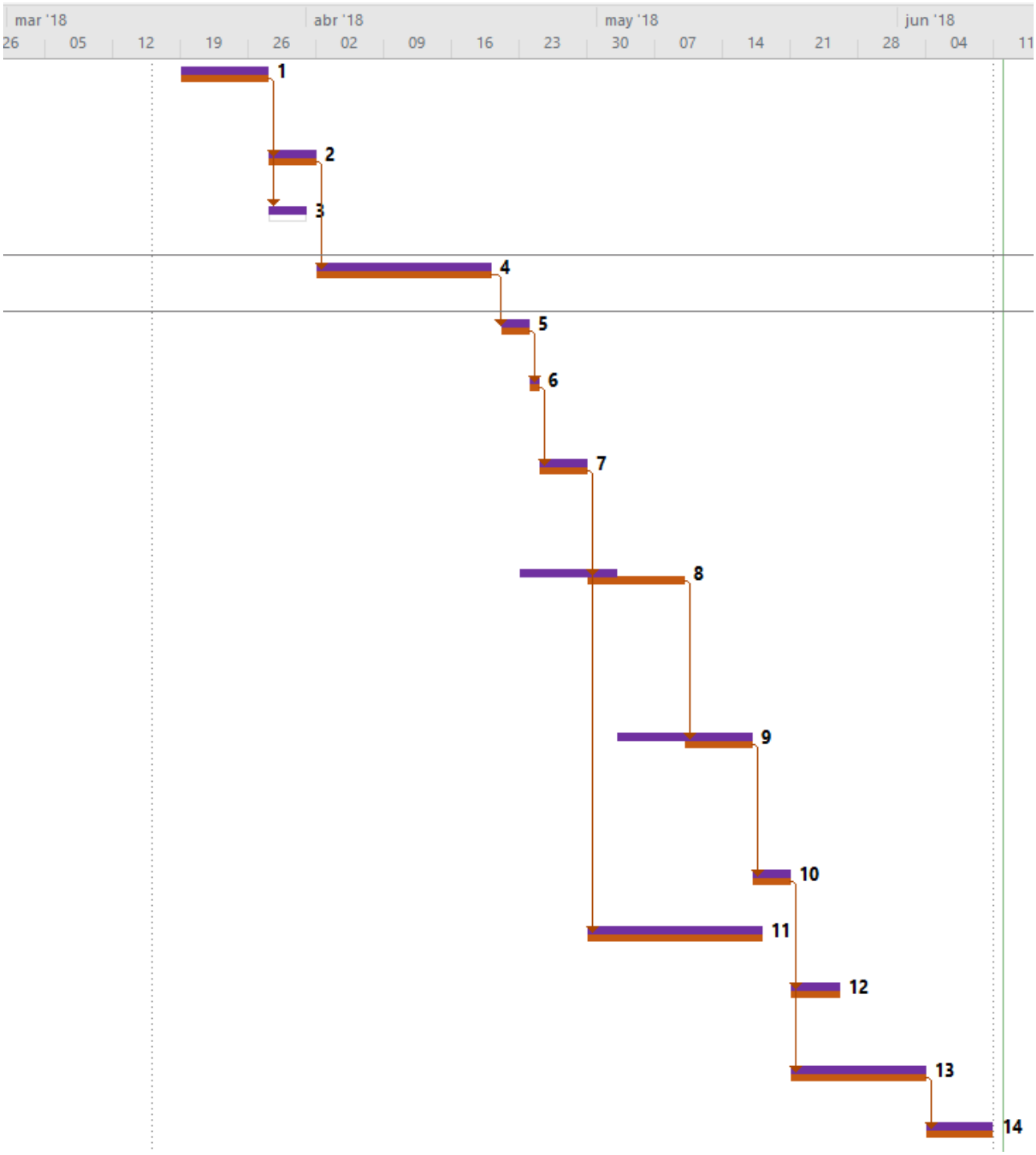

Ilustración 2: Diagrama de Gantt (parte 2)

En el diagrama de Gantt el color morado indica la duración de las tareas planificadas, el blanco las tareas eliminadas y el naranja las tareas realizadas.

1.5. Breve sumario de productos obtenidos.

Método: el resultado principal es una metodología para determinar el mejor algoritmo de agrupamiento para usar sobre datos de expresión génica.

Memoria: en esta se presentará toda la metodología seguida, de forma que pueda ser replicada, junto a los resultados obtenidos de la comparativa, dejando claro cuál ha sido elegido como método más eficaz y el por qué.

Producto: se presentará un informe dinámico con todo el código utilizado y la razón de su uso. El código se podrá aplicar a modo de método para otros datos.

Presentación virtual.

Autoevaluación.

1.6. Breve descripción de los otros capítulos de la memoria.

2. Métodos de agrupamiento

Se indican los algoritmos elegidos para realizar el proyecto, justificando su elección. También se describe el funcionamiento de cada uno de ellos. Por último se analiza la actualidad relacionada con estos métodos, como por ejemplo sus usos actuales.

3. Tratamiento y análisis de los datos.

Se describe la carga y procesado de los datos. Se plantea un análisis sobre la estructura de estos.

4. Implementación de los métodos.

En esta sección se realiza el proceso de selección de k y la implementación de los algoritmos elegidos. También se incluyen las representaciones de estos y unas tablas indicando los porcentajes de agrupamiento de cada clase en cada grupo.

5. Análisis comparativo de los métodos.

Se describen los métodos de comparación que se van a aplicar.

Se realiza una comparación de los algoritmos seleccionados para determinar cual es el mejor trabajando con los datos.

6. Resultados.

Se describen y se valoran los resultados obtenidos en el proceso de comparación.

7. Conclusiones.

En esta sección se plantean las conclusiones finales basadas en los resultados obtenidos.

8.Glosario.

Esta sección está dedicada a definir términos utilizados comúnmente durante el proyecto y que pueden no ser de conocimiento general.

9. Bibliografía.

En esta sección se incluye la bibliografía que se ha revisado durante el proyecto.

2. Métodos de agrupamiento

2.1. Definición y actualidad.

Los diferentes métodos de agrupamiento o algoritmos de clustering que se utilizan en el presente trabajo pueden agruparse dentro de la rama del aprendizaje no supervisado en el campo de la inteligencia artificial.

El término inteligencia artificial es en resumidas cuentas la capacidad de una máquina para percibir su entorno y realizar acciones que maximicen sus posibilidades de éxito durante la realización de una tarea con el fin de mejorar la eficiencia con que realiza esta. Para el desarrollo del proyecto se requiere la capacidad de aprendizaje que puedan presentar muchos de los métodos dentro de este campo, además se utilizará la capacidad de estos para extraer información de los datos de forma autónoma.

La rama del aprendizaje no supervisado congrega las metodologías que no requieren un conocimiento previo a la hora de desarrollarse, en este caso concreto los algoritmos utilizados tratan los datos como un conjunto de variables aleatorias sin atender a las clases en que están agrupadas las muestras y sin valorar la procedencia de estas.

Las técnicas de agrupamiento o clustering se basan en varios conceptos de forma general. El principal objetivo de estos es dividir o agrupar los datos en grupos homogéneos, utilizando para ello bien la distancia entre los objetos o bien la similitud entre estos. Para realizar estas divisiones los algoritmos tratan de obtener la mínima distancia o máxima similitud posible entre los objetos dentro de un mismo grupo y a la vez la máxima distancia o mínima similitud posible entre los diferentes grupos [5].

Cada uno de los diferentes grupos asignados por el algoritmo se conocen como clusters, siendo asignados de forma general a la letra *k*, por lo que si se

dice que un algoritmo tiene un *k* igual a 3, está indicando que los datos se van a dividir en 3 grupos diferentes.

Los algoritmos de clustering utilizan diferentes caminos para llevar a cabo la formación de grupos, en función de como se realiza el proceso se genera una división dentro de estos algoritmos en dos grupos, los algoritmos jerárquicos y no jerárquicos.

El grupo de algoritmos jerárquicos agrupa diferentes técnicas que basan su funcionamiento en la incorporación una por una de cada muestra incluyéndola en el grupo con mayor similaridad. La mayoría de estos algoritmos generan dendrogramas donde los objetos se van uniendo hasta conformar un árbol con todos ellos.

En el caso de los algoritmos no jerárquicos el funcionamiento parte de una distribución de los objetos, los cuáles serán asignados a uno u otro grupo en función de diferentes motivos, como pueda ser la cercanía a un punto medio o a un punto mediano. Las representaciones de estos algoritmos suelen llevarse a cabo mediante una nube de puntos, los cuales se reúnen en grupos de diferentes formas.

Actualmente, la inteligencia artificial se reconoce como uno de los pilares en la innovación en casi todos los campos de conocimiento, ya que se puede utilizar desde para el desarrollo de coches autónomos hasta para la investigación de diferentes enfermedades. Particularmente, las técnicas de clustering se utilizan para encontrar patrones en grupos de datos de gran tamaño que aparentemente no se observan; esto puede llevarse a cabo en campos como la Economía y el Marketing [6], y más relacionados con la Bioinformática, como la investigación de enfermedades [7], el análisis de perfiles genéticos [5] o la clasificación taxonómica de especies [8].

Dentro del programa R se han desarrollado multitud de funciones que replican los métodos implementados por la mayoría de algoritmos de clustering planteados hasta el momento. Debido a la gran cantidad de estos disponibles

se han seleccionado varios, atendiendo a varios motivos. En primer lugar se ha valorado la posibilidad de poder comparar estos algoritmos entre sí. Esta comparación es más compleja que para otras técnicas de inteligencia artificial, debido a que se realiza sin supervisión. En segundo lugar se ha valorado la capacidad del algoritmo de trabajar con datos de expresión génica, por lo cual se han descartado aquellos que no son capaces de trabajar correctamente este tipo de datos. En la siguiente sección se recogen los algoritmos seleccionados para el proyecto, junto al motivo de su elección y una definición de como trabaja cada uno.

2.2. Algoritmos elegidos.

Como se explica en el anterior apartado, la primera premisa para seleccionar los algoritmos ha sido que pudieran ser comparados con alguna metodología técnicamente correcta. Por ello, todos los algoritmos seleccionados se encuentran disponibles en la función *optCluster()* [9], perteneciente al paquete del mismo nombre, la cual será la encargada de realizar las comparaciones pertinentes en función de diversos factores que se indicarán posteriormente. En dicha función se encuentran disponibles 16 algoritmos diferentes, de los cuales se han seleccionado 7 para trabajar con datos de expresión génica.

Los 7 algoritmos seleccionados se describen a continuación, agrupados en las divisiones comentadas anteriormente:

- Algoritmos jerárquicos:
  - Hierarchical clustering [10,11,12,13]:
    - Descripción: En este algoritmo los datos se van agrupando en clusters formados por parejas cuya similaridad es mas alta hasta que solo hay un cluster que reúne a todos los demás. Es necesario elegir un método de comparación entre los clusters. La importancia de este algoritmo está en la facilidad de representación que tiene en un dendrograma y su fácil interpretación.

        - Función en R: *hclust()*
        - Forma de los datos: requiere una matriz de distancias de los datos, la cual se obtiene aplicando la función *dist()* sobre los datos ya tratados y ordenados con las muestras a colocar en las filas.
    - AGNES (Agglomerative Nesting)[13,14]:
        - Descripción: El funcionamiento de este algoritmo es similar al Hierarchical, en un primer paso asigna un cluster a cada muestra y a partir de entonces va uniendo los clusters en función de su similitud, hasta que todos forman un único cluster o hasta llegar al límite definido (por ejemplo la *k* definida).
        - Función en R: *agnes()*
        - Forma de los datos: esta función utiliza los datos en una matriz de distancias, pero la entrada debe ser con los datos originales tratados y con las muestras en las filas, ya que es la propia función la que crea la matriz de distancias.
    - DIANA (Divisive Analisis)[13,15]:
        - Descripción: El algoritmo DIANA trabaja de forma opuesta a los anteriores, basa su funcionamiento en un primer paso donde todas las muestras están unidas en un único cluster y desde ese punto va dividiendo en pares de clusters a medida que sus distancias aumentan, el proceso termina al llegar al estado donde todas las muestras pertenecen a un sólo cluster.
        - Función en R: *diana()*
        - Forma de los datos: en este caso la función trabaja de forma idéntica a AGNES, por lo que los datos de entrada serán los originales.
- Algoritmos no jerárquicos o basados en particiones:
    - K-means[16,17]:
        - Descripción: El algoritmo divide los datos en *k* clusters. Su funcionamiento se basa en formar los clusters de forma que la varianza interna de cada uno sea mínima, para ello utiliza una medida virtual de la media de las observaciones del cluster. Su

importancia radica en la sencillez que tiene y la velocidad a la que se desarrolla, por otro lado sufre problemas al tratar datos con gran cantidad de outliers.

- Función en R: *kmeans()*
- Forma de los datos: para el desarrollo de esta función no se requiere ningún tratamiento extra de los datos. Una vez estos han sido normalizados pueden utilizarse, de forma similar a los anteriores, para una división de las muestras, las cuales deben estar en las filas.

◦ PAM (Partition Around Medoids)[18,19,20]:

- Descripción: Este algoritmo es similar al k-means, con la diferencia de que en este la medida a partir de la cual se desarrolla cada cluster es la mediana de las observaciones de ese cluster (medoid). La importancia de este algoritmo es que funciona de forma similar al k-means pero mejorando el tratamiento de outliers.
- Función en R: *pam()*
- Forma de los datos: similar a k-means.

◦ CLARA (Clustering Large Aplications)[21,22]:

- Descripción: El algoritmo CLARA es una versión de PAM diseñada para ser más eficiente con grupos de datos mayores. Su funcionamiento se basa en la selección de subgrupos aleatorios dentro el grupo completo de datos y aplicar sobre estos el algoritmo PAM. Los grupos que muestran menor error cuadrático son seleccionados como clusters. Su eficiencia aumenta al aumentar el tamaño de los datos. Un inconveniente que posee es que al seleccionar parte de los datos puede crear un sesgo.
- Función en R: *clara()*
- Forma de los datos: similar al algoritmo k-means.

