Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte Final del Proyecto de Integración

Predicción del estado de cáncer de mama utilizando modelos de aprendizaje automático

Proyecto Tecnológico

Trimestre 2024 Otoño

Garcia Contreras Francisco Hugo
2182000391

Dra. Beatriz Adriana González Beltrán
Profesora Titular
Departamento de Sistemas

23 de febrero de 2024

Declaratoria

Yo, Beatriz Adriana González Beltrán, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Francisco Hugo Garcia Contreras, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Dedicatoria

Agradezco a mis padres por haber sido mi cimiento en todo momento. A mis hermanos por haber sido mi apoyo emocional. A mis amigos por haber compartido buenos momentos conmigo y haber estado en el trayecto. A la doctora Beatriz y al doctor Leonardo por haberme instruido en mi formación académica. Agradezco a Dios por permitirme concluir con una etapa muy importante de mi vida, así como el hecho de rodearme de personas maravillosas en el transcurso.

Resumen

El cáncer de mama es uno de los problemas que más aumento ha tenido en los últimos años, siendo foco de atención en todo el mundo debido a su severidad. Gracias a la recopilación de información genética hecha por clínicas, laboratorios y los profesionales de la salud, es posible el análisis de las propiedades que se encuentran en las lesiones producidas por el cáncer de mama.

Este proyecto realizó el análisis de datos genéticos correspondientes a los conjuntos de datos de cáncer de mama GSE10810, GSE61304, GSE45255, GSE20194, GSE20271, GSE22093 y GSE25066 obtenidos desde la página del *National Center for Biotechnology Information* (NCBI). Se utilizó una metodología para la normalización, reducción de dimensiones utilizando el Análisis de Componentes Principales (PCA), clasificación con el Método Empírico de Bayes (EB) y predicción de genes con mayor presencia en cada conjunto de datos de cáncer de mama, utilizando la Máquina de Vectores de Soporte (SVM) con una predicción del 85.71 %.

Palabras clave. Cáncer de mama, Aprendizaje automático, Predicción, SVM.

Índice general

Índice de figuras

1. Introducción

En la actualidad, el cáncer de mama es un problema relevante, ya que se considera que 1 de cada 12 mujeres pueden contraer esta enfermedad [1].

Es importante mencionar que el cáncer de mama suele clasificarse en etapas y el estado de los genes permite predecir la etapa en la que se encuentra el cáncer. La predicción del estado de cáncer de mama a través de los genes resulta un trabajo complejo, puesto que se necesita identificar los genes que se encuentran presentes en las etapas. Al día de hoy, existe una gran cantidad de datos genéticos de pacientes que han sido recolectados en hospitales, consultorios médicos y laboratorios, los cuales envían la información correspondiente a un centro de datos, de modo que esta información es sumamente extensa. El análisis de secuencias genéticas puede ser llevado a cabo a través del uso del aprendizaje automático.

El aprendizaje automático es una subcategoría de la inteligencia artificial, la cual consiste en la imitación de la inteligencia humana a través de acciones que podría realizar alguna persona [2]. El aprendizaje automático permite analizar, relacionar o descartar información. Cabe mencionar que el hecho de poder realizar el aprendizaje o la imitación de acciones facilita las tareas complejas como el análisis de grandes cantidades de datos. Es por esto que el uso del aprendizaje automático puede ser de mucha utilidad para analizar registros con respecto a pacientes que tienen cáncer de mama. Obteniendo los registros y analizando los genes que se encuentran en cada etapa del cáncer, es posible predecir, mediante modelos de aprendizaje automático, la etapa en que se encuentra el paciente. Esta predicción puede servir de apoyo a los profesionales de la salud para que decidan cuál sería el mejor tratamiento para el paciente (e.g. cirugía, quimioterapia, terapia hormonal).

2. Trabajos relacionados

- **Modelo en Machine Learning para el Diagnóstico del Cáncer de Mama** [3]. Este trabajo desarrolla un sistema usando modelos de aprendizaje automático para poder clasificar y diagnosticar cáncer de mama a través de información recolectada de muestras de masas mamarias del tipo FNA, donde la información recolectada será procesada por modelos de aprendizaje automático tomando en cuenta que la clasificación a realizar es con respecto a las medidas y características del área correspondiente a la lesión.

- **Comparación de técnicas de machine learning y estadística multivariante para la predicción del cáncer de mama utilizando biomarcadores obtenidos a partir de imágenes de resonancia magnética** [4]. Este proyecto compara técnicas de aprendizaje automático y estadísticas multivariante para la predicción de cáncer usando biomarcadores que fueron obtenidos mediante imágenes de resonancia magnética, tomando en cuenta secuencias DTI y DCE-MRI como biomarcadores. Los autores utilizaron los siguientes algoritmos: árboles de decisión, clasificadores bayesianos ingenuos (*Naïve Bayes*), Máquinas de Vectores de Soporte, regresión lineal, regresión logística, regresión de mínimos cuadrados parciales (PLS) y el algoritmo de los

método de los k vecinos más cercanos (kNN).

- **Reducción de dimensionalidad en Machine Learning. Diagnóstico de cáncer de mama basado en datos genómicos y de imagen** [5]. Este trabajo utiliza dos bases de datos *Breast Cancer Wisconsin Dataset* y *Breast Cancer Proteomes Dataset*. La primera contiene características extraídas de mamografías que permite detectar la enfermedad y la segunda tiene datos genómicos y se usa para la clasificación de tipos de cáncer de mama. Para realizar la detección, utilizan tres modelos: regresor bayesiano, regresión logística y máquinas de vectores de soporte (SVM) y después utilizan la técnica de análisis de componentes principales (PCA) para reducir la dimensionalidad del conjunto de datos y repiten el experimento y posteriormente comparan los resultados. Para resolver el problema de clasificación de tipos de cancer de mama, los autores utilizan técnicas de agrupamiento (*clustering*).
- **Análisis comparativo de predicción dentro de bases de datos de cáncer: una aplicación de aprendizaje automático** [6]. Los autores comparan el rendimiento de métodos para la predicción de cáncer a través del aprendizaje automático, destacando cuáles son los mejores del aprendizaje no supervisado como del aprendizaje supervisado mediante los siguientes algoritmos: kNN, SVM, métodos de Naïve Bayes y el perceptrón/retropropagación multicapa.
- **Metodología para el análisis e identificación de genes relacionados con cáncer de mama utilizando Machine Learning y datos de secuenciación masiva** [7]. Los autores analizan e identifican genes relacionados con el cáncer de mama haciendo uso del aprendizaje automático, utilizan un modelo de validación externo para la evaluación del rendimiento y utilizan técnicas de selección de características para determinar la secuencia de ADN de los genes relacionados con la lesión a través del uso del paquete Bioconductor, disponible para el lenguaje R.
- **Predicción de cáncer de mama utilizando BI-RADS y un clasificador binario basado en redes neuronales artificiales** [8]. Los autores predicen el cáncer de mama haciendo uso de un sistema de manejo de datos del reporte en la imagen mamaria para clasificar el tipo o clase de la lesión y a su vez se utiliza un clasificador binario mediante redes neuronales artificiales para el entrenamiento de RNA mediante datos de entrada.
- **Breast cancer stage prediction: a computational approach guided by transcriptome analysis** [9]. Los autores hacen predicción de genes a través de una metodología que consiste en reducción de genes usando una técnica de Análisis de Componentes Principales (PCA), posteriormente una selección de genes con el método Empírico de Bayes paramétricos (PEB) y una predicción de genes con el uso de una Máquina de Vectores de Soporte (SVM).

3. Justificación

La clasificación de genes en cada una de las etapas del cáncer de mama puede ser de mucha utilidad al momento de hacer el diagnóstico del paciente. Cada etapa del tumor

tiene una numeración distinta que va desde la etapa 0 hasta la etapa 4. La identificación de biomarcadores facilitaría la predicción de cáncer porque el uso de biomarcadores es un indicador de los eventos que ocurren en un sistema biológico.

Cabe mencionar que además de las etapas del cáncer de mama, se debe de considerar las formas en que se tratan estas lesiones. Si se pueden llegar a conocer los genes asociados a cada etapa del cáncer de mama, los profesionales de la salud pueden asociar la forma de curar la lesión, ya que estos juegan un papel muy importante en la caracterización de las lesiones y su recuperación depende mucho de la detección temprana del cáncer.

4. Objetivos

4.1. Objetivo General

Predecir el estado de cáncer de mama utilizando modelos de aprendizaje automático a través de la identificación de los genes asociados a cada etapa en que se encuentra el cáncer.

