Aplicación y comparativa de cuatro modelos de clustering para datos GTEx

Victoria López Sánchez
Trabajo de Fin de Máster

Carles Ventura Royo
Daniel Fernández Martínez
Enero 2019

Tabla informativa TFM

Título del trabajo	Aplicación y comparativa de cuatro modelos de clustering para datos GTEx
Nombre del autor	Victoria López Sánchez
Nombre del consultor	Carles Ventura Royo
Nombre del PRA	Daniel Fernández Martínez
Fecha de entrega	02/02/2012
Titulación	Máster en Bioinformática y Bioestadística
Área del trabajo final	Genomica computacional
Idioma del trabajo	Español
Palabras clave	Clustering, ARN-seq y GCN

Resumen

Las redes de coexpresión génica (GCN) son una herramienta fundamental para caracterizar genes mediante el estudio de sus patrones de correlación. Las GCN son modelos gráficos no dirigidos que constan de un conjunto de nodos y aristas, que representan los genes y las interacciones entre los genes respectivamente. Los genes se agrupan en función de su similitud formando módulos (grupos). Se asume que los genes que se encuentran en el mismo módulo están relacionados con un fenotipo determinado, una enfermedad o tienen una función similar. Los métodos habituales para generar GCN usan algoritmos de clustering basados en distancias o correlaciones para generar los diferentes módulos de genes. Estos procedimientos heurísticos tienen dos principales ventajas, su construcción es intuitiva y un corto tiempo de computación, sin embargo, tiene como limitación la falta de base estadística sobre las que aplicar inferencia para una población a partir de una muestra. Las limitaciones de los procedimientos heurísticos crea la necesidad de estudiar y evaluar dichos algoritmos de clustering justificando su uso o posibilitando el descubriendo de una posible mejora de dicho método para optimizar el agrupamiento de genes. Una alternativa a procedimientos heurísticos son los métodos basados en probabilidades (agrupamiento probabilístico). Es decir, se asume que el método de clasificación es generado por una distribución de probabilidad subyacente. Una de las características de esta metodología es que el clustering es 'fuzzy', es decir, cada observación tiene una probabilidad de pertenecer a un grupo (cluster). Uno de los modelos más populares del agrupamiento probabilístico es el modelo de mezcla finito (*finite mixture model*, FMM).

En el presente TFM se estudian y evalúan cuatro diferentes algoritmos de clustering: método coexp (formado por WGCNA y k-means), k-means convencional y dos modelos de mezcla finitos. En primer lugar se implementan todos los modelos de clustering, se busca el número de clusters óptimo en base a diferentes criterios estadísticos como BIC, AIC y el método de codo y por último, las particiones generadas por dichos métodos se visualizan y se comparan entre sí con la ayuda de una serie de métricas (o medidas) de validación externa. Se han usado medidas basadas en el conteo de pares (*counting based measures*) como el índice de Rand ajustado (*Adjusted Rand Index*, ARI) y en información teórica (*information theoretic-based measures*) como la variación de la información normalizada (*Normalized Variation of Information*, NVI) y distancia de información normalizada (*Normalized Distance of Information*, NID).

Abstract

Constructing a gene co-expression network (GCN) is an effective way to characterize the correlation patterns among genes. A gene co-expression network is an undirected graph, where each node corresponds to a gene, and each edge connects a pair of genes that are significantly correlated. Genes are grouped according to their similarity forming modules and it is assumed that the genes found in the same module are related to a specific phenotype, a disease or have a similar function. Common methods for constructing gene coexpression network use clustering algorithms based on distances or correlations to generate different gene modules. These heuristic procedures have two main advantages, their construction is intuitive and short computation time, however, has as limitations the lack of statistical base on which to apply inference for a population from a sample. Heuristic procedures limitations involve the evaluation of clustering algorithms justifying their use or enabling the discovery of a possible improvement of common gene coexpression networks methods.

A choice to heuristic procedures is the probability-based methods (probabilistic clustering). In other words, it is assumed that classification method is generated by underlying probabilistic distribution. An essential feature of those methods is clustering is "fuzzy", each observation has a possibility belonging to a group (clusters). The models more popular for probabilistic clustering are finite mixture models, FMM. Finite mixture models are probabilistic models that represent subpopulations within a general population without the need to identify the subpopulation to which an individual observation of the total dataset belongs.

Presently TFM, it has been studied and evaluated four different clustering algorithms: coexp method (WGCNA + k-means), k-means conventional and two mixture models. First of all, it will implement all clustering models, search optimum clusters numbers statistical criteria-based as BIC, AIC and elbow method, at last, the final partitions generated by those methods are visualized are compared via external validation metrics (measures). Various clustering comparison measures have been used, counting-based measures as Adjusted Rand index (ARI) and information theoretic-based as Normalized Variation of Information (NVI) and Normalized Distance of Information (NID).

Índice de Contenidos

Resumen **3**

Abstract **4**

1. Introducción **10**

1.1. Descripción general y contexto de TFM 10

1.2. Justificación del TFM 11

1.3. Objetivos generales y específicos 12

1.4. Enfoque y fases del TFM 13

1.5. Descripción del resto de capítulos de la memoria del TFM 14

2. Conceptos teóricos **16**

2.1. ADN, ARN y gen 16

2.2. Transcriptoma y expresión génica 18

2.3. Secuenciación y ARN-seq 19

2.4. Base de datos del área científica 22

2.4.1. Base de datos y proyectos 22

2.5. GTEx 22

3. Descripción de los modelos de clustering y MCC **24**

3.1. Análisis de conglomerados 24

3.1.1. Definición 24

3.1.2. Tipos de métodos de análisis de conglomerados 25

3.1.3. Gene-based clustering 25

3.2. Elección del número óptimo de clusters 25

3.2.1. Selección del modelo 25

3.2.2. Método de codo 26

3.2.3. BIC 26

3.2.4. AIC . . . 27
3.3. K-means convencional . . . 27
3.3.1. Conceptos básicos del algoritmo k-means . . . 27
3.3.2. Etapas algoritmo k-means . . . 28
3.3.3. Ventajas e inconvenientes del algoritmo k-means . . . 29
3.4. Método coexp . . . 30
3.4.1. Definición del método coexp . . . 30
3.5. Modelo de mezcla finito . . . 32
3.5.1. Definición del modelo de mezcla finito . . . 32
3.5.2. Estructura del modelo de mezcla finito . . . 33
3.5.3. FMM para clustering . . . 34
3.5.4. Ajustando modelos de mezcla mediante el algoritmo EM . . . 36
3.6. Métricas de comparación de clustering . . . 41
3.6.1. Descripción de las MCC . . . 41
3.6.2. Métricas de validación externa . . . 42
3.6.3. Variación de la información normalizada . . . 45
3.6.4. Distancia de información normalizada . . . 46
3.7. Visualización del clustering . . . 46
3.7.1. Método componentes principales . . . 46
3.7.2. T-SNE . . . 47

4. Metodología y herramientas **49**
4.1. El lenguaje R . . . 49
4.2. Accediendo a los datos via método coexp . . . 51
4.3. Descripción y exploración de los datos . . . 52
4.4. K-means . . . 55
4.5. FMM . . . 57
4.5.1. Mclust . . . 57
4.5.2. EMMIX . . . 59
4.5.3. Elección de paquetes de R para FMM . . . 61
4.5.4. Método coexp . . . 61
4.5.5. Visualización del clustering en R . . . 62
4.5.6. Medidas de comparación de clustering . . . 64

5. Resultados **65**
5.1. Selección del número de clusters en función de criterios estadísticos . . . 66
5.2. Visualización del clustering para coexp, k-means y FMM . . . 69

5.3. Medidas de comparación de clustering aplicadas a coexp, k-means y FMM 70
5.4. Conclusiones y futuras direcciones 70

Bibliografía **77**

Anexo guía **82**

Índice de Figuras

1. Estructura de las redes de las redes de coexpresión. 11
2. Diagrama de las fases del TFM. 14

3. Estructura química del ADN y ARN 17
4. Transcripción y traducción del ADN 19

5. Resumen estadístico de las primeras 10 columnas de Cortex. 54
6. Heatmap para una muestra de Cortex. 55
7. Tamaño de los módulos método coexp. 62
8. Histograma de los componentes principales del dataset Cortex 63

9. Gráfica método de codo k-means. 66
10. Gráfica valores BIC y AIC k-means 67
11. Gráfica valores BIC mclust. 68
12. Gráfica valores AIC mclust. 68
13. Valores BIC y AIC emmix 69

Índice de Tablas

1. Tabla de contingencia para las particiones U y V de X. 43

2. Tejidos disponibles GTEx. 52
3. Estructura de los datos transpuestos Cortex. 53
4. Valores BIC y AIC k-means 57
5. Valores BIC y AIC mclust. 58
6. Valores BIC y AIC emmix. 61

7. K óptimo y valores WCSS, BIC y AIC. 66
8. Resultados de las métricas de comparación del clustering. 70

Capítulo 1

Introducción

1.1. Descripción general y contexto de TFM

Conocer la secuencia del ADN (Ácido Desoxirribonucleótido) y ARN (Ácido Ribonucleótido) nos ha permitido caracterizar a los seres vivos de manera morfológica, fisiológica, bioquímica y conductual.

Para entender la estructura y función de los genes no sólo es necesario el análisis aislado de las secuencias genómicas (genómica) sino también el análisis de la expresión génica de una célula o tejido (transcriptómica) y la caracterización de la presencia de proteínas, sus asociaciones y/o modificaciones post-traduccionales en células normales, enfermas o sujetas a diversos estímulos (proteómica). La genómica, transcriptómica y proteómica se ha ido perfeccionando notoriamente en estos últimos años gracias al desarrollo de metodologías de secuenciación masiva de alto rendimiento (*next-generation sequencing*, NGS) generandose grandes bancos de secuencias de ADN, ARN o peptídicas, las cuales se almacenan en bases de datos construidas por grandes consorcios como por ejemplo GenBank, UCSC (*University of California*, Santa Cruz) o GTEx (*Genotype-Tissue Expression*). La extraordinaria gran cantidad de información genómica, transcriptómica y proteómica da lugar al desarrollo de herramientas bioinformáticas capaces de analizar, integrar y manipular dicha información en un corto periodo de tiempo [2].

El trancriptoma es todo el conjunto ARN de una célula, tejido u órgano concreto que sirve para medir la cantidad de actividad de los genes, es decir, la expresión génica. Se han desarrollado numerosas tecnologías con el objetivo de cuantificar los cambios en la expresión génica en diferentes condiciones pero la metodología ARN-seq (*RNA sequencing*) y ARN-sc (*RNA single cell sequencing*) dan lugar a medidas más precisas del transcriptoma [55].

Teniendo perfiles de expresión génica de varios genes para varias muestras o condiciones experimentales, se puede construir una **red de co-expresión génica** buscando pares de genes que muestren un patrón de expresión similar en todas las muestras, ya que los niveles de expresión de dos genes co-expresados suben o bajan juntos a través de ellas [50].

Las redes de co-expresión génica son modelos gráficos no dirigidos, que constan de un conjunto de nodos y aristas, que representan los **genes** y las **interacciones entre los genes** respectivamente (figura 1) [3].

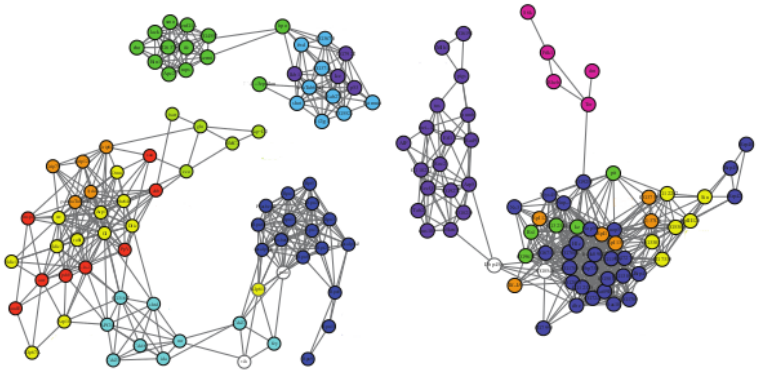

Figura 1: Estructura de las redes de las redes de coexpresión.

La información procedente de las redes de coexpresión génica puede ser usada para caracterizar genes (describir la función y estructura de un gen) asumiendo que genes fuertemente coexpresados tienen la misma función o se expresan en el mismo tipo celular [3].

En el presente TFM se estudian y evalúan cuatro diferentes algoritmos de clustering: método coexp (formado por WGCNA y k-means), k-means convencional y dos modelos de mezcla finitos. En primer lugar se implementan todos modelos de clustering, se busca el número de clusters óptimo en base a diferentes criterios estadísticos como BIC, AIC y el método de codo (MCD) y por último, las particiones generadas por dichos métodos se visualizan y se comparan entre sí con la ayuda de una serie de métricas (o medidas) de validación externa. Se han usado medidas basadas en el conteo de pares (*counting based measures*) como el índice de Rand ajustado (*Adjusted Rand Index*, ARI) y en información teórica (*information theoretic-based measures*)como la variación de la información normalizada (*Normalized Variation of Information*, NVI) y distancia de información normalizada (*Normalized Distance of Information*, NID).

1.2. Justificación del TFM

Los análisis de conglomerados (*clustering analysis*) son técnicas estadísticas que buscan agrupar y clasificar un conjunto de elementos en función de los valores que tomen las variables que los definen. Por este motivo y ante el desconocimiento *a priori* de la asignación correcta de los elementos a grupos, el análisis de conglomerados forma parte las denominadas técnicas de aprendizaje no supervisado (*unsupervised learning techniques*). Los grupos (clusters) que se forman deben de estar definidos de tal manera que los elementos que se encuentren en el mismo grupo presenten características comunes y una alta homogeneidad. Generalmente, estos métodos están basados en criterios geométricos, utilizados fundamentalmente como técnica descriptiva y exploratoria pero no explicativa. La mayoría de las técnicas de agrupamiento

están basados en distancias o procedimientos heurísticos que tienen dos principales ventajas, su construcción es intuitiva y un corto tiempo de computación, sin embargo, tiene como limitación la falta de base estadística sobre las que aplicar inferencia para una población a partir de una muestra.

Los resultados de estas técnicas de agrupamiento dependen del conjunto de datos, el algoritmo de agrupamiento seleccionado y la medida de similitud utilizada para comparar objetos (p.ej distancia).

Las limitaciones de los algoritmos basados en distancias y correlaciones crea la necesidad de evaluar los algoritmos habituales para generar GCN justificando su uso o posibilitando el descubriendo de una posible mejora de dicho método para optimizar el agrupamiento de genes. Por lo tanto, se estudia una alternativa a procedimientos basados en distancias y correlaciones que son los métodos basados en probabilidades (agrupamiento probabilístico). Es decir, se asume que el método de clasificación es generado por una distribución de probabilidad subyacente. Una de las características de ésta metodología es que el clustering es "fuzzy", es decir, cada observación tiene una probabilidad de pertenecer a un grupo (cluster). Uno de los modelos más populares clasificado como método de agrupamiento probabilístico es el modelo de mezcla finito (*finite mixture model*, FMM) [4].

1.3. Objetivos generales y específicos

Los objetivos generales del presente TFM son los siguientes:

1. Familiarizarnos con la metodología habitual para generar redes de coexpresión génica (WGCNA + k-means) , k-means convencional, la base de datos GTEx, el modelo de mezcla finito y las métricas de validación externa (ARI, NVI y NDI).
2. Implementar los métodos coexp, k-means y FMM utilizando datos procedentes de GTEx.
3. Comparar las particiones generadas por los métodos coexp, k-means y FMM mediante métricas de validación externa (ARI, NVI y NDI).

En base a los objetivos generales, se definen los objetivos específicos:

1. Obtención y análisis exploratorio de los datos GTEx.
2. Diseño y automatización de los métodos coexp, k-means y FMM para datos GTEx.
   - Implementación del método coexp.
   - Diseño y implementación de un modelo k-means.
   - Diseño y implementación de un modelo de mezcla finito (FMM).
   - Obtener el número de clusters K, en base a diferentes criterios (método de codo, BIC y AIC) para los modelos k-means y FMM y compararlos con el número de clusters obtenido por el método coexp.
3. Visualización de los datos GTEx mediante reducción de la dimensionalidad.

- Reducción de dimensionalidad mediante el análisis de componentes principales (PCA) y representación gŕafica de los datos en función de los diferentes clusters.
- Reducción de dimensionalidad mediante el análisis T-SNE y representación gŕafica de los datos en función de los diferentes clusters.

4. Comparación de las particiones generadas por los métodos coexp, k-means y FMM con las métricas ARI, NVI y NID.
   - Método coexp vs FMM (emmix).
   - Método coexp vs FMM (mclust).
   - Método coexp vs FMM (k-means).
   - FMM (emmix) vs FMM (mclust).
   - FMM (emmix) vs k-means.
   - FMM (mclust) vs k-means.
5. Tabla de resultados, conclusiones y direcciones a seguir.

1.4. Enfoque y fases del TFM

La realización del presente TFM se ha realizado de forma secuencial y ordenada, estableciendo y llevando a cabo una serie de fases, las cuales son imprescindibles para optimizar la elaboración de dicho trabajo. Las fases se han establecido en función de los objetivos generales y específicos. Algunas fases, como por ejemplo la fase de búsqueda bibliográfica y metodológica (fase A) a sido decisiva para detectar *a priori* algunos factores de riesgo que podrían haber obstaculizado la realización del TFM. A pesar de todo, era de esperar encontrarse con otros factores de riesgo que se han subsanado aplicando el elemento corrector necesario, intentando no modificar el planteamiento del TFM y sin alterar el plazo de entrega del mismo. El esquema de las fases del TFM se presenta en la figura 2.

- Fase A: Búsqueda bibliográfica y metodológica general y específica.

  La fase A es imprescindible para la realización del TFM ya que te permite familiarizarte con los datos, herramientas y metodología para realizar con éxito el TFM. Como se ha explicado anteriormente, esta fase también te permite detectar factores de riesgo reduciendo el tiempo requerido para subsanarlos.

- Fase B: Selección, pre-procesado y exploración de los datos de partida.

  GTEx proporciona una base de datos de transcriptoma muy extensa, por lo tanto, de todos los tejidos incluidos en dicha base de datos hemos elegido Cortex cerebral. El pre-procesado de los datos lo realiza directamente GTEx y para explorar los datos se utilizan los estándares básicos estadísticos comunes (estadística descriptiva).

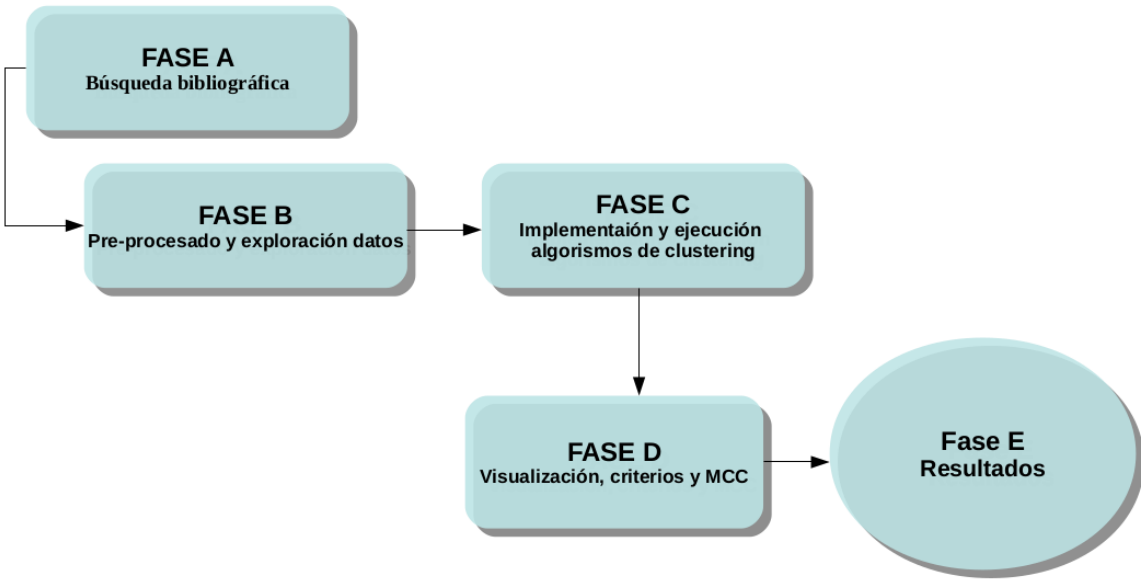

Figura 2: Diagrama de las fases del TFM.

- Fase C: Implementación y ejecución de los algoritmos de clustering.

  Construcción e implementación del método coexp, k-means y dos modelos de mezcla finitos para datos de transcriptoma.

- Fase D: Visualización, criterios estadísticos y medidas de comparación de clustering.

  Se aplican dos métodos de reducción de dimensionalidad (PCA y T-SNE) para visualizar los datos Cortex en función del agrupamiento generado por los cuatro métodos de clustering (método coexp, k-means y los dos FMM), se obtiene el número de clusters óptimo en base a criterios estadísticos como el método de codo, BIC y AIC y también se comparan las particiones finales generadas por dichos métodos de dos en dos.

- Fase E: Elaboración de resultados, conclusiones finales y direcciones.

1.5. Descripción del resto de capítulos de la memoria del TFM

Una vez explicado el capítulo 1, se resumen los demás capítulos del TFM para facilitar su comprensión.