◦ Model-Based clustering[23,24]:

- Descripción: Este tipo de algoritmo asume que cada observación proviene de una distribución la cual puede ser la combinación de varios clusters. La distribución de cada cluster se realiza mediante

el algoritmo de maximización de expectacion. Su importancia radica en que los valores asignados no son fijos a un cluster, sino que pueden pertenecer a varios a la vez y permite cierta flexibilidad a la hora de definir los grupos. Una característica diferencial de este método es que congrega varios modelos posibles en una sola función, al implementar el algoritmo este elige cual de los modelos es más adecuado para las características de los datos y el número de grupos asignado.

- Función en R: *Mclust()*
- Forma de los datos: similar a k-means.

3. Tratamiento y análisis de los datos

3.1. Carga de los datos

En secciones anteriores se ha hablado sobre el origen de los datos y el estudio al que pertenecen; en este punto se describe cómo y de donde se extraen los datos. Para obtener la información sobre los datos se han aplicado dos métodos distintos, de forma que las clases de cada muestra se obtienen de un fichero diferente a los datos de la matriz de expresión.

En primer lugar se obtienen las clases; para ello, se ha descargado desde el enlace (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5764) el archivo “GSE5764_series_matrix.txt.gz”, que contiene la información fenotípica de las muestras junto con los datos de expresión. Para cargar este archivo se utiliza la función *getGEO()* del paquete *GEOquery* [25].

Para poder extraer la información sobre las clases del objeto generado se utiliza la función *phenoData()* y la función *pData()* sobre el resultado de la anterior. En este punto se obtiene un data frame con toda la información de cada paciente, y en concreto, en la primera columna, se encuentra la información del tipo celular de cada muestra. Debido a que están representadas con una nomenclatura especial, se tratan para que sean homogéneas y además se generan unas abreviaturas para cada muestra que se utilizarán posteriormente en las representaciones.

La función de estos objetos con las clases será de dos tipos. Se utilizarán para facilitar la identificación de las muestras en los gráficos, de forma que se pueda observar mejor como se reúnen las muestras de tipos similares. También se utilizarán para realizar una comparativa sobre la eficiencia a la hora de reunir muestras de la misma clase en el mismo grupo.

A continuación se muestra una tabla resumen de las clases de cada muestra, así como la abreviatura asignada a cada una.

```
##           estadocel tipocel abreviatura
## GSM134584    normal  ductal         1ND
## GSM134586    normal lobular         1NL
## GSM134587   tumoral lobular         1TL
## GSM134588    normal  ductal         2ND
## GSM134589    normal lobular         2NL
## GSM134591   tumoral lobular         2TL
## GSM134687    normal  ductal         3ND
## GSM134688    normal lobular         3NL
## GSM134689   tumoral lobular         3TL
## GSM134690    normal  ductal         4ND
## GSM134691    normal lobular         4NL
## GSM134692   tumoral lobular         4TL
## GSM134693    normal  ductal         5ND
## GSM134694    normal lobular         5NL
## GSM134695   tumoral lobular         5TL
## GSM134696    normal  ductal         6ND
## GSM134697    normal lobular         6NL
## GSM134698   tumoral  ductal         1TD
## GSM134699    normal  ductal         7ND
## GSM134700    normal lobular         7NL
## GSM134701   tumoral  ductal         2TD
## GSM134702    normal  ductal         8ND
## GSM134703    normal lobular         8NL
## GSM134704   tumoral  ductal         3TD
## GSM134705    normal  ductal         9ND
## GSM134706    normal lobular         9NL
## GSM134707   tumoral  ductal         4TD
## GSM134708    normal  ductal        10ND
## GSM134709    normal lobular        10NL
## GSM134710   tumoral  ductal         5TD
```

Ilustración 3: Tabla resumen de las clases y sus abreviaturas

Anteriormente se ha comentado que en el mismo archivo de donde se extraen las clases se encuentra también la matriz de expresión. La razón para no utilizar esta para el proyecto reside en que el objeto que se obtiene al cargarla no es compatible con algunas funciones del paquete *affy* que se utilizarán después para tratar los datos. Por tanto, se ha optado por obtener esta matriz de expresión de los archivos .CEL. Estos pueden ser descargados desde el mismo enlace de GEO, pero en este caso el archivo a descargar es "GSE5764_RAW.tar"

Los archivos .CEL se utilizan de forma común para guardar resultados de estudios de expresión de Affymetrix. Están divididos de forma que en cada archivo se guarda la información de una sola muestra. Para cargarlos, además del directorio donde se encuentran, es necesario un archivo con los fenotipos, en este caso el archivo ha sido generado a la hora de obtener las clases. La función utilizada es *read.affybatch()*, perteneciente al paquete *affy* [26].

3.2. Tratamiento de los datos

En este punto están cargados y guardados en objetos tanto las clases en distintas formas, como la matriz de expresión sin tratar, por lo cual se puede comenzar el tratamiento de los datos que facilitará la implementación de los algoritmos de clustering sobre ellos.

El primer paso del tratamiento consiste en una normalización o escalado de los datos. Para realizar este proceso se ha elegido el método de RMA. Este método es uno de los más comunes a la hora de tratar matrices de expresión génica de Affymetrix. Su funcionamiento consiste en un primer paso donde se realiza una correción de fondo de los datos, seguido de un proceso de normalización por cuantiles en cada una de las muestras, y se finaliza con un cálculo de la expresión de cada probe. El resultado de este proceso es un objeto de clase “Expression Set”, cuyos datos de las muestras están correctamente normalizados.

Una vez se tienen los datos en la clase “Expression Set” se puede aplicar sobre ellos un filtro. La razón para utilizar un filtro es con el objetivo de reducir ampliamente el tamaño de los datos. En un primer momento el objetivo era utilizar la matriz completa, lo cual conllevó que algunos de los algoritmos no pudieran ejecutarse debido a que los objetos generados eran de tal volumen que la memoria RAM del equipo utilizado no podía abarcarlos. Además, durante revisiones de trabajos similares [27], se observó cómo los tiempos de procesado eran muy altos incluso con matrices de proporción 1:250 respecto a

la que aquí se trabaja; por ello se tomó la decisión de reducir de forma drástica el tamaño de los datos. La opción elegida para realizar este proceso fue la función *nsFilter()*, del paquete *genefilter.* La razón para escoger esta función es que no produce un filtrado aleatorio, sino que utiliza tanto las tablas de anotación de genes como los niveles de expresión de estos para conservar aquellos genes que tienen mayor importancia en los datos; de este modo, la información que se pierde al reducir la matriz es menor. Esta función trabaja en dos pasos; en un primer momento valora para cada gen si su varianza o media deben de aceptarse, los genes que son aceptados en este paso son sometidos a un test estadístico (t-test) que valora cuales son más importantes [28]. Se debe aportar a la función un valor de corte, que representa el porcentaje de genes que se van a eliminar, en este caso el valor dado es de 0.99, es decir, el 99% de los genes se eliminarán, conservando el 1% más influyente. Este porcentaje de filtrado se ha elegido basándose en el trabajo comentado anteriormente donde se utilizaba optCluster, el objetivo ha sido tener una matriz de datos de tamaño similar a la utilizada en ese estudio. En términos numéricos la matriz original constaba de 54675 variables o probes, mientras que la matriz filtrada consta de 202 variables.

En un último paso del tratamiento se obtiene un nuevo objeto con la matriz de expresión, utilizando la función *exprs()* sobre los datos filtrados. También se sustituyen en la matriz transpuesta los nombres de las muestras por las abreviaturas, conservando otra matriz con los nombres originales para utilizarla en el análisis de los datos.

3.3. Análisis de los datos

La mayoría de proyectos que utilizan datos de expresión génica realizan en mayor o menor medida un análisis descriptivo de estos con el fin de observar posibles errores graves o desviaciones debidas a aspectos técnicos o ruido, que puedan conllevar que el trabajo realizado posteriormente pierda validez. Por ello, en este caso, se realiza un análisis basado en representaciones gráficas de distinto tipo que darán una idea general de la estructura de los datos.

En primer lugar se valora el grado de normalización en las muestras, ya que este proceso se realizó antes del filtrado es posible que la normalización no sea perfecta, pero es importante que sea buena. Para comprobar este dato se utiliza la función *summary(),* que muestra la distribución en cuantiles y la mediana de cada muestra, sobre los datos ya filtrados y normalizados.

```
##   "GSM134584"      "GSM134586"      "GSM134587"      "GSM134588"
##  Min.   : 2.477   Min.   : 2.671   Min.   : 2.845   Min.   : 2.555
##  1st Qu.: 3.615   1st Qu.: 4.341   1st Qu.: 4.523   1st Qu.: 5.152
##  Median : 5.095   Median : 7.086   Median : 7.017   Median : 6.920
##  Mean   : 6.096   Mean   : 6.788   Mean   : 6.766   Mean   : 6.775
##  3rd Qu.: 8.581   3rd Qu.: 8.709   3rd Qu.: 8.231   3rd Qu.: 8.310
##  Max.   :13.126   Max.   :13.107   Max.   :12.618   Max.   :11.739
##   "GSM134589"      "GSM134591"      "GSM134687"      "GSM134688"
##  Min.   : 2.700   Min.   : 2.672   Min.   : 2.681   Min.   : 2.583
##  1st Qu.: 5.113   1st Qu.: 5.215   1st Qu.: 5.614   1st Qu.: 4.060
##  Median : 6.675   Median : 6.568   Median : 7.030   Median : 6.401
##  Mean   : 6.728   Mean   : 6.762   Mean   : 7.126   Mean   : 6.518
##  3rd Qu.: 8.156   3rd Qu.: 8.325   3rd Qu.: 8.524   3rd Qu.: 8.691
##  Max.   :12.003   Max.   :13.060   Max.   :13.461   Max.   :13.347
##   "GSM134689"      "GSM134690"      "GSM134691"      "GSM134692"
##  Min.   : 2.431   Min.   : 2.554   Min.   : 2.473   Min.   : 2.625
##  1st Qu.: 4.316   1st Qu.: 6.172   1st Qu.: 5.595   1st Qu.: 3.482
##  Median : 6.617   Median : 7.079   Median : 7.492   Median : 4.248
##  Mean   : 6.623   Mean   : 7.214   Mean   : 7.223   Mean   : 5.009
##  3rd Qu.: 8.346   3rd Qu.: 8.307   3rd Qu.: 8.740   3rd Qu.: 5.685
##  Max.   :13.146   Max.   :13.488   Max.   :13.446   Max.   :12.203
##   "GSM134693"      "GSM134694"      "GSM134695"      "GSM134696"
##  Min.   : 3.008   Min.   : 2.721   Min.   : 2.462   Min.   : 2.432
##  1st Qu.: 4.839   1st Qu.: 4.880   1st Qu.: 4.304   1st Qu.: 4.581
##  Median : 6.318   Median : 6.413   Median : 5.786   Median : 6.890
##  Mean   : 6.597   Mean   : 6.506   Mean   : 6.100   Mean   : 6.676
##  3rd Qu.: 7.760   3rd Qu.: 7.897   3rd Qu.: 7.296   3rd Qu.: 8.437
##  Max.   :11.898   Max.   :11.795   Max.   :12.327   Max.   :12.884
##   "GSM134697"      "GSM134698"      "GSM134699"      "GSM134700"
##  Min.   : 2.741   Min.   : 2.594   Min.   : 2.623   Min.   : 2.720
##  1st Qu.: 4.990   1st Qu.: 4.128   1st Qu.: 5.846   1st Qu.: 6.271
##  Median : 7.013   Median : 5.948   Median : 7.128   Median : 7.574
##  Mean   : 6.917   Mean   : 6.020   Mean   : 7.101   Mean   : 7.372
##  3rd Qu.: 8.698   3rd Qu.: 7.501   3rd Qu.: 8.552   3rd Qu.: 8.690
##  Max.   :13.088   Max.   :12.340   Max.   :12.437   Max.   :12.095
##   "GSM134701"      "GSM134702"      "GSM134703"      "GSM134704"
##  Min.   : 2.820   Min.   : 2.865   Min.   : 2.792   Min.   : 2.661
##  1st Qu.: 5.256   1st Qu.: 5.857   1st Qu.: 4.424   1st Qu.: 3.978
##  Median : 7.187   Median : 6.987   Median : 5.816   Median : 5.947
##  Mean   : 6.973   Mean   : 7.139   Mean   : 6.194   Mean   : 6.038
##  3rd Qu.: 8.556   3rd Qu.: 8.247   3rd Qu.: 7.467   3rd Qu.: 7.497
##  Max.   :12.835   Max.   :12.949   Max.   :12.967   Max.   :12.613
##   "GSM134705"      "GSM134706"      "GSM134707"      "GSM134708"
##  Min.   : 2.923   Min.   : 2.845   Min.   : 2.767   Min.   : 2.522
##  1st Qu.: 5.623   1st Qu.: 4.342   1st Qu.: 4.294   1st Qu.: 5.302
##  Median : 6.620   Median : 5.540   Median : 5.442   Median : 7.233
##  Mean   : 6.745   Mean   : 5.796   Mean   : 5.947   Mean   : 7.063
##  3rd Qu.: 7.912   3rd Qu.: 6.708   3rd Qu.: 7.116   3rd Qu.: 8.621
##  Max.   :12.112   Max.   :12.007   Max.   :13.155   Max.   :13.214
##   "GSM134709"      "GSM134710"
##  Min.   : 2.529   Min.   : 2.837
##  1st Qu.: 5.395   1st Qu.: 4.510
##  Median : 7.424   Median : 6.094
##  Mean   : 7.143   Mean   : 6.208
##  3rd Qu.: 8.523   3rd Qu.: 7.430
##  Max.   :13.152   Max.   :12.863
```

Ilustración 4: Sumario completo de los datos

En la tabla anterior se observa que las muestras se mantienen de una forma mas o menos constante en los mismos valores. Como se comentó anteriormente, las desviaciones que se observan pueden deberse al proceso de filtrado que sufrieron los datos y que al eliminar variables los rangos de algunas muestras hayan tendido a aumentar y los de otras a disminuir.

