Informe de práctica III
Algoritmos Genéticos
Selección de genes para clasificación en casos de leucemia aguda

Ana Medina García

1 de marzo de 2016

Introducción

El problema de la selección de variables

Los microarrays de ADN permiten monitorizar y medir los niveles de expresión de decenas de miles de genes simultáneamente en una muestra celular. Esta tecnología hace posible considerar el dignóstico de clasificación de cáncer basado en la expresión génica.

Podemos dividir la clasificación de muestras de cáncer en dos grandes retos: descubrimiento de clases y predicción de clases. El descubrimiento de clases se refiere a definir subtipos de tumor no reconocidos previamente.La predicción se refiere a la asignación, a muestras concretas de tumor, de clases ya definidas que pueden reflejar estados actuales o futuras consecuencias.

Dado el elevado número de genes, para esto es necesario seleccionar una cantidad más limitada o relevante de los mismos. El problema de la selección de características es un problema de optimización en el que debemos:

1. Buscar el espacio de posibles subconjuntos de características. Las estrategias de búsqueda pueden ser: exhaustivas (casi impracticables), heurísticas, o aleatorizadas.
2. Elegir el subconjunto que óptimo (o casi óptimo) con respecto a una función objetivo. Las estrategias de evaluación pueden ser: métodos “Filter” o métodos “Wrapper”.

Métodos Filter

Los métodos de filtrado consiguen la selección de genes independientemente del modelo de clasificación a utilizar. Se basan en un criterio que depende sólo de los datos para definir la importancia o relevancia de cada gen en la clasificación.

Como la mayoría son univariables, ignoran las correlaciones entre genes y obtienen subconjuntos que pueden contener información redundante.

Métodos Wrapper

Los métodos de envoltura seleccionan un subconjunto de genes mediante iteraciones con un clasificador. El objetivo es encontrar el subconjunto que alcanza el mayor rendimiento de predicción para un modelo de aprendizaje concreto. Estos métodos son computacionalmente intensivos, ya que se construye un clasificador para cada subconjunto candidato.

Algoritmos Genéticos

Entre las diferentes opciones, los algoritmos genéticos son probablemente la elección más popularizada. Los métodos empleados para la selección de genes utilizando algoritmos genéticos, comparten un conjunto de características comunes: representan un conjunto de genes preseleccionados con un vector binario, emplean procesos de entrecruzamiento y mutación estándar o específicos, y utilizan un clasificador concreto para la evaluación de la aptitud (función de fitness) de cada conjunto.

En esta práctica, proponemos un sencillo desarrollo, utilizando algoritmos genéticos, para seleccionar subconjuntos de genes para la clasificación de datos de microarrays, más concretamente, para la predicción de clases de cáncer.

Sistemas y métodos

Conjunto de datos de leucemia

Consideramos los datos de microarrays de pacientes con Leucemia, utilizados por T. R. Golub *et al.* (1999), para la clasifiación de leucemias agudas. El objetivo es la predición entre dos posibles tipos de leucemia aguda, siendo éstos ALL (leucemia aguda linfoide) y AML (leucemia aguda mieloide).

Nuestro conjunto inicial de datos consiste en 38 muestras (27 de ALL, 11 de AML) obtenidas de pacientes de leucemia aguda en el momento del diagnóstico. Se ha hibridado ARN de células mononucleares de médula ósea con microarrays de alta densidad de oligonucleótidos, producidos por Affymetrix y que contienen pruebas de 7129 genes. En cada una de las muestras, por cada gen, se tiene un nivel cuantitativo de expresión.

Para ello, en primer lugar cargamos los datos de train mencionados, dándoles el formato necesario para su posterior procesamiento, y comprobamos que las muestras se corresponden con las esperadas consultando las dimensiones de la estructura de datos (esperamos 38 muestras - 7129 genes para cada una) y el número de muestras de cada clase:

```
[1] "data.frame"
```

```
[1]   38 7129
```

```
ALL AML
 27  11
```

Pre-procesamiento de datos

Normalización

Una vez tenemos los datos, el primer paso es la normalización y/o escalado de los mismos. En nuestro caso, vamos a aplicar un procedimiento de normalización sencillo de forma que transformamos los datos a una distribución en el intervalo [0,1] según la siguiente fórmula:

$$v = x - min(x)/max(x) - min(x)$$

Eliminación de redundancias

Debemos tener en cuenta que existen muchos conjuntos de genes que están muy relacionados entre sí, de tal forma que el nivel de expresión es casi directamente proporcional entre unos y otros dentro del grupo. Esto puede dar lugar a redundancias en los datos.