En el capítulo 2 se presentan los conceptos teóricos más relevantes (ADN, ARN, gen, transcriptoma, secuenciación etc) y la base de datos utilizada (base de datos GTEx) para evaluar la metodología descrita e implementada en este TFM.

El capítulo 3 tiene tres partes, en la primera parte se explica el concepto general de análisis de

conglomerados (clustering) y los criterios utilizados para obtener el número de clusters K como por ejemplo BIC, AIC etc. En la segunda parte, se explica de manera teórica el **método coexp**, **k-means convencional** y el **modelo de mezcla finito** y en la tercera parte se presentan las métricas (ARI, NVI y NID) utilizadas para comparar las particiones generadas por los cuatro modelos implementados en este TFM y las técnicas usadas para reducir la dimensionalidad de los datos con el objetivo de visualizar el clustering realizado.

En el capítulo 4 se presentan la metodología y herramientas utilizadas en este TFM y tiene tres partes. En la primera parte, se ejecuta el método coexp y se implementan el método k-means y los dos modelos de mezcla finitos, también, se obtiene el número de clusters óptimo en base a los criterios estadísticos BIC, AIC y el método de codo. En la segunda parte se realizan dos técnicas de reducción de dimensionalidad (PCA y TSNE) para visualizar más fácilmente los datos en función de las particiones generadas por los cuatro métodos de clustering (coexp, k-means y los dos modelos de mezcla finitos) y en la tercera parte se implementan las medidas de comparación de clustering para enfrentar las particiones generadas por dichos modelos de clustering.

En el capítulo 5 se presentan los resultados, conclusiones y direcciones a seguir.

Capítulo 2

Conceptos teóricos

2.1. ADN, ARN y gen

El ADN (ácido desoxiribonucleótido) es una macromolécula química que contiene la información genética necesaria para el desarrollo y funcionamiento de todos los organismos vivos y algunos virus. Desde el punto de visto químico, el ADN esta constituido por dos cadenas (dos hélices) formadas por un elevado número de nucleótidos. Las dos hélices tienen direcciones opuestas, es decir, la dirección desde el extremo 5' al 3' de cada cadena es opuesta (cadenas antiparalelas). Cada nucleótido está formado por un glúcido (desoxirribosa), una base nitrogenada (puede ser Adenina, Timina, Citosina, y Guanina) y un grupo fosfato (derivado del ácido fosfórico). Lo que distingue a un polinucleótido de otro es la base nitrogenada, y por ello la secuencia del ADN se especifica nombrando solo la secuencia de sus bases [27]. A diferencia del ADN, el ARN en lugar de desoxiribosa tiene ribosa y tiene cuatro bases nitrogenadas, tres en común con el ADN (Adenina, Guanina y Citosina) pero en lugar de Timina tiene Uracilo (figura3).

Generalmente, el ARN es monocatenario (cadena simple), aunque también pueden puede formar estructuras bicatenarias o formar un híbrido ADN-ARN[1].

La disposición y orden de estas cuatro bases a lo largo de la cadena de ADN es lo que se entiende como información genética, la cual puede ser heredada (replicación) o utilizada por la maquinaria celular mediante dos procesos bioquímicos llamados transcripción y traducción genética [62].

El proceso de **replicación** permite al ADN duplicarse, es decir, sintetizar una copia idéntica. La replicación del ADN comienza en una secuencia de nucleótidos particular del cromosoma llamado origen de replicación (ORIc). La replicación ocurre bidireccionalmente, por medio de dos horquillas de replicación que se mueven en direcciones opuestas. El proceso de replicación necesita una secuencia ARN cebador (sintetizado por una enzima ARN primasa) con sus bases correctamente apareadas con la cadena molde, una serie de enzimas helicasas que desenrollan la doble hélice en cada horquilla de replicación y proteínas de unión a la cadena simple de ADN que estabilizan las cadenas separadas.

[1] `https://lidiaconlaquimica.wordpress.com/2015/07/20/estructura-y-tipos-de-arn/`

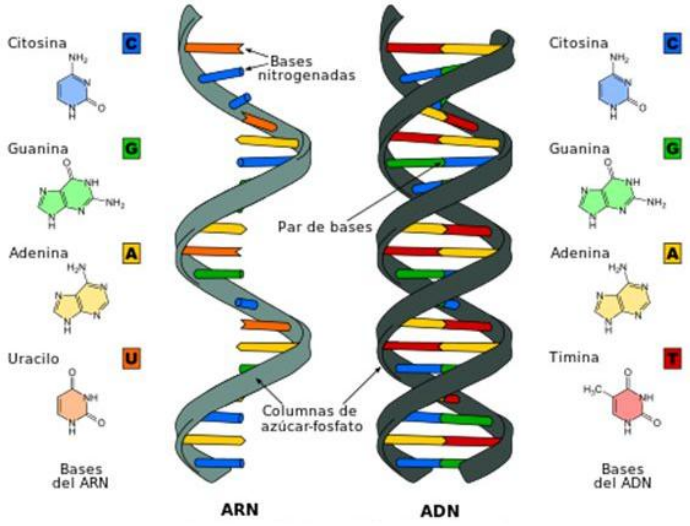

Figura 3: Estructura química del ADN y ARN

También, las topoisomerasas, relajan el superenrollamiento de la hélice, ya que cortan las cadenas por delante de las horquillas de replicación y luego las vuelven a unir [7].

La **transcripción** del ADN es el proceso en el que se transfiere la información genética contenida en el ADN, mediante la síntesis de otra molécula llamada ARN (figura 4). Este proceso es llevado a cabo, principalmente, por una serie de enzimas llamadas ARN polimerasas y tiene tres etapas, que son iniciación, elongación y terminación [7].

1. Iniciación: la ARN polimerasa se une a una secuencia de ADN llamada promotor (con la ayuda de factores de transcripción), que se encuentra al inicio de un gen (cada gen tiene su propio promotor) y se genera un cebador de ARN al que se le van a unir los nucleótidos complementarios de la cadena molde de ADN [7].

2. Elongación: la ARN polimerasa 'lee' base a base el molde de ADN y va uniendo secuencialmente, a partir del cebador, los nucleótidos de ARN complementarios a la secuencia molde formando una cadena que crece en dirección $5' \rightarrow 3'$. Esa cadena de ARNm (ARN mensajero), se llama transcrito y tiene la misma información, pero complementaria, que la cadena de ADN molde y en lugar de la base timina (T) tiene Uracilo (U) [7].

3. Terminación: una vez que la ARN polimerada llega a las secuencias de terminación del ADN, la ARN polimerasa libera el transcrito [7].

Después de estas tres fases, en eucariotas, el transcrito se modifica. Primero se le añade un caperuza al extremo 5'(CAP 5') y una cola de adeninas (poli-A) al extremo 3'. También, se produce la escisión

(eliminación) de los intrones (secuencias no codificantes) y la unión de los exones (secuencias codificantes) en un proceso llamado *splicing*.
El ARNm maduro luego va al citoplasma y se produce su **traducción** a polipéptido (figura 4). En el proceso de traducción se genera una cadena polipeptídica a partir del transcrito maduro utilizando el código genético como referencia y con la ayuda del ARNt (ARN transferente) y las dos subunidades del ribosoma. Al igual que la transcripción, la traducción tiene tres etapas, iniciación, elongación y terminación [62].

1. Iniciación: el ribosoma y el ARNt se unen al ARNm y forman el complejo de iniciación. El ARNt lleva incorporado la secuencia de iniciación (AUG) que se une a su secuencia complementaria en el ARNm liberando así el primer aminoácido, la metionina (eucariotas), la cual estaba anclada al otro extremo del ARNt [7].

2. Elongación: es la etapa donde la cadena de aminoácidos se extiende. La elongación se produce gracias al ARNt, ya que en un extremo tiene una secuencia de tres pares de bases, llamada anticodón, la cual reconoce su secuencia complementaria en el ARNm en el sitio A del ribosoma, llamada codón, permitiendo la formación de un enlace peptídico (gracias a la enzima peptidil-transferasa) entre la metionina y el aminoácido que lleva el ARNt en el otro extremo de la molécula. Este proceso se va repitiendo con todos los aminoácidos que se les va añadiendo a la cadena peptídica [7].

3. Terminación: la terminación de la cadena peptídica ocurre cuando entra al sitio A del ribosoma uno de los codones de stop del ARNm, estos codones no son reconocidos por ningún ARNt pero sí son reconocidos por factores de terminación de la cadena, que producen la liberación del polipeptido y el desacoplamiento de las subunidades del ribosoma [7].

En la actualidad, se acepta como definición de **gen** a todo segmento de ADN que se encuentra a continuación de un promotor y que puede ser transcrito, originando un ARN funcional (ARNm, ARNt, ARNsn...). Estos ARN funcionales son necesarios para la síntesis de macromoléculas como por ejemplo proteínas. [7].

2.2. Transcriptoma y expresión génica

Toda la información genética contenida en el ADN es copiada en forma de ARN, mediante un proceso denominado transcripción. A todos los transcritos generados en una célula o tejido en una condición fisiológica determinada se le llama **transcriptoma**.

El estudio del transcriptoma es esencial para entender la **función génica** ya que nos proporciona la capacidad para medir la actividad de los genes, es decir, la expresión génica. Podemos suponer, entonces, que si un gen tiene una expresión muy elevada en una determina condición fisiológica o célula/tejido es porque dicho gen cumple una determinada función bajo esa condición o en esa célula/tejido. El estudio

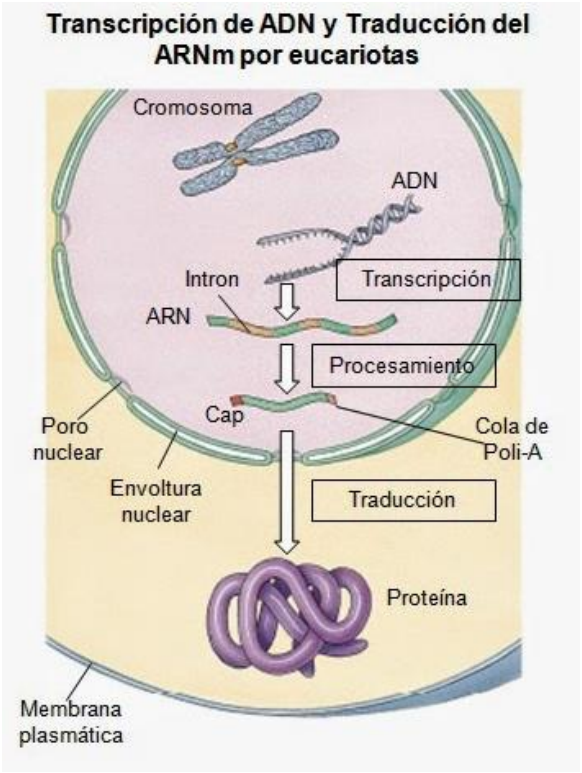

Figura 4: Transcripción y traducción del ADN

global del transcriptoma permite también establecer patrones de regulación génica coordinada, lo que contribuye no solo a dilucidar la función y agrupamiento de varios genes bajo un estímulo o condición específica, sino también a identificar elementos promotores comunes a varios genes [48].

2.3. Secuenciación y ARN-seq

El aumento de conocimiento en transcriptómica se produjo en primer lugar por el desarrollo de las primeras técnicas de secuenciación, que son la secuenciación química de Maxam y Gilbert y de "terminación de la cadena"de Sanger. Estas técnicas eran muy lentas y necesitaban reactivos muy tóxicos. Para solventar estos problemas se desarrollaron, poco después, técnicas basadas en hibridación de secuencias, como por ejemplo la técnica SAGE (*Serial Analysis of Gene Expression*), pero presentaban también muchas limitaciones como por ejemplo su baja cobertura, alta confianza en el conocimiento existente del genoma completo para su implementación o la necesidad de emplear métodos de normalización de datos muy complejos [55]. Actualmente, se han desarrollado nuevas tecnologías de secuenciación masiva llamadas NGS (*Next Generation Sequencing*). La tecnología NGS más popular en estudios de transcriptoma es la secuenciación directa de ARN, conocida como **metodología ARN-seq** que posibilita la caracterización completa de la expresión génica a lo largo del tiempo, condiciones experimentales o patológicas, sin

necesidad de un gran conocimiento del genoma completo [55].

Los experimentos ARN-seq, en general, consisten en las siguientes fases:

1. **Extracción del ARN total desde la muestra biológica**

   El ARN es aislado y purificado con la ayuda de una enzima llamada deoxyribonucleasa, una nucleasa que hidroliza el enlace fosfodiéster de los nucleótidos. Para generar ADNc (ADN circular) a partir de ese ARN es necesario que este último sea de calidad. La calidad del ARN se mide normalmente con un bioanalizador *Agilent*, que produce un número RIN (*RNA integrated number*) cuyo valor está entre 1 y 10, teniendo el valor de 10 las muestras de mayor calidad (muestras menos degradadas). Se consideraran muestras con poca calidad si se obtiene un valor RIN <6 [55].

2. **Fragmentación del ARN**

   La fragmentación del ARN normalmente se realiza mediante digestión con enzimas o por nebulización, sonicación etc. Este proceso permite generar secuencias con una longitud adecuada para la maquinaria de secuenciación, reducir el sesgo y la influencia de la secuencia de nucleótidos de los cebadores utilizados.

3. **Generación de librerías ARN-seq**

   Esta fase puede variar dependiendo de la tecnología NGS utilizada. En general, la construcción de bibliotecas de secuenciación implica dos pasos:

   - **Selección del ARN objeto a estudio**

     Antes de la construcción de librerías de ARN-seq se debe de elegir el protocolo de actuación que dependerá de la especie de ARN a estudio. El ARN total está formado por ARN ribosomal (95 % del ARN), ARNm, ARNpre (ARNm precursores), ARNnc (ARN no codificantes) etc. Lo más habitual en estudios de transcriptoma, es aislar el ARNm, para eso, se deben de eliminar de la muestra de ARN total las otras especies de ARN. Hay numerosas técnicas para realizar el enriquecimiento de las moléculas de ARN, por ejemplo, su captura usando polinucleótidos de timina que se unen covalentemente al ARNm. Para realizar estas tareas existen numerosos kits comerciales como por ejemplo RiboMinus [21]. En la mayoría de las tecnologías de secuenciación necesitas librerias de ADN y es por eso por lo que el ARN-seq se debe de transformar a ADN circular, el cual representa la cadena ARN original y su complementaria.

   - **Transcripción inversa**

     La transcripción inversa se realiza mediante una enzima llamada transcriptasa reversa (enzima vírica termoresistente). Gracias a su tiple función (ADN polimerasa dependiente de ARN, ribonucleasa H y ADN polimerasa) es capaz de transformar moléculas simples de ARN en ADN circular. Una de las estrategias para preservar toda la información de la secuencia de ARN monocatenario o de ADN circular es añadirles unas moléculas llamadas adaptadores

(en los extremos 3' y 5') o marcadores químicos (desoxi-UTP [2]) durante la síntesis de ADN circular, los cuales pueden ser degradados fácilmente por enzimas digestivas [21].

4. **Amplificación del ADN circular y secuenciación**.

   El siguiente paso es la amplificación de las moléculas de ADN circular-adaptadores. Este procedimiento, generalmente se realiza mediante el método PCR (reacción en cadena de la polimerasa). En general, consiste en tres fases que se pueden repetir entre 25-40 veces. En la primera fase, el termociclador genera en torno a 95 °C lo que permite la separación de los ácidos nucleicos de doble cadena. En la segunda, genera a una temperatura en torno a los 50-60 °C que permite el alineamiento de los cebadores al ADN molde y la tercera genera una temperatura entre 68-72 °C que facilita la polimerización de los nucleótidos gracias a una enzima llamada taq polimerasa generando nuevas cadenas de ADN [63].

   Las tecnologías de secuenciación están disponibles comercialmente y desarrolladas por compañías como Roche/454, Solexa/Illumina etc. Por ejemplo, Roche/454 emplea el secuenciador 454-Genome-Sequencer FLX, el cual realiza una pirosecuenciación. En esta tecnología, los adaptadores del ADN circular (muestra) se fijan a una superficie (microperlas) mediante su unión con una serie de oligonucleótidos complementarios adheridos a dicha superficie. Posteriormente, se lleva a cabo la amplificación de fragmentos mediante una PCR en emulsión para producir una gran cantidad de fragmentos de la misma secuencia y así aumentar la intensidad de la señal. Una vez finalizada la amplificación de fragmentos, se desnaturalizan y las microperlas son transferidas a un chip de fibra óptica en donde se incuban con las enzimas ADN polimerasa, ATP sulfurilasa, luciferasa y apirasa, así como con los sustratos luciferina y adenosin-5-fosfosulfato (APS). Sobre el chip se adicionan diferentes nucleótidos que al ser incorporados por la ADN polimerasa a la cadena naciente (se unen a los fragmentos) liberan moléculas de pirofosfato. Éste pirofosfato proviene de la formación del enlace fosfodiéster y es convertido a adenosin trifosfato (ATP), en presencia de la APS. El ATP producido reaccionará con la enzima luciferasa y emitirá un haz de luz que será detectado por una cámara CCD (*charge coupled device*) lo que permitirá la cuantificación de la señal [42].

   En el caso de Solexa/Illumina, la secuenciación esta basada en el principio de amplificación 'en puente' y el uso de un marcaje por fluorescencia de nucleótidos modificados que serán llamados terminadores reversibles. Los adaptadores del ADNc se unen a oligonucleótidos adheridos a una superficie sólida o *flow cell*. Estos oligonucleótidos, actúan como primers o cebadores sentido o antisentido, y crean puentes que favorecen la amplificación. Los amplicones permanecerán adheridos y después de la desnaturalización formarán otro puente para permitir la siguiente amplificación. Estos pasos se repiten hasta generar millones de grupos de un fragmento determinado. Una vez formados estos grupos se desnaturalizan nuevamente para iniciar la polimerización o síntesis de cada fragmento. La polimerización se produce se incorporarn una mezcla de cuatro nucleótidos marcados con 32-O-azidometil (terminadores reversibles) [42].

[2] Desoxi uridina tri fosfato.

Los terminadores reversibles (dideoxinucleótidos) detendrán la síntesis de ADN una vez la ADN polimerasa integre a la cadena naciente el nucleótido correspondiente. Los fluoróforos de los terminadores reversibles integrados a la cadena naciente son activados por un láser lo que provoca una emisión de luz que será diferencial de acuerdo con el nucleótido incorporado. La información será registrada y almacenada para su detencción [42].

En resumen, de la secuenciación masiva, obtenemos, finalmente, pequeñas secuencias de nucleótidos, llamadas **reads**, que corresponden a cada fragmento perteneciente a la librería de fragmentos de ADNc generada. El número de reads varía en función de la expresión de un determinado gen, es decir, si un gen se expresa más, tendrá más ADNc y, por tanto, es más probable que tenga un mayor número de reads. La lectura (depende de la tecnología utilizada) de estas reads generan los llamados perfiles de expresión génica.

2.4. Base de datos del área científica

2.4.1. Base de datos y proyectos

El aumento de la actividad científica, las mejoras tecnológicas y el incremento de la inversión en instalaciones científicas y tecnológicas han permitido la obtención de una gran cantidad de datos que crecen de manera exponencial. En la actualidad, existen grandes consorcios científicos (GenBank, BioCyc GTEx...) públicos o privados capaces de proporcionar no sólo una gran cantidad de datos de transcriptoma, sino también de genoma y proteoma. Todos estos recursos son proporcionados por y para la comunidad científica con el objetivo de agilizar el descubrimiento de la función y estructura de todos los genes y sus productos.

El **proyecto del genoma humano**, completado en 2003, inició una revolución ya que por primera vez se pudo obtener la secuencia completa de nuestro ADN lo que permitió el abaratamiento de las técnicas que hicieron posible su lectura a gran escala. Sin embargo, ese conocimiento estaba limitado ya que no permitía entender la biología de las enfermedades ni su relación con el genoma. Poco después, se desarrolló el **proyecto ENCODE** (*ENCyclopedia Of DNA Elements*), el cual permitió el mapeado de elementos funcionales codificados por el genoma humano (ARN cortos, largos, nucleares...), el **proyecto GTEX** (*genotype-tissue expression*) cuyo objetivo principal es estudiar la expresión génica en forma de ARN en los diferentes tejidos de un organismo vivo y el **proyecto epigenoma** que estudia los mecanismos de regulación de la expresión génica.

Para realizar el presente TFM hemos seleccionado datos procedentes del proyecto GTEx.

2.5. GTEx

El proyecto GTEx comenzó en 2010 pero fue en 2013 cuando el portal GTEx proporcionó acceso libre a todos sus recursos. El NIH (*National Institutes Health*) de EEUU lanzó el proyecto GTEx con el

objetivo de crear una base de datos y un banco de tejidos que permitiera a los científicos estudiar cómo las variaciones genómicas pueden afectar a la actividad de los genes y a la susceptibilidad de padecer enfermedades. La fuente de datos GTEx consiste en secuencias de genoma completo y de RNA obtenidas de una gran cantidad de tejidos (por ejemplo piel, cerebro, corazón etc) de más de 700 donantes adultos port-morten. Cada año va aumentando el número de donantes y de muestras. El estudio piloto de GTEx identificó una variante génica que influye en cómo los genes se activan y desactivan en los tejidos y órganos humanos denominada **loci de rasgos cuantitativos de expresión** (*expression quantitative trait loci*, eQTL). Es decir, se han obtenido, de las secuencias de ARN generadas en los diferentes tejidos, unas firmas transcripcionales características que se mantienen estables en tejidos post-mortem. Estas firmas están dominadas por un grupo de genes exclusivos de un tejido en particular y varían más entre órganos que entre individuos [58].