Además de los datos de la matriz de expresión es importante valorar las clases que se obtienen, para ello se analiza el objeto con las clases y se observan tanto la proporción de cada clase como la cantidad de muestras pertenecientes a cada una.

```
## clase.a
##  normalductal normallobular  tumorlobular   tumorductal
##            10            10              5             5
```

Ilustración 5: Tabla con la cantidad de muestras de cada clase

```
## clase.a
##  normalductal normallobular  tumorlobular   tumorductal
##     0.3333333     0.3333333     0.1666667     0.1666667
```

Ilustración 6: Tabla con el porcentaje de cada clase

Continuando con el análisis de la matriz de expresión, una de las representaciones gráficas más comunes para determinar la distribución de datos de este tipo es el histograma. Este tipo de representación agrupa los valores de expresión, en este caso en función de su densidad, lo que permite conocer si la distribución de valores es homogénea o no.

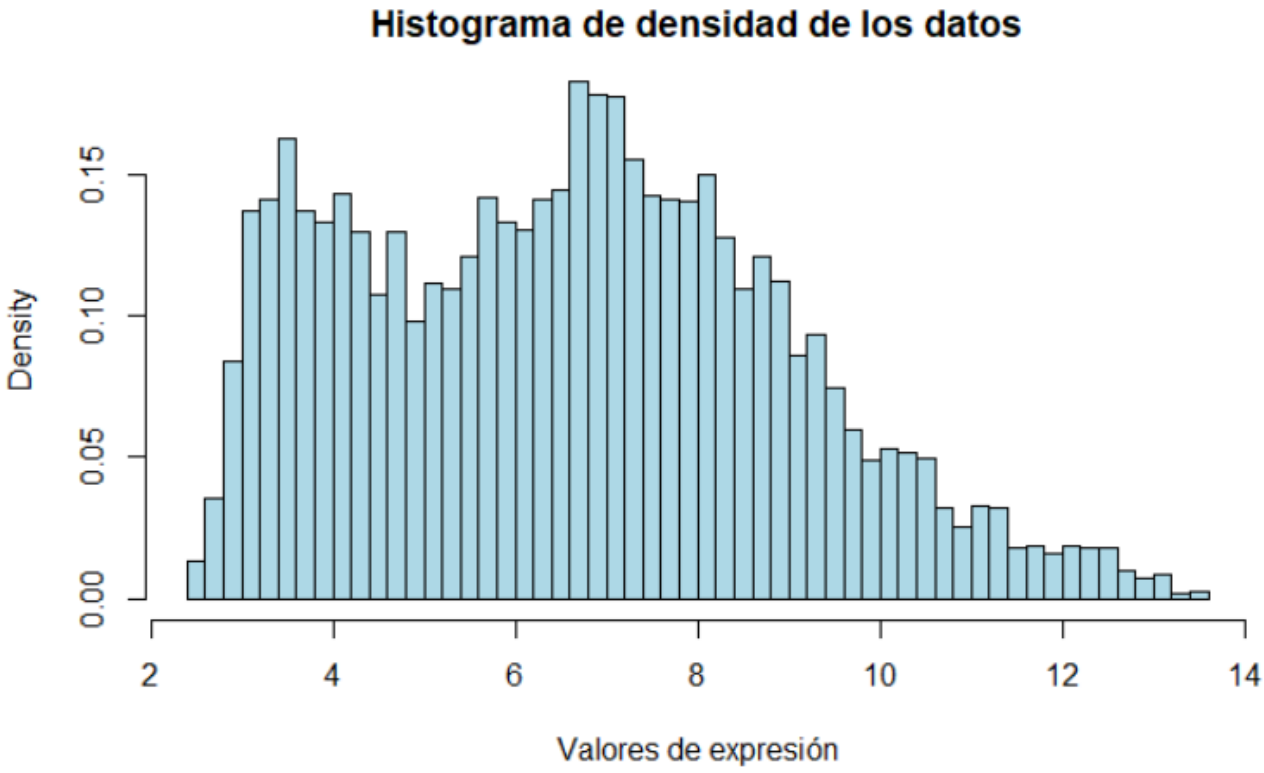

Ilustración 7: Histograma de los datos

En la representación en forma de histograma se observan dos picos de concentración de valores, uno en torno al 3 y otro en torno al 7, lo que indica que la mayoría de valores se concentran en torno a estos dos. Por otro lado la gran mayoría están en el rango entre 2,5 y 13, que como se vio en el sumario son los valores máximos y mínimos que aparecen en los datos.

En datos con una cantidad grande de variables como es este caso se puede, a la hora de representar gráficamente, unir varias de estas variables en lo que se conoce como componentes principales. Estas componentes reúnen las variables que cubren el mayor porcentaje de variabilidad posible, formando la componente principal. En la representación que se muestra a continuación se enfrentan las dos componentes principales, una en cada eje, y sobre el gráfico se distribuyen las muestras en forma de punto, con lo que se puede inferir como es la distribución de las muestras sobre estas componentes principales. La importancia de este gráfico reside en que es el que se utilizará para las representaciones de los algoritmos divisivos o basados en particiones.

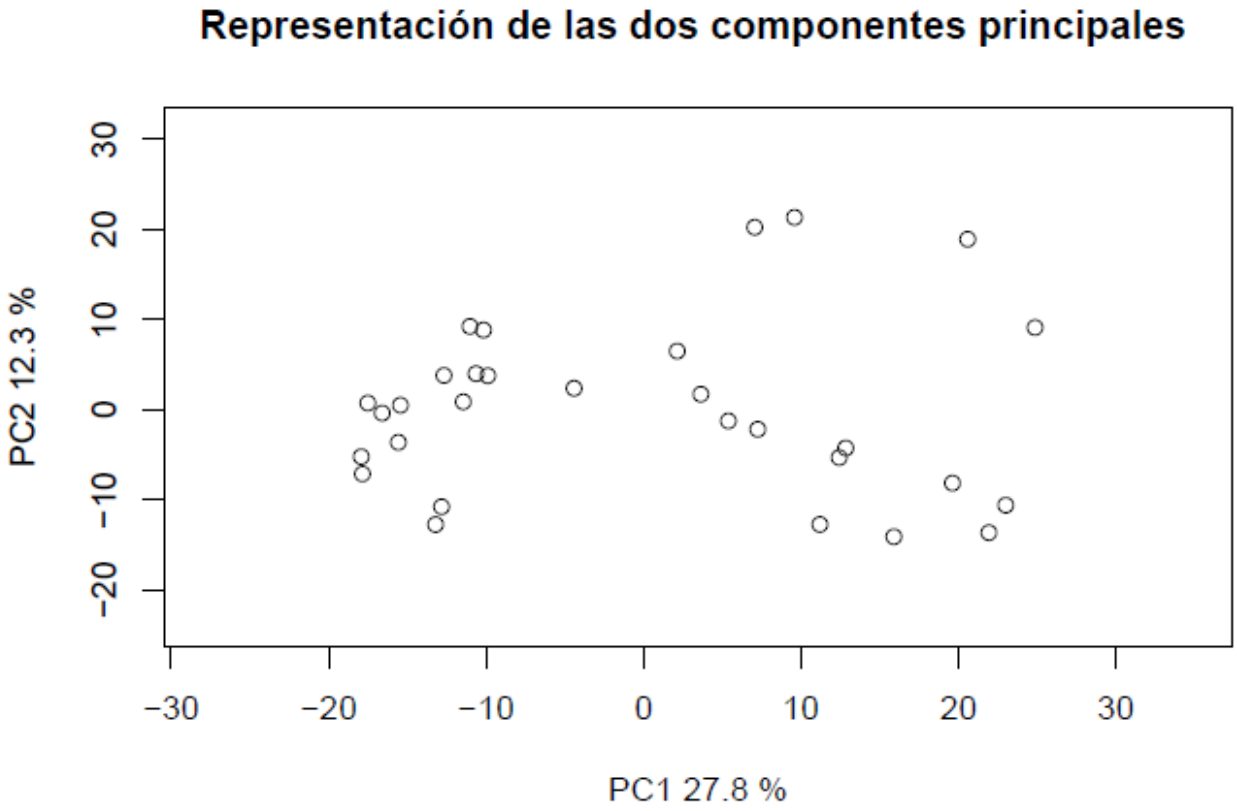

Ilustración 8: Representación gráfica de los componentes principales

En el gráfico se observa como las muestras se encuentran entre el -20% y 25% de la componente 1 y -20% y 20% de la componente 2.

Otra representación clásica de este tipo de datos es el boxplot. En este tipo de gráfico se pueden observar los cuantiles recogidos en la tabla de sumario de una forma más clara y comparable. Para facilitar la diferenciación se han marcado por colores según la clases de cada muestra.

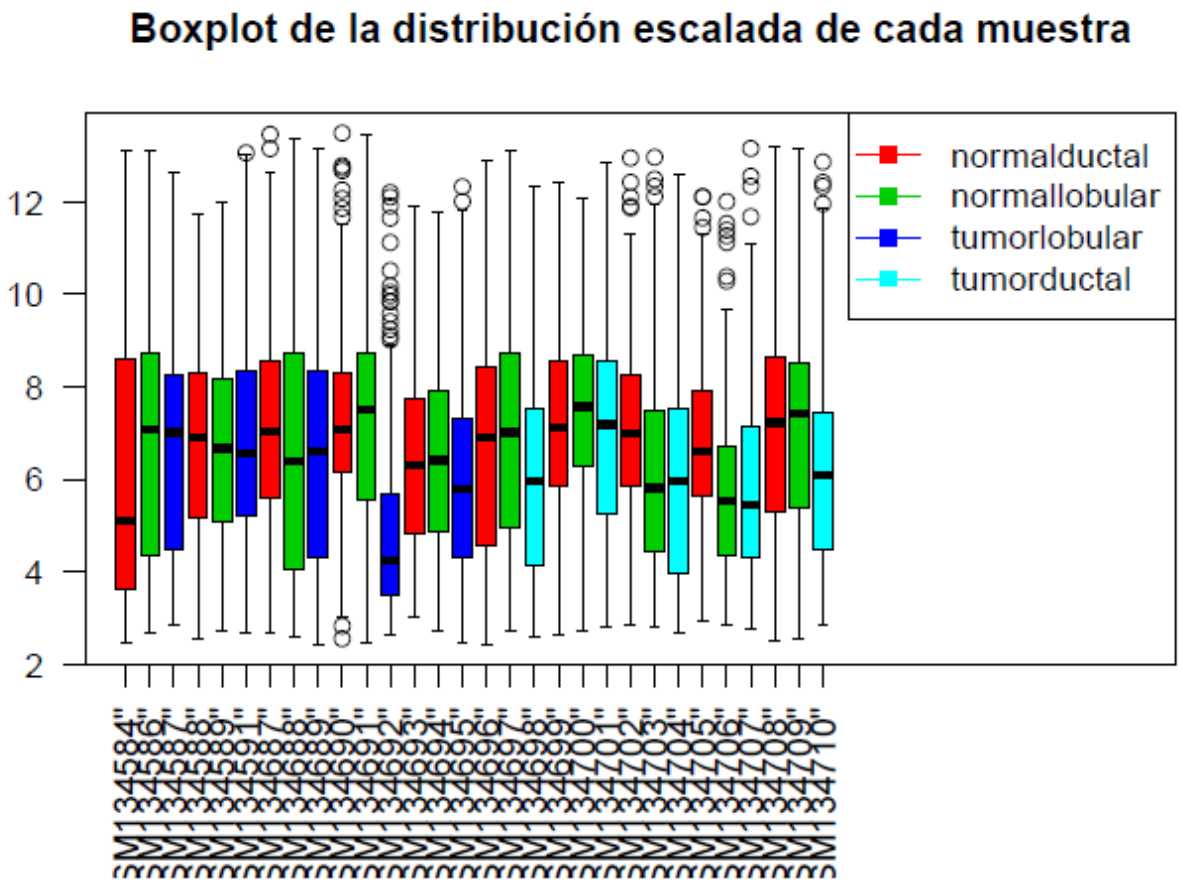

Ilustración 9: Diagrama de cajas de los datos

Lo que se explicó sobre el sumario se podría aplicar de forma igual en el boxplot; las muestras tienen cierta homogeneidad en su distribución, con variabilidad posiblemente debida al proceso de filtrado.

Para finalizar el proceso de análisis se va a generar una matriz de distancias para las muestras, de forma que se pueda observar la distancia euclídea entre cada una. Para facilitar su interpretación se representa gráficamente en un mapa de calor con la función *fviz_dist()* del paquete *factoextra* [29].

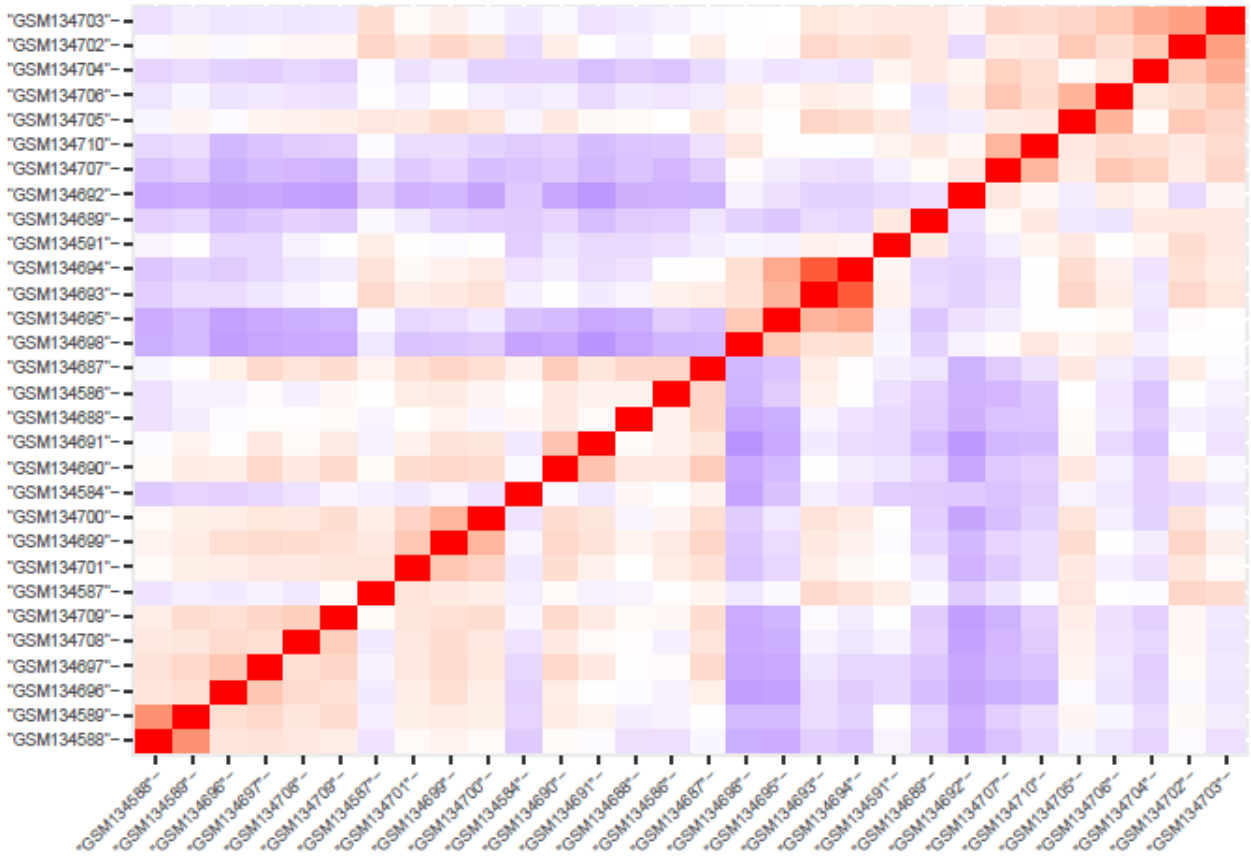

Ilustración 10: Mapa de calor de las muestras

En el mapa de calor las regiones con colores más rojizos representan mayor cercanía entre las muestras, mientras que las zonas más azules indican mayor distancia.