Para eliminar los genes que supongan redundancias, buscamos las correlaciones entre los genes y, ordenándolos, eliminaremos los que se tengan un coeficiente de correlación mayor a 0.9 con otro gen que represente mejor el patrón de expresión de ese conjunto. De esta forma, nos quedamos con un subconjunto de 6890 genes.

Filtrado por correlación

Para afrontar el problema, en primer lugar debemos explorar si existen genes cuyo patrón de expresión parece correlacionado con la distinción de clases que se pretende predecir. Con este objetivo, ordenamos los 6890 genes por su grado de correlación.

Según estudios previos sobre los datos, podemos confiar en que existe una correlación elevada de muchos de los genes con la distinción de clases AML-ALL. *"Alrededor de 1100 genes muestran más correlación de lo que se espera por azar"* (Golub *et al.*).

Por lo tanto, para realizar una preselección de los genes con potencial para la predicción de clases, ordenamos todos los genes por su correlación. La ordenación debe realizarse utilizando el valor absoluto de correlación, ya que debe tener la misma relevancia un gen muy sobreexpresado para una clase que uno muy infraexpresado.

Escogeremos el subconjunto de los 250 primeros genes. Esta cantidad ha sido elegida con el fin de optimizar la ejecución del posterior algoritmo genético.

Selección de genes mediante Algoritmos Genéticos

El segundo reto a afrontar del problema, es cómo utilizar un conjunto de muestras conocidas para crear un "predictor de clases" capaz de asignar su correspondiente clase a una nueva muestra. Para llevar a cabo esta tarea, se ha desarrollado un modelo aplicando un algoritmo genético con un clasificador sencillo, sobre los 250 genes preseleccionados.

Definición de la función de fitness

Con el objetivo de encontrar el subconjunto de genes que obtiene un mayor rendimiento predictivo con el mínimo número de genes seleccionados, se ha implementado una función de fitness que tiene en cuenta estos dos parámetros.

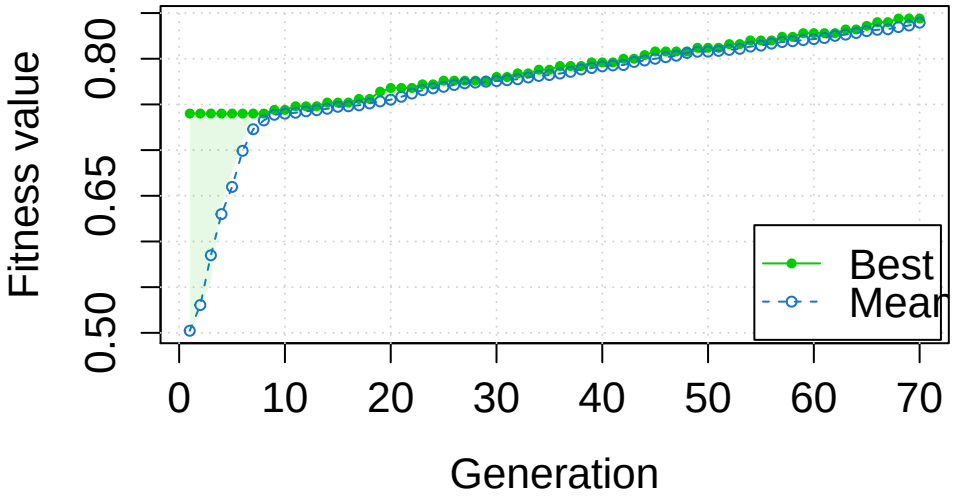
La función utiliza la regresión logística para medir la precisión de predicción de cada subconjunto candidato. El cálculo del valor de fitness persigue maximizar esta capacidad de predicción y minimizar el número de genes necesario para la misma:

```
> fitness.glm <- function(chromosome) {
+     forml <- as.formula(paste("Class~", paste(colnames(leukemia.best.corr[,
+         which(chromosome == 1)]), collapse = "+"), sep = ""))
+     regLog <- glm(forml, data = leukemia.best.corr, family = binomial("logit"))
+     pred <- predict(regLog, leukemia.best.corr, type = "response")
+     pred.th <- pred
+     pred.th[pred.th < 0.5] <- 0
+     pred.th[pred.th >= 0.5] <- 1
+     confMatrix <- confusionMatrix(pred.th, leukemia.best.corr$Class)
+     accuracy <- confMatrix$overall[1]
+     result <- accuracy - sum(chromosome)/gene.number
+     return(result)
+ }
```