4.2. Objetivos Específicos

1. Identificar los datos genéticos que estén relacionados con las etapas del estado de cáncer de mama.
2. Implementar métodos para la limpieza de los datos genéticos.
3. Procesar el conjunto de datos para obtener los genes con mayor presencia en las etapas utilizando aprendizaje automático.
4. Evaluar los resultados obtenidos por los modelos utilizados y agruparlos por etapas.
5. Visualizar los genes con mayor presencia en cada etapa del estado de cáncer de mama.

5. Marco teórico

En esta sección se describen los conceptos relacionados con el cáncer de mama y el aprendizaje automático.

5.1. Predicción del cáncer de mama

La propagación del cáncer de mama es una enfermedad que requiere de una asignación de gravedad, es por ello que hay un proceso llamado estadificación que consiste en la determinación de la etapa de cáncer de mama [10]; es decir, describe cuanto cáncer hay en el cuerpo del paciente para saber la gravedad del mismo y así saber el tipo de tratamiento

se requiere para poder erradicar con las lesiones presentadas. Las etapas designadas para la estadificación son 5, partiendo de la etapa 0 siendo esta la etapa más temprana del cáncer de mama, hasta la etapa 4 donde esta es la más avanzada, debida a su propagación. Es importante mencionar que los tipos de cáncer con etapas similares por lo regular tiene el mismo pronóstico aunque la experiencia que vive el paciente con el cáncer es única.

Los características clave de la información del cáncer de mama son [10]:

Tamaño del tumor (T): tamaño del tumor primario correspondiente a su propagación a la piel o a la pared torácica inferior en el seno.

- T0: No hay evidencia del tumor primario.
- T1: Tumor con un tamaño de 2 cm o menos de ancho.
- T2: Tumor con un tamaño mayor de 2cm y menor de 5 cm ancho.
- T3: El tumor mide más de 5 cm ancho.
- T4: El tumor es de cualquier tamaño y crece hacia la pared torácica o la piel.

Propagación hacia los ganglios linfáticos adyacentes (N): Indica si el cáncer se propagó a los ganglios linfáticos adyacentes al seno, en caso afirmativo, el número de ganglios linfáticos fueron repercutidos.

- N0: El cáncer no se ha propagado a los ganglios linfáticos cercanos.
- N1: El cáncer se propagó a entre 1 y 3 ganglios linfáticos donde podrían ser ganglios linfáticos axilares o ganglios linfáticos mamarios.
- N2: El cáncer se propagó a entre 4 y 9 ganglios linfáticos donde podrían ser ganglios linfáticos axilares o ganglios linfáticos mamarios.
- N3: El cáncer se ha propagado a 10 o más ganglios linfático. Se encuentra cáncer en por lo menos un ganglio linfático axilar o el cáncer se ha propagado a los ganglios linfáticos que están sobre la clavícula.

Propagación a sitios distantes (M): Indica si el cáncer se ha propagado a órganos distantes como podrían ser los pulmones, huesos, cerebro o hígado.

- M0: No hay rastro de propagación a distancia en los estudios de imagen o exámenes médicos.

- M1: Hay rastro de propagación a distancia en los estudios de imagen o exámenes médicos.

La predicción del cáncer de mama es un gran reto debido a la complejidad genómica del ser humano. Si se quiere predecir el cáncer, es necesario tener los datos necesarios así como características muy puntales que compartan los pacientes desde el punto de vista genético. El comportamiento de los genes es difícil de predecir, ya que pueden mutar, crecer y sobrevivir, haciendo complicado el proceso de la identificación de biomarcadores predictivos, los cuales sirven como incentivos para darle seguimiento al comportamiento de los genes involucrados en los tumores. Entre otros inconvenientes para la predicción se encuentran la limitación de la recopilación de datos, la progresión del cáncer y la heterogeneidad del cáncer [11] .

5.2. Aprendizaje Automático

El aprendizaje automático es una subcategoría de la inteligencia artificial que consiste en la imitación de la inteligencia humana a través de acciones que podría realizar alguna persona [2]. El hecho de poder realizar el aprendizaje o la imitación de acciones facilita las tareas complejas como el análisis de grandes cantidades de datos. El aprendizaje automático fue de suma importancia para el proyecto presentado, ya que se utilizaron modelos de aprendizaje automático, tanto supervisados como no supervisados. A continuación se explican ambas categorías.

Aprendizaje automático supervisado: Se le conoce como supervisado debido a que este modelo necesita de la supervisión humana, ya que requiere de datos de entrada y de salida etiquetados durante la fase de entrenamiento.

Aprendizaje automático no supervisado: Se trata del entrenamiento de modelos de datos que no han sido procesados ni etiquetados. En este tipo de aprendizaje automático no se requiere de la intervención humana, ya que solo se necesita colocar parámetros del modelo y colocar los datos de entrada, puesto que el modelo es capaz de hacer el procesamiento de una gran cantidad de los conjuntos de datos.

5.3. Algoritmos utilizados

Existen diferentes algoritmos para llevar a cabo el aprendizaje automático. A continuación se explican los tres que se utilizaron en este proyecto.

Análisis de Componentes Principales
Es una técnica estadística que se utiliza en el análisis exploratorio de los datos y para hacer modelos predictivos. Consiste en la reducción de dimensionalidad de un conjunto de datos con muchas variables, tratando de mantener la mayor cantidad de información (varianza) relevante posible a través de la creación de componentes principales, donde

las componentes principales explican gran parte de la variabilidad de los datos. Cada componente principal es la combinación lineal de las variables originales. Es importante destacar es que en un análisis de datos mientras haya más componentes principales, el margen de error de los datos será mayor por el ruido de los datos, por lo tanto siempre se tiene que considerar un número óptimo de componentes principales. [12].

Método Empírico de Bayes
El Método Empírico de Bayes es una técnica que utiliza procedimientos bayesianos estándar para hacer una distribución. Se compara con etiquetas previamente asignadas a muestras y genes para determinar en qué grupo deben ser clasificadas las muestras. [13]. Una visualización que permite comprender la inferencia realizada es a través de un diagrama de Venn, donde los contrastes se muestran a partir de los genes totales siendo estos el universo total del diagrama, puesto que al hacer los contrastes habrá genes que no entren dentro de las inferencias.

Máquina de Vectores de Soporte
Es un método de aprendizaje automático supervisado que ha destacado por su precisión. Se considera uno de los mejores clasificadores dentro del ámbito del aprendizaje estadístico y aprendizaje automático. El funcionamiento de una máquina de vectores de soporte consiste en correlacionar los datos en un espacio de características de grandes dimensiones de forma que se pueda hacer una asignación de categorías. Cuando se desea asignar elementos a diferentes categorías, se necesita establecer una frontera entre ellas. Esta frontera, llamada hiperplano, divide el espacio de nuestros valores de manera que esté equidistante de cada categoría. Cuando añadimos elementos cercanos a esta frontera, el algoritmo los asigna a la categoría más cercana y trata de mantenerlos lo más alejados posible de los puntos cercanos, que llamamos observadores. [14].

5.4. Métricas de evaluación de modelos de aprendizaje automático

Los resultados de los modelos de aprendizaje automático pueden ser de cuatro tipos:

- Verdadero Positivo (VP). El resultado predicho es verdadero y el resultado real también lo es.
- Verdadero Negativo (VN). El resultado predicho es falso y el resultado real también lo es.
- Falso Positivo (FP). El resultado predicho es verdadero y el resultado real es falso.
- Falso Negativo (FB). El resultado predicho es falso y el resultado real es verdadero.

Los modelos de aprendizaje automático se pueden evaluar utilizando diferentes métricas.