GTex investiga los patrones de variación del transcriptoma entre individuos y tejidos o células permitiendo a la comunidad científica adquirir conocimiento sobre enfermedades neurodegenerativas (Alzheimer, Parkinson...), cáncer o contribuir al impulso de la medicina personalizada.

Las muestras para GTEx se obtienen a través de autopsias en el PMI (*Post-morten Interval*, intervalo post-mortem) o de trasplantes de órganos y tejidos. Para comparar la calidad de los tejidos derivados de la autopsia y cirugía, también se recopilan un pequeño subconjunto de tejidos que se descartan de forma rutinaria durante la amputación quirúrgica, como la piel, la grasa y el músculo. La cuantificación de la expresión se realiza mediante secuenciación masiva en paralelo (MPS) o secuenciación de alto rendimiento (NGS). La comunidad científica no solo puede acceder a los datos de expresión génica sino también a la información de las muestras, los donantes y también a los protocolos de actuación [24].

Capítulo 3

Descripción de los modelos de clustering y MCC

En este capítulo se explican teóricamente el concepto de clustering, los modelos de clustering implementados en este TFM (k-means, coexp, FMM emmix y FMM mclust) y las medidas de comparación de clustering (MCC), ARI,VI, NVI y NID.

3.1. Análisis de conglomerados

3.1.1. Definición

El análisis de conglomerados o clustering es el estudio de los métodos y algoritmos que agrupan y clasifican objetos (o individuos) en función de una característica intrínseca común o similaridad. Son técnicas de aprendizaje no supervisado multivariante, basado en criterios geométricos y aplicable en numerosas áreas de investigación como la biología, biomedicina, astrofísica etc.

Uno de los primeros pasos para hacer el análisis de conglomerados es la medición del grado de similitud o diferencia entre los objetos que queremos clasificar. La similitud se puede representar en forma de distancia, correlación etc. Hay varias formas de medir la distancia entre dos puntos, la más usada para variables continuas es la distancia euclídea (una línea recta entre 2 puntos), sin embargo, también tenemos la distancia 'de cuadras de ciudad' o *city-block distance* (la suma de las diferencias absolutas de los valores de las variables) y la distancia de Chebychev (la máxima diferencia absoluta de los valores de las variables). Como las unidades de las variables pueden ser muy diferentes es recomendable estandarizar o normalizar los valores de las variables antes del análisis [53].

3.1.2. Tipos de métodos de análisis de conglomerados

Existen dos tipos de métodos para el análisis de conglomerados, los **métodos jerárquicos** y los **métodos no jerárquicos** o de partición.

- **Métodos jerárquicos:**

  Realizan una agrupación inicial y a partir de ésta va añadiendo o separando conjuntos de observaciones dando origen a grupos nuevos minimizando la distancia entre las observaciones pertenecientes a un grupo o cluster. Los métodos jerárquicos se subdividen en aglomerativos y disociativos. Los métodos **aglomerativos**, conocidos como métodos ascendentes, comienzan teniendo tantos grupos como observaciones haya y a partir de estas unidades iniciales se van formando grupos, de forma ascendente, hasta que al final del proceso todas las observaciones estan englobadas en un único conglomerado. En cambio, los métodos **disociativos**, también llamados métodos descendentes, comienzan teniendo un gran grupo que engloba a todas las observaciones y en pasos sucesivos se van agrupando diferencialmente cada una las observaciones [1].

- **Métodos no jerárquicos o de partición:**

  Son métodos que dividen el conjunto de observaciones en K grupos. K es elegido inicialmente por el usuario [1].

3.1.3. Gene-based clustering

Para datos de expresión génica, el clustering se puede realizar en función de los genes o de las muestras. En la agrupación basada en genes (*gene-based clustering*), los genes se consideran objetos mientras que las muestras se consideran características, mientras que en la agrupación basada en muestras, las muestras se pueden segregar en grupos idénticos donde los genes se tratan como características y las muestras como objetos. El clustering puede ser parcial (CP) o completo (CC). El CP suele ser el más adecuado para los datos de expresión ya que estos datos contienen algunos genes o muestras irrelevantes (ruido) pero en el caso de los datos GTEx (apartado 4.3), todos los genes son relevantes (se han eliminado el ruido y se han filtrado los datos) y por lo tanto, en este TFM se va a realizar un clustering completo para asignar cada uno de los genes del dataset a un cluster [6].

3.2. Elección del número óptimo de clusters

3.2.1. Selección del modelo

El primer paso para realizar clustering no jerárquico es la elección del número de clusters, K.

Se debe de asumir que no se puede calcular el **modelo verdadero**, es decir, el mejor modelo que se ajusta a los datos, por lo tanto, se estima el modelo que más se acerca al verdadero. Ese modelo se utiliza posteriormente para hacer inferencias a partir de los datos.

Para seleccionar el modelo que más se acerca al verdadero, es imprescindible la elección adecuada del número de clusters y para ello el usuario suele ejecutar el algoritmo de clustering para un rango de valores bastante alto y elige el modelo más adecuado en base a algún criterio. Hay muchos tipos de criterios para elegir el número de clusters óptimo, por ejemplo el "método de codo"(Elbow Method), BIC, AIC, método Calinsky etc.

Los tres criterios que se van a presentar en el presente TFM son el método de codo (k-means), criterio de información bayesiana (BIC) y criterio de información de Akaike (AIC).

3.2.2. Método de codo

El "método de codo"(*elbow method*) utiliza el valor **WCSS** (apartado 3.3) es decir, la suma de la distancia al cuadrado de cada uno de las observaciones a su centroide.

Una vez obtenidos los valores WCSS, se representan linealmente en función de un rango alto de valores K (figura 9). En esta gráfica se debe de apreciar un cambio brusco en la línea continua que une cada uno de los puntos. A ese cambio brusco en la gráfica se le denomina codo. El punto en el eje y donde se observa el el codo es el número de clusters K [23].

3.2.3. BIC

BIC (*Bayesian Information Criterium*) fue desarrollado por Gideon E. Schwarz. Es un criterio que determina la capacidad predictiva de un modelo en base a un medida (por ejemplo la verosimilitud). Más concretamente, busca el modelo más estable (más parecido al verdadero, el cual no se puede obtener) de un conjunto de modelos. Es capaz de medir la capacidad explicativa de un modelo y penalizarlo en función de su complejidad (número de parámetros del modelo) [4] .

BIC se define como:

$$BIC = -2 \times ln(\hat{L}) + ln(n) \times k \quad (1)$$

siendo $\hat{L}$ el *log-likelihood* del dataset del modelo a estudio, n el número de observaciones del dataset y k el número de parámetros estimados por el modelo a estudio. Se suele elegir el modelo que tenga el valor BIC más pequeño.

BIC penaliza los parámetros libres más fuertemente que el criterio de información de Akaike (AIC), aunque depende del tamaño de n y la magnitud relativa de n y k [4].

Características

- A priori es un criterio independiente.
- Mide la eficiencia de un modelo en términos predictivos.
- Penaliza la complejidad del modelo, donde la complejidad se refiere a el número de parámetros.
- Puede ser usado para elegir el número óptimo de clusters segun la complejidad intrínseca de un conjunto de datos.
- Es un criterio cercano a otros criterios como AIC [4].

3.2.4. AIC

AIC (*Akaike information criterion*) es una medida desarrollada por Hirotugu Akaike. AIC es un estimador insesgado de la distancia entre el modelo estimado y el verdadero. Al igual que BIC, AIC se utiliza para seleccionar el mejor modelo en base a algún criterio, como por ejemplo la verosimilitud. AIC no sólo recompensa la bondad de ajuste, sino también incluye una penalización, directamente proporcional al aumento del número de parámetros estimados [9].

AIC se define como:

$$AIC = -\ 2ln(\hat{L}) + 2 \times k \ (2)$$

siendo $\hat{L}$ el log-likelihood del dataset del modelo a estudio y k el número de parámetros estimados por el modelo a estudio [9].

Existe un único conjunto de valores de los parámetros que maximiza la función de verosimilitud, por lo tanto, cuanto más alto sea el valor de máxima verosimilitud mejor ajustado estará el modelo a los datos. Esto depende tanto de la forma del modelo como del grado de dispersión de los datos. La medida de bondad de ajuste en este caso es $2ln(\hat{L})$ y esa medida es una función creciente, es decir, la bondad de ajuste aumenta de forma logarítmica respecto al valor de máxima verosimilitud ($\hat{L}$). La complejidad de AIC viene dada por k, que es el número de parámetros del modelo [9].

3.3. K-means convencional

3.3.1. Conceptos básicos del algoritmo k-means

El algoritmo k-means fue propuesto por Stuart Lloyd en 1957 pero fue Hartigan y Wong entre 1975 y 1979 los que publicaron una versión más eficiente de dicho algoritmo.

K-means es un algoritmo heurístico de clasificación no supervisada que agrupa un conjunto de N datos en K grupos basándose en sus características. El agrupamiento se realiza minimizando la distancia intra-cluster y maximizando la distancia inter-cluster, es decir, divide los datos en clases o grupos homogéneos de modo que los elementos de la misma clase son tan similares como sea posible mientras que elementos de diferentes clases son tan diferentes como sea posible [26].

La distancia más utilizada en el algorimo k-means es la **distancia euclídea** y se define en un espacio de R dimensiones, siendo $P = p_1, ..., p_r$ y $Q = q_1, ..., q_r$ como [23]:

$$d(P, Q) = \sqrt{\sum_{i=1}^{R} (p_i - q_i)^2} \quad (3)$$

Más concretamente, k-means construye una partición de las observaciones (de acuerdo con el diagrama de Voronoi [3]) en K grupos con el fin de minimizar la suma de las distancias cuadráticas (*within-cluster sum of squares*, WCSS) de cada dato al centroide de su cluster. Es decir, minimiza la función conocida como función de error cuadrático (J) que se define como:

$$J(M) = \sum_{i=1}^{k} \sum_{j=1}^{c_i} || \ x_i - m_j \ ||^2 \quad (4)$$

siendo $|| \ x_j - m_i \ ||^2$ la distancia euclídea entre un dato (x_i) y un centroide (m_j), k el número de centroides, c_i el número de datos en un cluster i, $X = x_1, x_2, ..., x_n$ el conjunto de datos y $M = m_1, m_2, ..., m_k$ el conjunto de centroides [26].

3.3.2. Etapas algoritmo k-means

K-means es un algoritmo iterativo que se divide en 3 etapas, iniciación (*initialization*), asignación (*assigment step*) y actualización (*update step*).

- Inicialización: Se elige aleatoriamente el número de clusters K y los centroides ($m^{(k)}$) iniciales del conjunto de datos.
- Asignación: Cada dato es asignado a su centroide más cercano y se calcula la distancia euclídea entre ambos. Cada uno de los centroides define un cluster. Formalmente, si m_i es un conjunto de centroides perteneciente a M, cada dato x_i es asignado a un cluster en base a la fórmula:

$$\underset{m_i \in M}{argmin} \ dist(m_i, x)^2 \quad (5)$$

  donde $dist()$ es la distancia euclídea [61].
- Actualización: Se actualiza la posición del centroide de cada grupo tomando como nuevo centroide el valor medio de todos los datos que define un cluster.

[3] https://en.wikipedia.org/wiki/Voronoi_diagram

$$m_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_i \quad (6)$$

Después, se recalcula la distancia entre cada punto y su nuevo centroide.

Se repiten las etapas 2 y 3 hasta que no se pueda hacer más reasignaciones. Aunque el algoritmo termine siempre (converge), no se garantiza el obtener la solución óptima [26].

3.3.3. Ventajas e inconvenientes del algoritmo k-means

Ventajas

- Rápido y fácil de entender.
- Robusto.
- Flexible, es decir, se adapta situaciones atípicas mediante ajustes simples del modelo.
- Muy eficiente cuando los datos están distribuidos en grupos bastante dispersos [20][23].

Inconvenientes

- Es muy sensible a la elección aleatoria de los k centros iniciales. Esto perjudica la eficacia del algoritmo ya que en la práctica, no se conoce *a priori* el número de clusters final.
- Es susceptible a valores extremos porque distorsionan la distribución de los datos.
- El algoritmo falla para conjuntos de datos no-lineales.
- No es tan sofisticado como otros algoritmos de clasificación [20].

3.4. Método coexp

3.4.1. Definición del método coexp

Las redes de co-expresión génica (*gene coexpression networks*, GCN) son modelos gráficos no dirigidos, que constan de un conjunto de nodos y aristas, que representan los genes y las interacciones entre los genes respectivamente (figura 1) [50].

El método habitual de construcción de redes de co-expresión génica es el *Weighted Gene Co-expression Network Analysis*, **WGCNA**. WGCNA se utiliza para encontrar grupos (módulos) de genes altamente correlacionados (coexpresados) con el objetivo de caracterizar dichos genes [57]. Es decir, se busca asociar un conjunto de genes a un fenotipo determinado [3].

WGCNA parte de una matriz $G_{m\times n}$, donde m es el número total de muestras y n el número total de genes. Las observaciones de la matriz G corresponden a datos de expresión génica también llamados, perfiles de expresión génica. El primer paso para construir una GCN es definir una medida de similitud entre pares de genes que calcula el nivel de **concordancia** entre los perfiles de expresión génica a través de los experimentos (o muestras). Para cada par de genes i y j la medida de similitud se denota como s_{ij} [57] y en el caso del estándar WGCNA es la **correlación**.

Por lo tanto, la medida de similitud para cada par de genes se define como:

$$s_{ij} =\mid cor(g_i, g_j) \mid \quad (7)$$

donde g_i g_j son los perfiles de expresión de los gen i y el gen j respectivamente.
Para el conjunto de datos completo se genera una matriz de similitudes: S_{ij} $n \times n$.

Esta matriz de similitudes es transformada en una matriz adyacente también $n \times n$ que codificará la fuerza de la correlación entre dos genes. Al ser un método ponderado se utiliza un umbral suave (*soft thresholding*) para reflejar la información continua de los datos de expresión génica [22].

Por lo tanto, la matriz adyacente se define como:

$$A_{ij} = s_{ij}^{\beta} =\mid cor(g_i, g_j) \mid^{\beta} \quad (8)$$

El parámetro β indica como de suave es la transición entre el valor más bajo y el más alto de la coexpresión entre los genes. La metodología WGCNA permite elegir β de tal manera que la red muestre una propiedad llamada escala de topología libre (*Scale Free Topology*, SFT).

A partir de los valores de la matriz adyacente se crea la matriz **TOM** (*topological overlap matrix*, O_T) la cual amortigua el efecto de los genes ruidosos (*outliers*) y da una medida de proximidad de dos genes, teniendo en cuenta los genes en la vecindad. La matriz TOM combina la adyacencia de dos genes y la fortaleza de conexión que tienen esos dos genes con los otros genes contruyendo las redes de coexpresión.

TOM se tranformará en una matriz de distancias (**1-TOM**) la cual se usa como base para el agrupamiento jerárquico (*hierarchical clustering*, HC). En este caso se ha utilizado **flashclust**. Los módulos se definen como las ramas del dendograma generado por el HC, que se pueden identificar utilizando el algoritmo *'dynamic tree-cutting'* (paquete dynamicTreeCut) que aplicado al dendograma (*cutreeDynamic()*) genera una partición de un conjunto de genes disjuntos, $P = P_1, ..., P_k$ [22].

Los genes dentro de un módulo se resumen con el **módulo eigengene**, que puede considerarse como el mejor resumen de los datos de expresión y se define como el primer componente principal de los perfiles de expresión de los genes que pertenecen a un módulo [3].

Como ya se ha mencionado antes, el método WGCNA utiliza el HC cuya principal ventaja es que la estructura del dendrograma (el árbol de distancias que se construye con todos los genes) facilita el problema de encontrar un buen número de grupos (clusters), K. Además, los desarrolladores de WGCNA incluyen un método automatizado para generar el número apropiado de agrupaciones (el mencionado *'dynamic tree cutting'*). Por otro lado, un punto débil de la HC es que los resultados finales dependen en gran medida de cómo se comparan las distancias entre los grupos. Además, una vez que la decisión sobre a qué rama del dendrograma pertenece un gen, esto no se puede deshacer [3].

Para superar estas limitaciones, Botía et al propone una mejora del método convencional WGCNA mediante el refinamiento de los procesos computacionales que generan los diferentes clusters (módulos). Es decir, se le aplica un agrupamiento k-means al método convencional WGCNA (implementado en el paquete de R **km2gcn**, llamado k-means para redes de coexpresión). Este método modifica las particiones pero no a la matriz TOM. En el presente TFM se le llamará método coexp.

El algoritmo k-means convencional establece un número de grupos (clusters) y un número centroides (uno para cada grupo) aleatoriamente. Los centroides son los representantes de cada grupo, de tal manera que un gen (g) pertenece al grupo l si la distancia de dicho punto al centroide de ese grupo es mínima.

El centroide de un grupo l se representa como c_l y se define como:

$$c_l = \frac{1}{T} \sum_{k=1}^{T} g_l \quad (9)$$

siendo T el número de observaciones del cluster l.

En WGCNA la idea de centroide es sustituido por el de eigengene, por lo tanto, en este algoritmo híbrido el componente k-means usará eigengenes como centroides. La distancia en k-means siempre es definida entre un punto perteneciente al dataset y un centroide (eigengen) y es muy común utilizar la distancia euclídea pero esta algoritmo usa como medida de similitud (o distancia) la correlación de Pearson.

Más concretamente, se define la distancia para las redes de co-expresión como $1 - co(g_i, eg_j)$ siendo $co(g_i, eg_j)$ la medida normalizada de la co-expresión entre el perfil de expresión de un gen g_i y el eigengen eg_j, es decir [3]:

$$co(g_i, eg_j) = \frac{1}{2}(1 + cor(g_i, eg_j)) \quad (10)$$

Vale la pena señalar que HC necesita una matriz de distancias entre todos los genes, es decir, $1-TOM$, en cambio k-means necesita una definición de distancia computable entre el gen y el eigengene. Los genes se reasignan de forma iterativa a las particiones inducidas por los nuevos centroides. Si se cumple un criterio de parada (convergencia), el algoritmo finaliza. De lo contrario, se realiza una nueva iteración [3].

El objetivo principal del método coexp es optimizar la generación de los diferentes grupos de genes que genuinamente están relacionados con una función biológica, tipo celular, condición fisiológica determinada etc.

3.5. Modelo de mezcla finito

3.5.1. Definición del modelo de mezcla finito

Las primeras referencias que tenemos con respecto a las distribuciones de mezcla son proporcionadas por Karl Pearson en 1894. K. Pearson fue el que introdujo por primera vez la palabra 'mezcla' en estadística, con el objetivo de resolver el problema de asimetría de una muestra (procedente de un estudio de cangrejos) mediante una distribución normal simétrica compuesta por una mezcla de dos funciones de densidad de probabilidad normal [56]. Posteriormente, en diversas áreas de conocimiento se han desarrollado numerosos estudios estadísticos relacionados con modelos de mezcla. Los matemáticos Rao (1948) y Hasselblad (1966, 1969) utilizaron la estimación de máxima verosimilitud en el contexto de mezcla y McLachlan y Basford (1988), McLachlan y Jones (1988), McLachlan y Krishnan (1997), y McLachlan y Peel (2000) describieron de forma generalizada los modelos de mezcla finitos [49].

En general, los modelos de mezcla son modelos probabilísticos que representan subpoblaciones dentro una población general sin la necesidad de identificar la subpoblación a la que pertenece una observación individual del conjunto total de los datos. También, trata de modelar la distribución estadística de las observaciones pertenecientes a una población general mediante una mezcla (o suma ponderada) de distribuciones (distribución de mezcla) [29]. Las distribuciones de mezcla pueden ser continuas, binarias, ordinales, de recuento, categóricas, fraccionarias, censuradas, truncadas o incluso de supervivencia [49].

Los FMM (modelos de mezcla finitos) se utilizan para clasificar variables en grupos proporcionando una representación natural de la heterogeneidad de los datos con un número finito de "variables latentes"(variables que no se observan directamente, sino que se deducen, a través de un modelo matemático, de otras variables que se observan (se miden directamente)). Es decir, estos modelos están relacionados con el análisis de clases latentes (*latent class analysis*, LCA) que se usan para identificar clases usando información de variables observadas. Debido a su flexibilidad, los FMM se han utilizado ampliamente

para clasificar las observaciones, ajustar el agrupamiento (clustering) y modelar la heterogeneidad de los datos no observados. Se pueden utilizar modelos de mezcla de densidades normales con varianzas iguales para aproximar cualquier distribución continua arbitraria, lo que los convierte en una herramienta muy popular para modelar datos multimodales, sesgados o asimétricos [49].

3.5.2. Estructura del modelo de mezcla finito

Es conveniente proporcionar una formulación paramétrica para la representación del modelo de mezcla, y se adoptará en este apartado, la notación de McLachlan y Peel (2000).

En este tipo de modelos se parte de una muestra aleatoria ($Y = Y_1, Y_2, ...Y_n$) de tamaño n que se supone extraída de una población que es una mezcla aditiva de P subpoblaciones distintas (cada subpoblación definida como p siendo $p = 1, 2, ..., P$) donde cada una está en una proporción (o peso) que se representa como $\pi = \pi_1, ..., \pi_{P-1}$. Estas proporciones están caracterizadas por un parámetro θ siendo $\theta = \theta_1, .., \theta_P$. Concretamente, $\theta = (\mu, \sigma^2)$ y es un parámetro desconocido porque no conocemos ni la media ni la varianza [38].