Una vez definida la función de fitness, ejecutamos el algoritmo genético mediante el siguiente código. Los parámetros del mismo, los cuales han sido seleccionados mediante un prolongado proceso de pruebas, son:

- Tamaño de población: 40
- Número de generaciones: 70
- Probabilidad de entrecruzamiento: 0.8
- Probabilidad de mutación: 0.1

El algoritmo alcanza un valor de fitness de 0.844. A continuación podemos ver la gráfica de resumen de su evolución:

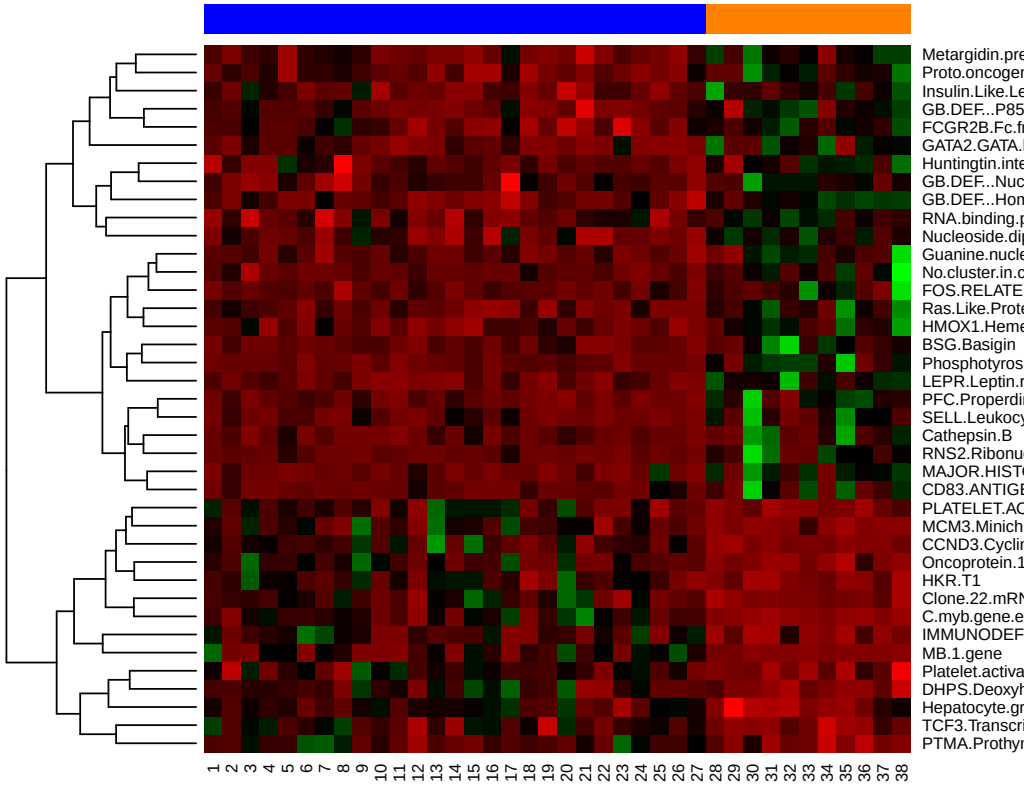


Los genes seleccionados por nuestro algoritmo genético son los siquientes:

```
 [1] "LEPR.Leptin.receptor.1"
 [2] "GB.DEF...Homeodomain.protein.HoxA9.mRNA"
 [3] "Phosphotyrosine.independent.ligand.p62.for.the.Lck.SH2.domain.mRNA"
 [4] "MAJOR.HISTOCOMPATIBILITY.COMPLEX.ENHANCER.BINDING.PROTEIN.MAD3"
 [5] "PFC.Properdin.P.factor..complement"
 [6] "C.myb.gene.extracted.from.Human..c.myb..gene..complete.primary.cds..and.five.complete.al
 [7] "RNS2.Ribonuclease.2..eosinophil.derived.neurotoxin..EDN."
 [8] "Metargidin.precursor.mRNA"
 [9] "Hepatocyte.growth.factor.like.protein.gene"
[10] "Ras.Like.Protein.Tc10"
[11] "BSG.Basigin"
[12] "HMOX1.Heme.oxygenase..decycling..1"
[13] "GATA2.GATA.binding.protein.2"
[14] "Nucleoside.diphosphate.kinase"
[15] "Proto.oncogene.BCL3.gene"
[16] "No.cluster.in.current.Unigene.and.no.Genbank.entry.for.U77396..qualifier.U77396_at."
[17] "HKR.T1"
[18] "Oncoprotein.18..Op18..gene"
[19] "Platelet.activating.factor.acetylhydrolase.IB.gamma.subunit"
[20] "CCND3.Cyclin.D3"
[21] "DHPS.Deoxyhypusine.synthase.1"
[22] "RNA.binding.protein.CUG.BP.hNab50..NAB50..mRNA"
[23] "Clone.22.mRNA..alternative.splice.variant.alpha.1"
[24] "MCM3.Minichromosome.maintenance.deficient..S..cerevisiae..3"
[25] "MB.1.gene"
[26] "Cathepsin.B"
[27] "SELL.Leukocyte.adhesion.protein.beta.subunit.1"
[28] "Insulin.Like.Leydig.Hormone"
[29] "GB.DEF...P85.beta.subunit.of.phosphatidyl.inositol.3.kinase"
[30] "TCF3.Transcription.factor.3..E2A.immunoglobulin.enhancer.binding.factors.E12.E47."
[31] "Huntingtin.interacting.protein..HIP1..mRNA"
```

```
[32] "FCGR2B.Fc.fragment.of.IgG..low.affinity.IIb..receptor.for..CD32."
[33] "Guanine.nucleotide.regulatory.protein..G13..mRNA"
[34] "PLATELET.ACTIVATING.FACTOR.ACETYLHYDROLASE.45.KD.SUBUNIT"
[35] "GB.DEF...Nuclear.factor.kappa.B2..NF.KB2..gene..partial.cds"
[36] "PTMA.Prothymosin.alpha"
[37] "CD83.ANTIGEN.PRECURSOR"
[38] "IMMUNODEFICIENCY.VIRUS.TYPE.I.ENHANCER.BINDING.PROTEIN.2"
[39] "FOS.RELATED.ANTIGEN.2"
```

El número de genes seleccionados es 39. Para visualizar la expresión de los mismos, utilizaremos un mapa de calor, en el cual podemos observar a simple vista si los genes se expresan diferencialmente entre las dos clases de leucemia aguda (en la barra superior: ALL -azul-, AML -naranja). Las columnas representan las 38 muestras y las filas la expresión de los genes escogidos:

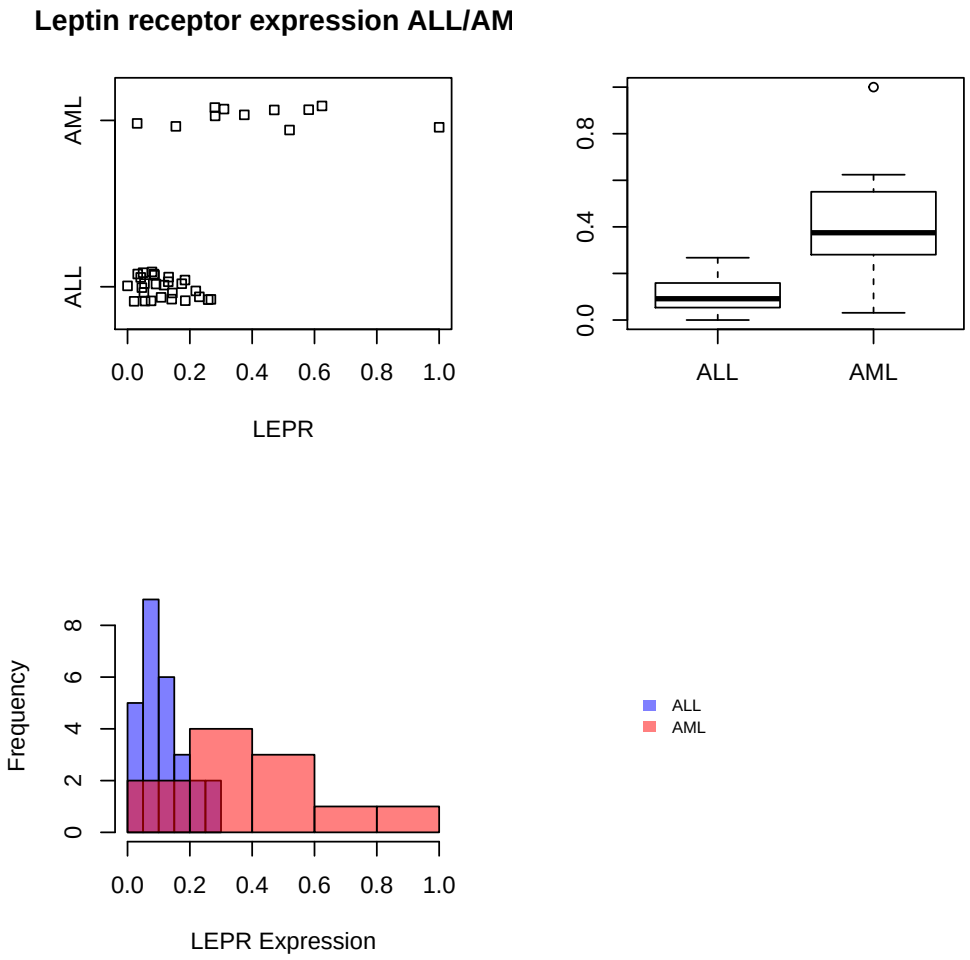


Como podemos observar, entre los genes seleccionados encontramos algunos que coinciden con los seleccionados por Golub *et al.* como pueden ser: *MB-1* (cuya utilidad en la diferenciación de células de linaje linfoide o mieloide ya fue demostrada anteriormente), *Leptin receptor* (se ha demostrado que el receptor de leptina tiene función anti-apoptótica en células hematopoyéticas), *Op18*, *Cyclin D3* y *MCM3* (las tres codifican proteínas críticas para la fase S del ciclo celular),o *E2A* y *HoxA9* (ambos son conocidos oncogenes); así como factores de transcripción.

En cambio, nuestro algoritmo no ha seleccionado varios de los genes nombrados en el artículo como el *Zyxin*, el *CD33* y el *CD11c*, que tienen relación con la adhesión celular. Aunque sí podemos encontrar en la lista otros genes relacionados con la adhesión celular como el *CD83* (un receptor de

adhesión de la lectina de tipo I que se une a monocitos y un subconjunto de células-T CD8+ activadas, (`http://www.ncbi.nlm.nih.gov/pubmed/11238630`), o el *selectin L* (otro receptor de adhesión, en este caso de lectina tipo C). El producto de este útimo gen se uiliza para la unión y la posterior circulación de los leucocitos en las células endoteliales, lo que facilita su migración hacia los órganos linfoides secundarios y los sitios de inflamación (`http://www.ncbi.nlm.nih.gov/gene/6402`).

A continuación, podemos observar con más detalle la expresión diferencial de algunos de estos genes:

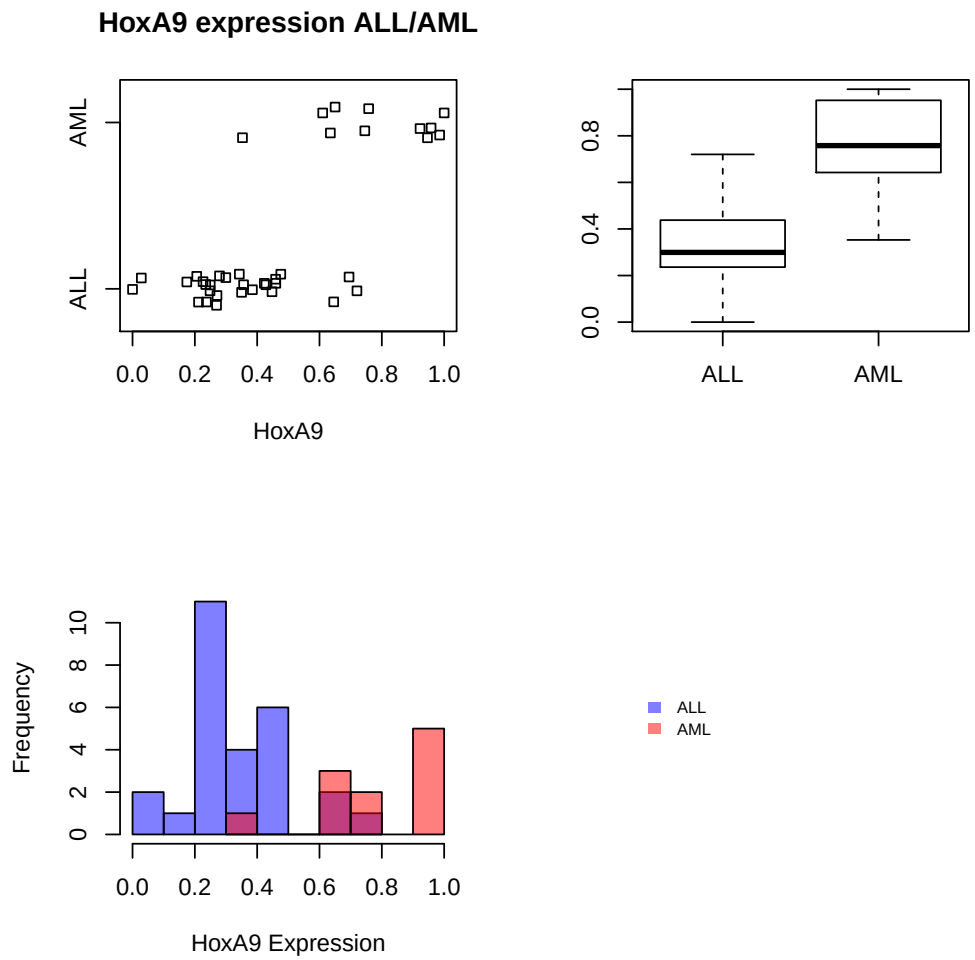


El gen recector de leptina, se expresa diferencialmente como cabía esperar, siendo sus niveles de expresión mucho más altos para pacientes con leucemia aguda mieloide.

En un estudio previo, ya comentado por Golub *et al.* en su artículo, se demostraba que, además de su relación con la regulación del peso, también tiene función antiapoptótica en células hematopyéticas.

La proteína codificada por este gen pertenece a la familia gp130 de receptores de citocinas, que estimulan la transcripción mediante la activación de las proteínas citosólicas STAT. Esta proteína es un receptor de leptina (una hormona que regula el peso) y está involucrado en la regulación del metabolismo, así como en una ruta hematopoyética requerida para la linfopoyesis normal.

