Metodología.

Para el desarrollo de dicho apartado, se utilizaron paquetes del lenguaje de programación R en su versión 4.0.0, el cual forma parte de una plataforma de software libre enfocado en el análisis estadístico (Ihaka, 1993). Así mismo, para llevar a cabo el análisis de datos de *microarrays*, se implementa el uso de paquetes de R contenidos en *bioconductor*, un proyecto de código abierto desarrollado en 2004 como una colaboración para el desarrollo de *softwares* enfocados en la *bioinformática* y la *biología computacional* (Gentleman, 2004).

El script de análisis de la presente investigación es una adaptación basada en las enseñanzas del profesor Francisco J. Romero Campero de la Universidad de Sevilla en España (Romero, 2020). El script original del profesor se encuentra en el repositorio https://github.com/fran-romero-campero/miscellanomics y el que se desarrolla en este trabajo de investigación se encuentra disponible en el repositorio de github https://github.com/YLestrade/Analisis\_expresion\_genica\_LMC.

Tipo de datos

El tipo de datos con los que se trabajaron para realizar esta investigación son perfiles de expresión génica de 36 muestras pacientes adultos con diagnóstico de leucemia mieloide crónica. De los cuales 24 son sensibles al tratamiento con imatinib por un periodo de 12 meses y 12 pacientes que desarrollaron resistencia a imatinib en el mismo periodo de terapia. Dichos perfiles de expresión se generaron mediante la amplificación de ácido ribonucleico (ARN) en un chip de expresión génica de la casa comercial Affymetrix HG-U133 Plus 2.0. Cada chip de expresión detecta 54675 genes guardados en un fichero con extensión .CEL. La determinación de la expresión génica de los *chips de microarrays* la llevó a cabo el equipo de trabajo del Dr. Michael Deininger en el Hospital Universitario Salud y Ciencia Óregon en Estados Unidos y fueron publicados en enero del 2010 en la revista Blood, volumen 115(12), páginas 315-325, bajo el título *A gene expression signature of CD34+ cells to predict major cytogenetic response in chronic-phase chronic myeloid leukemia patients treated with imatinib*, mencionando como autor principal a la Dra Shannon K. McWeeney. Ese mismo año, los datos se registraron en el set genómico GSE14671 en la página del banco de datos del Centro Nacional para la Información Biotecnológica, por sus siglas en inglés National Center for Biotechnology Information (NCBI).

Análisis pre-analítico computacional

Este apartado tiene como propósito llevar a cabo un análisis pre-analítico que permita descargar y evaluar la calidad de los ficheros que contienen los datos de la expresión génica en los *chips de microarrays*, para identificar sí algún fichero se encuentra dañado y así poder descartarlo del estudio. Una vez que verificó que la calidad de los datos en bruto es buena y no se detectaron *chips* dañados, se procede a eliminar la variabilidad experimental de los datos. Posteriormente, se llevó a cabo la estimación de los niveles de expresión génica a partir de la intensidad de la fluorescencia y así, seleccionar los genes que se expresan de forma diferencial mediante estadística inferencial. finalmente, se realiza un análisis exploratorio de los genes expresados y reprimidos para identificar la ruta metabólica, funciones celulares implicadas, algún tipo de proteína o componente celular. El script completo se encuentra disponible en (poner liga de github).

Descarga de ficheros

Las líneas de comando para la descarga de los paquetes que se van a requerir para el análisis pre-analítico se encuentran en la página [www.bioconductor.org](http://www.bioconductor.org).