- **Exactitud** (*Accuracy*): Esta medida nos permite conocer el porcentaje de predicciones correctas realizadas por el modelo en relación con el total de predicciones.

$$Exactitud = \frac{(VP + VN)}{(VP + FN + FP + VN)} \tag{1}$$

- **Sensibilidad o recuperación** (*Sensitivity or Recall)*: Es la proporción de instancias positivas que fueron identificadas correctamente por el modelo, y se calcula con:

$$Sensibilidad = \frac{VP}{VP + FN} \tag{2}$$

- **Especificidad** (*Specificity*): Es la proporción de instancias negativas que fueron identificadas correctamente por el modelo, y se calcula con:

$$Especificidad = \frac{VN}{VN + FP} \tag{3}$$

- **Precisión** (*Precision*):Es la proporción de predicciones positivas que son verdaderas.

$$Precisión = \frac{(VP)}{(VP + FP)} \tag{4}$$

- **F1-Score**: Es la media armónica de la precisión y recall, brindando un equilibrio entre ellas y se calcula con:

$$F1 = \frac{(1 + beta^2) * precisión * recall}{((beta^2 * precisión) + recall)} \tag{5}$$

  Donde beta = 1 para esta función

- **Prevalencia** (*Prevalence*): Se trata de la frecuencia con la que ocurre el evento de interés en el conjunto de datos.

$$prevalencia = \frac{(VP + FN)}{(VP + FP + FN + VN)} \tag{6}$$

- **Valor Predictivo Positivo** (*Pos Pred Value*): Es la proporción de instancias clasificadas como positivas que realmente son positivas y se calcula con:

$$PPV = \frac{(sensibilidad * prevalencia)}{((sensibilidad * prevalencia) + ((1 - especificidad) * (1 - prevalencia)))} \tag{7}$$

- **Valor Predictivo Negativo** (*Neg Pred Value*): Es la proporción de instancias clasificadas como negativas que realmente son negativas y se calcula con:

$$NPV = \frac{(especificidad * (1 - prevalencia))}{(((1 - sensibilidad) * prevalencia) + ((especificidad) * (1 - prevalencia)))} \tag{8}$$

- **Tasa de Detección** (*Detection Rate*): Es la proporción de casos positivos que fueron correctamente identificados por el modelo y se calcula con:

$$TasadeDetección = \frac{VP}{VP + FP + FN + VN} \tag{9}$$

- **Prevalencia de Detección** (*Detection Prevalence*): Es la proporción de casos que el modelo clasificó como positivos y se calcula con:

$$PrevalenciadeDetección = \frac{(VP + FP)}{(VP + FP + FN + VN)} \tag{10}$$

- **Exactitud Balanceada** (*Balanced Accuracy*): Es una medida de exactitud que tiene en cuenta el desequilibrio de clases y se calcula con:

$$ExactitudBalanceada = \frac{(sensibilidad + especificidad)}{2} \tag{11}$$

6. Metodología

Para poder llevar la predicción de la presencia de genes específicos en cada una de las etapas del cáncer, se llevó a cabo el proceso que se muestra en la Figura 1. Cabe señalar que este proyecto utiliza los mismos conjuntos de datos como la metodología que se encuentra en [9]. La variante radica en que se hizo una inferencia de genes para solo conservar los genes que se encontraran en los 7 conjuntos de datos, se eliminaron los genes que no se encontraban en los demás conjuntos de datos para después unir las 7 matrices y obtener una matriz unificada donde los renglones corresponden a los genes y las columnas a las 1077 muestras. Posteriormente, se usó la técnica de Análisis de Componentes Principales (PCA) usando el método de Elbow para obtener el número óptimo de la matriz unificada para la obtención de las cargas con el número de PC que proporciona el método de Elbow. Es importante menciona que solamente se usó el algoritmo de clasificación SVM, puesto que fue la técnica que tuvo mayor precisión en cuestión de la predicción de genes en [9].

6.1. Obtención del conjunto de datos de cáncer de mama

El objetivo de esta etapa consiste en encontrar información correspondiente a los genes que se encuentran en el cáncer de mama a través de conjuntos de datos de pacientes

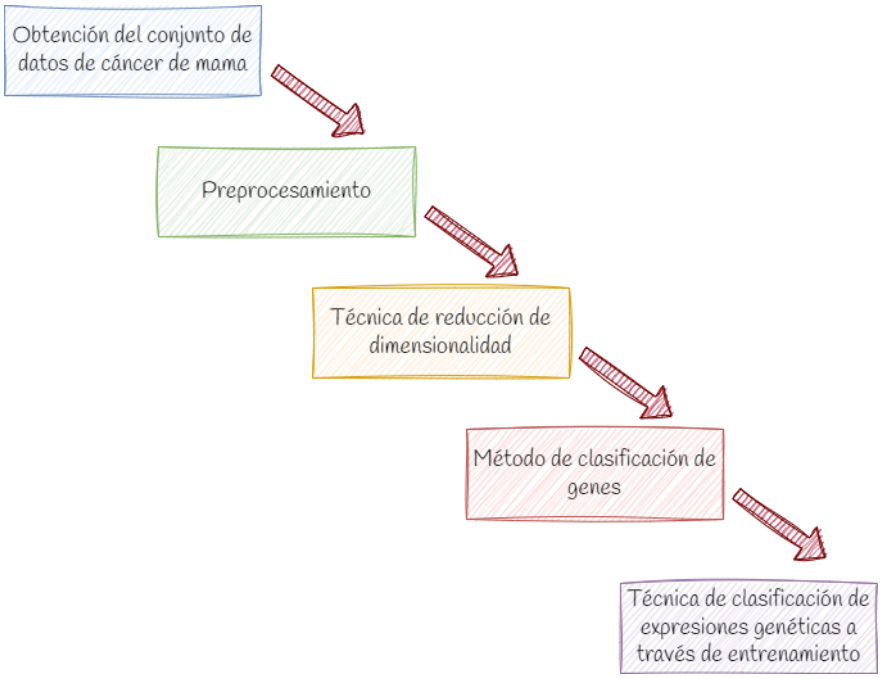

Figura 1. Diagrama en cascada de las etapas del proyecto

que fueron descargados desde la página del *National Center for Biotechnology Information* (NCBI) que contienen, por cada paciente, la etapa de cáncer y los genes correspondientes al paciente. Es importante mencionar que los conjuntos de datos a utilizar son los siguientes: GSE10810, GSE20271, GSE61304, GSE45255, GSE20194, GSE22093 y GSE25066.

6.2. Preprocesamiento

Esta etapa es la encargada de limpiar y obtener solo la información relevante con respecto a los criterios de interés, los cuales radican en obtener información relacionada con las etapas de cáncer, todo esto haciendo uso de un proceso de normalización y escalamiento con el objetivo de estandarizar los datos.

6.3. Técnica de reducción de dimensionalidad

En esta etapa se utiliza el análisis de componentes principales (PCA) para reducir la dimensionalidad de los conjuntos de datos y poder identificar la relevancia de los genes con respecto al valor de la varianza y la obtención de los genes que tengan un valor absoluto dentro del rango de retención establecido.

6.4. Método de clasificación de genes

Su principal función es agrupar los resultados obtenidos en la etapa anterior, todo esto para estimar parámetros y realizar inferencias de los grupos de las muestras que se encuentran con mayor relevancia en cada etapa haciendo uso del método Empírico de Bayes.

6.5. Técnica de clasificación de expresiones genéticas a través de entrenamiento

En esta fase se entrena una Máquina de Vectores de Soporte con genes preseleccionados a través de los pasos anteriores, con la finalidad de evaluar el rendimiento de la misma para determinar que tan bien generaliza la máquina con los datos no vistos.

7. Hardware y software necesario

7.1. Hardware

La computadora que se utilizó para realizar el proyecto tiene las siguientes características:

- Procesador Intel Core i3-3240 3.40GHz
- SSD Kingston 240gb
- 8 GB de memoria RAM
- Windows 10

7.2. Software

El software utilizado para la realización del proyecto es el siguiente:

- Lenguaje de programación R, versión 4.3.2, disponible en `https://cran.r-project.org/bin/windows/base/`
- Entorno de desarrollo integrado Rstudio, versión 2023.06.2-561, disponible en `https://posit.co/download/rstudio-desktop/`
- Bibliotecas pertenecientes a Bioconductor, versión 3.16, disponible en `https://www.bioconductor.org/install/`

8. Desarrollo del proyecto

Como punto de partida se instaló R 4.3.2 y RStudio 2023.06.2-561.

8.1. Obtención del conjunto de datos de cáncer de mama

Una vez instalados los programas, se descargaron los conjuntos de datos de cáncer de mama con los siguientes identificadores: GSE10810, GSE61304, GSE45255, GSE20194, GSE20271, GSE22093 y GSE25066. Los conjuntos de datos (datasets) fueron descargados desde la página del *National Center for Biotechnology Information* (NCBI)[1]. De cada conjunto de datos solo se seleccionaron las muestras que tenían las etapas de cáncer 1, 2 y 3. La etapa 0 significa que no hay evidencia de tumor y la etapa 4 fue descartada debido a que existían pocas muestras de esas etapas.

El sitio del NCBI cuenta con una herramienta de selección que permitió seleccionar las muestras requeridas, proporcionando un código inicial en lenguaje R para poder tener acceso a los conjuntos de requeridos. Las muestras obtenidas para cada conjunto de datos se muestran en la Figura 2.