Y_t es un vector aleatorio q-dimensional con función de densidad de probabilidad $f(yt)$ en R^q. Por lo tanto, $y = y_1, y_2..., y_n$ representa una muestra observada o realización de Y, donde y_t representa un valor observado del vector aleatorio Y_t [38].

La distribución de una variable aleatoria Y_t cuya **función de densidad de probabilidad (FDP)**, $f(y_t; \psi)$ o $p(y_t; \psi)$ se define como:

$$f(yt; \psi) = \sum_{p=1}^{P} \pi_p f(yt; \theta_p) \quad (11)$$

también puede llamarse **distribución de mezcla finita** de P componentes y tiene un vector de parámetros $\psi = (\pi_1, ...\pi_{P-1}, \theta_1, ..., \theta_P)$.

Las proporciones (π_p) están sujetas a la restricción:

$$\sum_{p=1}^{P} \pi_p = 1$$

Si definimos $X_t = Yt, Zt$ como el vector de datos completos cuyo único componente que se observa es Y_t la función de densidad sería [32]:

$$g(x_t; \psi) = \prod_{p=1}^{P} \pi_p f(yt; \theta_p)^{z_{tp}} \quad (12)$$

Los modelos de mezcla se reformulan como un problema de datos incompletos, ya que la asignación de los datos observados es desconocida.

3.5.3. FMM para clustering

Probabilidades posteriores

Cuando se utilizan modelos de mezcla en el contexto de análisis de conglomerados el objetivo es proporcionar una partición de los datos en P grupos, siendo P una cantidad fija. Los 'pesos' o proporciones de la población son interpretados como la probabilidad previa de pertenecer a una población dada. Por lo tanto, $Pr(Z_{tp} = 1) = \pi_p$, representa la probabilidad de asignar una observación a una subpoblación determinada p cuando la única información disponible de los datos son los pesos de cada uno de los grupos [33].

El procedimiento de agrupación (clustering) tiene como objetivo la recuperación de las "variables latentes"$z = z_1, ..., z_n$ habiendo observado $y = y_1, ...y_n$. Una vez que se ha ajustado el modelo de mezcla y se ha estimado su parámetro ψ, se proporciona un agrupamiento probabilístico de las observaciones en términos de sus **probabilidades posteriores** de pertenecer a un subgrupo [33]:

$$\hat{\tau}_{tp} = Pr(Z_{tp} = 1 \mid Y_t = y_t) = \frac{\hat{\pi}_p f(y_t; \hat{\theta})}{\sum_{l=1}^{P} \hat{\pi}_l f(y_t; \hat{\theta}_l)} \quad (13)$$

Las probabilidades $(\hat{\tau}_{t1}, ..., \hat{\tau}_{tP})$ son las probabilidades estimadas de que una observación (y_t) pertenezca al primer, segundo...,P componente de la muestra [45].

Cada punto u observación de los datos puede ser asignado a una población particular mediante la **estimación de la probabilidad a posteriori máxima** (*maximun a posteriori probability*, MAP), definida como [45]:

$$\hat{z}_{tp} = \begin{cases} 1 & if\ p\ = \underset{\ell}{\text{Argmax}}(\hat{\tau}_{tl}) \\ 0 & \text{otherwise} \end{cases}$$

MAP se define, en estadística bayesiana, como la elección de x que maximiza la función de densidad posterior. MAP esta estrechamente relacionada con el **método de estimación de máxima verosimilitud (ML)**, pero incorpora una distribución previa (que cuantifica la información adicional disponible a través del conocimientos previos de un evento relacionado) sobre la cantidad que se desea estimar. Por lo tanto, la estimación de MAP puede verse como una regularización de la estimación ML [45].

Los estudios más comunes de modelos de mezcla son los modelos de mezcla Gaussianos (Fraley and Raftery, 2002) y los modelos de clases latentes (Lazarsfeld y Henry,1968) [38] .

Selección del número de clusters en FMM

La elección del número de componentes (grupos) que constituyen un modelo de mezcla no es un problema trivial y ha sido objeto de estudio en las últimas décadas. Esta elección depende de varios factores, aunque los más relevantes son la distribución de los datos que son modelizados y la forma de las componentes [45]. La elección del número de clusters es siempre la primera pregunta a contestar al realizar un análisis de conglomerados.

La elección del número de clusters se puede enforcar de dos maneras: eligiendo un número de grupos proponiendo diferentes clasificaciones o asimilarlo como un problema de selección del modelo en base a un **criterio de penalización** [45]:

$$\ell_P(y;\hat{\psi}) - \beta pen(P)$$

P es el número de clusters, β es una constante y $pen(P)$ es la **función incremento** con respecto al número de clusters.

El criterio de penalización más utilizado en FMM es el criterio de información bayesiana (apartado 3.2.3). La selección del mejor modelo (m_i) de una colección de modelos ($m_1, ..., m_l$) mediante el enfoque bayesiano fue desarrollado por Schwartz(1978) y tiene como objetivo, en el contexto de los modelos de mezcla finitos, seleccionar el modelo que maximice la probabilidad posterior ($Pr\{m_i \mid Y\}$) [45].

$Pr\{m_i \mid Y\}$ se puede definir, utilizando la fórmula de Bayes como:

$$Pr\{m_i \mid Y\} = \frac{Pr\{Y \mid m_i\} Pr\{m_i\}}{Pr\{Y\}} \quad (38)$$

Considerando el caso donde la $Pr\{m_i\}$ no es informativa, BIC responde a la expresión :

$$BIC = -2Pr(Y \mid m_i) \cong -2\ell(\phi) + klog(n) \quad (39)$$

siendo k número de parámetros de la mezcla, n número de observaciones de la muestra y $\ell(\phi)$ función de verosimilitud.

Utilizando, el método de aproximación de Laplace (Lebarbier and Mary-Huard, 2004),la ecuación 39 se escribiría como:

$$BIC = -2Pr(Y \mid m_i) \cong -2logg_{m_i}(Y;\hat{\psi}_i) + v_i log(n) \quad (40)$$

Otro de los criterios más utilizados en FMM es el criterio de verosimilitud completa integrada (*integrated complete likelihood*, **ICL**) fue propuesto por Biernacky et al. (2010) y es similar al BIC, excepto que añade un nuevo término de penalización denominado **entropía media** que se define como [14]:

$$Ent(g) = -\sum_{i=1}^{g}\sum_{j=1}^{n} \hat{\tau_{ij}} log\hat{\tau} \geq 0$$

g es un número de cluster o componentes, $Ent(g)$ representa la capacidad del modelo de mezcla para dar una partición relevante de la distribución, de tal forma que si los componentes están bien separados, este término tiende a 0 y $\hat{\tau}_{ij}$ denota la probabilidad condicional de que la observación y_j pertenezca a la g-ésimo componente de la mezcla. Así, el número de clúster g' estimado por ICL es siempre menor o igual que el de BIC, debido a este término de penalización adicional que incorpora [14].

3.5.4. Ajustando modelos de mezcla mediante el algoritmo EM

Generalidades del algoritmo EM

El algoritmo EM (*Expectation-Maximization*) fue descrito y presentado por Dempster (1977) pero la idea básica fue introducida anteriormente por Newcomb (1886) quien desarrollo un modelo de mezcla de dos componentes.

El algoritmo EM se usa en estadística para encontrar estimadores de máxima verosimilitud (*maximum likelihood estimators*) de parámetros en modelos probabilísticos que dependen de variables no observables (variables latentes) como por ejemplo los modelos de mezcla finitos. Al plantear modelos de mezcla se crea la necesidad de reformular el conjunto de datos observados como un caso de datos incompletos [45]. Más concretamente, el algoritmo EM trata de obtener la FDP (Función de Densidad de Probabilidad) desconocida a la que pertenecen el conjunto completo de datos.

La FDP (apartado 3.5.2) se puede definir mediante una combinación lineal de P componentes (subpoblaciones) y una serie de parámetros desconocidos, θ. Por lo tanto, FDP queda definida como:

$$FDP = P(y_t) = \sum_{p=1}^{P} \pi_p f(y_t; \theta_p) \quad (14)$$

siendo p un componente dado, π_p las probabilidades a priori de cada grupo (cluster) cuya suma tiene que ser 1 y $f(y_t; \theta_p)$ la función de densidad del componente y.

Cada cluster se corresponde con las respectivas muestras de datos que pertenecen a cada una de las densidades que se mezclan. Se pueden estimar las FDP de forma arbitraria, utilizándose FDP normales n-dimensionales, t-Student, Bernoulli, Poisson y log-normales.

El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el "likelihood"(verosimilitud) de los datos.

Sea $y = y_1, y_2, ..., y_n$ observaciones independientes de una variable aleatoria Y con función de densidad $f(y \mid \theta)$, donde θ es el vector de parámetros desconocidos que queremos estimar; entonces, la **función de densidad conjunta** se define como:

$$f(y \mid \theta) = \prod_{j=1}^{n} f(y \mid \theta) = L(\theta \mid y) \quad (15)$$

donde $L(\theta \mid y)$ es la función de verosimilitud.

El algoritmo EM trata entonces de estimar los parámetros θ maximizando este likelihood. Normalmente, lo que se calcula es el logaritmo de este likelihood, conocido como **log-likelihood** y se define como [32]:

$$\ell(y; \psi) = logL(\theta \mid y) = log \prod_{j=1}^{n} f(y_j \mid \theta) = \sum_{j=1}^{n} logf(y_j \mid \theta) \quad (16)$$

Como se ha explicado anteriormente, si definimos X como el conjunto de datos completos de donde procede x, Y como un conjunto de muestras observadas y Z como el conjunto de muestras latentes se puede establecer que $X = Y * Z$, por lo tanto $x = (y, z)$.

La densidad de los datos observados X puede ser escrita como:

$$g(x;\psi) = f(y,\psi)k(z \mid y;\psi) \quad (17)$$

donde $f(y,\psi)$ es la densidad de los datos observados y $k(z \mid y;\psi)$ es la densidad condicionada de los datos ausentes por elecciones observadas en la muestra.

Esto da lugar a la definición de dos verosimilitudes diferentes, verosimilitud de datos observados/incompletos ($\ell(y;\psi)$) y verosimilitud de los datos no-observados/completos ($\ell^c(x;\psi)$.

Estas dos verosimilitudes están vinculadas mediante la ecuación siguiente [45]:

$$log\ell^c(x;\psi) = log\ell(y;\psi) + logk(z \mid y;\psi) \quad (18)$$

siendo

$$\ell^c(x;\psi) = \sum_{t=1}^{n} logf(x_t;\psi) \quad (19)$$

y

$$logk(z \mid y;\psi) = \sum_{t=1}^{n}\sum_{p=1}^{P} logE(Z_{tp} \mid Y_t = y_t) \quad (20)$$

Formulación general del algoritmo EM

Como se ha comentado en el apartado anterior, el algoritmo EM es un método que sirve para encontrar la estimación de máxima verosimilitud, es decir, calcula los parámetros desconocidos θ de una distribución. Concretamente, consiste en la optimización indirecta de las verosimilitudes de los datos incompletos a través de la optimización iterativa de la expectativa condicional de las verosimilitudes de los datos completos utilizando el ajuste actual para ψ [31].

Si definimos ψ^h como el valor del parámetro en la iteración h, se puede definir la función log-verosimilitud [31]:

$$\ell(y;\psi) = Q(\psi;\psi^h) - H(\psi;\psi^h) \quad (21)$$

$$Q(\psi;\psi^h) = E_{\psi^h}\left\{\ell^c(X;\psi) \mid Y\right\} \quad (22)$$

$$H(\psi;\psi^h) = E_{\psi^h}\left\{logk(Z \mid Y;\psi) \mid Y\right\} \quad (23)$$

$E_{\psi^h}\{.\}$ es el operador 'expectación' o esperanza dado el ajuste actual ψ^h para ψ.

Para el caso especial de los modelos de mezcla las log-verosimilitudes también pueden ser escritas como [31]:

$$\ell(y;\psi) = \sum_{t=1}^{n} log f(y_t;\psi) = \sum_{t=1}^{n} log \left\{ \sum_{p=1}^{P} \pi_p(y_t;\theta_p) \right\} \quad (24)$$

$$\ell^c(x;\psi) = \sum_{t=1}^{n} log g(x_t;\psi) = \sum_{t=1}^{n} \sum_{p=1}^{P} z_{tp} log \left\{ \pi_p f(y_t;\theta_p) \right\} \quad (25)$$

Etapas del algoritmo EM

El algoritmo EM se divide en dos etapas, la etapa expectación o paso E y la etapa de maximación o paso M.

1. **Etapa de expectación:** Utiliza los valores de los parámetros, iniciales o proporcionados por el paso maximization, obteniendo diferentes formas de la FDP buscada. En primer lugar se calcula un valor inicial para θ^h. y después se calcula $Q(\psi;\psi^h)$. Esta función es necesaria para maximizar $log\ p(x \mid \theta)$ y queda definida como:

$$Q(\psi;\psi^h) = \sum_{t=1}^{n} \sum_{p=1}^{P} E_{\psi^h} \left\{ Z_{tp} \mid Y_t = yt \right\} log \left\{ \pi_p f(y_t;\theta_p) \right\} \quad (26)$$

con

$$E_{\psi^h} \left\{ Z_{tp} \mid Y_t = yt \right\} = \tau_{tp}^h$$

donde

$$\hat{\tau}_{tp}^h = \frac{\pi_{h-1} f(y_t;\theta_p^{h-1})}{\sum_{l=1}^{P} \pi_l^{h-1} f(y_t;\theta_l^{h-1})} \quad (27)$$

$\hat{\tau}_{tp}^h$ representa las probabilidades estimadas de que el punto y_t pertenezca a los componentes $p = 1, ..., P$ de la mezcla, siendo la mayor probabilidad entre ellas la que es elegida. Por lo tanto, la función Q queda definida como:

$$Q(\psi;\psi^h) = \sum_{t=1}^{n} \sum_{p=1}^{P} \tau_{tp}^h \left\{ Z_{tp} \mid Y_t = y_t \right\} log \left\{ \pi_p f(y_t;\theta_p) \right\} \quad (28)$$

Q es el valor esperado (*expected value*) del log-verosimilitud de θ con respecto a la distribución de Z dada una X y a la estimación de los parámetros θ^h [31].

En ésta etapa se busca maximizar $log f(y_t;\theta_p)$ pero desconocemos x y la función Q depende de θ pero también de θ^h que es la suposición inicial.

2. **Etapa de maximización:** Obtiene nuevos valores de los parámetros a partir de los datos proporcionados por el paso expectación [45].

   El paso M requiere una maximización global de la función Q con respecto a ψ dada una estimación actualizada de ψ^{h+1}. Es decir, en este paso se elige $\psi^{h+1} = \underset{\psi}{Argmax}\left\{Q(\psi;\psi^h)\right\}$ que es el parámetro que maximiza $Q(\psi;\psi^h)$.

   La estimación de las proporciones de mezcla se calculan a través de la maximización restringida del log-verosimilitud de datos incompletos [31]:

$$\hat{\pi_p}^{h+1} = \sum_{t=1}^{n} \tau_{tp}^h \quad (29)$$

Matriz de información usada en el algoritmo EM

La matriz de información es una manera de medir la cantidad de información que una variable observable aleatoria tiene de los parámetros desconocidos θ. Formalmente, es la varianza del *score* de la información observada. Se denomina "score.ª la derivada parcial con respecto a θ del logaritmo natural de la función de probabilidad Bajo ciertas condiciones de regularidad, si θ es el parámetro verdadero (es decir, X se distribuye realmente como $f(X;\theta)$, se puede mostrar que el valor esperado de la puntuación es 0 [45].

Por lo tanto, una vez estimado los parámetros θ, se evalúa los errores estándar de $\hat{\psi}$ y se hace mediante la matriz de información esperada (I), la cual, queda definida como [45]:

$$I(\psi) = E_Y\left\{\frac{\partial^2}{\partial\psi\partial\psi^T}\ell(Y;\psi)\right\} \quad (30)$$

En la práctica, esa cantidad se estima con la **matriz de información observada**, $I(\psi;Y)$, con la relación [45]:

$$I(\psi) = E_Y\left\{I(\psi;Y)\right\}$$

La derivación de la matriz de información, $I(\psi;y))$, es simplificada por el principio de información faltante (*missing information principle*) introducido por Woodbury (1971).

Principio de información faltante

Considerando la formulación de los modelos de mezcla como un problema de datos faltantes, definimos la **matriz de infomación de datos completos** en función de los log-verosimilitudes de los datos completos como [45]:

$$I^c(\psi; x) = \left\{ \frac{\partial^2}{\partial\psi\partial\psi^T} \ell^c(x; \psi) \right\} \quad (31)$$

Como los datos incompletos y los datos completos se relacionan con la fórmula 32 [45]:

$$\ell(y; \psi) = \ell^c(x; \psi) - logk(z \mid y; \psi) \quad (32)$$

la **matriz de información de datos observados** se define como:

$$I(\psi; y) = I^c(\psi; x) - I^m(\psi, z) \quad (33)$$

A partir de las ecuaciones 32 y 33 se define la **matriz de información de datos faltantes** como:

$$I^m(\psi; z) = \frac{\partial^2}{\partial\psi\partial\psi^T} logk(z \mid y; \psi) \quad (34)$$

La matriz de información de datos faltantes es la 'información perdida' como consecuencia de haber observado sólo y y no z.

Al añadirle a ambos lados el término expectación en función de Y, la ecuación 33 se puede escribir como:

$$I(\psi; y) = E_{X|Y} \left\{ I^c(\psi; X) \right\} E_{Z|Y} \left\{ I^m(\psi; x) \right\} \quad (35)$$

Por otra parte, si extraemos la matriz de información observada en función de los la verosimilitud de los datos completos (X) podemos introducir los conceptos:

$$S(y; \psi) = \frac{\partial^2}{\partial\psi} \ell(y; \psi)$$

$$S^c(x; \psi) = \frac{\partial^2}{\partial\psi} \ell^c(x; \psi)$$

dando lugar a la re-difinición de la matriz de datos faltantes (Louis, 1982) [45]:

$$E_{z|Y} I(\psi; Z) = E_{X|Y} \left\{ S^c(X; \psi) S^c(X; \psi)^T \right\} - S(Y; \psi) S(y; \psi)^T \quad (36)$$

La ecuación 36 nos indica que todas las expectaciones condicionales calculadas (ecuación 35) pueden ser calculadas en el algoritmo EM usando una expectación de gradiente y curvatura de la verosimilitud de los datos completos y por lo tanto la ecuación 35 puede ser reformuladar de la siguiente manera:

$$I(\hat{\psi}; y) = E_{X|Y}\left\{I^c(\psi; X)\right\} \mid_{\psi=\hat{\psi}} - E_{X|Y}\left\{S^c(X;\psi)S^c(X;\psi)^T\right\} \mid_{\psi=\hat{\psi}} \quad (37)$$

considerando que $\psi = \hat{\psi}$ y $S(y;\psi) = 0$.

Convergencia del algoritmo EM

La convergencia del algoritmo EM fue descrita por Wu 1983 y Dempster (1977) y se define como el cambio lo suficientemente pequeño en los parámetros θ o en la función de log-verosimilitudes que permite la finalización del algoritmo EM.

La secuencia de valores log-verosimilitudes es monótona, es decir que no decrece tras una iteración del algoritmo EM, $\ell(\phi^{t+1}) \geq \ell(\phi^t)$ [14].

3.6. Métricas de comparación de clustering

3.6.1. Descripción de las MCC

En las últimas décadas, se han escrito miles de artículos que proponen cientos de algoritmos de agrupamiento. El esfuerzo por encontrar mejores métodos de agrupación es casi imposible sin el desarrollo de medidas efectivas para la comparación de agrupaciones, un área de investigación que también ha recibido mucha atención.

Cuando se realiza un análisis de conglomerados es importante evaluar el resultado de los algoritmos que implementamos para realizar dicho análisis, sin embargo, es muy difícil definir cuando el resultado de un agrupamiento es aceptable. Por esta razón, existen una serie de técnicas y/o métricas para la **validación del agrupamiento** realizado.

La validación del agrupamiento o 'validación de las particiones' nace como la necesidad de encontrar los parámetros de entrada óptimos que apuntan a un buen agrupamiento, además de disminuir la subjetividad del clustering.

Hay dos tipos de validación:

1. Validación interna: utiliza métricas que evalúan el agrupamiento teniendo en cuenta solo la información que nos proporcionan los datos.

2. Validación externa: utiliza métricas que miden la calidad del agrupamiento conociendo información externa de antemano (distorsión, verosimilitude etc). Esta validación se utiliza para escoger un algoritmo de clustering óptimo sobre un conjunto de datos específicos.

En este TFM nos vamos a centrar en las métricas de validación externa para comparar cómo de parecidos son los k-means, FMM y coexp.