Se comienza con la instalación del paquete *BiocManager* (Bioconductor, 2003) desde el cran de R, usando la siguiente línea:

install.packages(“BiocManager”)

Posteriormente, se descargará el paquete *affy* (Gautier L, 2004) que permite leer archivos .CEL del *chip de microarray de Affymetrix.* Se abre la librería correspondiente con las líneas de comando que se muestra a continuación.

|  |  |  |
| --- | --- | --- |
| BiocManager::install(c("affy", “Biobase”,”GEOquery”)) | |  |
|  | library(affy) | |
| library(Biobase) | |
| library(GEOquery) | |

Los datos se descargan desde la dirección <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE14671> , en la parte final de la página se localiza el botón de descarga. En la página se muestran 59 muestras en total, estas se dividen en 12 muestras del set de entrenamiento resistente al tratamiento, 24 muestras del set de entrenamiento sensible al tratamiento, 6 muestras que pertenecen al set de validación resistente al tratamiento y 17 muestras del set de validación sensible al tratamiento. Para este estudio se analizó el set de muestras del conjunto de entrenamiento, divididas en 12 muestras resistentes y 24 muestras sensibles al tratamiento. Se utiliza la función *ReadAffy()* para descargar los ficheros desde el ordenador. El argumento verbose=TRUE, permite visualizar los ficheros en formato .CEL que se van agregando al objeto Base de Datos Microarray (GSE14671). Así, se puede tener una mejor visualización y certeza de que el objeto que se va a analizar se encuentre completo. Para saber el tipo de placa que se implementó basta con usar la función *cdfName()*. En este caso, los datos se obtuvieron mediante placas de hibridación HG-U133 Plus-2 de la casa comercial Affymetrix. Hay otra opción para hacer más amigable la selección de archivos sin tener que filtrar desde el ordenador, se descarga el paquete tkWidgets (Zhang, 2024) y desde una ventana emergente, se seleccionan los ficheros y se descargan de forma automática.

GSE14671<- ReadAffy(verbose=TRUE, widget=TRUE)

cdfName(GSE14671)

Análisis de la calidad de los datos

Visualización de placas

En este punto, se reproducen las imágenes de las placas que fueron obtenidas del escaner para identificar algún daño estructural o físico. En caso de encontrar placas dañas se tendrán que eliminar del estudio. Para esto, se utiliza la función *image* que se encuentra en el paquete *affy* (Gautier, Cope, Bolstad, & Irizarry, 2004) en la página de Bioconductor. Dicha función, traduce el valor de fluorescencia y lo asocia a un color específico con el argumento *col* y la función *rainbow* para generar 100 colores diferentes.

Se debe indicar la matriz de datos que contiene el conjunto de microarrays en formato .CEL, en el siguiente argumento se identifica el número de la columna que se desea mostrar. Para este caso, se ejemplifica con el comando que se utiliza para la visualización de la imagen obtenida del escaner de la primera muestra identificada como GSM366179 y corresponde a un paciente resistente a *imatinib*. Este procedimiento se debe de repetir para cada una de las placas.

image(GSE14671[,1], col=rainbow(100))

Control de calidad de la hibridación

En este punto, permite llevar a cabo un análisis de la calidad de los datos determinando el porcentaje de detección. Lo anterior corresponde al porcentaje de las sondas que lograron hibridarse en las que se detecta fluorescencia. El segundo valor determinado es la fluorescencia de fondo, que indica los sitios en donde no hubo hibridación, con el propósito de identificar las muestras que son viables para el estudio.

El control de calidad de hibridación se obtiene con la función *qc* dentro del paquete *simpleaffy* y se aplicar a la matriz de datos GSE14671 que contiene los ficheros .CEL como se muestra a continuación.

GSE14671\_qc<-qc(GSE14671)

plot(GSE1467\_qc)

Preprocesamiento de los datos

En este punto, se pretende eliminar la *variabilidad experimental* o de fondo (Fisher R.A., 1937) que se genera por la manipulación de las muestras, el proceso de hibridación, lectura de las placas o algún otro factor de aleatoriedad con respecto a los cambios genéticos inducido por el tratamiento de 12 meses con imatinib en pacientes adultos diagnosticados con LMC. Desde el comienzo de los estudios con microarrays las personas dedicadas al análisis genético, han enfrentado grandes retos, ya que la variabilidad en la señal de la sonda suele ser mayor en comparación con la variabilidad entre muestras o réplicas. Lo anterior sugiere, la presencia de un sesgo en la inexactitud de la medición de la fluorescencia contra la exactitud de la hibridación de la sonda (Jasik, Marczyk, Polanska, & Rzeszowska-Wolny, 2013).