Con esta representación se da por concluido el análisis de los datos, dando por buena su validez. Por ello, se puede comenzar con el proceso de aplicación de los algoritmos elegidos previamente.

4. Implementación de los métodos

4.1. Planteamiento general

Llegado este punto ya se tienen los datos preparados para aplicar sobre ellos los algoritmos seleccionados, pero se requiere un proceso previo antes de ejecutar y representar cada uno para poder seleccionar los mas eficientes y ahorrar tiempo de procesado. A continuación se describe el proceso que se seguirá para cada algoritmo, de forma que para cada uno se seleccionarán los mejores valores de *k*, es decir el valor óptimo de número de grupos en que se dividen los datos.

El proceso constará de los siguientes pasos:

- En primer lugar se utiliza la función *optCluster()* indicando el algoritmo seleccionado correspondiente, así como un número de ciclos suficientemente grande como para no dejar lugar a dudas sobre el resultado. También se indica que se utilicen todos los métodos comparativos disponibles (explicados en la sección sobre comparativa de los algoritmos) . Por último se incluye una semilla fija que permitirá reproducir los resultados sin aleatoriedad.
- Se incluye un rango para los valores de *k* siguiendo el modelo *n*:*m*, donde n es el valor mínimo que puede poseer k y m es un valor mayor que m y que formará un intervalo donde se espera encontrar el óptimo.
- Al analizar la salida de la función, si el valor de *k* óptimo ha sido el valor *m*, este se aumenta, si es menor que este se asume como valor óptimo.
- El ciclo se repite hasta que el valor para *k* obtenido sea menor que *m*.
- Se realiza el algoritmo utilizando su función propia, planteando el valor de *k* obtenido anteriormente.
- Con el objeto obtenido de cada función propia se realiza una representación gráfica que muestra como ha dividido los datos el algoritmo [30].

- Utilizando los datos obtenidos en estas gráficas se redacta una tabla definiendo el porcentaje de cada clase en cada grupo. Esta comparativa se valorará en las conclusiones.

Puntualizaciones sobre el proceso de selección de *k*:

- En el código no se encuentra representada esta selección, sino que ha sido realizada y se han conservado sólo el rango de valores de *k* dentro del cual todos los algoritmos tienen su óptimo con el fin de disminuir el tiempo de procesado.
- Para cada caso se han seleccionado los dos mejores valores, buscando dar más amplitud a la prueba y obteniendo mejor capacidad comparativa dentro del mismo algoritmo.

4.2. Algoritmos jerárquicos

- Hierarchical :

Los valores óptimos de *k* para este algoritmo son 2 y 4, en este orden.

Como se planteó en la descripción de estos algoritmos, su representación se realiza en forma de dendrograma, por lo que para diferenciar los grupos se ha implementado una separación por colores así como unos recuadros que delimitan las muestras de cada grupo.

  - K=2

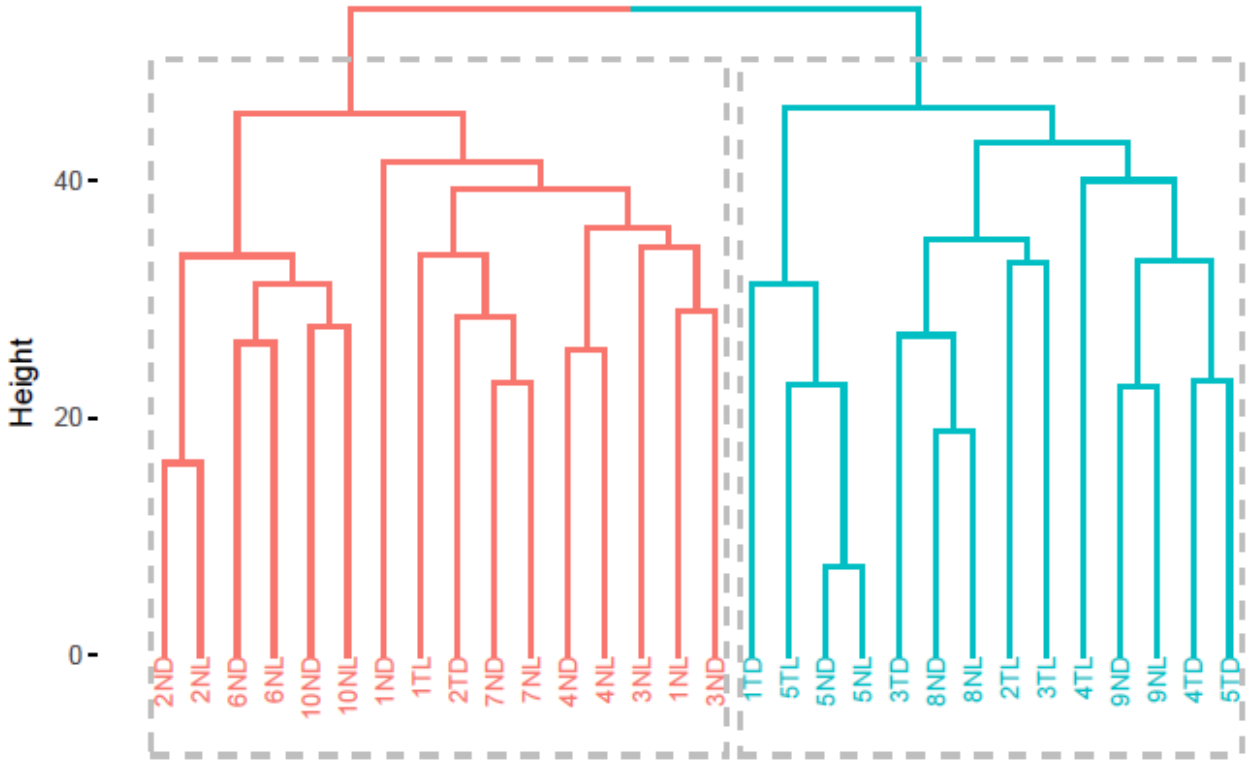

Ilustración 11: Representación Hierarchical k=2

	Grupo 1	Grupo 2
Normal Lobular	44%	21%
Normal Ductal	44%	21%
Tumoral Lobular	6%	29%
Tumoral Ductal	6%	29%

- K=4

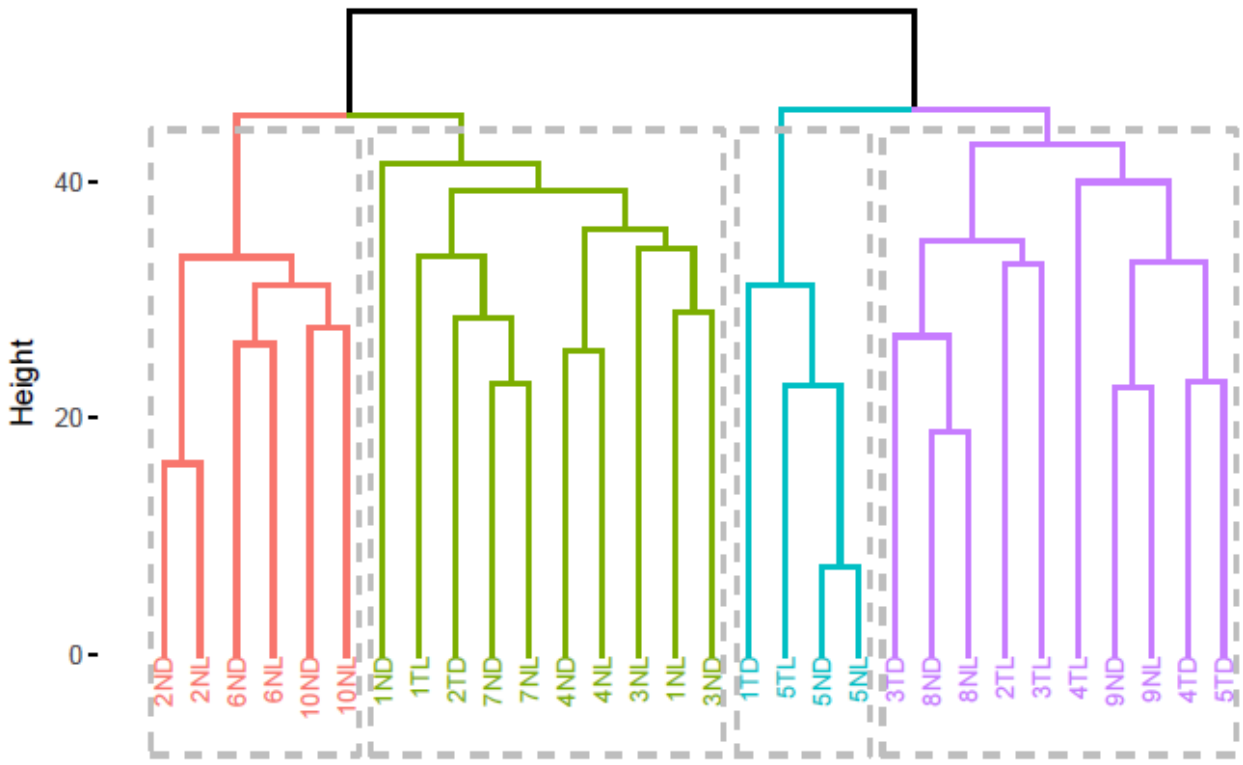

Ilustración 12: Representación Hierarchical k=4

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Normal Lobular	50%	40%	25%	20%
Normal Ductal	50%	40%	25%	20%
Tumoral Lobular	0%	10%	25%	40%
Tumoral Ductal	0%	10%	25%	40%

- AGNES :

Los valores óptimos obtenidos para el algoritmo AGNES han sido los mismos que para el algoritmo Hierarchical. Esto es normal debido a que ambos algoritmos trabajan de forma similar, por lo que sus valores de comparación son casi iguales. La representación de este se realiza de la misma forma que en el caso anterior, se genera un dendrograma con los grupos separados en colores y rodeados de un recuadro.

  - K=2

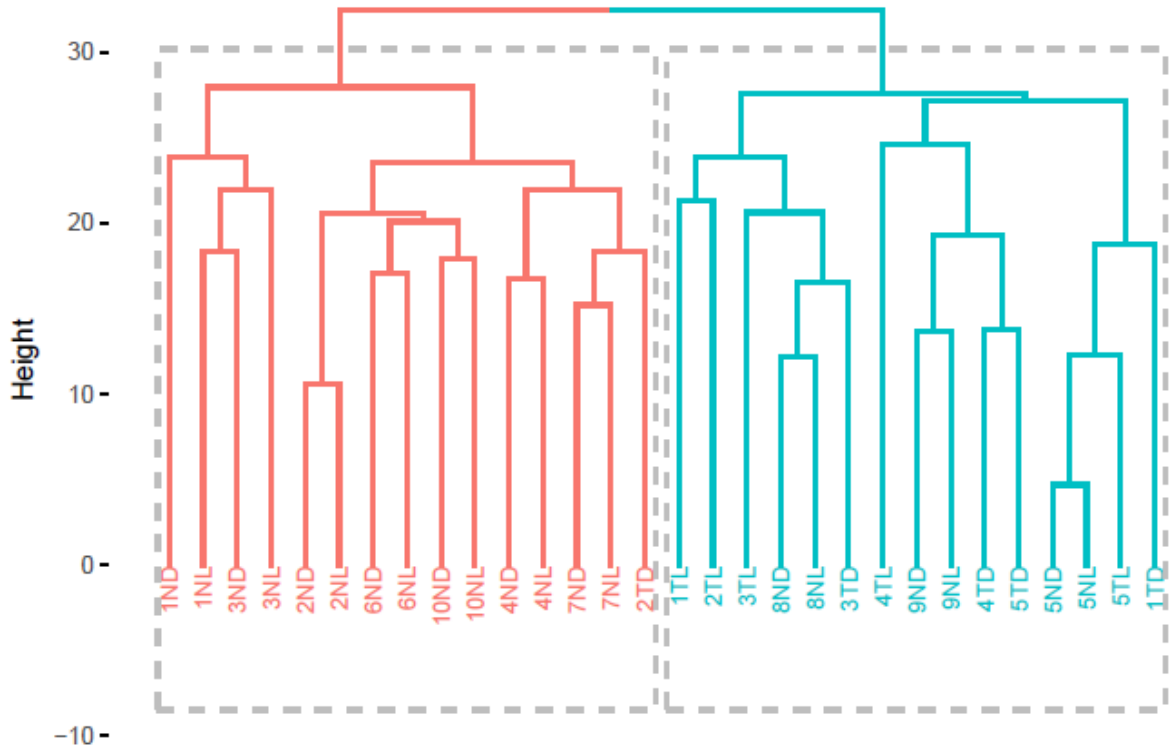

Ilustración 13: Representación AGNES k=2

	Grupo 1	Grupo 2
Normal Lobular	46,6%	20%
Normal Ductal	46,6%	20%
Tumoral Lobular	0%	33%
Tumoral Ductal	6,6%	27%