Su información en NCBI: `http://www.ncbi.nlm.nih.gov/gene/3953`

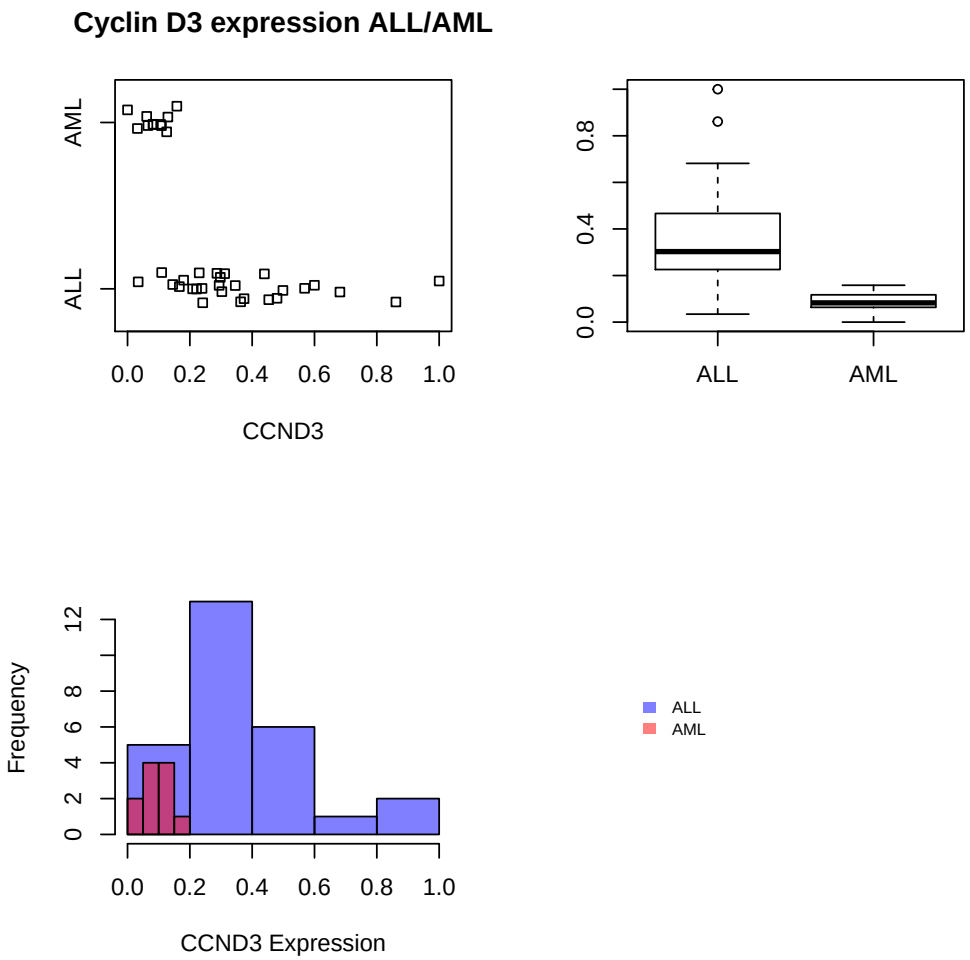


Podemos observar como el gen Homeobox A9 se expresa diferencialmente para los dos tipos, apreciando unos niveles más elevados en los casos de leucemia aguda mieloide.

El gen Homeobox A9 codifica un factor de transcripción que regula la expresión génica, la morfogénesis y la diferenciación. Se ha demostrado que una traslocación específica de este gen y el gen NUP98 está relacionada con la leucemogénesis mieloide.