Dataset	Número de muestras	Número de genes totales
GSE10810	29	18382
GSE61304	56	54675
GSE45255	139	22268
GSE20194	220	22283
GSE20271	124	22262
GSE22093	79	22283
GSE25066	430	22283

Figura 2. Conjuntos de datos de cáncer de mama

8.2. Preprocesamiento

Para realizar el preprocesamiento de los datos, se utilizó un proceso llamado RMA (del inglés, *Robust Multichip Average*), que permite pre-procesar los datos de un microarreglo con el fin de obtener estimaciones precisas de las expresiones génicas. Este método ayuda a corregir los defectos técnicos y biológicos inherentes a los microarreglos. Gracias a la herramienta proporcionada por la página del NCBI, no fue necesario realizar este proceso, ya que al descargar el código de R que proporciona NCBI, este proceso estaba incluido en las líneas de código que se proporcionan para obtener el conjunto de datos en cuestión.

Una vez obtenidos los conjuntos de datos, se llevó a cabo un proceso para obtener solo los genes que se encontraban en todos los conjuntos de datos. Para ello, se obtuvieron las etiquetas de los genes de cada conjunto de datos y se compararon para conservar únicamente los genes presentes en los 7 conjuntos de datos. Posteriormente, se realizó una reducción de genes y la unificación de los datos de los genes con respecto a las muestras. Como resultado, se obtuvo una matriz con un total de 11821 genes y 1077 muestras.

[1]https://www.ncbi.nlm.nih.gov

8.3. Técnica de reducción de dimensionalidad

Una vez obtenida la matriz unificada se llevó a cabo una técnica de reducción de dimensionalidad denominada PCA (del inglés, *Principal Component Analysis*) utilizando la biblioteca *PCAtools* de *Bioconductor*. La función usada fue `pca()`, en donde el parámetro de entrada que se colocó fue la matriz unificada. Ya obtenido nuestro análisis de componentes principales fue necesario saber cuál era el número óptimo de componentes principales (PC). Para ello, se utilizó el método `Elbow`, para obtenerlo se usó la función *findElbowPoint()*, dándonos 19 componentes principales como el número óptimo para el análisis. Puesto que se obtuvo el número óptimo de componentes principales sigue la reducción de características mediante el valor del rango de retención para los genes de interés, es por ello que se obtuvo los genes con respecto al valor absoluto menor a 0.08 con la función *plotloadings()*, dando como resultados un listado de genes dentro del rango establecido, dando un total de 144 genes, ya que el listado tiene la secuencia de los genes en cada PC y los genes que aparecen en el primer PC son los mismos que habrá en los siguientes. La relación componentes principales vs cargas de componentes se puede ver en la Figura 3.

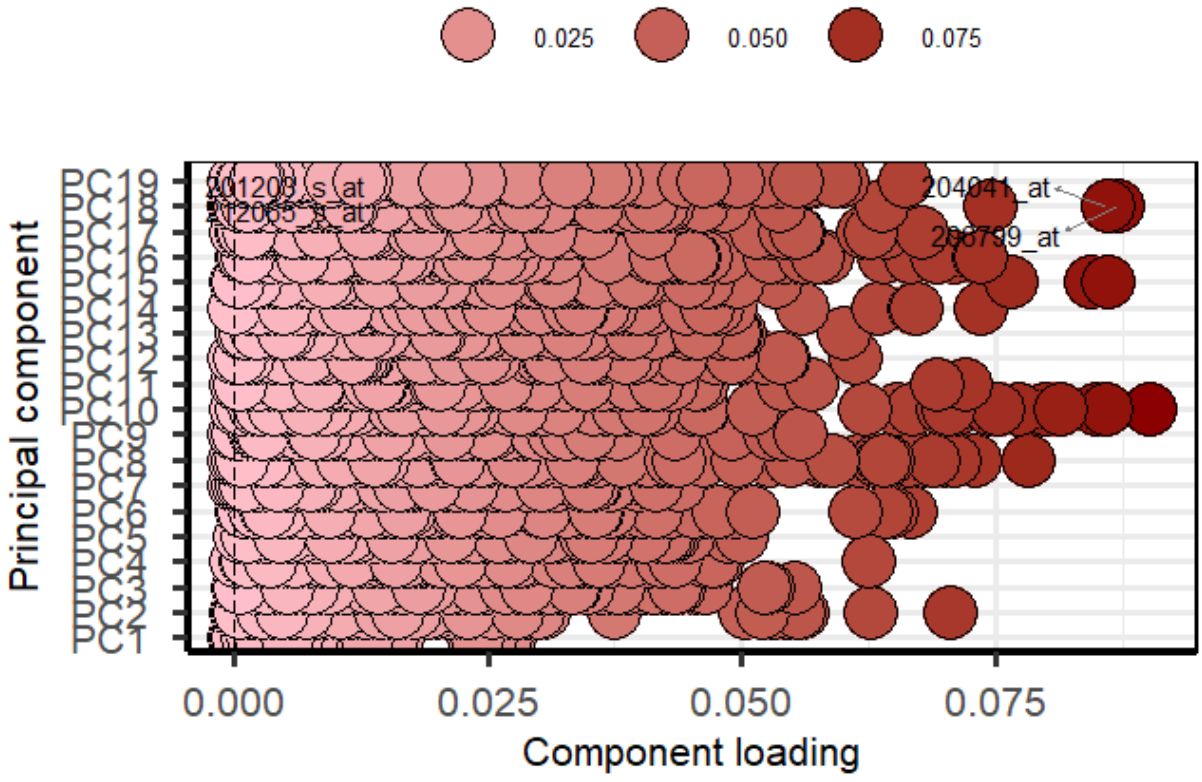

Figura 3. Gráfica de componentes principales vs cargas de componentes

Tras esto se hizo una reducción, haciendo una proyección de los 144 genes resultantes contra la matriz unificada para poder conservar los datos de expresión genética de los 144 genes con respecto a las 1077 muestras.

8.4. Método de clasificación de genes

Para poder clasificar los genes y asignarles la etapa de cáncer con mayor ranking se utilizó el método PEB (del inglés, *Parametric Empirical Bayes*). Para realizar este paso, fue necesario obtener las etapas de cáncer de cada muestra, ordenadas conforme a la matriz completa para poder crear un modelo de diseño para un ajuste de modelo lineal. Con la función `model.matrix()` se creó el modelo de diseño en donde solo van las etapas, más no la matriz de expresión. Posteriormente, se usó el paquete Limma de Bioconductor para hacer el modelo de ajuste lineal con la función `lmfit()`. Después se creó un contraste de matriz para hacer la comparativa por pares de cada columna designada; es decir, las columnas T1,T2 y T3 hechas en el modelo de diseño. El contraste se creó con la función `makecontrasts()`, luego se hizo el contraste lineal con la función `contrasts.fit()` usando el modelo lineal y el contraste obtenido de `makecontrasts()`. Una vez que se obtuvo el contraste para el modelo lineal se aplicó la función `eBayes()` para la expresión diferencial de bayes empírico, dándole como parámetro el modelo de contraste lineal de este proyecto. Para obtener la importancia de cada grupo se hizo uso de la función `topTable()` teniendo en cuenta solo los que tuvieran un valor ajustado de p valor menor a 0.05[2]. Los genes resultantes con sus respectivas inferencias fueron visualizados en un diagrama de Venn como se muestra en la Figura 4.

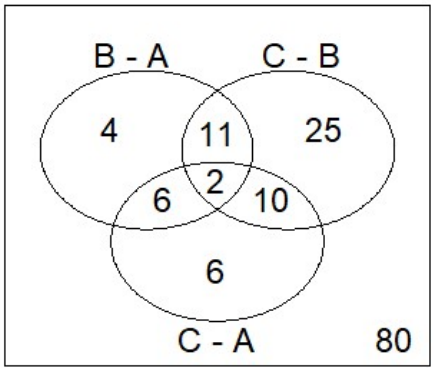

Figura 4. Diagrama de Venn de los genes con mayor presencia

Se agrupó cada conjunto de genes resultantes para después hacer un descarte de genes duplicados, ya que solo debe existir un gen en una sola etapa. Se realizó una proyección desde la etapa 3 hasta la etapa 1 donde se buscó que no hubiera ningún gen repetido en los otros conjuntos de genes. Una vez que se obtuvieron los genes resultantes se vuelve a hacer una proyección con la matriz de los 144 genes resultantes para solo conservar los genes que aparecen en los resultados de `topTable()`. Es importante mencionar que para acceder a cada conjunto o círculo del diagrama de Venn se tiene que cambiar el parámetro "coef = n" donde n es el número de contraste que se colocó, dando como total 38 genes resultantes los cuales se encuentran en la Figura 5 donde los primeros 13 genes corresponden a la etapa 1, los siguientes 20 genes pertenecen a la etapa 2 y los últimos 5 pertenecen a la etapa 3.