Para valorar la similitud entre dos particiones (p.ej **U** y **V**) se debe de introducir una serie de cantidades que miden su información en común [54].

$$H(U) = -\sum_{i=1}^{K} \frac{a_i}{N} log(\frac{a_i}{N}) \text{ (Entropía para U) (41)}$$

$$H(V) = -\sum_{j=1}^{K'} \frac{b_j}{N} log(\frac{b_j}{N}) \text{ (Entropía para V) (42)}$$

$$H(U,V) = -\sum_{i=1}^{K}\sum_{j=1}^{K'} \frac{n_{ij}}{N} log(\frac{n_{ij}}{N}) \text{ (Entropía conjunta para U y V) (43)}$$

$$I(U,V) = -\sum_{i=1}^{K}\sum_{j=1}^{K'} \frac{n_{ij}}{N} log(\frac{n_{ij}/N}{a_i bj/N^2}) = H(U) + H(V) - H(U,V) = MI \text{ (44)}$$

La **Información Mútua** (*Mutual Information*, MI) o transinformación mide la cantidad de independencia mutua que hay entre dos particiones, es decir, mide la reducción de la incertidumbre (entropía) de una partición (U) debido al conocimiento del valor de otra partición (V). Por lo tanto, la información mutua es usada para medir la información compartida de dos particiones y evaluar su similaridad [54].

En este TFM se utilizaran medidas de validación externa, más concretamente, el índice de Rand ajustado (*Adjusted Rand Index*,ARI), variación de la información (*Variation of information*, VI), variación de la información normalizada (*normalized variation of information*, NVI) y la distancia de información normalizada (*normalised information distance*, NID) que seran presentadas en el apartado 3.6.2.

3.6.2. Métricas de validación externa

Índice de Rand Ajustado

Este índice fue propuesto por Hubert and Arabie (1985) como modificación del índice de Rand (Rand, 1971).

El índice de Rand (RI) es muy sensible tanto al número de clusters como a la cantidad de elementos y su valor esperado para dos particiones aleatorias no toma un valor constante (p.ej cero). Con la intención de resolver estas limitaciones se crearon otras métricas como por ejemplo el índice de Rand ajustado (ARI) cuya función principal es medir el grado de similitud entre dos particiones generadas a través de un algoritmo de agrupamiento. ARI es un valor numérico que oscila entre 0 y 1, y cuanto mayor sea, mayor será la semejanza entre los resultados de las dos particiones que se comparan.

ARI asume una distribución hipergeométrica generalizada como hipótesis nula, es decir, se definen aleatoriamente las dos particiones con un número fijo de grupos clusters y un número fijo de elementos en cada grupo (el número de clusters de las dos particiones no necesariamente tiene que ser el mismo). Entonces, el ARI es la diferencia (normalizada) del RI y su valor esperado bajo la hipótesis nula [54].

$$ARI = \frac{RI - RI_{esperado}}{RI_{maximo} - RI_{esperado}} \quad (45)$$

Dado un conjunto de N objetos, $X = x_1, ..., x_N$, sea $U = (u_1, ..., u_k)$ y $V = v_1, ..., v_{k'}$ particiones de X, la información de la superposición (overlaps) entre U y V queda resumida con una matriz de contingencia $K \times K'$, la cual queda resumida en la tabla 1 donde n_{ij} es interpretado como el número de observaciones de X comunes para los clusters U_i y V_j, a_i el número de objetos que hay en u_i y b_j el número de objetos que hay en v_j [10].

U/V	v_1	v_2	...	$v_{K'}$	Total
u_1	n_{11}	n_{12}	...	$n_{1K'}$	a_1
u_2	n_{21}	n_{22}	...	$n_{2K'}$	a_2
...	...	...	...		
u_K	n_{K1}	n_{K2}	...	$n_{KK'}$	a_K
Total	b_1	b_2	...	$b_{K'}$	$\sum n_{i,j} = N$

Tabla 1: Tabla de contingencia para las particiones U y V de X.

Dadas dos particiones U y V, ARI se define en función de la tabla 3.1 como:

$$ARI(U,V) = \frac{\sum_{j,i}\binom{n_{i,j}}{2} - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{N}{2}}{1/2(\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}) - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{N}{2}} \quad (46)$$

Si el resultado es cero, indica que las particiones son independientes.

Variación de la información

Esta métrica introducida por Meila (2003) mide la distancia entre dos particiones del mismo conjunto de datos.

VI se define como (Strehl y Ghosh):

$$VI(U,V) = H(U,V) - I(U,V) = 2H(U,V) - H(U) - H(V) \quad (47)$$

$$VI(U,V) = H(U \mid V) + H(V \mid U) \quad (48)\text{(entropías condicionales)}$$

Propiedades VI

1. La variación de la información satisface los axiomas métricos.

   - No negatividad.

   $$VI(U,V) \geq 0$$

   - Simetría.

   $$VI(U,V) = VI(V,U)$$

   - Desigualdad triangular.

     Dadas tres particiones cualesquiera (U, V, W):

   $$VI(U,V) + VI(V,W) \geq VI(U,W)$$

2. Invarianza.

   El valor de $VI(U,V)$ depende solo del tamaño relativo de los grupos. No depende directamente del número de observaciones del conjunto de datos.

3. $VI(U,V) \leq log(n)$.

4. Si U y V tienen como mucho una cantidad de $K*$ grupos cada una, siendo $K* \leq \sqrt{n}$ entonces $VI(U,V) \leq 2log(K*)$.

5. Asume que V es obtenido desde U mediante la división de U_k dentro de grupos $V_{k1}, ..., V_{km}$, por lo tanto:

   $$VI(U,V) = VI(U, U \times V) + VI(V, U \times V)$$

6. Dados dos particiones cualesquiera, U y V, se establece que:

   $$VI(U,V) \geq VI(U, U \times V)$$

7. Aditividad con respecto al refinamiento.

   Dada una partición cualesquiera, U, cuya refinación[4] es V y la refinación de esta es W, se estable que:

$$VI(U,W) = VI(U,V) + VI(V,W)$$

8. Aditividad con respecto a la unión.

   Dados dos particiones U y V (V se obtiene a partir de U), se establece que:

$$VI(U,V) = VI(U,U \times V) + VI(V,U \times V)$$

9. Aditividad convexa.

   Dadas 3 particiones U,V y W(donde W se obtiene de V y V de U) y $P(k)$ (representa la proporción de observaciones que pertenecen a U_k) se define que:

$$VI(V,W) = \sum_{k=1}^{K} P(k)VI(V_k, W_k)$$

El VI está limitado por 0 (cuando los dos grupos son idénticos) y siempre están delimitados por $log(N)$, aunque se pueden alcanzar límites más estrechos según el número de agrupaciones [37].

3.6.3. Variación de la información normalizada

NVI es la variación de la información normalizada. La normalización facilita la interpretación y comparación de las particiones en diferentes condiciones (Strehl y Ghosh, 2002; Luo et al., 2009). NVI también tiene la propiedad de aditividad convexa como la VI pero no sigue los axiomas métricos. El valor de NVI se encuentra entre 0 y 1, obteniendo 1 al comparar dos particiones idénticas.

NVI se define como [10]:

$$NVI(U,V) = 1 - \frac{I(U,V)}{H(U,V)} \quad (50)$$

[4]Cuando una partición se obtiene de la división de otra se dice que la segunda es la refinada de la primera.

3.6.4. Distancia de información normalizada

La distancia de información normalizada (*normalised information distance*, NVI) es una medida de 'propósito general' para la validación del agrupamiento, la comparación y el diseño de algoritmos, ya que posee simultáneamente varias propiedades útiles e importantes. El valor de NID se encuentra entre 0 y 1, obteniendo 1 al comparar dos particiones idénticas.

NID se define como [10]:

$$NID = 1 - \frac{I(U,V)}{max\{H(U)H(V)\}} \quad (51)$$

3.7. Visualización del clustering

Para representar los datos Cortex es necesario **reducir la dimensionalidad** ya que mejora su visualización en función de las particiones generadas por los diferentes métodos de clustering utilizados en el presente TFM (k-means, FMM y coexp).

3.7.1. Método componentes principales

El método de componentes principales (*principal component analysis*, PCA) descrito por Karl Pearson (1857-1936) transforma las variables originales de un conjunto de datos en unas variables denominadas **componentes principales**, las cuales no estan correlacionadas linealmente. Los componentes principales se ordenan en función de la varianza original, es decir, de la cantidad de información de los datos que llevan incorporada. Cuanto mayor sea su varianza mayor es la cantidad de información que lleva incorporado dicho componente. Es un método particularmente útil para reducir la dimensionalidad de un conjunto de datos y representarlos adecuadamente con la mínima pérdida de información. Los primeros componentes principales describen la mayor parte de la varianza de los datos [19].

Siendo p variables originales, $X = X_1, X_2..X_p$ y k variables çomponentes principales", $Z = Z_1, Z_2.., Z_k$, el primer componente principal, al igual que los restantes, se define como la combinación lineal de las variables originales:

$$Z_{1i} = U_{11}, X_{1i}, U_{12}X_{2i}...U_{1p}X_{pi} \quad (52)$$

Para el conjunto de las n observaciones muestrales, la ecuación puede expresarse matricialmente como:

$$\begin{bmatrix} Z_{11} \\ Z_{12} \\ ... \\ Z_{1n} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{21} & ... & X_{p1} \\ X_{12} & X_{22} & ... & X_{p2} \\ ... & ... & ... & ... \\ X_{1n} & X_{2n} & ... & X_{pn} \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{12} \\ ... \\ u_{1p} \end{bmatrix}$$

siendo su notación abreviada: $Z_1 = Xu_1$.

La media de Z_1 es cero, es decir, $E(Z_1) = E(Xu_1) = 0$ y su **varianza** se define como:

$$V(Z_1) = \frac{\sum_{i=1}^{n} Z_1^2}{n} = \frac{Z_1' Z_1}{n} = \frac{u_1' X' X u_1}{n} = u_1' \left[\frac{X'X}{n}\right] u_1 = u_1 V u_1 \ (53)$$

Existen dos formas básicas de aplicar el PCA:

1. Método basado en la matriz de covarianzas (el más común), que se usa cuando los datos son dimensionalmente homogéneos y presentan valores medios similares.

   La expresión $\frac{X'X}{n}$ (matriz de inercia) es la matriz de covarianzas denominadas V (ecuación 53).

   Para obtener Z_1 maximizamos la varianza con la restricción $\sum_{j=1}^{p} u_1 u_1 = 1$ lo que da lugar a la aplicación del método de los multiplicadores de Lagrange, es decir:

   $$L = u_1' V u_1 - \lambda(u_1' u_1 - 1) \ (54)$$

   que derivando u_1 e igualando a 0, obtenemos que $v(Z_1) = \lambda$.

   Por tanto, para maximizar $V(Z_1)$ hay que tomar el mayor valor propio λ de la matriz V.

   Tomando λ_1 como el mayor valor propio de V y u_1 como su vector propio asociado normalizado ($u_1' u_1 = 1$) se define el vector de ponderaciones que se aplica a las variables iniciales para obtener el primer componente principal, componente que se expresa como [19]:

   $$Z_1 = Xu_1 \ (55)$$

2. Método basado en la matriz de correlación se utiliza cuando los datos no son dimensionalmente homogéneos o el orden de magnitud de las variables aleatorias medidas no es el mismo. La expresión $\frac{X'X}{n}$ es la matriz de correlaciones llamada R.

Este método da buenos resultados y tiene numerosas aplicaciones pero no es muy adecuado para datos no lineales.

3.7.2. T-SNE

La herramienta T-SNE (ïncrustación de vecinos estocásticos distribuidos en t", *t-Distributed Stochastic Neighbour Embedding*) es una de las técnicas estadísticas más adecuadas para reducir la dimensionalidad con el objetivo de pre-procesar o visualizar un conjunto de datos de alta dimensionalidad.

Esta herramienta fue desarrollada por Laurens van der Maaten y Geoffrey Hinton (2008). Es una técnica que reduce el espacio dimensional a 2 o 3 dimensiones (menos dimensiones que PCA) usando una

medida de similitud, como por ejemplo la distancia euclídea, que considera las similitudes y discrepancias entre pares de datos pero conservando las estructuras locales de estos.

La metodología T-SNE se divide en dos etapas principales [25]:

1. Se define la probabilidad condicional sobre pares de objetos de alta dimensión de tal manera que objetos similares tienen una alta probabilidad de ser seleccionados, mientras que los objetos diferentes tienen una probabilidad extremadamente pequeña de serlo.

   Si consideramos como medida de similitud la distancia euclídea, partiendo de un conjunto de objetos, $X = x_1, ...x_N$, se define como la similitud del punto x_j al punto x_i como la probabilidad condicional, $p_{j|i}$), es decir:

   $$p_{j|i} = \frac{exp(- \mid\mid x_i - x_j \mid\mid^2)/2\sigma_i^2)}{\sum_{k \neq i} exp(- \mid\mid x_i - x_j \mid\mid^2)/2\sigma_i^2)} \quad (56)$$

   donde σ_i es la varianza centrada en x_i.

2. Se define una distribución de probabilidad de similitud, $q_{j|i}$, sobre los puntos en el mapa de baja dimensión y se minimiza la divergencia de **Kullback-Leibler** (es una medida no simétrica de la similitud o diferencia entre dos funciones de distribución de probabilidad) entre las dos distribuciones con respecto a las ubicaciones de los puntos en el mapa.

   La distribución de probabilidad de similitud, $q_{j|i}$, se define como [25]:

   $$q_{j|i} = \frac{(1+ \mid\mid y_i - y_j \mid\mid^2)^{-1}}{\sum_{k \neq i}(1+ \mid\mid x_i - x_j \mid\mid^2)^{-1}} \quad (57)$$

   siendo y_i e y_j dos puntos en el mapa [17].

   Para minimizar la divergencia entre dos distribuciones P y Q se cumple :

   $$C = KL(P \mid\mid Q) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}} \quad (58)$$

   siendo $KL(P \mid\mid Q)$ la divergencia de Kullback-Leibler [17].

Capítulo 4

Metodología y herramientas

En este capítulo se describe la metodología computacional y todas las herramientas usadas para obtener los resultados y completar los objetivos del TFM.

4.1. El lenguaje R

La herramienta principal usada en el presente TFM es el lenguaje de programación R versión 3.5.1 (2018-07-02).

R es un entorno y lenguaje de programación enfocado al análisis estadístico. Es el lenguaje de código abierto y software libre más utilizado por la comunidad científica y popular en el campo de la minería de datos, biomedicina, bioinformática y analítica financiera. Ofrece una gran cantidad de técnicas para la exploración, manejo, análisis, procesamiento y visualización de una gran cantidad de datos [18].

El lenguaje R fue diseñado en por Robert Gentleman y Ross Ihaka en 1993 en la Universidad de Auckland utilizando los lenguajes de programación S y Scheme. R se distribuye libremente (tiene una licencia GNU/GLP) y está disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux. Los usuarios de R software pueden ser capaces de consultar, compartir y modificar el código fuente. Los miembros originales que desarrollaron R crearon la fundación R (**R Foundation**). El desarrollo actual de la plataforma de R es responsabilidad del **R Development Core Team** (Equipo Central de Desarrolladores R). Si tecleas en la consola de R la función *contributors()* aparecerá una lista con los nombres de los autores iniciales del equipo central de desarrolladores R [51] .

R permite a los usuarios:

- Definir funciones.
- Integrar bases de datos mediante paquetes de R o externas (con diferentes extensiones).
- Cálculos numéricos, programación orientada a objetos (POO).
- Visualizar y explorar gran cantidad de datos.
- Integrar Latex u otros editores de texto.
- Utilizar herramientas de econometría específicas.
- Generar documentos interactivos, utilizando la herramienta R markdown. Esta herramienta te permite integrar en dichos documentos otros lenguajes de programación como python, Perl, Ruby o html.

Una de las características de R más importantes es la capacidad de incorporar en cualquier momento nuevas funciones capaces de realizar nuevas tareas.

Existen diversas interfaces gráficas (**graphical user interface**, GUI) que facilitan el trabajo con el lenguaje R, como por ejemplo R Commander, JGR (*Java GUI for R*) o **R studio** [62].

R studio es un entorno de desarrollo integrado (*Integrated Development Enviroment*, IDE) para el lenguaje de programación R que incluye una consola de comandos (donde se ejecuta el código), un panel en el que puedes guardar las variables declaradas, los datasets y las funciones (*Environment*) y consultar el historial y las conexiones. También hay un panel que muestra los diferentes directorios de trabajo, visualizar gráficas o consultar información sobre los diferentes paquetes de R [52].

R incluye un repositorio de paquetes llamado **CRAN** (*Comprensive R Archive Network*) con 13.528 paquetes disponibles diferentes [63]. Por otra parte, hay paquetes que se pueden integrar en R para su utilización que no estan incluidos en ese repositorio (fuentes externas). Se denomina paquete de R al conjunto de funciones, datos y código que se almacenan en una carpeta con un estructura bien definida. Se pueden instalar todos los paquetes incluidos en R mediante la función *install.packages()*.

R studio te permite utilizar una herramienta llamada R markdown (a partir de la versión 0.98.932) con la que puedes generar documentos que integran texto, código, tablas y gráficas interactivas e imágenes (archivos .Rmd). Esta herramienta fue creada por John Gruber en 2004.

Dichos documentos puede exportarse en tres formatos:

- html: existe la posibilidad de que el documento se exporte como una página web o como una presentación html (similar a Power Point). Las últimas versiones de Rstudio permiten incluso generar una página web completa.
- doc: para ser directamente editados con Microsoft Word o LibreOffice Writer.
- pdf: para ésta última opción es preciso que el ordenador del usuario disponga de una instalación válida de LaTeX. LaTeX es un completo (y complejo) sistema de edición de textos de código abierto que puede descargarse libremente para Windows (Miktex), para Mac (Mactex) y para Linux (TexLive).

Esto es posible gracias a los paquetes **knirt** (Yihui Xie, 2007) y **pandoc** (John McFarlane, 2006) del repositorio CRAN. R markdown tiene una sintaxis simple basada en una serie de 'marcas' para incorporar al documento encabezados, subencabezados, cambiar el tipo de letras o insertar gráficos y enlaces etc.

La plataforma R tiene un soporte muy bueno ya que se actualiza con regularidad. Su última versión estable es la 3.5.1 (2 de julio de 2018).

En el presente TFM se ha utilizado la plataforma R studio en el sistema operativo Linux/Ubuntu. El método coexp requiere exclusivamente el sistema operativo Linux/Ubuntu o Mac mientras que los demás métodos, k-means o FMM pueden ser ejecutados en el sistema operativo Windows.

4.2. Accediendo a los datos via método coexp

Los datos que se utilizan en el presente TFM pertenecen al consorcio GTEx (archivos .rds) y estan incluidos en el paquete **km2gcn** (paquete externo). Éste paquete también incluye una gran cantidad de funciones para acceder a los datos, analizarlos, procesarlos y crear redes de coexpresión génica (GCN) a partir de dichos datos. Este paquete se encuentra en el repositorio Github. Se van a utilizar los datos GTEx V6 y para acceder a ellos es imprescindible acceder a la dirección [5] y clonar la carpeta **coexp**. Una vez adquirido todos los archivos y carpetas puedes acceder a ellas utilizando R studio. La carpeta coexp contiene numerosas subcarpetas, archivos .R y .rds (datos GTEx).

[5] `https://github.com/juanbot`

El primer paso para acceder a nuestros datos es cargar en R studio los archivos: geneannot.R, coexpression.R y todos los archivos .rds que se encuentran en la carpeta **gtexv6**. Para cargarlo se utiliza la función básica de R llamada *source()*. Al ejecutar dichos archivos en tu consola, se guardan los archivos .rds que contienen datos de expresión génica de 47 tejidos procedentes de GTEx y un gran número de funciones y objetos.

Para consultar qué datos de expresión están disponibles en la carpeta coexp puedes ejecutar la función *coexp.initDb()* y como se trata de datos GTEx, debemos de poner como argumento para esta función dicha palabra entre comillas.

Para acceder a los datos de expresión génica y guardarlos en un objeto para su posterior utilización se debe de utilizar la función *coexp.getExprDataFromTissue()*. De los **47 tejidos disponibles** (tabla 2) puedes acceder a los datos del tejido/os que quieras simplemente cambiando los parámetros de la función. En nuestro caso vamos a utilizar los datos de **Cortex**.

Tejidos disponibles GTEX				
AdiposeSub	CellsFirbroblasts	FCortex	Ovary	Stomach
AdiposeVisceral	CellsLymphocytes	HeartAtrialApp	Pancreas	Substantianigra
AdrenalGland	Cerebellum	HeartLeftVent	Pituitary	Testis
Amygdala	CerebHemisphere	Hippocampus	Prostate	Thyroid
AntCingCortex	ColonSigmoid	Hypothalamus	Putamen	Uterus
ArteryAorta	ColonTransverse	Liver	SkinLowerLeg	Vagina
ArteryCoronary	**Cortex**	Lung	SkinSuprapubic	WholeBlood
ArteryTibial	EsophGastJunction	MuscleSkeletal	SmallIntestine	
Breast	EsophMucosa	NerveTibial	Spinalcord	
Caudate	EsophMuscularis	NucAccumbens	Spleen	

Tabla 2: Tejidos disponibles GTEx.

Por otra parte, el archivo naming.R de coexp-master contiene otras funciones para convertir las identidades Ensembl (nombres de las columnas) en sus símbolos u otras nomenclaturas para facilitar su estudio o consulta.

4.3. Descripción y exploración de los datos

En este apartado, se describen y exploran los datos de partida que se utilizan en el presente TFM.