Debido a que la disfunción de HOXA9 parece relacionada con la leucemia aguda mieloide y que la expresión del gen parece ser marcadamente diferente entre linajes de eritrocitos en distintas etapas de desarrollo, este gen es de particular interés para el estudio de la hematopoyesis.

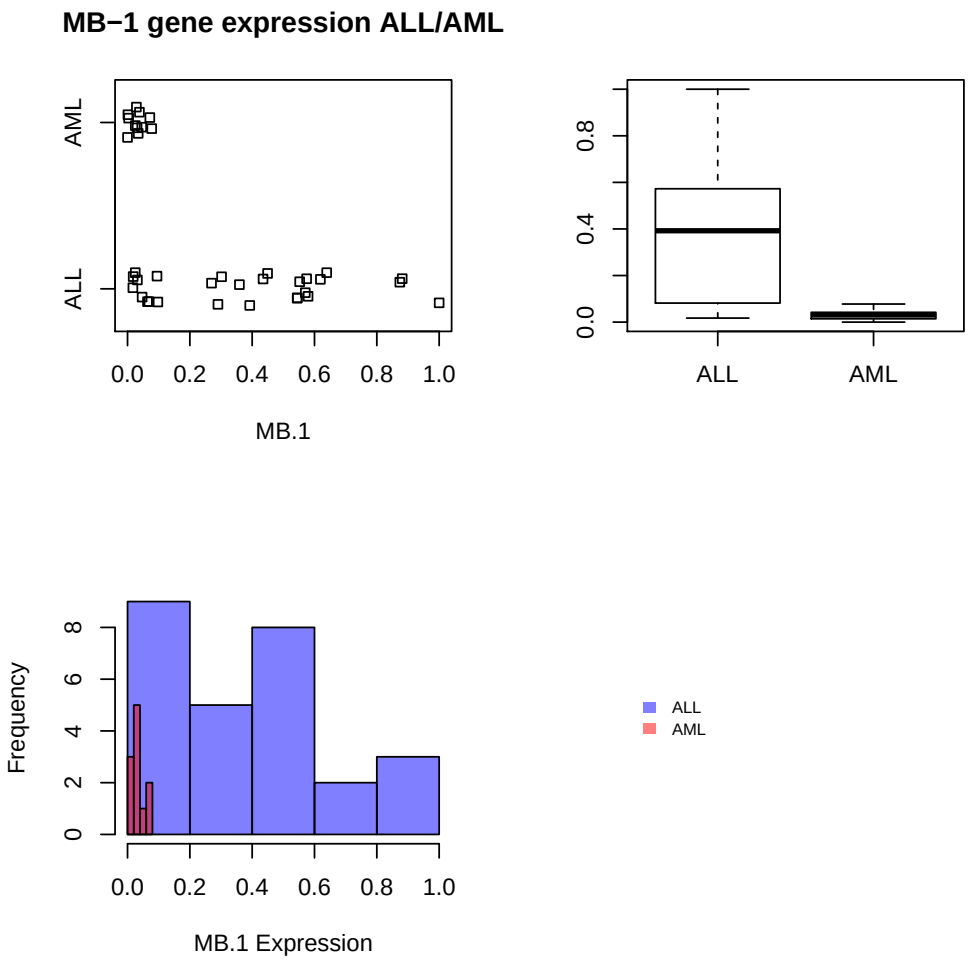
Su información en NCBI: `http://www.ncbi.nlm.nih.gov/gene/3205`