[2]Las dudas con respecto al manual de uso de Limma y sus funciones están en la guía de usuario: `https://bioconductor.org/packages/release/bioc/html/limma.html`

```
 [1] "204018_x_at" "209458_x_at" "211699_x_at" "211745_x_at" "217414_x_at" "204848_x_at" "209116_x_at"
 [8] "211696_x_at" "217232_x_at" "204351_at"   "211708_s_at" "208710_s_at" "216971_s_at" "204468_s_at"
[15] "219519_s_at" "222020_s_at" "201505_at"   "213764_s_at" "37892_at"    "204457_s_at" "208747_s_at"
[22] "215388_s_at" "203256_at"   "205157_s_at" "209156_s_at" "203153_at"   "208963_x_at" "216598_s_at"
[29] "204320_at"   "204734_at"   "201983_s_at" "204855_at"   "209351_at"   "219555_s_at" "213350_at"
[36] "209699_x_at" "211653_x_at" "203824_at"
```

Figura 5. Genes con mayor presencia

8.5. Técnica de clasificación de expresiones genéticas a través de entrenamiento

Para poder hacer uso de la máquina de vectores de soporte se convirtió la matriz resultante de las instrucciones anteriores en un *data frame* para que tuviera compatibilidad con los datos de entrada y no hubiera errores posteriores.

Después, se hizo una partición de muestras; es decir, las muestras resultantes de las instrucciones anteriores se dividieron en datos de entrenamiento y datos de prueba, siendo los datos de entrenamiento los datos brindados como entrenamiento a la Máquina de Vectores de Soporte, para esto se utilizó el 80 % de los 38 genes. Luego, se usó la función `svm()` de la biblioteca `e1071`, dándole como parámetro nuestros datos de entrenamiento y colocando el parámetro "kernel = linear". Una vez entrenada la máquina, se evaluó su desempeño con la función `ConfusionMatrix()` de la biblioteca `Caret`, dando como parámetro los datos de prueba. Una vez realizado este último paso se obtuvieron los resultados que se muestran en la Figura 6.

```
          Reference
Prediction 1 2 3
         1 2 0 1
         2 0 4 0
         3 0 0 0

$overall
      Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull AccuracyPValue
     0.8571429      0.7407407      0.4212768      0.9963897      0.5714286      0.1243408
 McnemarPValue
           NaN

$byClass
         Sensitivity Specificity Pos Pred Value Neg Pred Value Precision Recall  F1 Prevalence
Class: 1           1         0.8      0.6666667      1.0000000 0.6666667      1 0.8  0.2857143
Class: 2           1         1.0      1.0000000      1.0000000 1.0000000      1 1.0  0.5714286
Class: 3           0         1.0            NaN      0.8571429        NA      0  NA  0.1428571
         Detection Rate Detection Prevalence Balanced Accuracy
Class: 1      0.2857143            0.4285714               0.9
Class: 2      0.5714286            0.5714286               1.0
Class: 3      0.0000000            0.0000000               0.5
```

Figura 6. Resultados de la Máquina de Vectores de Soporte

9. Análisis y discusión de resultados

En primera instancia, la reducción de los conjuntos de datos de cáncer de mama consistió en la inferencia de los genes de cada uno, resultando en una matriz en la que solo los genes que se encontraban en los 7 conjuntos de datos fueron los resultantes. Este paso fue importante, puesto que se hizo un descarte primordial para la preparación de los datos genéticos para ser procesados con el método de Análisis de Componentes Principales, donde los resultados después de utilizar este método fueron retenidos, colocando un

rango máximo de retención de 0.08 del valor absoluto de la varianza de los genes, siendo estos un total de 144, ya que al hacer este proceso aparecen los mismos genes en todos los PC resultantes en el proceso de PCA, por lo tanto solo se tomaron los 144 genes que son lo que aparecen en el PC1.

La creación posterior del modelo de diseño depende de las etapas de cáncer de mama de cada una de las muestras, por lo tanto se creó un arreglo en donde vinieran las etapas correspondientes a cada muestra ordenadas de la misma manera en las que aparecen en la matriz resultante de los pasos anteriores; es decir, en la matriz unificada hay 1077 muestras las cuales corresponden a cada conjunto de datos de cáncer de mama, donde cada muestra tiene su respectiva etiqueta ordenada, entonces se hizo un arreglo que llevara el orden de estas muestras en donde solo existiera el dato de la etapa en que corresponde, en otras palabras u arreglo de 1077 datos con los valores ´1,2 o 3´ respectivamente.

Una vez que se obtuvo el arreglo de etapas de cáncer de mama y se colocó como parámetro de diseño de la función de `model.matrix()`, el resultado de esta función fue la asignación de la muestra a su etapa en una tabla. Esta tabla sirve como parámetro para el ajuste de modelo lineal de datos de la función `lmfit()` al igual que la matriz unificada donde solo nos interesan los 144 genes que obtuvimos en la retención. Es importante mencionar que la matriz unificada que necesitamos debe de contener las 1077 muestras y los 144 genes resultantes solamente. Después de haber ingresado esos parámetros, se obtuvo un ajuste de modelo lineal de los datos.

Como último requisito para proceder a la parte del uso de la expresión diferencial de Bayes se requiere de un contraste y dicho contraste requiere 3 casos para poder hacer una comparativa de pares, los 3 casos son: $B - A$, $C - B$ y $C - A$. Siendo A la etapa 1, B la etapa 2 y C la etapa 3. Posterior a la creación del contraste, se convirtió el contraste a un contraste de modelo lineal tomando como parámetros el ajuste de modelo lineal y el contraste.

Después se usó la función `eBayes()` para realizar la estimación de varianza empírica de los efectos diferenciales encontrados en las expresiones genéticas, dándole como parámetro nuestro contraste de modelo lineal. Luego se utilizó la función `topTable()` la cual devuelve las inferencias de cada contraste, por lo tanto obtuvimos 3 resultados, cada uno por cada contraste. Se tomó como rango máximo $0{,}05$ para los valores del `adj.p.values`, ya que corresponden a la varianza de los conjuntos de datos, donde los datos entran en un criterio que consiste en dos posibles categorías, hipótesis nula y la hipótesis alternativa. Ya que cuando un resultado radica cerca de cero significa que los resultados observados son estadísticamente significativos; es decir, que su concordancia no es casualidad y entran en la hipótesis alternativa. Por otro lado, cuando los datos son superiores al rango establecido significa que entran en favor a la hipótesis nula donde significa que los resultados no tienen mucha significancia en el análisis.
En el último paso de este proyecto se entrenó una Máquina de Vectores de Soporte para hacer predicción de datos a través de los datos obtenidos en los pasos anteriores, tomando solo el 80 % de los 38 genes resultantes, usando la función `svm()` se colocó el 80 % de las

muestras y la función `confusionMatrix` para obtener los resultados que se encuentran en la Figura 6.

La matriz de confusión permite observar las predicciones hechas correctamente con respecto a la etapa, en donde los renglones son las predicciones y las columnas corresponden a la referencia (etapa). La forma de interpretar los resultados es en diagonal; es decir, que los valores que se encuentren en las casillas (1,1), (2,2) y (3,3) son las predicciones que se realizaron correctamente. Eso quiere decir:

- se hizo correctamente la predicción de 2 genes pertenecientes a la etapa 1.
- se hizo correctamente la predicción de 4 genes pertenecientes a la etapa 2.
- en la etapa 3 la predicción fue errónea, ya que el resultado está en la casilla (1,3) en lugar de (3,3).

Por lo tanto, los números que están fuera de la diagonal de la matriz son las predicciones erróneas.

Las estadísticas de la matriz de confusión permiten observar el valor de cada métrica evaluada. Si se tiene un valor de 1.0 (100 %) significa que los genes fueron acertados en ese apartado, si el valor es 0.0 (0 %) significa que los genes no fueron acertados en ese apartado y Na indica que no se puede calcular el puntaje.

Cada apartado de la Figura 6 menciona lo siguiente:

- **Exactitud** (*Accuracy*):
  El valor de la exactitud es de 0.8571, eso quiere decir que el 85.71 % de las predicciones fueron correctas sobre el total de las predicciones realizadas.

- **Sensibilidad** (*Sensitivity*):
  Etapa 1: 1, significa que el modelo clasificó correctamente todas las muestras positivas.
  Etapa 2: 1, significa que el modelo clasificó correctamente todas las muestras positivas.
  Etapa 3: 0, significa que en esta etapa el modelo clasificó de manera errónea todas las muestras positivas.

- **Especificidad** (*Specificity*):
  Etapa 1: 0.8 significa que el modelo clasificó correctamente el 80 % de las muestras negativas.
  Etapa 2: 1.0 significa que el modelo clasificó correctamente el 100 % de las muestras negativas.
  Etapa 3: 1.0 significa que el modelo clasificó correctamente el 100 % de las muestras negativas.