--Texto sobre normalización de los datos-- (Kim K, 2010)

Se procede a realizar un gráfico boxplot para visualizar la distribución de la luminiscencia con el propósito de comprobar la existencia de variabilidad experimental de cada muestra con los siguientes comandos.

boxplot(GSE14671, col=rainbow(36), cex=0.8, las=2, ylab="Luminiscencia")

Posteriormente que se haya detectado la presencia de variabilidad experimental, Mediante el algoritmo Robust Multiarray Average (RMA) se logra eliminar el ruido de fondo haciendo una corrección de la fluorescencia, manteniendo así la variabilidad genética de los distintos pacientes.

El algoritmo RMA realiza las siguientes acciones:

1.- Corrección de fondo.

2.- Normalización los datos.

3.- Cálculo del nivel de expresión génica.

(poner referencias)

Los datos genómicos son transformados en logaritmo con base 2 mediante la expresión matemática ().

Se utiliza el siguiente comando para aplicar el algoritmo RMA a la matriz de datos GSE14671.

gse14671.norm<-rma(GSE14671)

Una vez realizado este paso, es conveniente generar otro gráfico de boxplot con la matriz gse14671 normalizada como método de verificación de que el algoritmo haya funcionado correctamente. En caso de contar con otro paquete que contenga la función rma, es conveniente utilizar el siguiente comando para realizar este paso.

gse14671.norm<-affy::rma(GSE14671)

Una vez que se aplicó el algoritmo correspondiente, se procede a generar la matriz de expresión que contiene los niveles estimados de la expresión génica de cada una de las muestras. Obteniendo una matriz *gxp* en donde *g* corresponde al identificador de la sonda y *p* a la muestra de cada paciente. Para esto, se requiere implementar la función *exprs* contenida en el paquete *Biobase* de *Bioconductor* a la matriz GSE14671 normalizada (gse14671.norm).

gse14671.exp<-exprs(gse14671.norm)

Matriz de expresión media

A partir de la matriz de expresión génica creada anteriormente y con el nombre gse14671.exp se procede a generar la matriz de expresión media de las muestras de los pacientes con LMC resistentes y sensibles a *imatinib*. Para poder simplificar este punto, se crea un vector que contiene los últimos dígitos del identificador de la muestra seguido de un guion (-) y la letra R para los pacientes resistentes al tratamiento y una S para los pacientes sensibles. Con la función *colnames*, se le asigna a la matriz de expresión gse14671.exp el vector que se creó para cambiar los nombres de las etiquetas. Finalmente, con la función *head*, se verifica que las etiquetas se hayan reemplazado correctamente. Cabe señalar que ambas funciones se encuentran en el paquete *base* de R.

ID<-c("179-R","180-R","181-R","182-R","183-R","184-R",

"185-R","186-R","187-R","188-R","189-R","190-R",

"191-S","192-S","193-S","194-S","195-S","196-S",

"197-S","198-S","199-S","200-S","201-S","202-S",

"203-S","204-S","205-S","206-S","207-S","208-S",

"209-S","210-S","211-S","212-S","213-S","214-S")

colnames(gse14671.exp)<-ID

head(gse14671.exp)

Se continuación, se calcula la media aritmética con la expresión matemática ().

Donde:

es la expresión génica de cada paciente. (columnas).

es la media aritmética.

Número total de pacientes.

Se suma cada columna de la matriz de expresión génica gse14671.exp del bloque de pacientes resistentes y se divide entre los 12 pacientes que forman el bloque.

resistente\_exp<-(gse14671.exp[,"179-R"]+

gse14671.exp[,"180-R"]+

gse14671.exp[,"181-R"]+

gse14671.exp[,"182-R"]+

gse14671.exp[,"183-R"]+

gse14671.exp[,"184-R"]+

gse14671.exp[,"185-R"]+

gse14671.exp[,"186-R"]+

gse14671.exp[,"187-R"]+

gse14671.exp[,"188-R"]+

gse14671.exp[,"189-R"]+

gse14671.exp[,"190-R"])/12

Se repite el mismo procedimiento con el bloque de pacientes sensibles, sólo que esta vez se va a dividir entre 24 que corresponde al total de pacientes.