- K=4

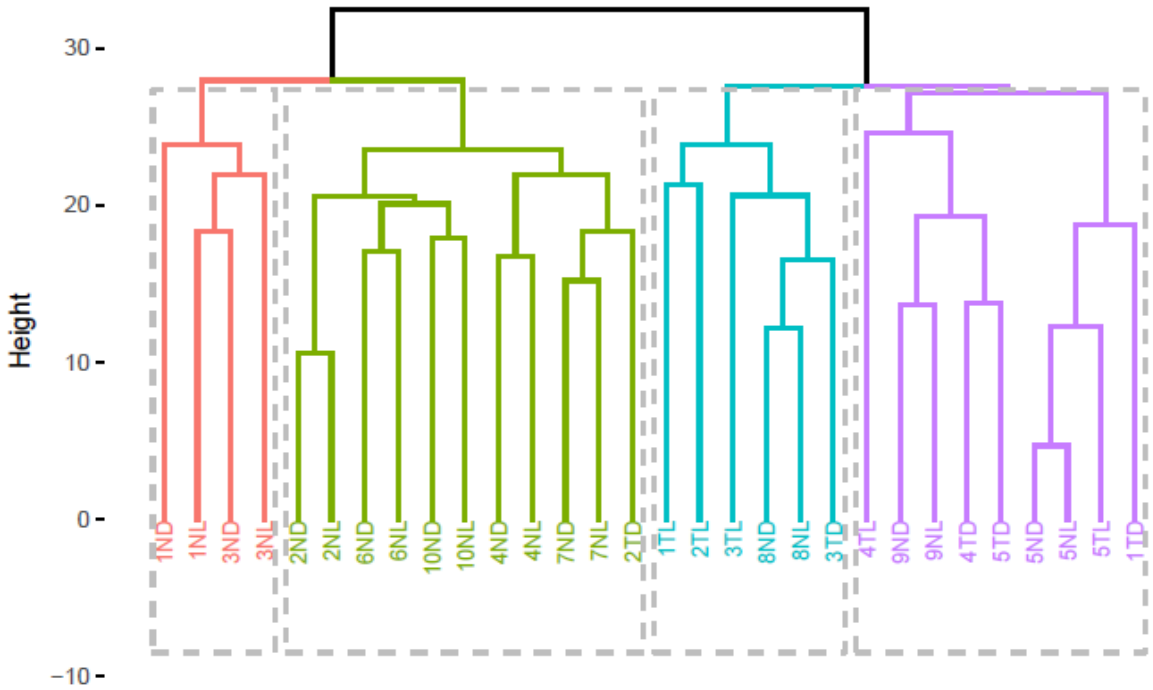

Ilustración 14: Representación AGNES k=4

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Normal Lobular	50%	45,5%	16,6%	22,2%
Normal Ductal	50%	45,5%	16,6%	22,2%
Tumoral Lobular	0%	0%	50%	22,2%
Tumoral Ductal	0%	9%	16,6%	33,3%

- DIANA:

Para el algoritmo DIANA los valores óptimos obtenidos han sido también 2 y 4. En este caso, a pesar de obtener los mismos valores de *k,* los valores comparativos no son similares a los anteriores, lo que muestra como este algoritmo funciona en sentido contrario que los otros dos algoritmos jerárquicos. Su representación si sigue la misma forma.

  - K=2

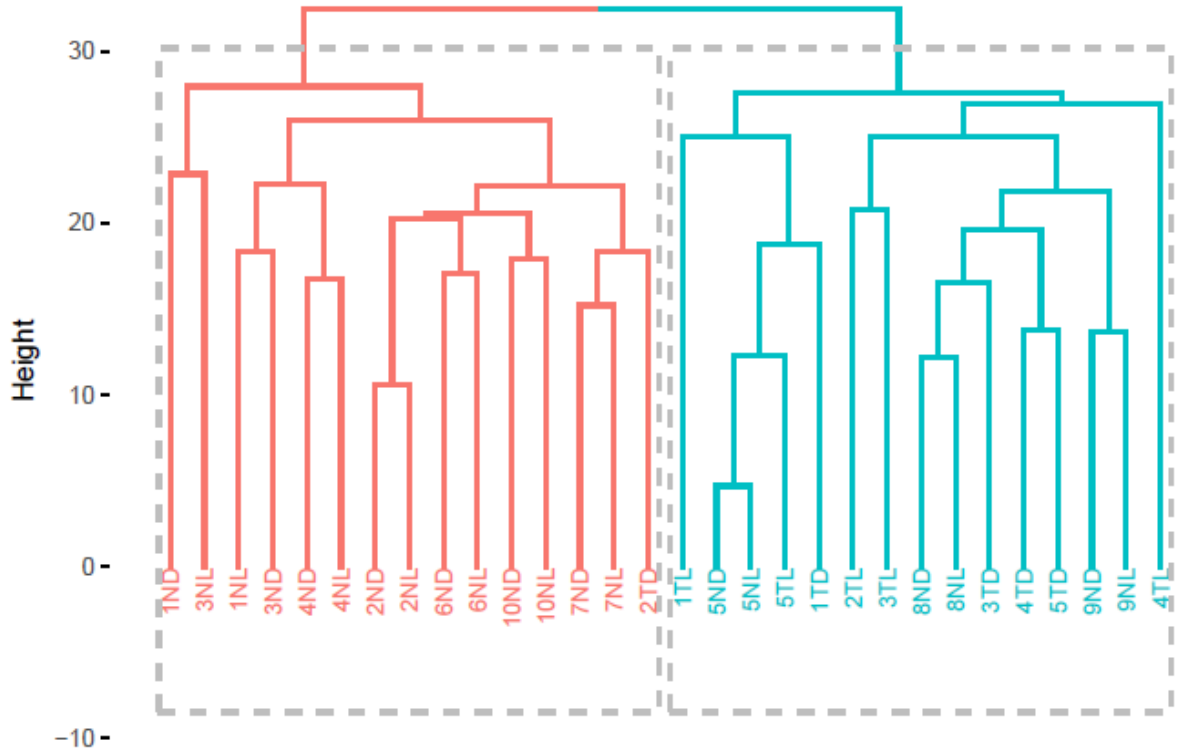

Ilustración 15: Representación DIANA k=2

	Grupo 1	Grupo 2
Normal Lobular	46,6%	20%
Normal Ductal	46,6%	20%
Tumoral Lobular	0%	33,3%
Tumoral Ductal	6,6%	26,6%

  - K=4

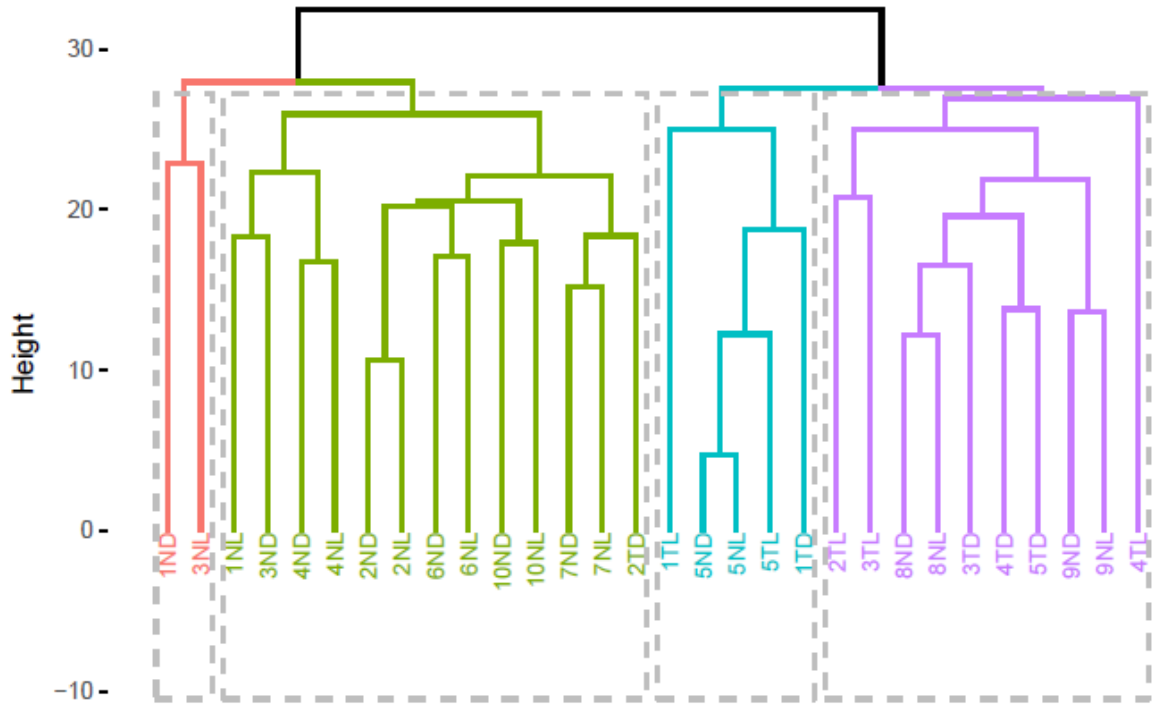

Ilustración 16: Representación DIANA k=4

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Normal Lobular	50%	46%	20%	20%
Normal Ductal	50%	46%	20%	20%
Tumoral Lobular	0%	0%	40%	30%
Tumoral Ductal	0%	8%	20%	30%

4.3. Algoritmos no jerárquicos

- K-means:

En este primer algoritmo basado en particiones se han obtenido valores óptimos de 2 y 3 grupos. En el análisis de los datos se presentó una gráfica basada en las componentes principales de este set de datos, sobre esta se van a representar las divisiones llevadas a cabo por este tipo de algoritmos. En la

mayoría de estas representaciones se recogen los grupos con forma elíptica que es una de las que mejor se adapta a su distribución.

- K=2

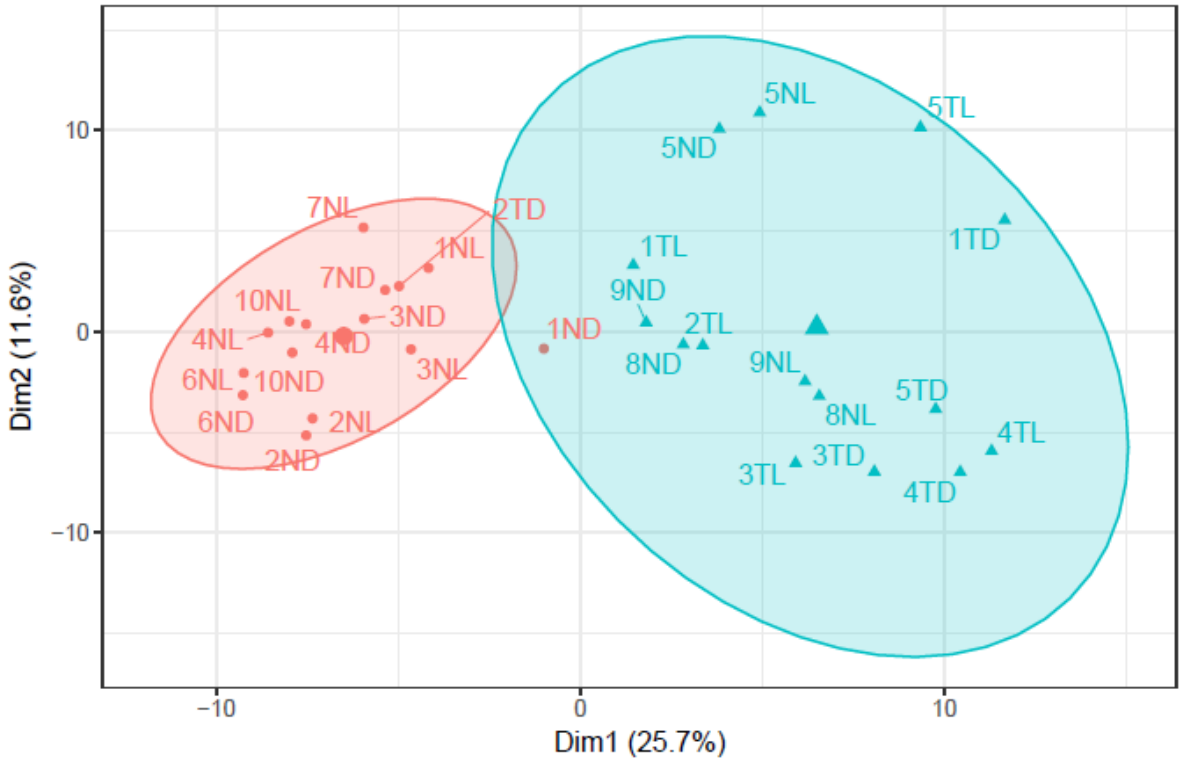

Ilustración 17: Representación K-means k=2

	Grupo 1	Grupo 2
Normal Lobular	46,6%	20%
Normal Ductal	46,6%	20%
Tumoral Lobular	0%	33,3%
Tumoral Ductal	6,6%	26,6%