- **Precisión** (*Precision*):
  Etapa 1: 0.6666667 significa que el 66.66 % de las predicciones positivas hechas por el modelo son correctas.
  Etapa 2: 1.0 significa que el 100.00 % de las predicciones positivas hechas por el modelo son correctas.
  Etapa 3: NaN significa que todas las predicciones positivas hechas por el modelo son incorrectas.

- **F1-Score**:
  Etapa 1: 0.8, indica un buen rendimiento aunque no fue perfecto en la precisión y la sensibilidad.
  Etapa 2: 1.0, indica un rendimiento perfecto en la precisión y la sensibilidad.
  Etapa 3: Na, indica que no se puede calcular el puntaje de F1 con respescto a la precisión y la sensibilidad.

- **Prevalencia** (*Prevalence*):
  Etapa 1: 0.2857 significa que el 28.57 % de las muestras positivas del conjunto de datos se encuentran en la etapa 1.
  Etapa 2: 0.5714 significa que el 57.14 % de las muestras positivas del conjunto de datos se encuentran en la etapa 2.
  Etapa 3: 0.1428 significa que el 14.28 % de las muestras positivas del conjunto de datos se encuentran en la etapa 3.

- **Valor Predictivo Positivo** (*Pos Pred Value*):
  Etapa 1: 0.6666667 significa que el 66.66 % de las predicciones positivas hechas por el modelo son correctas.
  Etapa 2: 1.0 significa que el 100.00 % de las predicciones positivas hechas por el modelo son correctas.
  Etapa 3: NaN significa que todas las predicciones positivas hechas por el modelo son incorrectas.

- **Valor Predictivo Negativo** (*Neg Pred Value*):
  Etapa 1: 1.0 significa que el 100.00 % de las predicciones negativas hechas por el modelo son correctas.
  Etapa 2: 1.0 significa que el 100.00 % de las predicciones negativas hechas por el modelo son correctas.
  Etapa 3: 0.8571 significa que el 85.71 % de las predicciones negativas hechas por el modelo son correctas.

- **Tasa de Detección** (*Detection Rate*):
  Etapa 1: 0.2857 indica que el modelo fue capaz de detectar correctamente el 28.57 % de todos los casos positivos en el conjunto de datos.

Etapa 2: 0.5714 indica que el modelo fue capaz de detectar correctamente el 57.14 % de todos los casos positivos en el conjunto de datos.
Etapa 3: 0.0000 indica que el modelo fue capaz de detectar correctamente el 0.00 % de todos los casos positivos en el conjunto de datos.

- **Prevalencia de Detección** (*Detection Prevalence*):
  Etapa 1: 0.4285 significa que el 42.85 % de las muestras positivas en el conjunto de datos se clasificaron correctamente.
  Etapa 2: 0.5714 significa que el 57.14 % de las muestras positivas en el conjunto de datos se clasificaron correctamente.
  Etapa 3: 0.0000 significa que el 0.00 % de las muestras positivas en el conjunto de datos se clasificaron de manera errónea.

- **Exactitud Balanceada** (*Balanced Accuracy*):
  Etapa 1: 0.9 indica que el modelo está clasificando correctamente el 90 % de las muestras en promedio
  Etapa 2: 1.0 indica que el modelo está clasificando correctamente el 100 % de las muestras en promedio
  Etapa 3: 0.5 indica que el modelo está clasificando correctamente el 50 % de las muestras en promedio.

En los apartados anteriores se mostró el porcentaje que tuvo cada etapa con respecto a la predicción realizada, arrojando una exactitud del 85.71 %. De igual forma, se mostró el porcentanje que se logró por cada etapa en los distintos apartados.

En [9] se obtuvo el 92.21 % de exactitud al utilizar el algoritmo de SVM. En este proyecto se esperaba lograr un mayor porcentaje de predicción; sin embargo, no fue así. Aunque el porcentaje que se obtuvo en este proyecto fue mayor al 80 % de exactitud, no se pudo superar la exactitud del reporte mencionado.

10. Conclusiones

Este proyecto presentó una clasificación de genes de cáncer de mama a través de una metodología que establece los pasos para obtener los resultados esperados. Se utilizaron técnicas de aprendizaje automático supervisadas y no supervisadas, incluyendo el análisis de componentes principales para reducir el conjunto de datos y preservar la varianza relevante. El proceso empírico de Bayes permitió inferir la relevancia de los genes según su etapa y clasificarlos en función de criterios específicos.

El uso de la máquina de vectores de soporte para este proyecto fue vital puesto que el proceso que realizó para la identificación de los elementos a considerar fue bastante bueno, ya que proporciono información bastante completa con respecto a los valores pertenecientes a la matriz de confusión.

En el trabajo [9] se obtuvo el 92.21 % de exactitud al utilizar el algoritmo de SVM. Sin embargo, en este proyecto se logró el 85.71 % de exactitud. Como trabajo a futuro, se puede utilizar aprendizaje profundo y otros algoritmos de clasificación para obtener mejores resultados tanto en la exactitud como en los genes que se encuentran en los conjuntos de datos así como los métodos de reducción de dimensiones de los datos de expresión genética, con la posibilidad de utilizar más de 7 conjuntos de datos genéticos para tener un mayor número de muestras y de datos genéticos.

Referencias

[1] "Cáncer de mama," julio 2023. [Online]. Available: https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer

[2] "Aprendizaje automático," 2023. [Online]. Available: https://www.nsnam.org/

[3] J. Robles Fajardo and J. Millan Gomez, "Modelo en machine learning para el diagnóstico del cáncer de mama," Repositorio Institucional de la Universidad Distrital Francisco José de Caldas, 2023.

[4] M. Alcalá Belmonte, "Comparación de técnicas de machine learning y estadística multivariante para la predicción del cáncer de mama utilizando biomarcadores obtenidos a partir de imágenes de resonancia magnética," Universitat Politècnica de València. Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, 2022.

[5] J. Galarza Hernández, "Reducción de dimensionalidad en machine learning. diagnóstico de cáncer de mama bsado en datos genómicos y de imagen," Universitat Politècnica de València. Departamento de Matemática Aplicada, 2017.

[6] E. R.-R. Gabriel MauricioMartínez-Toro, Dewar Rico-Bautista, "Análisis comparativo de predicción dentro de bases de datos de cáncer: una aplicación de aprendizaje automático," *RISTI*, vol. 19, no. E17, pp. 113–122, enero 2019.

[7] J. Liñares Blanco, "Metodología para el análisis e identificación de genes relacionados con cáncer de mama utilizando machine learning y datos de secuenciación masiva," Master's thesis, 2017.

[8] A. Hoyo, "Predicción de cáncer de mamas utilizando bi-rads y un clasificador binario basado en redes neuronales artificiales," Universidad Simón Bolívar Caracas, Venezuela, Dpto. Tecnología Industrial, 2008.

[9] G. G. Athira K, "Breast cancer stage prediction: a computational approach guided by transcriptome analysis." *Molecular Genetics and Genomics*, vol. 297, no. 6, p. 1467–1479, august 2022.

[10] "Etapas (estadios) del cáncer de seno," noviembre 2021. [Online]. Available: https://www.cancer.org/es/cancer/tipos/cancer-de-seno/comprension-de-un-diagnostico-de-cancer-de-seno/etapas-del-cancer-de-seno.html

[11] E. Maciá Antolinez, Núria; Bernadó Mansillas, "Métricas de complejidad para la transformación del problema de la detección de cáncer basado en mamografías," *Revista del Centro de Investigación. Universidad La Salle*, vol. 7, no. 28, pp. 69–92, dic 2007.