sensible\_exp<-(gse14671.exp[,"191-S"]+

gse14671.exp[,"192-S"]+

gse14671.exp[,"193-S"]+

gse14671.exp[,"194-S"]+

gse14671.exp[,"195-S"]+

gse14671.exp[,"196-S"]+

gse14671.exp[,"197-S"]+

gse14671.exp[,"198-S"]+

gse14671.exp[,"199-S"]+

gse14671.exp[,"200-S"]+

gse14671.exp[,"201-S"]+

gse14671.exp[,"202-S"]+

gse14671.exp[,"203-S"]+

gse14671.exp[,"204-S"]+

gse14671.exp[,"205-S"]+

gse14671.exp[,"206-S"]+

gse14671.exp[,"207-S"]+

gse14671.exp[,"208-S"]+

gse14671.exp[,"209-S"]+

gse14671.exp[,"210-S"]+

gse14671.exp[,"211-S"]+

gse14671.exp[,"212-S"]+

gse14671.exp[,"213-S"]+

gse14671.exp[,"214-S"])/24

Una vez que se tiene las dos matrices por separadas, se va a generar la matriz de expresión media (gse14671.exp.media) que contiene la media aritmética de la matriz resistente\_exp y sensible\_exp. Con la función *matrix*, se indican las matrices que vamos a conjuntar, para este caso son resistente\_exp y sensible\_exp. Con el argumento *ncol*, se le indican que la nueva matriz va a contener dos columnas. A continuación, se crea un vector llamado ID que contiene las etiqueta de las columnas. En el comando siguiente, se indica que los nombres de las filas que se encuentran en la matriz resistente\_exp se los adjudique como nombre de las filas a la nueva matriz. Para finalizar, al vector ID que contiene el nombre de las columnas, se las adjudique nuevamente a la matriz. Se comprueba que la construcción de la matriz de expresión media sea correcta usando la función head.

gse14671.exp.media<-matrix(c(resistente\_exp, sensible\_exp), ncol=2)

ID<-c("resistente\_exp","sensible\_exp")

rownames(gse14671.exp.media)<-names(resistente\_exp)

colnames(gse14671.exp.media)<-ID

head(gse14671.exp.media)

Esta debe devolver una matrix *g x p* donde *g*= 54675 filas y *p*=2 columnas.

Comparación de genotipos

Una vez que se obtuvo la matriz de expresión media, se comparan los dos genotipos con la finalidad de identificar la diferencia de expresión entre los pacientes con LMC sensibles y resistentes a imatinib. Para este caso, se consideran a los pacientes sensibles al tratamiento como grupo control y a los resistentes al medicamento como grupo problema. Se procede a generar un gráfico de dispersión, en el eje X se colocarán a los pacientes sensibles al tratamiento y en el eje Y, a los pacientes resistentes al tratamiento y observará la dispersión que presenten las muestras.

Selección de genes expresados.

Factor de proporcionalidad (Fold-Change) (falta fundamentar)

Para llevar a cabo la selección de genes expresados del set GSE14671 de pacientes que son resistentes o sensibles a imatinib después de un tratamiento de 12 meses, se realiza el cálculo del factor de proporcionalidad o Fold-Change (FC). Partiendo de una matriz *g* x *m*, en donde *g* se localizan los genes y *m* las muestras de los pacientes sensibles y resistentes a imatinib, se procede a calcular la media de expresión de cada fila que contienen los genes de pacientes sensibles a imatinib (e1s) y la media de expresión de las filas de los genes de pacientes resistentes a imatinib (e1r). Como los datos de expresión se encuentran transformados en logaritmo con base 2 (log2), la media de la expresión génica resiste y sensible a imatinib se resta como lo indica la expresión matemática () y esta operación se realiza para cada uno de los genes de la matriz de expresión que se encuentran en cada fila.