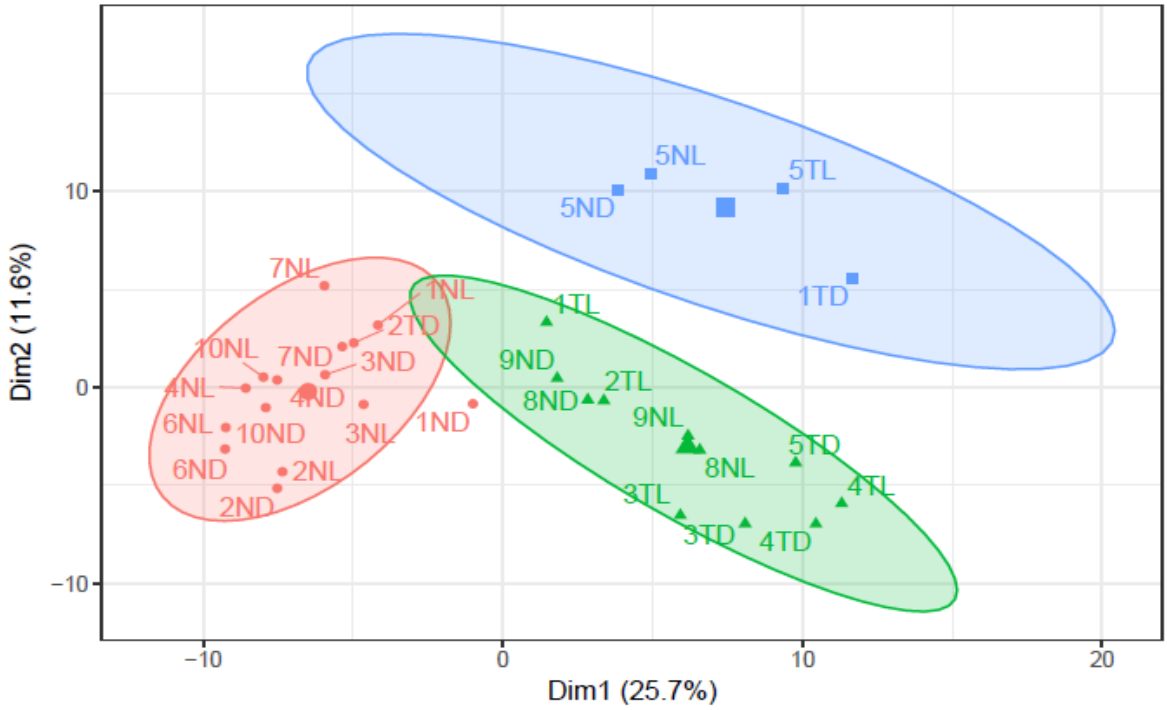

Ilustración 18: Representación K-means k=3

- K=3

	Grupo 1	Grupo 2	Grupo 3
Normal Lobular	46,6%	18%	25%
Normal Ductal	46,6%	18%	25%
Tumoral Lobular	0%	36%	25%
Tumoral Ductal	6,6%	28%	25%

- CLARA:

Para este algoritmo se obtienen valores óptimos de 2 y 3. A pesar de que los valores de comparación no son iguales al algoritmo anterior, la división de los datos si es similar.

- K=2

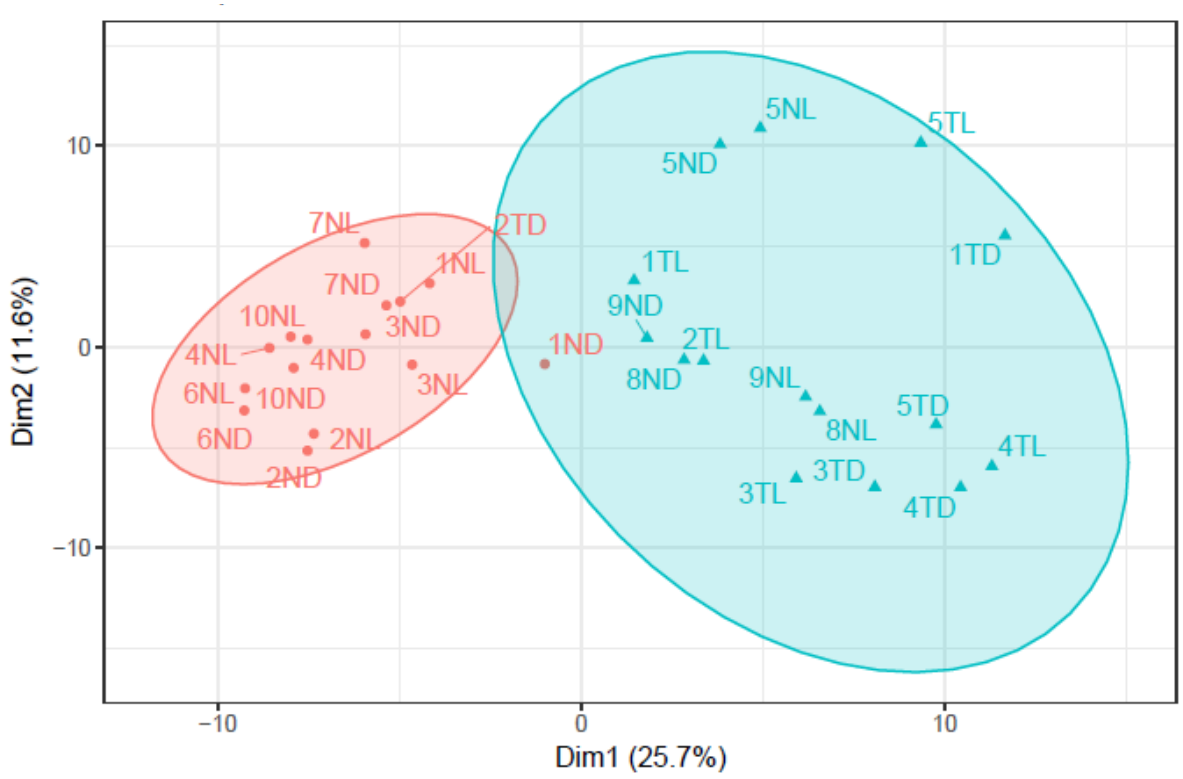

Ilustración 19: Representación CLARA k=2

	Grupo 1	Grupo 2
Normal Lobular	46,6%	20%
Normal Ductal	46,6%	20%
Tumoral Lobular	0%	33,3%
Tumoral Ductal	6,6%	26,6%

- K=3

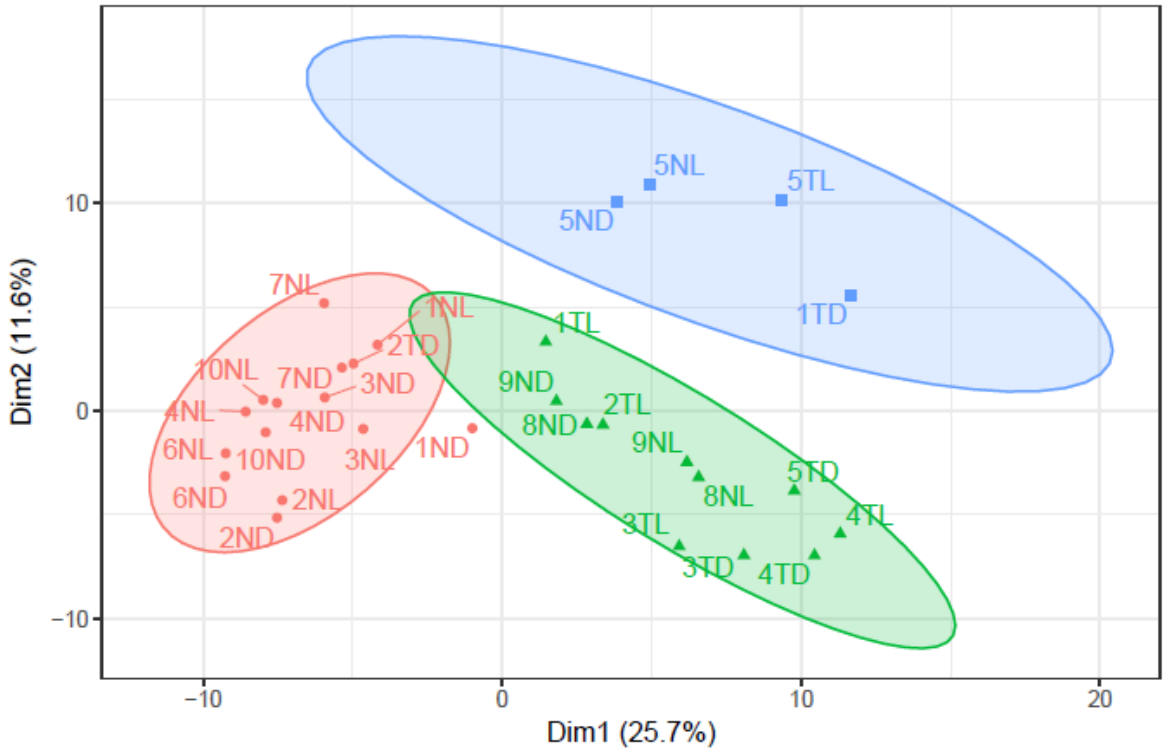

Ilustración 20: Representación CLARA k=3

	Grupo 1	Grupo 2	Grupo 3
Normal Lobular	46,6%	18%	25%
Normal Ductal	46,6%	18%	25%
Tumoral Lobular	0%	36%	25%
Tumoral Ductal	6,6%	28%	25%

- PAM:

El algoritmo PAM muestra diferencias con los anteriores a la hora de formar grupos, y obtiene de valores óptimos para *k* 3 y 5.

  - K=3

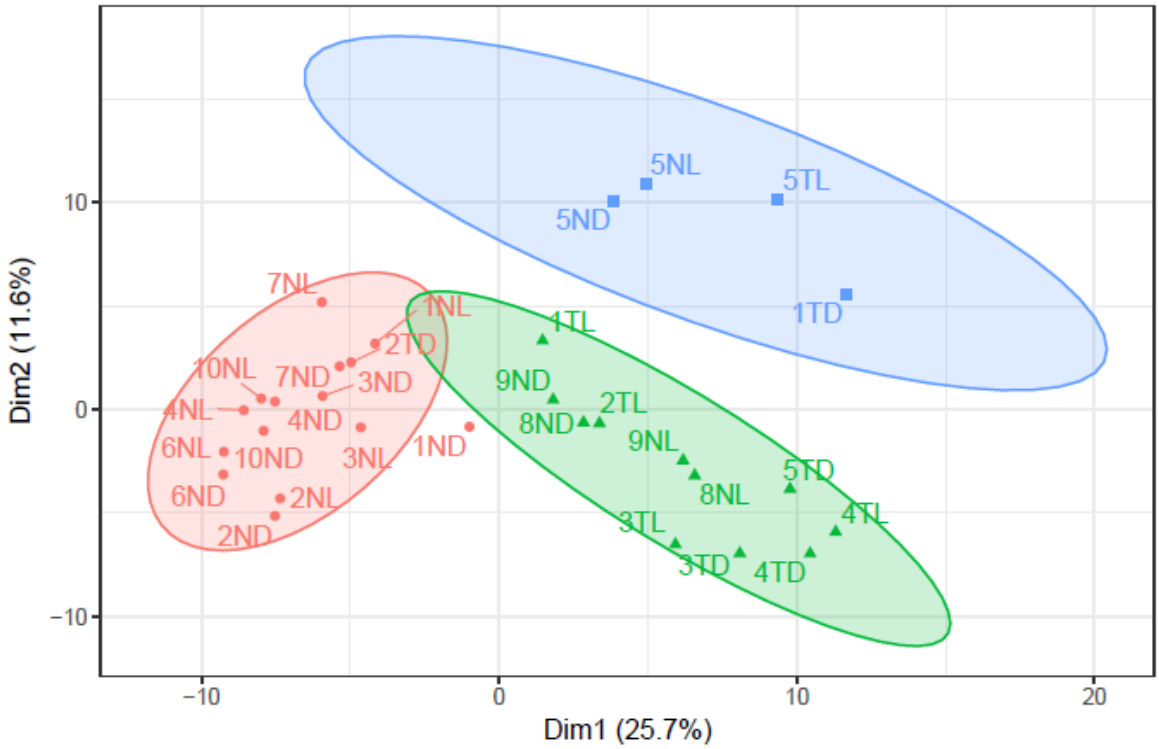

Ilustración 21: Representación PAM k=3

	Grupo 1	Grupo 2	Grupo 3
Normal Lobular	46,6%	18%	25%
Normal Ductal	46,6%	18%	25%
Tumoral Lobular	0%	36%	25%
Tumoral Ductal	6,6%	28%	25%

- K=5

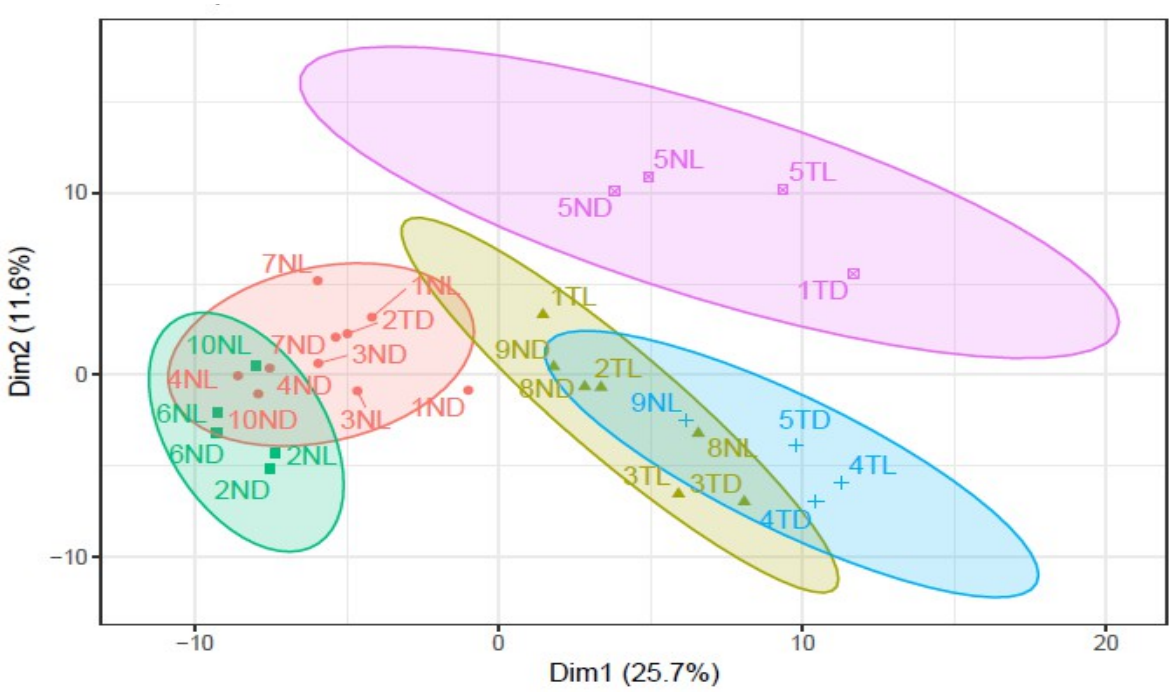

Ilustración 22: Representación PAM k=5

	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
Normal Lobular	60%	18%	14%	25%	25%
Normal Ductal	40%	18%	29%	0%	25%
Tumoral Lobular	0%	36%	43%	25%	25%
Tumoral Ductal	0%	28%	14%	50%	25%

- Model-Based:

En el caso del grupo de algoritmos model-based se han obtenido valores óptimos de 2 y 4. En las representaciones basadas en la función propia se indica además cual ha sido el modelo elegido como óptimo en cada caso, pero este tipo de selección no entra en las competencias de este proyecto.