[12] "AnÁlisis de componentes principales (pca)," junio 2018. [Online]. Available: https://rpubs.com/Cristina_Gil/PCA

[13] "Método empírico de bayes," febrero 2024. [Online]. Available: https://academia-lab.com/enciclopedia/metodo-empirico-de-bayes/#

[14] "Las máquinas de soporte vectorial: una explicación completa," marzo 2021. [Online]. Available: https://academia-lab.com/enciclopedia/metodo-empirico-de-bayes/#

11. Apéndice A. Código fuente

11.1. Paquetes

```
install.packages(c("ggplot2", "e1071", "caret"))

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(version = "3.18")

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("limma")

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("PCAtools")

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

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

11.2. Conjunto de datos de cáncer de mama

```
library(GEOquery)
library(ggplot2)
library(PCAtools)
library(limma)
library(umap)
library(e1071)
library(caret)

#Obtención del dataset GSE10810 a traves de la plataforma NCBI
gset <- getGEO("GSE10810", GSEMatrix =TRUE, getGPL=FALSE)

if (length(gset) > 1) idx <- grep("GPL570", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
#Conjutos de muestras, donde las de digito 0 son de interes, mientras que las que t
gsms <- "XXXXXXXXXXXXXXXXXXXXXXXXXXXX000000000000000000000000000000XX"
sml <- strsplit(gsms, split="")[[1]]

# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
```

```
sml <- sml[sel]
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("total")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE10810", "/", annotation(gset), sep ="")

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE10810", "/", annotation(gset), " value distribution", sep ="")

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates


ex10810 <- ex
gset10810 <- gset


##############################GSE61304################################################


# load series and platform data from GEO

gset <- getGEO("GSE61304", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL570", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

gsms <- "00X000000000X00000000X000000000X00000000000000X0000000000000X"
sml <- strsplit(gsms, split="")[[1]]
```

```
# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
sml <- sml[sel]
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("Group")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
dev.new(width=3+ncol(gset)/6, height=5)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE61304", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs)
legend("topleft", groups, fill=palette(), bty="n")
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE61304", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, group=gs, main=title, legend ="topright")

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE61304")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)


ex61304 <- ex
gset61304 <- gset
```

```
##################################GSE45255#########################################

# load series and platform data from GEO

gset <- getGEO("GSE45255", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL96", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

# box-and-whisker plot
dev.new(width=3+ncol(gset)/6, height=5)
par(mar=c(7,4,2,1))
title <- paste ("GSE45255", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.7, notch=T, main=title, outline=FALSE, las=2)
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE45255", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, main=title, legend=F)

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE45255")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)

ex45255 <- ex
gset45255 <- gset


##################################GSE20194#########################################
```

```
# load series and platform data from GEO

gset <- getGEO("GSE20194", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL96", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

gsms <- paste0("00X00X0000X0000000000X000000000000000X0X0X000000XX",
               "00XX000000XX00000000000000000000XX00000000X00XX00000",
               "XX0000X0XXX0000X0X0X00X00000X00X00XX00X0X00000X000",
               "00X0000000X0X0XX000000000X000000X0000XX0X0000000XX",
               "00000000X0000X0X000X000000X0X0000000000000000000000",
               "000000X00000000000XX0000X00X")
sml <- strsplit(gsms, split="")[[1]]

# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
sml <- sml[sel]
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("group")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
dev.new(width=3+ncol(gset)/6, height=5)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE20194", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs)
legend("topleft", groups, fill=palette(), bty="n")
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE20194", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, group=gs, main=title, legend ="topright")
```

```
# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE20194")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)

ex20194 <- ex
gset20194 <- gset


################################GSE20271#########################################

# load series and platform data from GEO

gset <- getGEO("GSE20271", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL96", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

gsms <- paste0("000000X00000X00000X00XX0X0X0X00XX0XX0XXX0X000XX0X0",
               "X00XXXX0X0000000000000X000000X00000X000000X000000",
               "0X00X00XXXX0000000000XXXXX00X0XXX0000000000XXXXXX",
               "0X00000000000X00X0000000000X")
sml <- strsplit(gsms, split="")[[1]]

# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
sml <- sml[sel]
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("group")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
```

```
dev.new(width=3+ncol(gset)/6, height=5)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE20271", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs)
legend("topleft", groups, fill=palette(), bty="n")
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE20271", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, group=gs, main=title, legend ="topright")

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE20271")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)


ex20271 <- ex
gset20271 <- gset



##################################GSE22093#########################################

# load series and platform data from GEO

gset <- getGEO("GSE22093", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL96", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

gsms <- paste0("0000000000X000000000000000000000000000000X0000X",
               "0000X0000X0X00X00000X00XXXXXXX00000XX0X000000XX00X",
               "XXX")
sml <- strsplit(gsms, split="")[[1]]

# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
sml <- sml[sel]
```

```
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("group")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
dev.new(width=3+ncol(gset)/6, height=5)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE22093", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs)
legend("topleft", groups, fill=palette(), bty="n")
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
title <- paste ("GSE22093", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, group=gs, main=title, legend ="topright")

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE22093")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)

ex22093 <- ex
gset22093 <- gset


################################GSE25066#############################################

# load series and platform data from GEO
```

```
gset <- getGEO("GSE25066", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL96", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]

gsms <- paste0("00000X000X00000X000X00000000X0X0X0X000000000000000",
               "000000000X0000000000000000000000000000000000000X",
               "00000000000000000000000XXXXX00X000X00000X000X0X0000",
               "000000X00X0000X00X000000X000XX000000X0X0000000000X",
               "X0000X0X00000X00000X000000XX0000X0000000000000000",
               "000000000000XX00000000000000000X0X0X000000000XX000",
               "00X0XXXX0X0000000000000000000000000000000000000000",
               "00000000000000000000000000000000000000000000000X",
               "00X00X00XX00000X0X0000000000000X0000000000000000",
               "000X00000000X0XX0X00X0X00XX0XXX0X00X0XXX0X00000000",
               "0X000000")
sml <- strsplit(gsms, split="")[[1]]

# samples selection within a series
sel <- which(sml != "X")  # eliminate samples marked as "X
sml <- sml[sel]
gset <- gset[ ,sel]

ex <- exprs(gset)
# log2 transform
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0)
if (LogC) { ex[which(ex <= 0)] <- NaN
ex <- log2(ex) }

groups <- c("group")
gs <- factor(sml)
levels(gs) <- groups

# box-and-whisker plot
dev.new(width=3+ncol(gset)/6, height=5)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",
          "#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE25066", "/", annotation(gset), sep ="")
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs)
legend("topleft", groups, fill=palette(), bty="n")
dev.off()

# expression value distribution plot
par(mar=c(4,4,2,1))
```

```
title <- paste ("GSE25066", "/", annotation(gset), " value distribution", sep ="")
plotDensities(ex, group=gs, main=title, legend ="topright")

# mean-variance trend
ex <- na.omit(ex) # eliminate rows with NAs
plotSA(lmFit(ex), main="Mean variance trend, GSE25066")

# UMAP plot (multi-dimensional scaling)
ex <- ex[!duplicated(ex), ]  # remove duplicates
ump <- umap(t(ex), n_neighbors = 15, random_state = 123)
plot(ump$layout, main="UMAP plot, nbrs=15", xlab="", ylab="", pch=20, cex=1.5)
library("maptools")  # point labels without overlaps
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)

ex25066 <- ex
gset25066 <- gset
```

11.3. Principal

```
############################## FUNCIONES ########################################


reducirMatrix <- function(ex,conservar){
  indices <- match(conservar,rownames(ex))
  salida <- ex[indices, , drop=FALSE]

  return(salida)
}

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


#REDUCCIÓN DE LAS MATRICES CON RESPESCTO A LOS GENES QUE COINCIDEN EN
#LOS 7 DATASETS
lista <- list(rownames(ex10810), rownames(ex61304),rownames(ex45255),
              rownames(ex20194),rownames(ex20271),rownames(ex22093),
              rownames(ex25066))
genesIdenticos <- Reduce(intersect,lista)

exReducido10810 <- reducirMatrix(ex10810,genesIdenticos)
exReducido61304 <- reducirMatrix(ex61304,genesIdenticos)
exReducido45255 <- reducirMatrix(ex45255,genesIdenticos)
exReducido20194 <- reducirMatrix(ex20194,genesIdenticos)
exReducido20271 <- reducirMatrix(ex20271,genesIdenticos)
exReducido22093 <- reducirMatrix(ex22093,genesIdenticos)
exReducido25066 <- reducirMatrix(ex25066,genesIdenticos)
```

```
#OBTENCIÓN DE LA MATRIX COMPLETA EN DONDE LOS RENGLONES SON TODOS
#LOS GENES QUE COINCIDEN
# Y LAS COLUMNAS SON LAS 1077 MUESTRAS RECOLECTADAS
matrixCompleta <- cbind(exReducido10810,exReducido61304,exReducido45255
                        ,exReducido20194,exReducido20271,exReducido22093
                        ,exReducido25066)


#OBTENCIÓN DE LOS COMPONENTES PRINCIPALES DEL DATASET COMPLETO USANDO
#EL PROCESO DE PCA(principal component analysis)
PCA <- pca(matrixCompleta)


#OBTENCIÓN DEL NUMERO OPTIMO DE COMPONENTES PRINCIPALES A UTILIZAR
elbow <- findElbowPoint(PCA$variance)
elbow <- as.numeric(elbow)
componentesPrincipales <- c(1:elbow)