Log2FC1 = e1s-e1c

Posteriormente, se fija un umbral en el factor de proporcionalidad para poder separar los genes activados de los silenciados. Para el caso de los genes activados se fija el umbral >2, 4 y 8. Esto significa que se identificarán los genes en ambos grupos de pacientes que presenten expresiones génicas mayores a dos, cuatro y ocho veces; como los datos que se utilizan se encuentra en log2 se identificaran los genes con expresiones equivalentes mayores a 1,2 y 3 respectivamente. Expresión matemática ()

FCk >2,4,8 ⎯→ log2FCk >1,2,3

Para la selección de genes silenciados o reprimidos se identifican las expresiones génicas que sean menores de la mitad, la cuarta parte y octava parte. Del mismo modo, como los datos se encuentran expresados en log2 se identifican las equivalencias de las expresiones menores a -1,-2 y-3 como lo indica la expresión matemática ().

FCk < ,, ⎯→ log2FCk <-1,-2,-3

Este método es intuitivo y es útil en diseños experimentales que presentan muy baja variabilidad. Además, no se requieren de numerosas replicas biológicas (Adler, 2018).

Inferencia estadística

Este segundo método de selección permite mantener un control sobre los genes que puedan arrojar un falso-positivo. Es decir, los genes que pueden tener expresiones génicas que se registren como activos cuando son genes silenciados. Para este procedimiento, partiendo de una matriz de expresión g x m en donde g son los genes expresados y m el número de paciente, se realiza un contraste de hipótesis de cada uno de los genes contenidos en las filas (g) donde la hipótesis nula se define como la igualdad de la expresión media de los genes en pacientes sensibles a imatinib (µ1s) y la expresión media de los genes de pacientes con resistencia a imatinib (µ1r). En contraste con la hipótesis alterna, se enuncia la diferencia entre la media de expresión de los pacientes sensibles y resistentes a imatinib, en la cual se busca esa diferencia que brinde significancia.

H0: µ1s = µ1r

H1: µ1s ≠ µ1r

Una vez que se tiene planteado el contrate de hipótesis, se llevó a cabo el cálculo de t-Student moderada. La ventaja de este estadístico es que utiliza el conjunto de genes completo para estimar la varianza de un gen en específico, partiendo del supuesto que todos los genes varían de alguna forma. El resultado que arroja el estadístico t-Student moderada es un p-valor con un nivel de significancia previamente establecido para cada uno de los genes que se encuentran en estudio. Para el estudio se consideró un nivel de significancia de p<0,01, lo que quiere decir que por cada 1000 genes analizados se obtendrá 10 falsos positivo. Para poder corregir el nivel de significancia debido a los múltiples tests, se implementó el método de Benjamin-Hochberg para la generación de un q-valor o False Discory Rate (FDR).

(explicar el método Benjamin-Hockberg)

Finalmente, se fija el umbral de significancia de 0,01 y 0,05 para los genes activados en pacientes sensibles y resistentes a imatinib mostrados en la expresión matemática ().Los genes que cumplan con la condición se clasifican como activados.

Log2FCk > 0 y q-valork < 0,05; 0,01

Para los genes silenciados se utiliza la expresión matemática (). Los que cumpla con dicha condición se identificarán como genes no expresados o silenciados.

Log2FCk < 0 y q-valork < 0,05; 0,01

Combinación de Fold-Change e inferencia estadística.

Para que la identificación de genes sea más confiable, se combinaron los métodos de Fold-Change e inferencia estadística. De tal forma, partiendo de la matriz g x m, en donde g son los genes contenidos en las filas y m los pacientes con leucémica mieloide crónica sensibles y resistentes a imatinib. Lo primero que se realiza es el cálculo de log2 del Fold-Change para cada uno de los genes, fijando un umbral de 2, 4, y 8. Para que se puedan identificar los genes activados con valores de log2FCk >1, 2 y 3. Así mismo, se calcula el q-valor mediante t-Student moderada y se fija un umbral de significancia de 0,05 y 0,01. Finalmente, queda la expresión matemática () que muestra la condición de deben de cumplir los genes para poder se clasificados como activados.