  - K=2

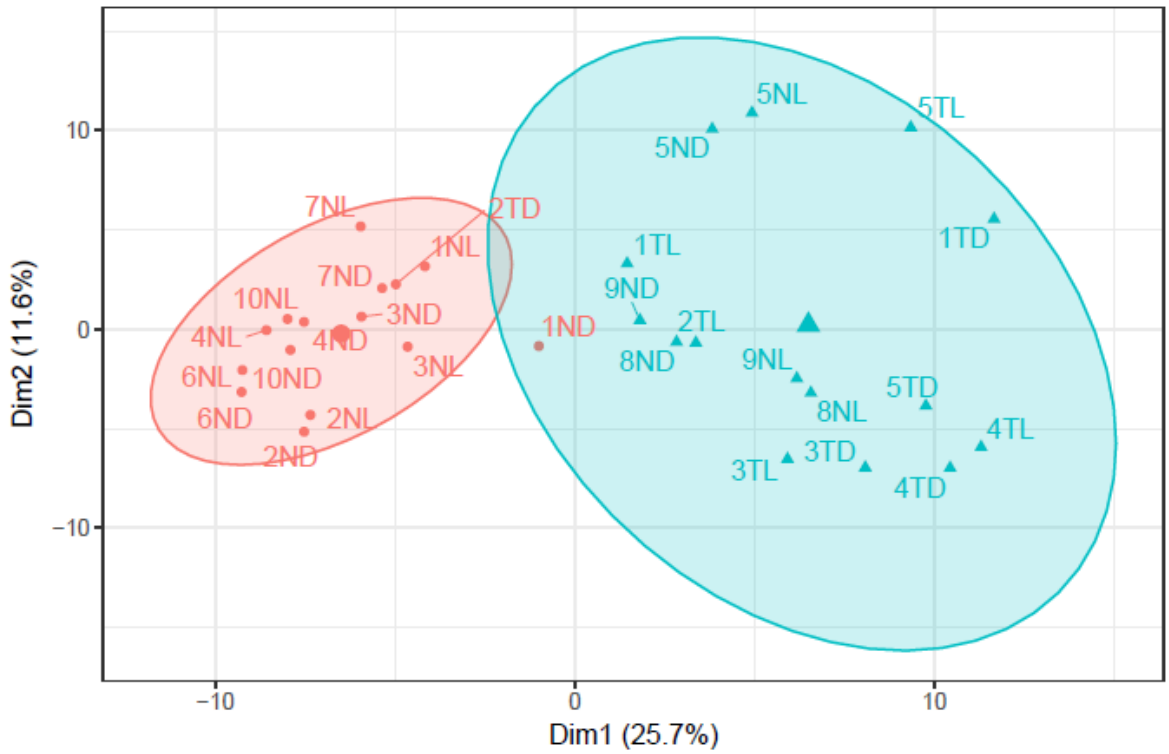

Ilustración 23: Representación Model-Based k=2

	Grupo 1	Grupo 2
Normal Lobular	46,6%	20%
Normal Ductal	46,6%	20%
Tumoral Lobular	0%	33,3%
Tumoral Ductal	6,6%	26,6%

- K=4

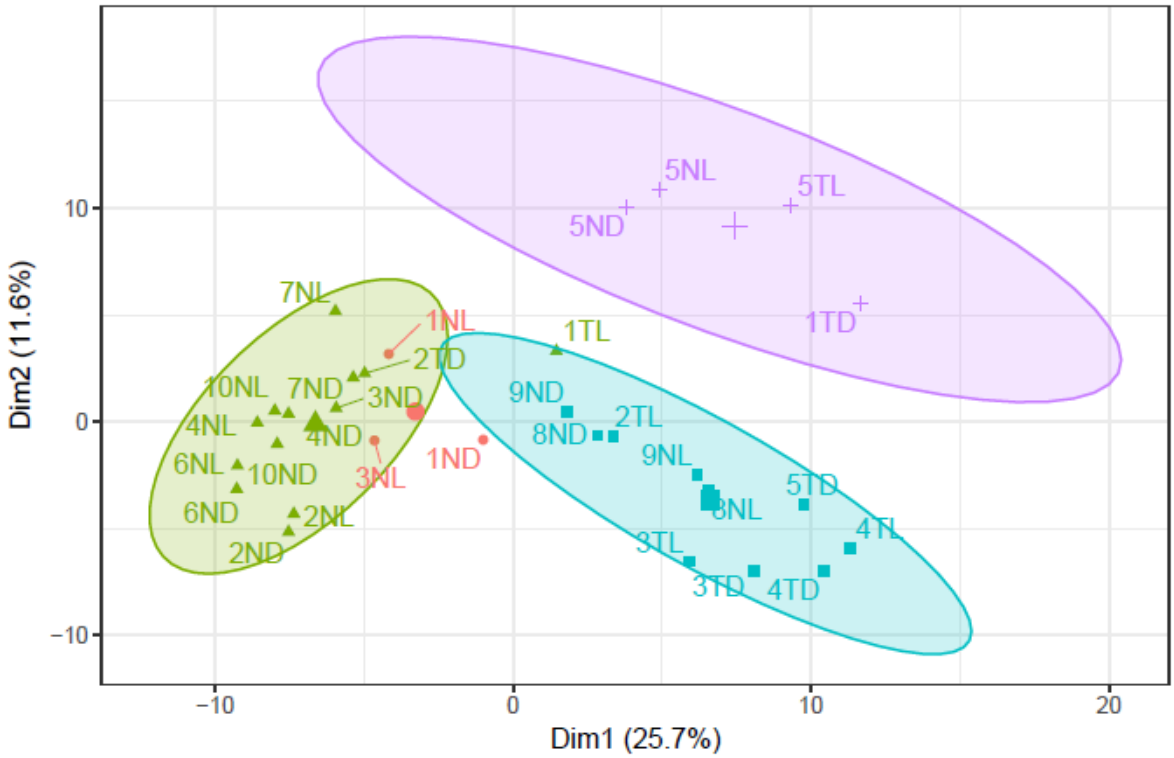

Ilustración 24: Representación Model-Based k=4

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Normal Lobular	38,4%	66,6%	20%	25%
Normal Ductal	46,4%	33,3%	20%	25%
Tumoral Lobular	7,6%	0%	30%	25%
Tumoral Ductal	7,6%	0%	30%	25%

Durante esta sección se han obtenido los mejores resultados para cada algoritmo, además se han representado y comparado para facilitar la decisión sobre su eficiencia. En la próxima sección se utilizará de nuevo la misma función comparativa, pero en esta ocasión se implementará para todos los algoritmos con sus valores óptimos, de forma que se conseguirá una medida de eficacia entre todos, pudiendo determinar los mejores para este set de datos.

5. Análisis comparativo de los métodos

En la sección previa se ha utilizado para obtener los valores óptimos de división de cada algoritmo la función *optCluster()*, pero no se ha indicado la manera que tiene esta de comparar los resultados de los diferentes casos. Antes de realizar el análisis final utilizando la función se va a proceder a describir las diferentes comparativas que utiliza con el fin de facilitar el entendimiento del proceso.

La función basa su análisis comparativo en 3 criterios principales, que abarcan a su vez un total de 9 variables. Cada una de las variables tomará un valor para cada caso, para cada valor de *k* en cada algoritmo. El caso que obtenga el menor valor para esa variable se considerará el mejor para era variable, mientras que el que obtenga los menores valores para todas las variables en conjunto será el mejor en general, es decir, el más eficiente trabajando con los datos [27].

Las nueve variables indicadas se agrupan por criterios de la siguiente forma:

- Criterio de internalidad:

    - Conectividad: esta variable busca que para cada observación su vecino más próximo se encuentre dentro del mismo cluster según su distancia Euclídea, de forma que si se cumple, el valor será 0. Los valores de esta variable están entre 0 e infinito, siendo el valor óptimo 0.
    - Ancho de la silueta: esta variable se basa en los valores en el rango [-1,1] de cada observación, utilizando el grado de confianza de que esa observación pertenezca al grupo donde está incluida. El valor óptimo de esta variable es el 1, mientras que el peor valor posible sería el -1.
    - Índice de Dunn: este índice se basa en el ratio entre la distancia Euclídea mínima entre observaciones de diferentes grupos y el

diámetro máximo de los cluster. Sus valores se mueven entre 0 e infinito, de forma que a mayor valor, mejor resultado.

- Criterio de estabilidad:

  - APN: esta variable mide la proporción de muestras que se colocan en diferentes grupos cuando se realiza el agrupamiento eliminando una columna de los datos. Sus valores están entre 0 y 1, siendo los cercanos a 0 los mejores.
  - AD: esta variable mide la distancia media de las observaciones agrupadas a partir de los datos completos comparandola con la distancia media obtenida en el mismo proceso pero eliminando una columna de los datos. Los valores están entre 0 e infinito debiendo ser lo menores posibles para mejorar la eficiencia.
  - ADM: en este caso el proceso es el mismo, se miden las medias de distancia entre las muestras de cada grupo y el centro de este y se comparan con las obtenidas a partir de los datos menos una columna. Los valores se mueven entre 0 e infinito, siendo los valores más bajos los más óptimos.
  - FOM: esta variable mide la diferencia de muestras presentes en cada grupo cuando se elimina una columna. Sus valores están entre 0 e infinito, siendo los cercanos a 0 los mejores.

- Criterio biológico:

  - Índice Biológico de Homogeneidad (BHI): esta variable utiliza los genes de la matriz de expresión para para definir biológicamente los grupos creados, para ello crea unas clases donde se agrupan los genes, si las clases entre genes dentro de cada grupo coinciden se asigna un valor de 1, sino se asigna 0. Los valores se mueven entre 0 y 1, siendo los valores más altos los que indican mayor homogeneidad.

- Índice Biológico de Estabilidad (BSI): en este caso se utilizan también los genes, pero el proceso es comparar los grupos formados con todos los datos frente a los grupos formados al quitar una columna de los datos. Los valores están entre 0 y 1, siendo mejores los cercanos a 1.

Una vez explicados los criterios que se utilizan para comparar los diferentes algoritmos se va a aplicar la función *optCluster()* con todos los algoritmos elegidos, utilizando también el rango de valores óptimos obtenido en la sección anterior.

El resultado de esta comparativa, así como de las referidas al porcentaje de agrupación correcta en función de las clases, se tratarán en la sección de resultados, de forma que se facilitará la interpretación de estos al agruparlos.

Además de la comparación interna realizada por la función *optCluster()* se planteó el uso de algún tipo de comparativa externa, utilizando las clases obtenidas para comparar los grupos creados con las clases originales. Existen multitud de métodos utilizados habitualmente en algoritmos de clasificación que pueden ser adaptados a algoritmos de agrupamiento, como pueden ser el Indice de Rand, F-Measure o el Indice de Fowlkes-Mallows [31]. A pesar de existir muchos métodos, no se suele recomendar su uso [32] debido entre otras características, a que para aplicarlos se requiere que el número de clases sea el mismo que el número de grupos, y sin esta condición el funcionamiento de estos métodos no es correcto.

Teniendo en cuenta estos factores y atendiendo a la necesidad de observar como ha agrupado las clases cada algoritmo, se han planteado unas tablas donde se indica el porcentaje de aparición de una clase dentro de un grupo. Estas tablas se utilizarán como apoyo para determinar si los mejores algoritmos obtenidos de la comparativa interna son de verdad útiles para tratar este tipo de datos.

6. Resultados

En primer lugar se muestran los resultados de la función *optCluster()* con todos los algoritmos. Esta función da como resultado un listado de todas las posibles combinaciones de algoritmo con *k ,* de forma que para cada una se muestran los valores de cada criterio de comparación. Estos datos no se muestran aquí debido a su volumen, pero se pueden observar en el archivo .Rmd adjunto.

Para poder valorar los resultados interesa otra parte de la salida de esta función, donde se muestra una lista ordenada de los algoritmos cuyos valores han sido mejores en la comparación. Esta lista se muestra a continuación.

```
## The overall optimal clustering method and number of clusters is:
##      diana-2
##
## The optimal list is:
##         diana-2 agnes-2 diana-4 hierarchical-2 kmeans-2 clara-2 diana-3 diana-5
##         kmeans-5 pam-3 kmeans-4 clara-3 pam-4 agnes-3 clara-4 hierarchical-3 agnes-4
##         hierarchical-4 agnes-5 pam-5 hierarchical-5 kmeans-3 model-4 clara-5
##         model-2 model-5 pam-2
##
##   Algorithm:   CE
##   Distance:    Spearman
##   Score:       63.1634
##   Iterations:  187
```

Ilustración 25: Resultado de la comparativa final

Con el resultado de la función queda claro que el algoritmo DIANA destaca sobre el resto. Los resultados sobre esta comparación se valorarán en la sección de conclusiones.

En este punto se pueden utilizar las tablas realizadas con cada algoritmo para valorar cómo han agrupado los datos los mejores algoritmos obtenidos en la función *optCluster()*. Para ello se resumen las agrupaciones de los 5 primeros resultados de la lista. A continuación se muestra una tabla unificada con los mejores resultados.