Como se ha comentado en el apartado anterior, en este TFM se utilizan datos de expresión génica incluidos en la base de datos GTEx V6 que incluye una gran cantidad de muestras distribuidas en 47 tejidos post-morten (47 matrices de expresión génica) cuya estructura se caracteriza por tener como variables lo genes (columnas) y como observaciones la expresión génica (ya transformados en valores RPKM,

Reads Per Kilobase Million) ordenados por filas que representan las diferentes muestras. La cantidad de genes (variables) depende de cada tejido ya que se seleccionan solamente aquellos que muestran valores de RPKM > 0.1 en más del 80 % de las muestras. Se descartan tejidos con menos de 60 muestras por razones de potencia estadística. Antes de que estos datos se utilicen para la creación de redes de co-expresión génica se corrige el efecto del batch, sexo, edad y el PMI y también el efecto de las covariables desconocidas mediante el uso de análisis de variables sustitutas (*Surrogate Variable Analysis*, SVA). Para la construcción de la red, utilizamos los residuos obtenidos mediante la regresión de los valores de la expresión RPKM con las covariables conocidas y desconocidas mediante un modelo lineal generalizado [3].

De los 47 tejidos disponibles, se escogen los datos de cortex cerebral que tienen 114 filas (muestras) y 20.601 columnas (genes). Se obtiene la transpuesta de estos datos porque nos interesa **clasificar los genes en clusters** (*gene-based clustering*) para aplicar los métodos de clustering k-means y FMM. Un esquema de la estructura de los datos se muestra en la tabla 3.

	Muestra 1	**Muestra 2**	...	**Muestra 114**
Gen 1	0.8528601	0.0434527	...	0.0656827
Gen 2	0.0578708	-0.0935688	...	-0.3012500
Gen 3	0.3483538	-0.1553445	...	-0.4947187
...	...	...	...	...
Gen 20601	0.03910096	-0.1712313	...	0.5447101

Tabla 3: Estructura de los datos transpuestos Cortex.

Los valores de los perfiles de expresión de cortex cerebral van de -6.434577 a 10.182974 y podemos ver que no hay valores nulos o faltantes ya que son eliminados en el pre-procesamiento. La alta dimensionalidad de los datos dificulta la exploración de estos, por tanto se considera seleccionar una muestra al azar por muestreo simple aleatorio ya que todas las variables son del mismo tipo (genes) y tienen la misma importancia (muestra de 20 filas y 10 columnas). Aplicando la función genérica *summary()* por columnas a esta muestra de los datos originales podemos obtener los valores de la media, mediana, el valor mínimo, máximo etc (figura 5).

```
GTEX.131XW.3126.SM.5LZUC GTEX.T5JC.2426.SM.3NMDB GTEX.13NYB.3026.SM.5IJD7 GTEX.X4XX.3026.SM.3NMB2
Min.   :-0.51620         Min.   :-0.06800        Min.   :-0.178343        Min.   :-0.44491
1st Qu.:-0.31355         1st Qu.: 0.03937        1st Qu.: 0.001397        1st Qu.:-0.01694
Median :-0.25222         Median : 0.12779        Median : 0.136483        Median : 0.04775
Mean   :-0.21302         Mean   : 0.16513        Mean   : 0.119301        Mean   : 0.09565
3rd Qu.:-0.06879         3rd Qu.: 0.23000        3rd Qu.: 0.188615        3rd Qu.: 0.23328
Max.   : 0.19362         Max.   : 0.70604        Max.   : 0.484761        Max.   : 0.47775
GTEX.Y8DK.0826.SM.4TT3T GTEX.1192X.3126.SM.5N9BY GTEX.13QIC.2926.SM.5J2NF GTEX.WVLH.3026.SM.3MJG9
Min.   :-0.63352        Min.   :-0.33179        Min.   :-0.310974        Min.   :-0.58923
1st Qu.:-0.15496        1st Qu.:-0.07124        1st Qu.:-0.110146        1st Qu.:-0.16742
Median : 0.01093        Median : 0.03792        Median : 0.004428        Median :-0.01388
Mean   :-0.03031        Mean   : 0.07531        Mean   : 0.008754        Mean   : 0.06016
3rd Qu.: 0.15338        3rd Qu.: 0.19273        3rd Qu.: 0.091771        3rd Qu.: 0.29566
Max.   : 0.41346        Max.   : 0.80405        Max.   : 0.464243        Max.   : 0.93995
GTEX.13S7M.3126.SM.5RQJQ GTEX.13NYS.3126.SM.5KLYV
Min.   :-0.7786          Min.   :-0.52777
1st Qu.:-0.3635          1st Qu.:-0.01186
Median :-0.1992          Median : 0.08292
Mean   :-0.1872          Mean   : 0.10070
3rd Qu.: 0.1023          3rd Qu.: 0.21607
Max.   : 0.3264          Max.   : 0.93327
```

Figura 5: Resumen estadístico de las primeras 10 columnas de Cortex.

Para representar datos de expresión génica se suele utilizar el gráfico heatmaps. Un heatmap es una representación de los datos (en forma de matriz) mediante una escala de colores (colores más oscuros o menos cálidos para valores más altos y colores menos oscuros o más cálidos para los valores más bajos). Es un gráfico útil para representar perfiles de expresión génica. Se utiliza el paquete gplots que contiene una función llamada *heatmaps.2()*(figura 6).

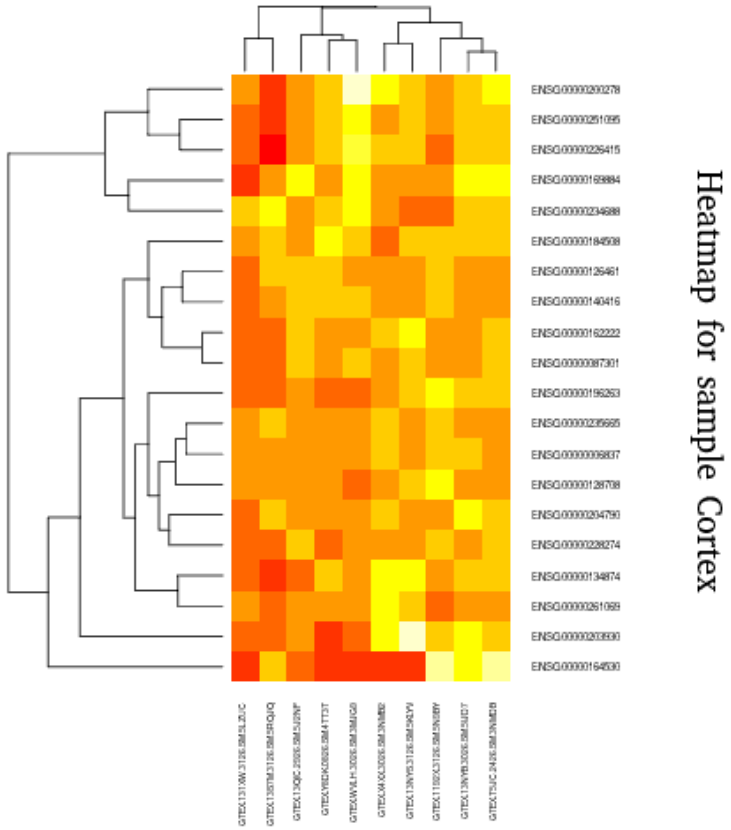

Figura 6: Heatmap para una muestra de Cortex.

4.4. K-means

En primer lugar se instalan y cargan todos los paquetes de R necesarios para implementar el algoritmo k-means. Para ejecutar dicho algoritmo hemos utilizado el paquete stats de R, cuyos datos de entrada de la función *kmeans()* son los siguientes:

```
kmeans(x, centers, iter.max, nstart , algorithm = c('Hartigan-Wong','Lloyd','Forgy','MacQueen'))
```

- x: datos numéricos (en forma de matriz numérica o dataframe).
- centers: número de clusters (K).
- iter.max: número máximo de iteracciones.
- algorithm: algoritmo/s utilizados para calibrar k-means.

Uno de los parámetros más importantes para k-means es el parámetro algorithm. R propone 4 posibles algoritmos que son: 'Hartigan-Wong', 'Lloyd','Forgy','MacQueen'. De los 4 posibles algoritmos, los más

habituales son el algoritmo Lloyd (Stuart Lloyd, 1957) (utilizado por defecto) y el algoritmo Hartigan-Wong (Hartigan,) pero para datasets con un gran número de variables o cuyas observaciones sean muy cercanas no funciona bien y puede dar errores al compilarlos, por eso, el algoritmo utilizado al implementar k-means es el de MacQueen (J.B. MacQueen, 1967).

El algoritmo MacQueen comienza considerando un número K de centroides en función de los primeros elementos del dataset como los **K centroides iniciales**. Después, se van asignando los objetos a un grupo o conglomerado en función del centroide más próximo con la caracteristica de que al efectuar cada asignación se recalculan las coordenadas del nuevo centroide. Esta característica es la que distingo el algoritmo MacQueen de los otros dos. Finalmente, una vez asignados todos los objetos, se recalculan los centroides para cada uno de los conglomerados y se reasigna cada objeto a su centroide más cercano [34].

Los datos de salida más importantes de la función *kmeans()* son los siguientes:

- WCSS: tot.withinss = sum(withinss).
- iter: número de iteracciones externas.
- cluster: particiones finales.
- centers: los valores de los centroides de cada clusters.

Una vez que la matriz de datos originales (procedente de GTEx) se transforma en su transpuesta, se consulta cual sería el mejor modelo para nuestros datos utilizando los valores WCSS ("método de codo"), BIC y AIC (apartado 3.2) con el objetivo de comprobar si obtenemos el mismo número de grupos K que al implementar el método coexp.

Para realizar la selección del mejor modelo utilizando los valores WCSS se ha implementado una función llamada *choose_kop()* para aplicar la función *kmeans()* en el rango de valores K = 2:50 recogiendo todos los resultados en una lista, kmeans_exp, con el objetivo de obtener todos los valores WCSS de los 49 modelos. Estos valores WCSS (eje x) se representan en función de todos lo valores K (eje y) para visualizar gráficamente el çodo.º valor de grupos K óptimo.

Utilizando la lista de resultados kmeans_exp podemos obtener también los valores BIC y AIC de todos los modelos ejecutados usando sus fórmulas matemáticas (apartado 3.2.3 y 3.2.4) mediante la función *save_bic_and_aic()*. En la tabla 4 se muestran las primeras 5 filas de la tabla con los valores BIC y AIC 49 modelos ejecutados. La tabla completa se encuentra en la carpeta results (anexo guía)

También, se puede obtener fácilmente los centroides (*obtain_centers()*) para su consulta o las particiones finales con la ayuda de la función *obtain_cluster()* (el primer parámetro es la lista de resultados kmeans_exp, y el segundo es un valor de K deseado. Esta función guarda la clasificación de los genes en clusters en archivos .csv para su consulta, posterior representación y evaluación (MCC), por lo tanto se obtienen las particiones finales de los modelos con K = 35 para compararlas con las particiones generadas aplicando el método coexp.

BIC	AIC	namesk
219558.2	217749.5	kmeans2
204385.8	201672.6	kmeans3
193467.4	189849.9	kmeans4
187742.9	183221.1	kmeans5
183093.3	177667.0	kmeans6

Tabla 4: Valores BIC y AIC k-means

4.5. FMM

Para implementar el modelo FMM se han utilizado dos paquetes mclust y EMMIX.

4.5.1. Mclust

Mclust fue desarrollado por Fraley y Raftery en 1999 aunque se ha ido actualizando con el tiempo. Este paquete nos proporciona funciones para estimar la densidad, realizar modelos de clustering y clasificación mediante el algoritmo EM basado en modelos de mezcla finitos y análisis discriminantes. También, podemos obtener los valores BIC e ICL en base a un modelo de clustering e incluye funciones para obtener gráficos de incertidumbre, proyecciones aleatorias, gráficos de contorno y perspectiva. Generalmente, para obtener modelos de mezcla finitos de un conjunto de datos se utilizan dos funciones principales, *Mclust()* y *mclustBIC()* [11].

Los datos de entrada de la función *Mclust()* son los siguientes:

```
Mclust(data, G, modelNames)
```

- data: matriz de datos (o dataframe). No se permiten datos categóricos.
- G: vector entero que especifica el número de componentes de la mezcla (agrupaciones) para cual el BIC se va a calcular. El valor predeterminado es G = 1:9.
- modelNames: vector de cadenas de caracteres que indica los modelos que se ajustarán en la fase EM del agrupamiento. Para datos multivariantes se utiliza mclust.options ('emModelNames').

Los datos de salida de la función *Mclust()* son los siguientes:

- data: matriz de datos de entrada.
- modelName: una cadena de caracteres que denota el modelo en el que se obtiene el mejor valor BIC.
- n: el número de observaciones en los datos.

- d: la dimensión de los datos.
- G: el número óptimo de componentes de la mezcla.
- BIC: todos los valores BIC.
- bic: valor óptimo de BIC.
- loglik: la probabilidad de registro correspondiente al BIC óptimo.
- df: el número de parámetros estimados.
- classification: particiones finales.

La función *mclustBIC()* tiene los mismos parámetros que *Mclust()* y sirve para obtener los tres mejores modelos según BIC. También, en este paquete hay una función equivalente en base al criterio ICL (*Integrated Completed Likelihood*) que es *mclustICL()*.

Se debe de tener en cuenta que la función *mclustBIC()* considera que el "mejor modelo.es el que tiene el valor BIC más alto entre los modelos ajustados, ya que no incluye el componente negativo de la ecuación 1 (apartado 3.2) [12]. Los paquetes stats y EMMIX no incluyen una función para obtener directamente los mejores valores BIC.

Una vez que la matriz de datos originales se transforma en su transpuesta generamos los modelos FMM en un rango de valores de K = 2:50 generando una lista con todos los modelos llamada mclust_models y a partir de esa lista se pueden obtener los valores BIC de cada uno de los modelos implementados. Por otra parte, utilizando su fórmula matemática, se obtienen los valores AIC de todos los modelos ejecutados anteriormente gracias a los objetos loglik y df. Todos los valores BIC y AIC de los 49 modelos son guardados en una tabla y se representan gráficamente (apartado 5.1) para consultar el número de clusters, K, según los valores BIC y AIC. En la tabla 5 se muestran las 5 primeras filas de la tabla con los valores BIC y AIC de los modelos utilizando el algoritmo mclust.

BIC	AIC	namesk
-374717.20	371091.78	mclust2
-122478.92	117036.82	mclust3
22325.25	-29584.03	mclust4
-475751.04	470293.08	mclust5
-613482.56	607112.29	mclust6

Tabla 5: Valores BIC y AIC mclust.

A partir de la función *Mclust()* se pueden obtener las particiones finales de cualquier modelo en función del valor K por lo tanto se obtienen las particiones generadas con un valor K igual a 35, que es el valor de clusters óptimo que obtenemos para nuestros datos utilizando el método coexp con el objetivo de compararlas con las métricas de comparación de clustering.

4.5.2. EMMIX

EMMIX es un paquete desarrollado por G. McLachlan en 1999 que proporciona funciones para estimar la densidad de los modelos de mezcla multivariados, distribuciones normales en función de la covarianza y ajusta modelos de mezcla mediante el algoritmo EM pudiendo elegir entre diferentes distribuciones disponibles generando las particiones finales en función de dicho modelo.

La función prncipal es **EMMIX()** y sus datos de entrada más importantes son los siguientes:

```
EMMIX(dat, g, distr="mvn",ncov=3, clust = NULL, itmax=1000)
```

- dat: conjunto de datos, una matriz numérica $n \times p$, donde n es el número de observaciones y p la dimensión de los datos.
- g: número de componentes del modelo de mezcla.
- distr: cadena de tres letras que indica el tipo de distribución que se ajustará, el valor predeterminado es 'mvn', la distribución Normal.
- ncov: número entero pequeño que indica el tipo de estructura de covarianza; el valor predeterminado es 3.
- clust: vector de enteros que especifican las particiones iniciales de los datos, el valor predeterminado es NULL.
- itmax: número de iteraciones máximas.

Los datos de salida de **EMMIX()** son los siguientes:

- aic: Akaike Information Criterion (AIC).
- bic: Criterio de información Bayes (BIC).
- icl: Criterio de verosimilitud completada de ICL (ICL).
- pro: un vector de proporciones de mezcla.
- mu: una matriz numérica con cada columna correspondiente a la media.
- sigma: una matriz de dimensión (p, p, g) con la primera matriz de covarianza de dos dimensiones correspondiente de cada componente.
- clust: un vector con las particiones finales.
- loglik: valores de log-verosimilitud.

Una vez que la matriz de datos originales se transforma en su transpuesta, generamos los modelos FMM en un rango de valores de K = 2:50. Todos ellos se guardan en un objeto emmix_models y a partir de éste se pueden obtener los valores BIC y AIC de cada uno de los modelos implementados directamente (sin utilizar en ninguno de los dos casos las formulas matemáticas). Todos estos valores son guardados en tabla para su consulta y posterior representación gráfica con el objetivo de obtener el número de clusters óptimo K según los valores BIC y AIC. En la tabla 6 se muestran las 5 primeras filas de la tabla que recoge todos los valores BIC y AIC de todos 49 modelos EMMIX ejecutados. Por otra parte, se obtienen las particiones finales generadas por el paquete EMMIX de K = 35 para su posterior comparación.

BIC	AIC	namesk
143425.1	198555.0	emmix_models2
141186.1	192217.9	emmix_models3
134058.8	163964.3	emmix_models4
126824.9	159206.2	emmix_models5
119546.7	155585.5	emmix_models6

Tabla 6: Valores BIC y AIC emmix.

4.5.3. Elección de paquetes de R para FMM

Mclust es un paquete de R muy popular para modelar los datos como mezclas finitas gaussianas y proporciona estrategias para clustering y clasificación de los datos.

Según Scrucca et al, mclust5 es la herramienta más poderosa para modelar datos como mezclas finitas gaussianas en comparación con otros paquetes como por ejemplo Rmixmod, flexmix o mixtools. Los datos utilizados en este TFM tienen un gran número de dimensiones y por lo tanto paquetes como Rmixmod y Flexmid quedaron descartados para realizar clustering ya que no funcionan bien con datos de alta dimensionalidad. Además, Flexmid tiene una capacidad de clustering muy limitada. Por otra parte, mixtool funciona bien con datos de alta dimensionalidad pero no puedes obtener las particiones finales de los modelos, por lo tanto también quedó descartado.

Por otra parte, se barajó la posibilidad de utilizar el paquete EMMIX-gene, desarrollado en los últimos años, pero para realizar el clustering con este paquete previamente se requiere seleccionar los 'genes más importantes' con la función *select_genes()*. El paquete EMMIX-gene fue descartado porque en este estudio es imprescindible modelar los datos originales (114×20601) ya que el método coexp lo hace particiendo de dichos datos (sin eliminar filas, es decir genes). Si uno de los objetivos del TFM es métodos de clustering no es recomendable comparar dos métodos de clustering diferentes utilizando datos de diferente estructura. Además, para generar redes de coexpresión eliminar genes supone eliminar información biológica relevante, ya que el objetivo de estas redes es encontrar relaciones entre los genes.

También se barajó la posibilidad de utilizar el paquete HDclassif pero antes de aplicar el algoritmo EM realiza una reducción de dimensionalidad, por tanto se obtiene un número clusters K muy bajo.

Como se ha comentado anteriormente, no solo se ha obtenido las particiones utilizando mclust sino también las generadas por el paquete EMMIX ya que éste funciona muy bien con datos con alta dimensionalidad y según Fraley et al mclust no.

4.5.4. Método coexp

Para obtener los clusters (módulos) a los que pertenece cada uno de los genes de un dataset procedente de GTEx mediante el método coexp se ejecuta la función *coexp.getNetworkFromTissue()*. Esta función necesita como parámetros el tejido (Cortex) y la red de coexpresión (gtexv6).

Los datos de salida de la función *coexp.getNetworkFromTissue()* son:

- modulesColors: particiones finales donde los clusters son etiquetas de caracteres (colores).
- modulesLabels: particiones finales donde los clusters son etiquetas de caracteres numéricos.

El método coexp tiene otras funciones útiles como por ejemplo *coexp.getGenesFromModule()* que te permite obtener el número de genes que tiene un módulo concreto o la función *coexp.plotModSizes()* con la cual puedes obtener una gráfica (figura 5) que te permite chequear el tamaño de los módulos pudiendo observar qué módulos tienen más o menos genes.

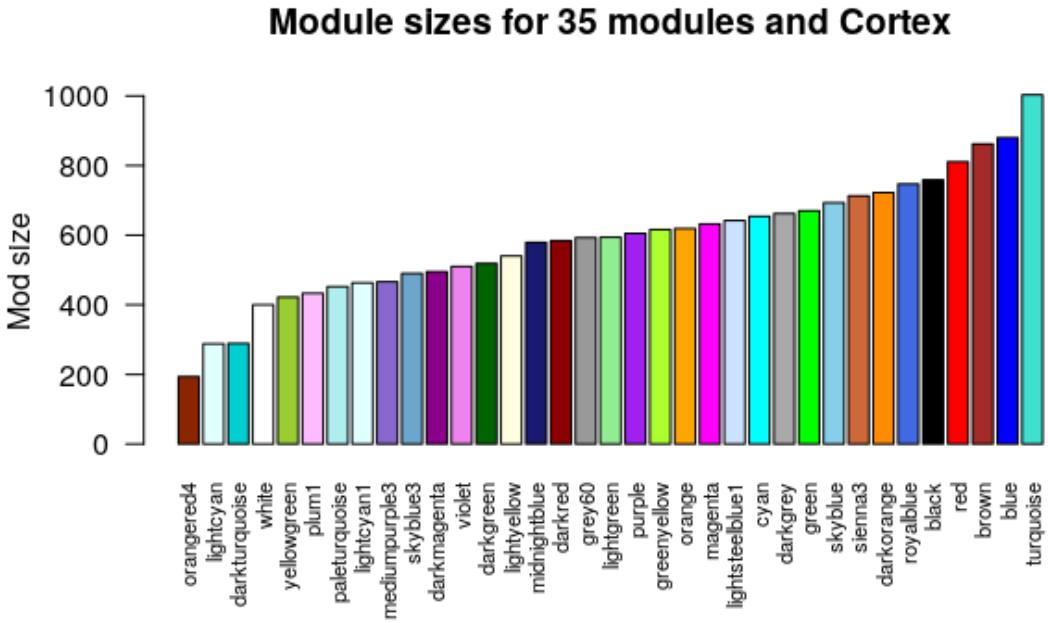

Figura 7: Tamaño de los módulos método coexp.