Log2FCk >1,2,3 y q-valork <0,05; 0,01

En el caso de los genes silenciados, se identificarán fijando un umbral de *Fold-Change* de , para obtener valores de log2FCk < -1,-2,-3. De la misma forma, se calculará el q-valor por medio de *t-Student moderada* y se fija un umbral de significancia de 0,05 y 0,01. Los genes que cumplan con la condición de la expresión matemática (), se identificarán como genes silenciados.

Log2FCk <-1,-2,-3 y q-valork <0,05 y 0,01

Para llevar a cabo la clasificación de los genes en activados y silenciados, se utiliza el paquete *Linear Model for Microarray Data (limma)* (Ritchie ME, 2015) y se siguen los siguientes pasos:

1.- Diseño de la matriz experimental.

2.- Cálculo de las medias de expresión para cada gen con respecto a cada paciente.

3.- Diseño de la matriz de contraste.

4.- Cálculo de *Fold-Change* y *p-valores*.

5.- Obtención de la matriz diferencial.

5.1.- Identificación de genes activados

5.2.- Identificación de genes silenciados.

7.- Generación del gráfico de dispersión.

Diseño de la matriz experimental

Con la ayuda de la función *model.matrix* que se encuentra en el paquete stats (falta referencia), se crea un matriz de diseño experimental que identifique las condiciones de los genotipos de los pacientes. Los que son resistentes a imatinib se identifican con la condición 1, los que son sensibles a imatinib con la condición 2 y su respectiva etiqueta. Finalmente, se obtener una matriz como se indica en la figura () con un total de 12 pacientes resistentes y 24 pacientes sensibles diferenciado la confición de cada uno.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Paciente | Resistente | Sensible |  | Paciente | Resistente | Sensible |
| 1 | 1 | 0 | 19 | 0 | 1 |
| 2 | 1 | 0 | 20 | 0 | 1 |
| 3 | 1 | 0 | 21 | 0 | 1 |
| 4 | 1 | 0 | 22 | 0 | 1 |
| 5 | 1 | 0 | 23 | 0 | 1 |
| 6 | 1 | 0 | 24 | 0 | 1 |
| 7 | 1 | 0 | 25 | 0 | 1 |
| 8 | 1 | 0 | 26 | 0 | 1 |
| 9 | 1 | 0 | 27 | 0 | 1 |
| 10 | 1 | 0 | 28 | 0 | 1 |
| 11 | 1 | 0 | 29 | 0 | 1 |
| 12 | 1 | 0 | 30 | 0 | 1 |
| 13 | 0 | 1 | 31 | 0 | 1 |
| 14 | 0 | 1 | 32 | 0 | 1 |
| 15 | 0 | 1 | 33 | 0 | 1 |
| 16 | 0 | 1 | 34 | 0 | 1 |
| 17 | 0 | 1 | 35 | 0 | 1 |
| 18 | 0 | 1 | 36 | 0 | 1 |

Figura(). Diseño de la matriz experimental.

En el script quedaría de la siguiente forma:

matriz.experimental<-model.matrix(~-1+factor(c(1,1,1,1,1,1,1,1,1,1,

1,1,2,2,2,2,2,2,2,2, 2,2,2,2,2,2,2,2,2,2,

2,2,2,2,2,2)))

A la matriz experimental se le asignan el nombre de las columnas “Resistente” y “Sensible”.

colnames(matriz.experimental)<-c("Resistente","Sensible")

Cálculo de las medias de expresión

Para poder obtener las medias de expresión de cada uno de los genes con respecto a cada paciente, se utiliza la función *lmFit* que ajusta un modelo lineal generalizado para cada gen dado en una serie de matrices. Se toman en consideración a la matriz que contiene los niveles de expresión (gse14671.exp) y la matriz del diseño experimental (matriz.experimental) como lo muestra la línea de comando a continuación.

mod.lineal.fit<- lmFit(gse14671.exp, matriz.experimental)

Diseño de la matriz de contraste

El siguiente paso que se realiza es la construcción de una matriz de contraste de las muestras de pacientes sensibles identificadas como *control* (-1) y las muestras resistentes como *problema* (1), como se indica en la tabla ().

|  |  |
| --- | --- |
| Nivel | Contraste |
| Resistente-Sensible |
| Resistente | 1 |
| Sensible | -1 |

Tabla (). Estructura de la matriz de contraste.