Algoritmo	Clase	Grupo 1	Grupo 2	Grupo 3	Grupo 4
DIANA k=2	Normal Lobular	46,6%	20%		
	Normal Ductal	46,6%	20%		
	Tumoral Lobular	0%	33,3%		
	Tumoral Ductal	6,6%	26,6%		
AGNES k=2	Normal Lobular	46,6%	20%		
	Normal Ductal	46,6%	20%		
	Tumoral Lobular	0%	33%		
	Tumoral Ductal	6,6%	27%		
DIANA k=4	Normal Lobular	50%	46%	20%	20%
	Normal Ductal	50%	46%	20%	20%
	Tumoral Lobular	0%	0%	40%	30%
	Tumoral Ductal	0%	8%	20%	30%
Hierarchical k=2	Normal Lobular	44%	21%		
	Normal Ductal	44%	21%		
	Tumoral Lobular	6%	29%		
	Tumoral Ductal	6%	29%		
K-means k=2	Normal Lobular	46,6%	20%		

	Normal Ductal	46,6%	20%		
	Tumoral Lobular	0%	33,3%		
	Tumoral Ductal	6,6%	26,6%		

- DIANA-K=2

Grupo 1: Igualmente representadas las clases Normal Lobular y Normal Ductal

Grupo 2: La mayoría de las muestras de las clases tumorales agrupadas.

- AGNES-K=2

Grupo 1: Igualmente representadas las clases Normal Lobular y Normal Ductal

Grupo 2: La mayoría de las muestras de las clases tumorales agrupadas.

- DIANA-K=4

Grupo 1: Igualmente representadas las clases Normal Lobular y Normal Ductal con mínima cantidad de muestras

Grupo 2: Igualmente representadas las clases Normal Lobular y Normal Ductal con muchas muestras.

Grupo 3: Igualdad en la representación con más de Tumoral Lobular.

Grupo 4: Mayoría de muestras tumorales.

- Hierarchical-K=2

Grupo 1: Mayoría de muestras normales, igualmente representadas entre ellas.

Grupo 2: Similitud entre todas las clases, mayoría poco destacable de muestras tumorales.

- Kmeans-K=2

Grupo 1: Igualmente representadas las clases Normal Lobular y Normal Ductal

Grupo 2: La mayoría de las muestras de las clases tumorales agrupadas aquí.

A partir de todos los resultados obtenidos se pueden llegar a varias conclusiones claras, que se formularán en la siguiente sección.

7. Conclusiones

Esta sección se va a dedicar a explicar las ideas que se pueden extraer de los resultados obtenidos, así como plantear ciertos aspectos relacionados con los algoritmos.

En primer lugar la comparación realizada por la función *optCluster()* ha dejado claro que el mejor algoritmo ha sido el DIANA , siendo su versión con *k*=2 más eficiente.

Otra aportación constatada es que los algoritmos jerárquicos trabajan de forma más eficiente con este tipo de datos, ya que los 4 mejores algoritmos pertenecen a este grupo.

El último dato que se puede extraer de esta comparación es que 5 de los 6 primeros algoritmos tienen en común que *k*=2, por lo que se puede pensar que la división en 2 grupos es la que más favorece a estos datos.

Las tablas generadas a partir de los resultados de los principales algoritmos dejan claro que estos algoritmos diferencian mucho mas fácilmente entre estado de la célula (normal o tumoral) que entre el tipo de esta (lobular o ductal), lo que confirmaría el planteamiento anterior de que la división más favorable es en dos grupos. También queda claro que ninguno es realmente eficiente a la hora de agrupar las diferentes clases, por lo que a la vista de estos datos no se puede recomendar su uso sobre matrices de expresión génica, al menos no para tomar decisiones relevantes. Sí podrían usarse para apoyar diagnósticos sobre la tumoralidad de ciertas células, ya que parece que diferencian parcialmente bien entre célula normal y tumoral.

A parte de las conclusiones sobre cómo han funcionado los algoritmos sobre los datos, cabe destacar ciertos problemas que se han generado al aplicarlos. Uno de estos problemas ha sido que el gran tamaño de la matriz de datos original conllevaba tiempos inasequibles para la aplicación de algunos algoritmos, así como la generación de objetos demasiado grandes. Otro de los problemas que han surgido es que algunos algoritmos no han sido capaces de trabajar con los datos, ya sea porque no pueden trabajar con datos de este tipo o bien porque no podían dividirlos ni siquiera en dos grupos. Estos problemas han requerido de la modificación de la planificación inicial, pero debido a que se planearon las tareas con margenes de tiempo amplios se han podido solventar sin mayor problema.

En este punto cabe añadir ciertos factores a modificar o implementar en futuros proyectos que sigan una línea similar a este. La función *optCluster()* limita los algoritmos que pueden utilizarse, por lo que sería recomendable plantear mecanismos de comparación similares a los desarrollados por esta para otros algoritmos. Un factor clave a la hora de desarrollar algunos algoritmos con grupos de datos grandes es el tiempo que requieren de procesado, sería por ello conveniente utilizar equipos de mayor potencia y que puedan abarcar los objetos de grandes volumenes que se generan, de esta forma se podría mejorar el desarrollo del proyecto.

En el desarrollo del proyecto se ha podido seguir la metodología que se planteó en un principio con pequeñas modificaciones indicadas anteriormente, pero de forma general se puede decir, que se han cumplido los objetivos iniciales sin grandes desviaciones y que los resultados obtenidos a pesar de no ser los que se buscaban dejan claras las ideas que se querían comprobar.

Para acabar y a modo de resumen se puede concluir que el algoritmo DIANA y por lo general los algoritmos jerárquicos son los mejores para trabajar con estos datos, pero no se recomendaría su uso debido a que la eficacia no es suficientemente alta.

8. Glosario

- Algoritmo: Conjunto de procesos que derivan en el tratamiento de un objeto para modificarlo u obtener información de él.
- Aprendizaje automático: Metodología aplicada a procesos informáticos que permite mejorar la eficiencia de un proceso sin intervención humana.
- Aprendizaje no supervisado: Rama del aprendizaje automático que se basa en permitir a los algoritmos trabajar sin incorporar condiciones por parte del humano.
- Grupo o cluster: cada una de las divisiones que lleva a cabo un algoritmo de agrupamiento sobre un set de datos.
- Matriz de expresión genética: Matriz de valores, donde cada valor indica el nivel de expresión de un probe en una muestra.
- Probe: Fragmento de DNA o RNA que hibrida con la muestra y puede posteriormente ser medido.

9. Bibliografía

1-.Hu, C. W., Kornblau, S. M., Slater, J. H., & Qutub, A. A. (2015).*Progeny Clustering*: A Method to Identify Biological Phenotypes.*Scientific Reports*,*5*, 12894. http://doi.org/10.1038/srep12894 .

2-.Autor: Nathan Lawlor (19/01/2018). Título: A Guide to multiClust . Nombre de la página: Bioconductor. Disponible en: http://www.bioconductor.org/packages/3.7/bioc/vignettes/multiClust/inst/doc/multiClust.html . Fecha consulta: 03/03/2018

3-.Yin, L., Huang, C.-H., & Ni, J. (2006). Clustering of gene expression data: performance and similarity analysis.*BMC Bioinformatics*,*7*(Suppl 4), S19. http://doi.org/10.1186/1471-2105-7-S4-S19 .

4-.Zhao, W., Zou, W., & Chen, J. J. (2014). Topic modeling for cluster analysis of large biological and medical datasets.*BMC Bioinformatics*,*15*(Suppl 11), S11. http://doi.org/10.1186/1471-2105-15-S11-S11.

5-.Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghiren, E., Ameh, F., … Adebiyi, E. (2016). Clustering Algorithms: Their Application to Gene Expression Data.*Bioinformatics and Biology Insights*,*10*, 237–253. http://doi.org/10.4137/BBI.S38316.

6-. Autor: Kimbarly Coffey (13/08/2016). Título: K-means clustering for customer segmentation: a practical guide. Nombre de la página: Kimberly Coffey Blog. Disponible en: http://www.kimberlycoffey.com/blog/2016/8/k-means-clustering-for-customer-segmentation. Fecha de consulta: 15/05/2018

7-. Zhao, W., Zou, W., & Chen, J. J. (2014). Topic modeling for cluster analysis of large biological and medical datasets.*BMC Bioinformatics*,*15*(Suppl 11), S11. http://doi.org/10.1186/1471-2105-15-S11-S11

8-.Sul, W. J., Cole, J. R., Jesus, E. da C., Wang, Q., Farris, R. J., Fish, J. A., & Tiedje, J. M. (2011). Bacterial community comparisons by taxonomy-supervised analysis independent of sequence alignment and clustering.*Proceedings of the National Academy of Sciences of the United States of America*,*108*(35), 14637–14642. http://doi.org/10.1073/pnas.1111435108

9-. Autor: Michael Sekula. Título: optCluster documentation Nombre de la página: rdocumentation. Disponible en: https://www.rdocumentation.org/packages/optCluster/versions/1.1.0/topics/optCluster. Fecha de consulta: 30/03/2018.

10-.Mackay, A., Weigelt, B., Grigoriadis, A., Kreike, B., Natrajan, R., A'Hern, R., … Reis-Filho, J. S. (2011). Microarray-Based Class Discovery for Molecular Classification of Breast Cancer: Analysis of Interobserver Agreement.*JNCI Journal of the National Cancer Institute*,*103*(8), 662–673. http://doi.org/10.1093/jnci/djr071

11-.Wang, Y., & Li, T.-Q. (2013). Analysis of Whole-Brain Resting-State fMRI Data Using Hierarchical Clustering Approach.*PLoS ONE*,*8*(10), e76315. http://doi.org/10.1371/journal.pone.0076315

12-.Sirinukunwattana, K., Savage, R. S., Bari, M. F., Snead, D. R. J., & Rajpoot, N. M. (2013). Bayesian Hierarchical Clustering for Studying Cancer Gene Expression Data with Unknown Statistics.*PLoS ONE*,*8*(10), e75748. http://doi.org/10.1371/journal.pone.0075748

13-.Autor: Perceptive Analitics. (18/12/2017) Título: How to perform Hierarchical Clustering using R. Nombre de la página: r-bloggers. Disponible en: https://www.r-bloggers.com/how-to-perform-hierarchical-clustering-using-r/. Fecha de consulta: 01/04/2018

14-.Procedencia: Documentación de R Título: Agglomerative Nesting (Hierarchical Clustering). Nombre de la página: stat.ethz Disponible en: https://stat.ethz.ch/R-manual/R-patched/library/cluster/html/agnes.html. Fecha de consulta: 02/04/2018

15-.Procedencia: Documentación de R Título: Divisive Analisis Clustering. Nombre de la página: stat.ethz Disponible en:https://stat.ethz.ch/R-manual/R-patched/library/cluster/html/diana.html. Fecha de consulta: 02/04/2018

16-.Kakushadze, Z., & Yu, W. (2017). *K-means and cluster models for cancer signatures. *Biomolecular Detection and Quantification*, *13*, 7–31. http://doi.org/10.1016/j.bdq.2017.07.001

17-..Autor: Andrea Trevino. (12/06/2016) Título: Introduction to k-means clustering. Nombre de la página: datascience. Disponible en: https://www.datascience.com/blog/k-means-clustering. Fecha de consulta: 06/04/2019

18-.Procedencia: Wikipedia Título: k-medoids. Nombre de la página: wikipedddia Disponible en:https://es.wikipedia.org/wiki/K-medoids. Fecha de consulta: 06/04/2018

19-.Hosseinzadeh, F., Ebrahimi, M., Goliaei, B., & Shamabadi, N. (2012). Classification of Lung Cancer Tumors Based on Structural and Physicochemical Properties of Proteins by Bioinformatics Models. *PLoS ONE*, *7*(7), e40017. http://doi.org/10.1371/journal.pone.0040017

20-.Broin, P. Ó., Smith, T. J., & Golden, A. A. (2015). Alignment-free clustering of transcription factor binding motifs using a genetic-k-medoids approach. *BMC Bioinformatics*, *16*, 22. http://doi.org/10.1186/s12859-015-0450-2

21-.Procedencia: Documentación de R Título: Clustering Large Applications. Nombre de la página: stat.ethz Disponible en:https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/clara.html. Fecha de consulta: 06/04/2018

22-.Autor: kassambara . (04/09/2017) Título: Clustering Large Applications. Nombre de la página: sthda. Disponible en: http://www.sthda.com/english/articles/27-partitioning-clustering-essentials/89-clara-clustering-large-applications/. Fecha de consulta: 07/04/2018

23-.Noel-MacDonnell, J. R., Usset, J., Goode, E. L., & Fridley, B. L. (2018). Assessment of data transformations for model-based clustering of RNA-Seq data. *PLoS ONE*, *13*(2), e0191758. http://doi.org/10.1371/journal.pone.0191758

24-.Pan, W., Lin, J., & Le, C. T. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, *3*(2), research0009.1–research0009.8.

25-.Autor: Sean Davis (10/05/2016) Título: GEOquery package Nombre de la página: rdocumentation Disponible en:https://www.rdocumentation.org/packages/GEOquery/versions/2.38.4. Fecha de consulta: 01/04/2018

26-.Autor: Rafael Irizarry Título: affy package Nombre de la página: rdocumentation Disponible en:https://www.rdocumentation.org/packages/affy/versions/1.50.0. Fecha de consulta: 01/04/2018

27-.Sekula, M., Datta, S., & Datta, S. (2017). optCluster: An R Package for Determining the Optimal Clustering Algorithm. *Bioinformation*, *13*(3), 101–103. http://doi.org/10.6026/97320630013101

28-.Autor: Bioconductor package mantainer Título: nsFilter Nombre de la página: rdocumentation Disponible en: https://www.rdocumentation.org/packages/genefilter/versions/1.54.2/topics/nsFilter. Fecha de consulta: 06/04/2018

29-.Procedencia: STDHA Título:Factoextra R Package: Easy Multivariate Data Analyses and Elegant Visualization. Disponible en: http://www.sthda.com/english/wiki/factoextra-r-package-easy-multivariate-data-analyses-and-elegant-visualization. Fecha de consulta: 16/04/2018.

30-.Procedencia: Rstudio-pubs Título: Heatmaps y aprendizaje no supervisado Disponible en: https://rstudio-pubs-static.s3.amazonaws.com/310338_fc5c392188a14507b6325570c6a5e821.html. Fecha de consulta: 10/04/2018.

31-.Wagner, Silke & Wagner, Dorothea. (2007). Comparing Clusterings - An Overview. Technical Report 2006-04.

32-.Steinley D. (2004) Properties of the Hubert-Arabie adjusted Rand index. Psychological Methods. Sep;9(3):386-96. http://doi.org/10.1037/1082-989X.9.3.386