El algoritmo híbrido (WGCNA + k-means) genera **35 módulos** diferentes. El número de módulos (K), como he comentado en el apartado 3.4, se obtiene realizando un clustering jerárquico estándar (flashclust) especificando el valor máximo que puede tomar K, que es 150.

4.5.5. Visualización del clustering en R

PCA en R

Para aplicar la metodología PCA en R solo es necesario el paquete de R base. La función *prcomp()* calcula los componentes principales de un dataset. Dicha función retorna una lista de valores llamados: sdev, rotation, center, scale y **x** . El objeto x es una matriz que contiene todos los componentes principales, pudiendo acceder a ellos fácilmente.

Es muy común, comprobar la variabilidad de los componentes realizando un histograma de dichos componentes. Como se muestra en la figura 8 los dos primeros componentes contienen la mayor información de las variables del dataset Cortex.

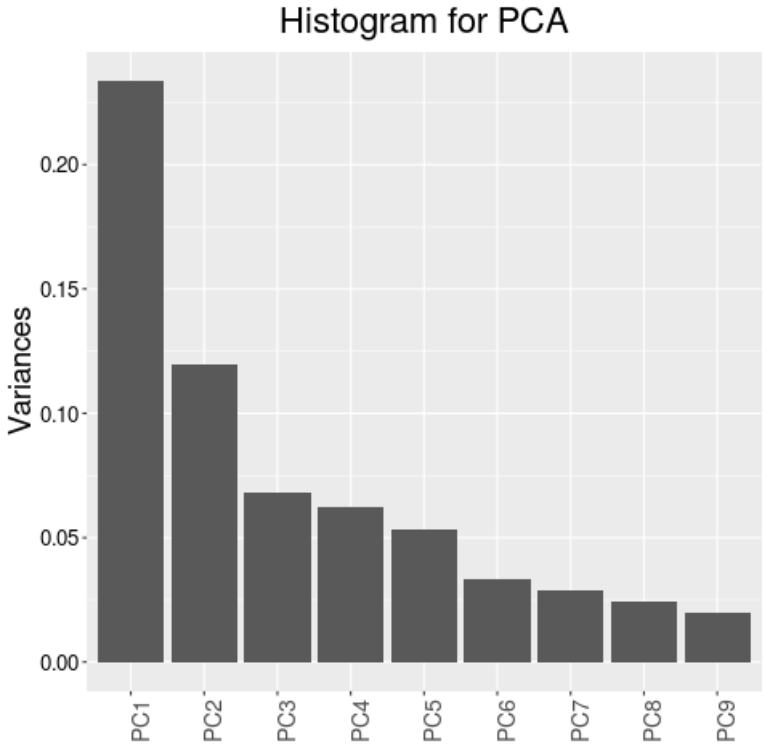

Figura 8: Histograma de los componentes principales del dataset Cortex

Para visualizar los datos, con sus respectivos clusters, hemos utilizado los dos primeros componentes principales (PC1 y PC2) y el paquete ggplot2.Los gráficos PCA se han realizado para todos los modelos (coexp, k-means, FMM mclust y FMM emmix) (anexo guía).

T-SNE en R

Para aplicar la metodología T-SNE en R solo es necesario el paquete **RTSNE** incluido en CRAN. El paquete RTSNE contiene una función llamada *tSNE()* con los siguientes datos de entrada:

```
tSNE(X, Y, k, perplexity, n.iter)
```

- X: dataframe, matriz o matriz de distancia.
- Y: configuración k dimensional inicial. Si es NULL el método usado es random.
- K: número de dimensiones, solo puede ser 2 o 3. Por defecto es 2.

- *perplexity*: es el límite superior aproximado del tamaño del vecindario.

  La perplexity es el parámetro principal. Los valores pueden ir de 5 a 100. Es muy común probar con varios valores y si el dataset es muy grande es recomendable utilizar un valor de perplexity alto.

  El parámetro perplexity se define como:

$$Perp(P_i) = 2^{H(P_i)} \quad (59)$$

  siendo $H(P_i) = -\sum_j p_{j|i}\ log(p_{j|i})$ la entropía Sannon de P_i.

- iter: número de iteracciones. Es el número de iteracciones que requiere un conjunto de datos para finalizar.

  Hemos aplicado la función *tSNE()* a nuestros datos para obtener las variables TSNE1 y TSNE2 que serán representadas con diferentes valores de perplexity (20,30,60 y 90) con el objetivo de visualizar los datos en función de las particiones generadas. Las gráficas se describen en anexo guía.

4.5.6. Medidas de comparación de clustering

Para obtener las medidas de comparación de clustering hemos utilizado el paquete **aricode** (repositorio de CRAN).

Con el paquete aricode, directamente podemos obtener las medidas de comparación de clustering ARI, NVI y NDI, entre otras, descritas en el artículo Vinh et al (2009). En este paquete hay implementadas tres funciones para obtener dichas medidas que son *ARI()*, *NVI()* y *NDI()* cuyos parámetros son las particiones (vector de etiquetas de las particiones). Vamos a obtener el valor NVI por lo tanto no es necesario obtener VI, el cual se puede obtener directamente con el paquete mcclust.

Las particiones generadas por lo cuatro modelos implementados en este TFM (coexp, k-means y FMM) son comparadas de dos en dos mediante las funciones de los paquetes aricode. Con la función *get_MCC()* cargo en R todas las particiones generadas por los tres modelos en una lista de vectores y computo las medidas de comparación de clustering de dos en dos guardando los resultados en una tabla. La primera columna de la tabla generada por *get_MCC()* son todos los modelos que se comparan y las otras tres son los valores ARI, 1-NVI y 1-NID de cada unos de los modelos (tabla 8, apartado 5.1).

Los datos de salida de las tres medidas de comparación de clustering, ARI, NVI y NID son valores numéricos que van de 0 a 1. Si al comparar dos particiones con cualquiera de las tres métricas obtienes un valor de 1 se considera que las particiones son iguales y si obtienes un valor de 0 se considera que las particiones son totalmente diferentes. Para obtener la misma escala de interpretación que ARI, se utilizan los valores 1-NVI y 1-NID (apartados 3.6.3 y 3.6.4).

Capítulo 5

Resultados

Los resultados se van a presentar en tres apartados diferentes. En el apartado 5.1 se presentan los resultados obtenidos al aplicar diferentes criterios ("método de codo", BIC y AIC) para obtener el número de clusters (K) óptimo de los modelos k-means y FMM con la finalidad de comparar dichos valores de K con el valor K = 35 obtenido aplicando el método coexp.

En el apartado 5.2 se presentan las gráficas creadas para la visualización de los datos Cortex cerebral en función de los diferentes clusters obtenidos por los métodos coexp, k-means y los dos FMM usando las metodologías estadísticas de reducción de dimensionalidad y se comparan entre sí.

En el apartado 5.3 se presentan los resultados obtenidos al aplicar las métricas de comparación de clustering (ARI, NVI y NID) utilizando las particiones generadas por los modelos coexp, k-means y FMM de dos en dos con el objetivo de elucidar si hay o no semejanzas entre ellos.

5.1. Selección del número de clusters en función de criterios estadísticos

En la tabla 7 se presentan los valores K óptimos obtenidos en base a los criterios "método de codo"(k-means), BIC y AIC.

	K			Valores		
Modelos	Método codo	BIC	AIC	WCSS	BIC	AIC
k-means	10	15	15	162651.1	171220.7	157655.2
FMM (mclust)	-	41	41	-	-2363.2	-35937.7
FMM (emmix)	-	50	50	-	-164943.8	-211455.5

Tabla 7: K óptimo y valores WCSS, BIC y AIC.

Como podemos ver en la tabla 7 los valores del número de clusters óptimo para k-means según el criterio método de codo es 10 tal y como muestra la figura 9 ya que el cambio brusco de la evolución en la inercia o codo se observa cuando K = 10.

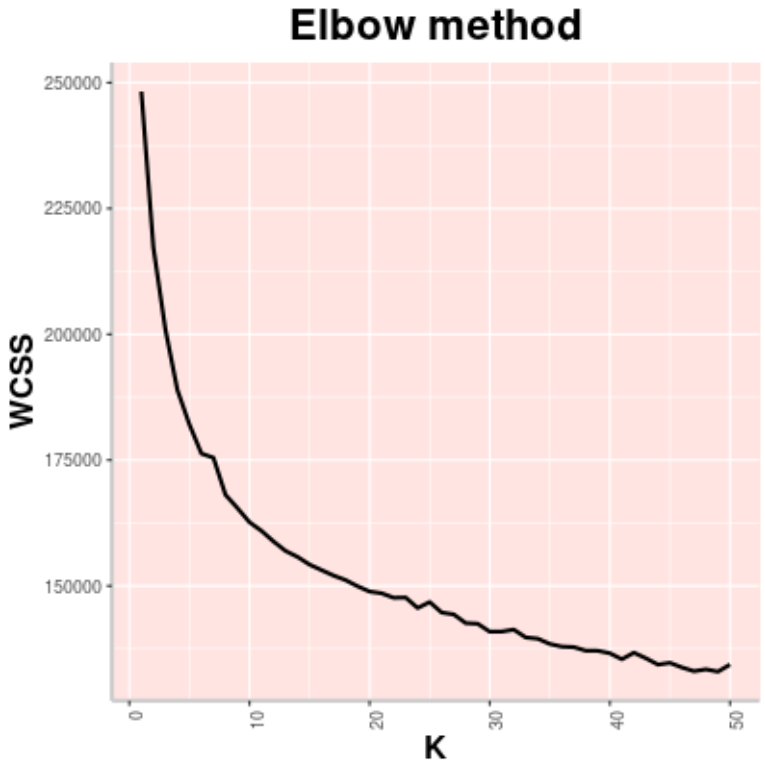

Figura 9: Gráfica método de codo k-means.

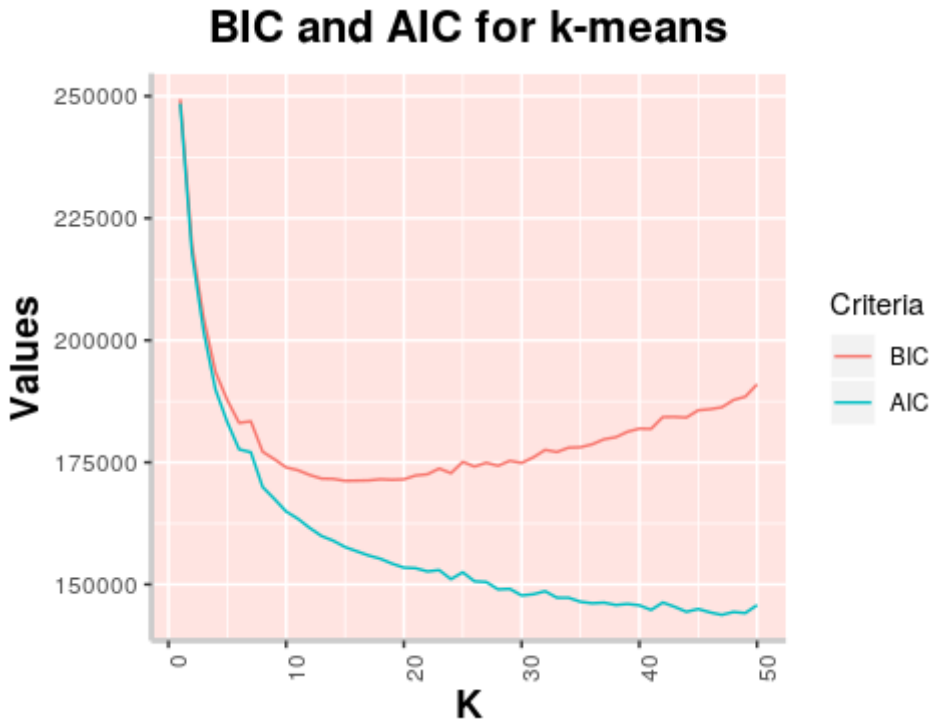

Figura 10: Gráfica valores BIC y AIC k-means

Según los criterios BIC y AIC el valor más adecuado para el número de clusters podría ser 15 ya que en este caso se suele escoger el valor K que tiene un valor BIC más pequeño cuando se produce el cambio en la evolución de la inercia en el gráfico. Se puede observar en la figura 10.

Para el modelo FMM (mclust) se ha obtenido el valor K = 41, este caso es especial, ya que mclust elimina el componente negativo de la fórmula general de BIC (apartado 3.2.3) para obtener todos los valores BIC de todos los modelos ejecutados y por esa razón la gráfica (figura 11) es ascendente y no descendente como en el caso de k-means. En cambio, la gráfica de los valores AIC es descendente ya que estos valores se han obtenido de la fórmula original (figura 12). Para obtener el número de clusters en base a BIC debemos de coger el valor K que tiene un BIC máximo y un AIC mínimo antes de que estos valores empiecen a ser constantes o que no haya un cambio alto en las unidades entre los valores BIC de un modelo al siguiente.

Se han representado los valores BIC y AIC en dos gráficas diferentes ya que no tienen la misma cinética y es más ilustrativo para su interpretación. También, a diferencia de las demás gráficas se han representado con un diagrama de puntos porque los valores no son tan uniformes como los demás modelos (k-means y FMM emmix) y su interpretación se hace más difícil.

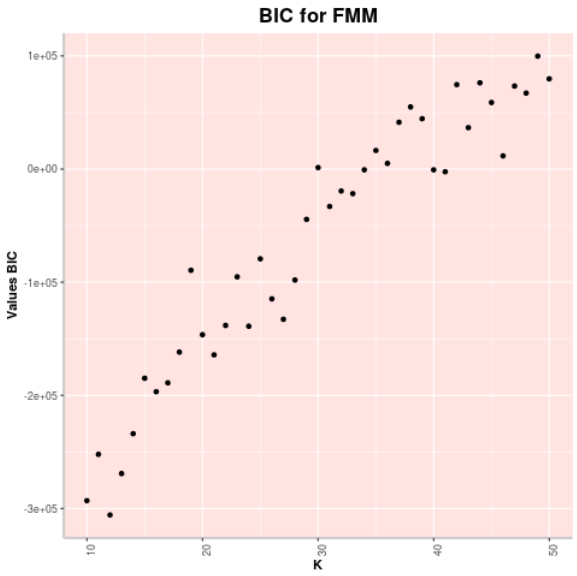

Figura 11: Gráfica valores BIC mclust.

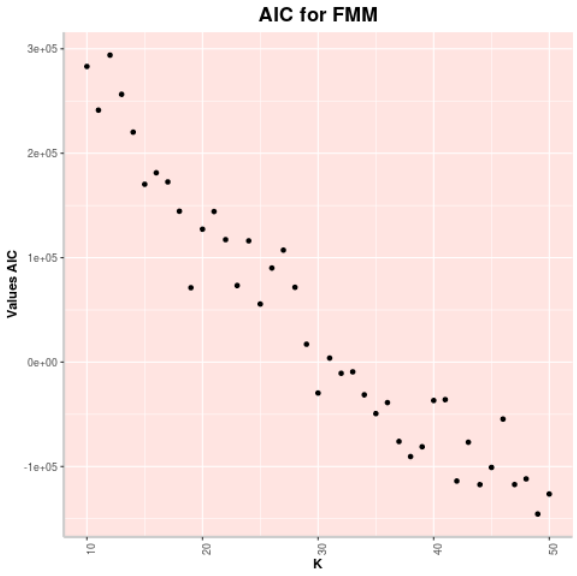

Figura 12: Gráfica valores AIC mclust.

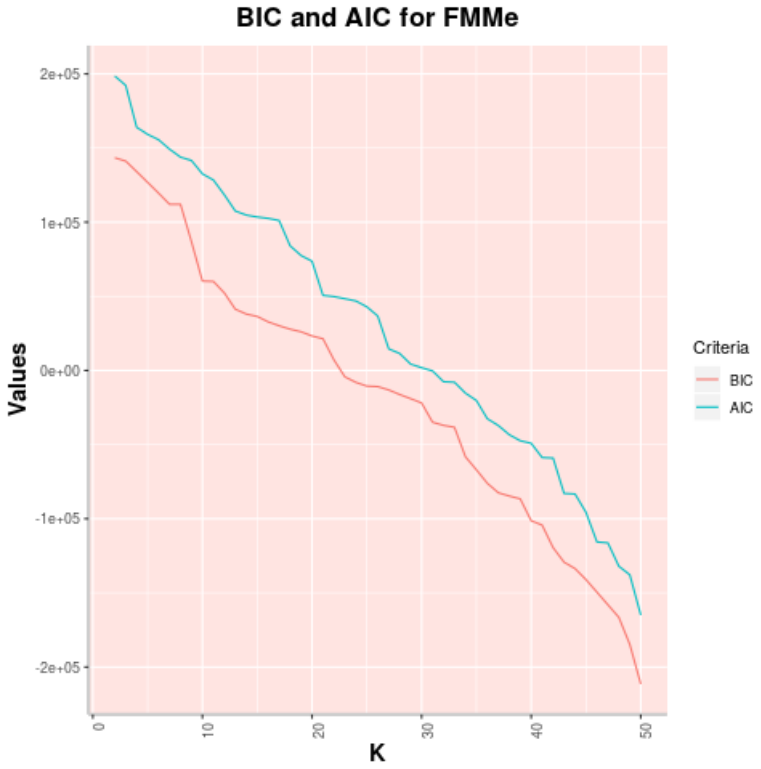

Figura 13: Valores BIC y AIC emmix

En el caso del modelo FMM (emmix), las dos gráficas tienen forma descendente ya que EMMIX obtiene los valores BIC y AIC desde la fórmula general (apartado 3.2.3). Se escoge un valor de K = 50 ya que los valores BIC y AIC son los más pequeños de todos los modelos. Podemos observar en la figura 13 que la cinética no llega a ser constante en ningún momento, como en el caso de los otros modelos de clustering, por lo tanto, se puede decir que en base a BIC y AIC un numero más alto de clusters sería más recomendable en el caso de FMM (emmix). Para eso se deberían de ejecutar el modelo FMM (emmix) con un rango de valores mayor a 2:50.

5.2. Visualización del clustering para coexp, k-means y FMM

Para representar gráficamente datos con valores muy similares y con una gran cantidad de columnas (variables o dimensiones) se utilizan métodos de reducción de dimensionalidad como el método PCA o TSNE implementados con la ayuda de paquetes de R (explicados en el apartado 4.4.5).

Las ocho gráficas resultantes al aplicar PCA y TSNE sobre los datos, en función de los diferentes clusters que generan los cuatro algoritmos de clustering usados en este TFM (coexp, k-means, FMM emmix y FMM mclust) se presentan en la carpeta *results* o haciendo click sobre sus respectivos enlaces en el anexo guía.

5.3. Medidas de comparación de clustering aplicadas a coexp, k-means y FMM

En la tabla 8 se presentan los resultados obtenidos al comparar las particiones generadas por los cuatro modelos, coexp, k-means , FMM (mclust) y FMM (emmix) con k = 35 utilizando las medidas de comparación de clustering ARI, NVI y NID. La primera columna hace referencia a los cuatro modelos comparados de dos en dos y en las tres columnas restantes se encuentran los valores ARI, 1-NVI y 1-NID.

Models	ARI	1-NVI	1-NID
Coexp vs FMM (emmix)	0.1489621	0.3027379	0.4389926
Coexp vs FMM (mclust)	0.1230960	0.2585097	0.3569734
Coexp vs k-means	0.1837863	0.3323517	0.4770123
FMM (emmix) vs FMM (mclust)	0.4819914	0.4521675	0.5697929
FMM (emmix) vs kmeans	0.3933624	0.4143893	0.5785088
FMM (mclust) vs k-means	0.3556966	0.3816616	0.4996562

Tabla 8: Resultados de las métricas de comparación del clustering.

5.4. Conclusiones y futuras direcciones

En éste apartado se van a explicar los resultados presentados en los apartados 5.1, 5.2 y 5.3.

1. **Selección del número de clusters en base a los criterios BIC, AIC y método de codo**

   Se han implementado diferentes métodos que nos ayudan a elegir un número apropiado de clusters para ajustar nuestro modelo/s de clustering. Una mala elección del número de clusters puede dar lugar a grupos muy heterogéneos (pocos clusters) o muy homogéneos (muchos clusters).

   En la tabla 7 podemos ver que para realizar k-means con nuestros datos obtenemos un número de clusters K = 10 utilizando el método de codo y K = 15 si usamos los criterios BIC y AIC. Los dos valores son muy bajos para estos datos ya que tienen 114 variables y 20601 filas.