Se genera la matriz de contraste (matriz.contraste) en donde se realiza la diferencia del tratamiento – control. Lo anterior se realiza implementando la función *makeContrasts* que se encuentra en el paquete limma y se especifican las etiquetas utilizadas para identificar a los pacientes como resistentes y sensibles. Posteriormente, se indican los niveles de dichas etiquetas en orden.

matriz.contraste<- makeConstrasts(Resistente-Sensible,

levels=c(“Resistente”,”Sensible”))

Cálculo de Fold-Change y p-valores.

Una vez que te obtiene la matriz de contraste, se procede a calcular el factor de proporcionalidad o Fold-Change mediante un producto matricial en donde se utiliza la función *constrasts.fit.* Para generar la nueva variable que contiene dicho producto matricial, se implementa la matriz mod.lineal.fit y matriz.contraste como indica el comando a continuación.

mod.lineal.fit.contras<-constrasts.fit(mod.linear.fit , matriz.contraste)

Para calcular el p-valor, se utiliza la función *eBayes* que implementa una *t de Student* *moderada* basada en *inferencia bayesiana*. Para llevar a cabo lo anterior, la función se implementa a la variable modlineal.fit.contras que contiene el resultado del cálculo de los Fold-Changes como se muestra en la siguiente línea de comando.

result.inferencia<-eBayes(modlineal.fit.contras)

Para poder visualizar los resultados de la expresión diferencial de los genes basados en Fold-Change e inferencia estadística, se utiliza la función toptable

Matriz de expresión diferencial

Para obtener la matriz de expresión diferencias se requiere extraer los resultados de los Fold-Change y los p-valores de los genes de importancia. Para eso, se utiliza la función *toptable*, se debe de indicar el número de genes, que en este caso corresponden a 54,675; para el argumento *coef*, se le indicará el número de comparaciones que se hicieron para crear el modelo y se encuentra contenido en la matriz de contraste, en este caso *coef*=1, porque sólo se tiene una comparación (Resistente-Sensible). Se declara el argumento sort.by=”logFC” para que los resultados sean ordenados de acuerdo al Fold-Change. La línea de comando final queda de la siguiente forma:

expresion.diferencial<-topTable(result.inferencia, number = 54675, coef=1,

sort.by = "logFC")

De esta manera, tendremos en un *dataframe* que contiene en las filas la sonda de cada uno de los genes expresados de forma diferencial y en las columnas se indicará el log2 del Fold-Change (logFC), la media de expresión (AveExpr), t de Student moderada (t), p-valor(P.Value) y p-valor ajustado (adj.P.Value) que corresponde al q-valor. Para su consulta, la matriz se guarda con el nombre de “expresión diferencia.csv” y se encuentra en el repositorio https://github.com/YLestrade/Analisis\_expresion\_genica\_LMC en GitHub.

Se continúa con la extracción de la columna que contienen los valores de Fold-Change y se crea una matriz llamada FC.genes. Finalmente, se extrae los nombres de los genes para crear la matriz genes.id.

FC.genes<-expresion.diferencial$logFC

genes.id<-rownames(expresión.diferencial)

Para identificar los genes activados dentro de la matriz FC.genes, se procede a fijar el umbral de expresión cuyo Fold-Change sea >1 y con la función *length* se muestra los genes que cumplen con esa condición.

genes.activados<- genes.id[FC.genes > 1]

length(genes.activados)

En el caso de la identificación de los genes silenciados, se fija el Fold-Change < -1 y se muestran el número de genes que cumplen con la condición.

genes.silenciados<- genes.id[FC.genes < -1]

length(genes.silenciados)

Obtención de gráfico de dispersión

Una vez que se identificaron los genes activados y silenciados siguiendo el criterio del Fold-Change. El siguiente paso, es obtener el gráfico de dispersión como medio de representación utilizando la función plot.