#OBTENCIÓN DE LOS GENES MAS RELEVANTES CON RESPECTO AL VALOR ABSOLUTO
#DE LOS VECTORES DE CARGA
#COLOCANDO UN RANGO DE RETENCION IGUAL A 0.08
cargas <- plotloadings(PCA,
                       components = getComponents(PCA, componentesPrincipales),
                       rangeRetain = 0.08, absolute = TRUE,
                       col = c('black', 'pink', 'red4'),
                       drawConnectors = TRUE, labSize = 4) + coord_flip()

cargasAbsolutas <- cargas$data


#EL NUMERO 144 CORRESPONDE A LOS GENES QUE SE OBTIENEN DE cargas$data,
#PERO COMO LA SECUENCIA DE
# LOS GENES SE REPITE SOLO SE TOMAN LOS QUE APARECEN
#EN EL PC1 YA QUE LOS GENES DEL PC2 EN ADELANTE SON LOS MISMOS
cargasnombres <-cargasAbsolutas$var[1:144]


# REDUCCIÓN DE MATRIX PARA CONSERVAR LOS GENES RELEVANTES
#Y SUS DATOS CORRESPONDIENTES DE EXPRESION GENETICA
matrixActualizada <- reducirMatrix(matrixCompleta,cargasnombres)


# OBTENCIÓN DE LA ETAPA DE CANCER DE MAMA DE CADA MUESTRA
#PARA SU POSTERIOR UNIFICACIÓN
```

```
etapas10810 <- c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,
                 2,2,2,2,2,3,3,3,3)

etapas61304 <- c(2,2,2,2,2,2,1,1,2,2,2,2,1,1,2,2,2,3,1
                 ,2,2,1,2,1,1,2,2,2,1,2,2,3,1,3,2,2,3,2,2,
                 2,2,2,2,3,2,2,1,3,3,2,3,2,1,1,2,2)

etapas45255 <-gset45255$`histological grade:ch1`
etapas45255 <- gsub("G2", "2", etapas45255)
etapas45255 <- gsub("G1", "1", etapas45255)
etapas45255 <- gsub("G3", "3", etapas45255)
etapas45255 <- gsub("NA", "1", etapas45255)
etapas45255 <- as.numeric(etapas45255)

etapas20194 <- gset20194$`tbefore:ch1`
etapas20194 <- as.numeric(etapas20194)

etapas20271 <- gset20271$`prechemo t:ch1`
etapas20271 <- as.numeric(etapas20271)

etapas22093 <- gset22093$`prechemo t:ch1`
etapas22093 <- as.numeric(etapas22093)

etapas25066 <- gset25066$`clinical_t_stage:ch1`
etapas25066 <- gsub("T1", "1", etapas25066)
etapas25066 <- gsub("T2", "2", etapas25066)
etapas25066 <- gsub("T3", "3", etapas25066)
etapas25066 <- as.numeric(etapas25066)

etapasTotales <- c(etapas10810,etapas61304,etapas45255,
                   etapas20194,etapas20271,etapas22093,etapas25066)


#CREACIÓN DEL MODELO DE DISEÑÓ PARA LAS ETAPAS
design <- model.matrix(~ 0+factor(etapasTotales))
colnames(design) <- c("A","B","C")


#AJUSTE DE MODELO LINEAL DE LOS DATOS
linear.fit <- lmFit(matrixActualizada,design)


#CREACIÓN DE CONTRASTE PARA EL MODELO LINEAL
contrast.matrix <- makeContrasts(B-A,C-B,C-A,levels = design)
contrast.linear.fit <- contrasts.fit(linear.fit,contrast.matrix)
```

```
#USO DE LA EXPRESSION DIFERENCIAL DE BAYES EMPIRICO
contrast.results <- eBayes(contrast.linear.fit)


#PARTICION DE LOS CONJUNTOS DE DATOS CON
#RESPECTO A CADA COMPARATIVA POR GRUPOS (1 VS 1)
etapa1 <- topTable(contrast.results,coef = 1,adjust.method = "BH",number = 144)
etapa2 <- topTable(contrast.results,coef = 2,adjust.method = "BH",number = 144)
etapa3 <- topTable(contrast.results,coef = 3,adjust.method = "BH",number = 144)
resultados1 <- etapa1[etapa1$adj.P.Val < 0.05,]
resultados2 <- etapa2[etapa2$adj.P.Val < 0.05,]
resultados3 <- etapa3[etapa3$adj.P.Val < 0.05,]

genesT1 <- rownames(resultados1)
genesT2 <- rownames(resultados2)
genesT3 <- rownames(resultados3)

analisisGenes <- decideTests(contrast.results)
print(analisisGenes)
summary(analisisGenes)


#DIAGRAMA DE VENN PARA LA VISUALIZACIÓN DE LA PRESENCIA DE GENES
vennDiagram(analisisGenes)


#GENES CON MAYOR RELEVANCIA B-A
genesUpT1 <- analisisGenes@.Data[,1]
genesUpT1 <- subset(genesUpT1,genesUpT1 == '1')

#GENES CON MAYOR RELEVANCIA C-B
genesUpT2 <- analisisGenes@.Data[,2]
genesUpT2 <- subset(genesUpT2,genesUpT2 == '1')

#GENES CON MAYOR RELEVANCIA C-A
genesUpT3 <- analisisGenes@.Data[,3]
genesUpT3 <- subset(genesUpT3,genesUpT3 == '1')

#PROYECCION DE LOS GENES PARA QUE SOLO PERTENEZCAN A UNA ETAPA Y NO A DOS
etiquetasT1 <- names(genesUpT1)
etiquetasT2 <- names(genesUpT2)
etiquetasT3 <- names(genesUpT3)

restantes3 <- setdiff(etiquetasT3,etiquetasT2)
restantes3 <- setdiff(restantes3,etiquetasT1)

restantes2 <- setdiff(etiquetasT2,etiquetasT3)
```

```
restantes2 <- setdiff(restantes2,etiquetasT1)

restantes1 <- setdiff(etiquetasT1,etiquetasT3)
restantes1 <- setdiff(restantes1,etiquetasT2)

replica1 <- rep(1,13)
replica2 <- rep(2,20)
replica3 <- rep(3,5)

unionGenes <- c(restantes1,restantes2,restantes3)

genEtapa <- c(replica1,replica2,replica3)

#REDUCCIÓN DE LA MATRIX PARA CONSERVAR UNICAMENTE LOS GENES DE INTERES
matrixSVM <- reducirMatrix(matrixActualizada,unionGenes)

matrixFinal <- as.data.frame(matrixSVM)
matrixFinal$etapa <- as.factor(genEtapa)


#ESTABLECER LA SEMILLA DEL GENERADOR DE NÚMEROS PSEUDOALEATORIOS
set.seed(123)

#CONVERSIÓN DE LA MATRIX TIPO MATRIX A TIPO DATAFRAME
matrixFinal <- as.data.frame(matrixFinal)

#UNION DE MATRIX CON DATOS DE ETAPA DE CANCER
matrixFinal$etapa <- as.factor(genEtapa)

#PARTICIÓN DE DATOS PARA TENER DATOS DE ENTRENAMIENTO Y DATOS DE PRUEBA
indices_entrenamiento <- createDataPartition(matrixFinal$etapa,
                                             p = 0.8, list = FALSE)
datos_entrenamiento <- matrixFinal[indices_entrenamiento, ]
datos_prueba <- matrixFinal[-indices_entrenamiento, ]

#CREACIÓN DE MODELO DE MAQUINA DE SOPORTE VECTORIAL
modelo_svm <- svm(etapa ~ ., data = datos_entrenamiento, kernel = "linear")

#EVALUACIÓN DE MAQUINA DE SOPORTE VECTORIAL
evaluacionSVM <-confusionMatrix(predict(modelo_svm, datos_prueba),
                                datos_prueba$etapa)

summary(evaluacionSVM)
```

12. Apéndice B. Manual del usuario

12.1. Instalación de bibliotecas

En primer lugar se instalan los paquetes necesarios para el proyecto. Por lo tanto se ejecutan las líneas de código que se encuentran en el apartado 11.1 paquetes del Apéndice A

12.2. Obtención de conjuntos de datos de cáncer de mama

Posteriormente, se descargan los conjuntos de datos de cáncer de mama de interés correspondientes a los datsets GSE10810, GSE61304, GSE45255, GSE20194, GSE20271, GSE22093, GSE25066. Este proceso se realiza con las líneas de código pertenecientes al apartado 11.2 Conjunto de datos de cáncer de mama del Apéndice A.

12.3. Ejecución del procedimiento del proyecto

Finalmente, para ejecutar el procedimiento completo del proyecto, se utilizan las líneas de código que se encuentran en el apartado 11.3 principal del Apéndice A.