   A priori se puede decir que un número de clusters más alto sería más adecuado para ajustar el modelo k-means a nuestros datos. De todas formas, k-means no es un buen método de clustering para datos de alta dimensionalidad por que el concepto de distancia se vuelve menos preciso a medida que crece el número de dimensiones. Por otra parte, K = 41 y K = 50 (modelos FMM mclust y FMM emmix) son valores más adecuados para estos datos. En la figura 12 (FMMe) se insinua que un número de clusters más alto sería más adecuado para ajustar el modelo FMM (emmix) a los datos Cortex según los criterios BIC y AIC. El algoritmo híbrido (WGCNA + k-means) genera un número de clusters K = 35, la ventaja que tiene éste método en comparación con

k-means y los modelos emmix y mclust es que el usuario no elige el número de clusters, sin embargo al ser datos tan grandes podría ser necesario un número de clusters mayor [23].

A pesar de todo, la elección del número de clusters es muy subjetivo ya que se **estima** el mejor modelo que se ajusta a nuestros datos ya que no podemos saber cuál es el mejor modelo.

2. **Representación gráfica de las particiones generadas los modelos de clustering**

   - PCA vs TSNE

     Comparando las gráficas generadas por los dos métodos de reducción de dimensionalidad se puede observar claramente que el método T-SNE es más efectivo ya que se pueden diferenciar mejor los diferentes clusters, es decir, los puntos están más dispersos y no tan solapados como en el caso de las gráficas de PCA. El uso del método TSNE es muy común cuando se quiere representar datos de expresión génica ya que se puede elegir el tamaño del vecindario con el parámetro *perplexibility*.

     A pesar de la reducción de dimensionalidad, los valores del dataset Cortex son muy cercanos entre sí, por eso puede haber en la gráfica un cierto solapamiento lo que dificulta la diferenciación de los diferentes clusters, por ese motivo, se le ha añadido etiquetas a los gráficos TSNE que corresponden a los diferentes clusters.

- Comparación de las particiones generadas por los modelos de clustering

  Se puede observar que en general el clustering que realizan los cuatro algoritmos es moderadamente diferente entre sí (como se confirma al aplicar las métricas de comparación de clustering), ya que los colores y etiquetas difieren ligeramente en los cuatro gráficos (coexp, k-means, FMM mclust y FMM emmix con K = 35).

  Se puede observar que el tamaño de los clusters (cantidad de genes que contine un grupo) en coexp es más uniforme que en los otros tres modelos, de hecho mclust tiene clusters con mucho genes y otros con muy pocos (menos de 5 genes) como es el caso de los grupos 4, 5 , 6, 16, 23, 31, 32 y 34. Este hecho también concuerda con los valores más cercanos a 1 (particiones independientes) al comparar coexp y mclust. Cuanto más parecidos en tamaño sean los grupos de dos modelos diferentes más posibilidades hay de que los mismos genes esten en el mismo grupo en ambos modelos. Las gráficas TSNE mas similares son por una parte las dos gráficas que representan las particiones generadas por los dos modelos FMM, lo cual es lógico porque son dos modelos que utilizan el mismo algoritmo (algoritmo EM) y por otro lado las gráficas que representan las particiones generadas por coexp y k-means respectivamente lo cual también es lógico porque coexp es un algoritmo híbrido WGCNA que utiliza k-means.

3. **Comparación del clustering mediante ARI, NVI y NID**

   En términos generales, los valores de ARI, NVI y NID estan en el rango de 0 (particiones diferentes) a 1 (particiones iguales) pero como podemos ver en la **tabla 8** en algunos casos el valor ARI no va en consonancia con los valores 1-NVI y 1-NID. Esto es debido a que ARI no funciona muy bien cuando el número de clusters es alto o el dataset tiene muchas variables [54]. Por lo tanto, en este apartado interpretaremos los resultados basándonos sobre todo en los valores 1-NVI y 1-NID.

   Como podemos ver en la **tabla 8** al comparar las particiones generadas por los modelos coexp y FMM (emmix) los valores 1-NVI y 1-NID no son muy cercanos a 0 (no similitud) pero tampoco muy cercanos a 1 (0.30 y 0.43), por lo tanto no podemos decir que sean muy parecidos. Cuando comparamos las particiones generadas por coexp y mclust obtenemos los valores más cercanos a 0 (0.25 y 0.35), es decir, las particiones no son muy similares, este hecho se puede observar en sus respectivas gráficas T-SNE. Al comparar coexp con k-means obtenemos los valores más cercanos a 1 (0.33 y 0.47) lo cual es lógico ya que coexp es un método híbrido WGCNA que incorpora k-means. Estan diferencias son producidas porque coexp emplea clustering jerárquico. Por otra parte, la particiones más similares son las generadas por ambos modelos de mezcla (emmix y mclust) ya que al compararlas obtienen los valores ARI,1-NVI y 1-NID más cercanos a 1 (0.48, 0.45 y 0.56).

   Dados estos resultados podemos decir que coexp no es muy similar a ambos modelos FMM, lo cual es de esperar ya que el método coexp esta basado en correlaciones y FMM es probabilidades.

   Asumiendo que WGCNA es el método por excelencia para la generación de GCN, se puede decir, en base a los resultados que un modelo probabilístico puede ser una buena opción para sustituir a k-means en el método coexp (WGCNA + k-means).

Si se hubieran parecido mucho más las particiones generadas por el método coexp y las de los modelos de mezcla finito quizás no podríamos decir que un módelo de mezcla finito podría sustituir a k-means en el método coexp.

4. **Futuras direcciones**

   - Considerar el modelo de mezcla finito para sustituir a k-means en el método coexp con el objetivo de mejorar la generación de redes de coexpresión génica.
   - Crear un paquete de R que recoja todos los análisis implementados en el presente TFM.
   - Planteamiento e implementación de un pseudocódigo para ajustar modelos de mezcla finitos a datos con un gran número de filas y columnas sin necesidad de reducir la dimensionalidad de estos para reducir el tiempo computacional empleado.

Notación

- Apartado 3.2

  K: número de clusters máximo.

  WCSS: suma de las distancias cuadráticas (*within-cluster sum*).

  $log(\hat{L})$: log de la verosimilitud (*log-likelihood*).

  n: número de observaciones de un dataset.

  k: número de parámetros estimados.

- Apartado 3.3

  $d(.)$: distancia.

  x_i: dato o punto de un dataset.

  m_j: centroide de un grupo.

  $m^(k)$: conjunto de centroides.

  $dist()$: distancia euclídea.

- Apartado 3.4

  G: matriz de datos de expresión.

  m: número total de muestras.

  n: número total de genes.

  $cor(.)$: correlación.

  S_{ij}: Matriz de similitud.

  A_{ij} Matriz adyacente.

  g_i y g_j: dos genes cualesquiera.

  β: umbral sueve ('*soft thresholding*').

  P: conjunto de particiones.

- Apartado 3.5

  y: vector de variables observadas.

  z: vector de variables latentes ('*missing variables*').

  x: vector de datos completos, $X = (Y, Z)$.

  P: cantidad de subpoblaciones (componentes o grupos).

  θ: parámetros desconocidos.

  π: vector de pesos o probabilidades.

  $\hat{\tau}_t$: probabilidad estimada de que una observación y_t pertenezca a una partición.

  FDP: Función de Densidad de Probabilidad.

  $\ell(\theta)$: verosimilitud de los datos observados (*observed-data likelihood*).

  $\ell_c(\theta)$: verosimilitud de los datos completos (*complete-data likelihood*).

  $f(y;\theta)$: función de verosimilitud de los datos observados, es igual a: $L(\theta; y)$.

  $f(x;\theta)$: función de verosimilitud de los datos completos, es igual a: $L(\theta; x)$.

  $\ell(\hat{\theta})$: Estimador de máxima verosimilitud.

  $\hat{\theta}$: parámetro estimado.

  $Q(\psi; \psi^h)$: Función Q.

  Algoritmo EM: Algoritmo expectación-maximización(*expectation-maximization algorithm*).

  EMV: estimadores de máxima verosimilitud (*maximum likelihood estimators*, MLE).

  $E.$: Operador esperanza/expectación.

  $I(\theta)$: matriz de información esperada.

  $I(\psi; y)$: matriz de información de datos observados.

  $I^c(\psi, x)$: matriz de información de datos completos.

  $I^m(\psi, z)$: matriz de información de datos faltantes.

- Apartado 3.6

  U y V: particiones U y V.

  $H(.)$: entropía.

  n_{ij}: número de objetos comunes entre u_i y v_j (dos particiones cualesquiera de U y V respectivamente).

  a_i y b_j: número de objetos que hay en u_i y v_j respectivamente.

  W: partición refinada.

- Apartado 4.7

  p: número total de variables originales (X).

  k: número total de variables componentes principales (Z).

  Z_1: varianza.

  V: matriz de inercia V (covarianzas).

  R: matriz de inercia R (correlaciones).

  $p_{j|i}$: probabilidad condicional.

  σ: varianza.

  $q_{i|j}$: distribución de probabilidad de similitud.

  y_i y y_j: dos puntos cualesquiera.

  P y Q : dos particiones cualesquiera.

  C: divergencia.

  $perp(.)$: perplexibility.

  $H(.)$: entropía.

Bibliografía

1. Bonner, R. E. (1964). On some clustering techniques. IBM journal of research and development, 8(1), 22-32.

2. Bolívar Zapata , F. G. B. (2004). Fundamentos y casos exitosos de la biotecnología moderna. El Colegio Nacional.

3. Botía, J. A., Vandrovcova, J., Forabosco, P., Guelfi, S., D'Sa, K., Hardy, J., ... & Weale, M. E. (2017). An additional k-means clustering step improves the biological features of WGCNA gene co-expression networks. BMC systems biology, 11(1), 47.

4. Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: understanding AIC and BIC in model selection. Sociological methods & research, 33(2), 261-304.

5. Cahoy, J. D., Emery, B., Kaushal, A., Foo, L. C., Zamanian, J. L., Christopherson, K. S., ... & Thompson, W. J. (2008). A transcriptome database for astrocytes, neurons, and oligodendrocytes: a new resource for understanding brain development and function. Journal of Neuroscience, 28(1), 264-278.

6. Chandrasekhar T, Thangavel K, Elayaraja E. Effective clustering algorithms for gene expression data. Int J Comput Appl. 2011;32(4):25–9.

7. Curtis, H., & Schnek, A. (2008). Biología. Ed. Médica Panamericana.

8. Durrant, W. E., Rowland, O., Piedras, P., Hammond-Kosack, K. E., & Jones, J. D. (2000). cDNA-AFLP reveals a striking overlap in race-specific resistance and wound response gene expression profiles. The Plant Cell, 12(6), 963-977.

9. Fabozzi, F. J., Focardi, S. M., Rachev, S. T., & Arshanapalli, B. G. (2014). The basics of financial econometrics: Tools, concepts, and asset management applications. John Wiley & Sons.

10. Fernández , .D, and Pledger, .S (2015).Categorising Count Data into Ordinal Responses with Application to Ecological Communities. Journal of Agricultural, Biological, and Environmental Statistics. DOI: 10.1007/s13253-015-0240-3.

11. Fop, M., & Murphy, T. B. (2018). Variable selection methods for model-based clustering. Statistics Surveys, 12, 18-65.

12. Fraley, C., & Raftery, A. E. (2003). Enhanced model-based clustering, density estimation, and discriminant analysis software: MCLUST. Journal of Classification, 20(2), 263-286.

13. Fraley, C., & Raftery, A. E. (2007). Model-based methods of classification: using the mclust software in chemometrics. Journal of Statistical Software, 18(6), 1-13.

14. Fraley, C., Raftery, A. E., Murphy, T. B., & Scrucca, L. (2014). Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation (Department of Statistics, University of Washington, 2012).

15. Gómez Losada, Á. (2014). Modelos de mixturas finitas para la caracterización y mejora de las redes de monitorización de la calidad del aire.

16. Hartigan, J. A. (1975). Clustering algorithms.

17. Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

18. Hershey, J. R., & Olsen, P. A. (2007, April). Approximating the Kullback Leibler divergence between Gaussian mixture models. In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on (Vol. 4, pp. IV-317). IEEE.

19. Ihaka, R., & Gentleman, R. (1996). R: a language for data analysis and graphics. Journal of computational and graphical statistics, 5(3), 299-314.

20. Jolliffe, I. (2011). Principal component analysis. In International encyclopedia of statistical science (pp. 1094-1096). Springer, Berlin, Heidelberg.

21. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis & Machine Intelligence, (7), 881-892.

22. Kukurba, K. R., & Montgomery, S. B. (2015). RNA sequencing and analysis. Cold Spring Harbor protocols.

23. Langfelder, P., & Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC bioinformatics, 9(1), 559.

24. Lantz, B. (2013).Machine learning with R. Packt Publishing Ltd.

25. Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., ... & Foster, B. (2013). The genotype-tissue expression (GTEx) project. Nature genetics, 45(6), 580.

26. Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), 2579-2605.

27. MacKay, D. (2003). An example inference task: clustering. Information theory, inference and learning algorithms, 20, 284-292.

28. Martínez-Frías, M. L. (2010). Estructura y función del ADN y de los genes. Tipos de alteraciones de la función del gen por mutaciones. SEMERGEN-Medicina de Familia, 36(5), 273-277.

29. McKenzie, A. T., Wang, M., Hauberg, M. E., Fullard, J. F., Kozlenkov, A., Keenan, A., ... & Zhang, B. (2018). Brain Cell Type Specific Gene Expression and Co-expression Network Architectures. Scientific reports, 8.

30. McLachlan, G. y Basford, K. 1988. Mixture models: inference and applications to clustering. Dekker, New York.

31. McLachlan, G. J. and Jones, P. N. (1988). Fitting mixture models to grouped and truncated data via the EM algorithm. Biometrics, 44(2): 571-78.

32. McLachlan, G. and Krishnan, T. (1997). The EM Algorithm and Extensions. Wiley Series in Probability and Statistics, New York.

33. McLachlan, G. and Peel, D. (2000). Finite Mixture Models. Wiley Series in Probability and Statistics, New York.

34. McLachlan, G., Wang, K., Ng ,A. and Peel , D. (2018). EMMIX: Model Based Clustering with Finite Mixture Models. R package version 1.0.3. https://github.com/suren-rathnayake/EMMIX.

35. Medina-Veloz, G., Luna-Rosas, F. J., Tavanez-Avendaño, J. F. y Narvaez-Murillo, R. U. (2016). Calibración y selección del modelo de aprendizaje no supervisado K-Medias, de una encuesta sobre factores de riesgo en el consumo de drogas entre estudiantes. Revista de Análisis Cuantitativo y Estadístico Junio, 2016 ,Vol.3 No.7 1-9.

36. Meila, M. (2002). Comparing clusterings. In Proceedings of the 22nd international conference on Machine learning-ICML'05 (pp. 577-584).

37. Meila, Marina (2003). Comparing Clusterings by the Variation of Information. Learning Theory and Kernel Machines: 173–187. doi:10.1007/978-3-540-45167-9_14.

38. Meila, M. (2007). Comparing clusterings—an information based distance. Journal of multivariate analysis, 98(5), 873-895.

39. Mengersen, K. L., Robert, C., & Titterington, M. (2011). Mixtures: estimation and applications (Vol. 896). John Wiley & Sons.

40. Melnykov, V., & Maitra, R. (2010). Finite mixture models and model-based clustering. Statistics Surveys, 4, 80-116.

41. Metzker, M. L. (2010). Sequencing technologies—the next generation. Nature reviews genetics, 11(1), 31.

42. Nguyen Xuan Vinh, Julien Epps and James Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. Journal of Machine Learning Research 11 (2010) 2837-2854.

43. Nyren, P., & Lundin, A. (1985). Enzymatic method for continuous monitoring of inorganic pyrophosphate synthesis. Analytical biochemistry, 151(2), 504-509.

44. 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, BBI-S38316.

45. Parikshak, N. N., Gandal, M. J., & Geschwind, D. H. (2015). Systems biology and gene networks in neurodevelopmental and neurodegenerative disorders*. Nature Reviews Genetics, 16(8), 441.

46. Picard, F. (2007). An introduction to mixture models. Statistics for Systems Biology Group. Research Report, (7).

47. Sabin, M., & Gray, R. (1986). Global convergence and empirical consistency of the generalized Lloyd algorithm. IEEE Transactions on information theory, 32(2), 148-155.

48. Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. The R journal, 8(1), 289.

49. Soto Sedano, J. C., & López Carrascal, C. E. (2014). RNA-seq: herramienta transcriptómica útil para el estudio de interacciones planta-patógeno. Fitosanidad, 16(2), 101-113.

50. StataCorp, L. P. (2013). Stata power and sample-size reference manual.

51. Stuart, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. science, 302(5643), 249-255.

52. Team, R. C. (2013). R: A language and environment for statistical computing.

53. Team, R. C. (2000). R language definition. Vienna, Austria: R foundation for statistical computing.

54. Vega-Dienstmaier, J. M., & Arévalo-Flores, J. M. (2014). Clasificación mediante análisis de conglomerados: un método relevante para la psiquiatría. Revista de Neuro-Psiquiatría, 77(1).

55. Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research, 11(Oct), 2837-2854.

56. Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. Nature reviews genetics, 10(1), 57.

57. Yangxin, H., & Zhang, H. (2015). Finite Mixture Models and Their Applications: A Review. Department of Epidemiology and Biostatistics, College of Public Health, University of South Florida, USA.

58. Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. Statistical applications in genetics and molecular biology, 4(1).

Otros recursos:

59. GTEx portal

60. http://sherrytowers.com

61. Juan Botía Github

62. www.DataScience.com

63. www.wikipedia.com/wiki/RStudio

64. https://cran.r-project.org/web/packages/

Anexo guía

En este apartado se explica de forma esquematizada en qué carpeta y/o archivo se pueden encontrar el código de R, tablas de resultados y gráficas que no se han podido incorporar a la memoria del presente TFM por su tamaño o complejidad.

Código de R

1. Funciones principales y de apoyo: **Functions_methods.R**

   Este archivo contiene todas las funciones principales de los cuatro métodos de clustering implementados en el presente TFM además de otras funciones de .ªpoyoçomo por ejemplo para realizar gráficas más elaboradas, modificar datos etc.

2. Preprocesado, análisis exploratorio y visualización de los datos: **analysis_mydata.R**

   En este archivo se encuentra el código y funciones en R para realizar el pre-procesado y exploración de los datos usando estándares estadísticos básicos. También, está todo el código y funciones de R para visualizar los datos mediante los métodos PCA y TSNE en función del clustering generado por los métodos coexp, k-means y los dos modelos de mezcla (mclust y emmix).

3. Método coexp: **get_datafromCOEXP.R**

   En este archivo se encuentra el código en R para obtener los datos Cortex (GTEx) y las particiones generadas por el método coexp.

4. k-means: **k-means_for_mydata.R**

   En este archivo se encuentra el código en R y las funciones necesarias para ejecutar k-means utilizando los datos Cortex y obtener el número de clusters óptimo en base a los criterios BIC, AIC y método de codo.

5. FMM mclust y emmix: **FMM_for_mydata.R**

   En este archivo se encuentra el código en R y las funciones necesarias para ejecutar los dos modelos de mezcla (mclust y emmix) utilizando los datos Cortex y obtener el número de clusters óptimo en base a los criterios BIC, AIC y método de codo.

6. Medidas de comparación de clustering: **measures_for_clusterincomparison.R**

   En este archivo se encuentra todo el código en R y las funciones necesarias para aplicar las medidas de comparación de clustering ARI, NVI y NID sobre las particiones generadas por los métodos de clustering implementados en el presente TFM.

Tablas de datos

Las tablas de datos en formato .csv se encuentran en la capeta results.

1. Tabla valores BIC y AIC k-means: **tableBA_k-means.csv**

   En esta tabla se encuentran los valores BIC y AIC de todos los modelos k-means con un número de K = 2:50.

2. Tabla valores BIC y AIC emmix: **tableBA_FMMemmix.csv**

   En esta tabla se encuentran los valores BIC y AIC de todos los modelos FMM (emmix) con un número de K = 2:50.

3. Tabla valores BIC y AIC mclust: **tableBA_FMMmclust.csv**

   En esta tabla se encuentran los valores BIC y AIC de todos los modelos FMM (mclust) con un número de K = 2:50.

Gráficas

En este apartado estan todos los **links** para visualizar directamente las gráficas en formato **.pdf** que no se presentan en la memoria de TFM por su forma y tamaño pero son una parte fundamental de este TFM.

1. Gráficas PCA

   - Grafica 1: **PCA35_coexp.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad PCA en función de las particiones generadas por el método coexp.

   - Gráfica 2: **PCA35_kmeans.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad PCA en función de las particiones generadas por el método k-means.

   - Grafica 3: **PCA35_FMMe.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad PCA en función de las particiones generadas por el método FMM (emmix).

- Grafica 4: **PCA35_FMMmc.pdf**

  Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad PCA en función de las particiones generadas por el FMM (mclust).

2. Graficas T-SNE

   - Gráfica 5: **TSE35_coexp.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad TSNE en función de las particiones generadas por el método coexp con K = 35.

   - Gráfica 6: **TSE35_kmeans.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad TSNE en función de las particiones generadas por el método k-means con K = 35.

   - Gráfica 7: **TSE35_FMMe.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad TSNE en función de las particiones generadas por el método FMM (emmix) con K = 35.

   - Gráfica 8: **TSE35_FMMmc.pdf**

     Gráfica que representa los datos Cortex mediante el técnica de reducción de dimensionalidad TSNE en función de las particiones generadas por el método FMM (mclust) con K = 35.

     Hemos generado también las gráficas con otros valores de perplexibility pero al tratarse de datos de alta dimensionalidad los gráficos más adecuados son los que se han generado con los valores más altos de dicho parámetro.