Podemos observar claramente como el gen Cyclin D3 se encuentra muy infraexpresado en los casos de leucemia aguda mieloide y no tanto así en los casos de linfoide. Por lo tanto, no sorprende que sirva de ayuda en la labor de predicción de clases.

El gen Cyclin D3 codifica una proteína que pertenece a una conservada familia caracterizada por una marcada periodicidad en la abundancia de proteínas a lo lardo del ciclo celular (de ahí su nombre). La proteína forma, como subunidad reguladora, un complejo con CDK4 o CDK6, cuya actividad es requerida para la transición entre las fases G1/S del ciclo celular. Además, esta proteína interacciona y está involucrada en la fosforilación de la proteína Rb de supresión tumoral.

Su información en NCBI: `http://www.ncbi.nlm.nih.gov/gene/896`



El gen Mb1 se puede apreciar muy diferencialmente expresado para los dos tipos de leucemia aguda de nuestras muestras, siendo su nivel de espresión casi nulo en los casos de leucemia aguda mieloide.

Este gen codifica la subunidad Ig-alfa de un receptor del antígeno de linfocitos B (*BCR, B-cell antigen receptor*), y se expresa exclusivamente en una etapa muy temprana de las células-B de la médula ósea.

En la leucemia linfoide aguda se producen cantidades excesivas de linfocitos inmaduros (linfoblastos). Estos linfocitos inmaduros invaden la sangre, la médula ósea y los tejidos linfáticos, haciendo que se inflamen. Las céululas cancerosas se multiplican rápidamente y desplazan a las células normales de la médula ósea.

La leucemia linfoblástica aguda de precursores B es un tipo de leucemia linfoide aguda que afecta en particular a los precursores de los linfocitos B que están localizados en la médula ósea. Constituyen cerca del 85 % de los casos de leucemias linfoblásticas agudas.

Por todo ello, no es de extrañar que la expresión del gen MB-1 sea mucho más elvada en los casos de leucemia aguda linfoide que en los de mieloide.

Predicción de clases

Conjunto de datos de test

Para comprobar si nuestro modelo es capaz de realizar una buena predicción del tipo de leucemia aguda dada una muestra desconocida, utilizaremos los datos de test del conjunto de datos de Golub *et al.*. Se trata de 34 muestras de pacientes con leucemia aguda, 20 de ALL y 14 de AML.

En primer lugar, cargamos los datos y comprobamos que se corresponden con lo esperado. Seguidamente normalizamos la distribución de los datos aplicando la misma fórmula que en los datos de entrenamientoo.

```
[1] "data.frame"
```

```
[1]   34 7129
```

```
ALL AML
 20  14
```

Clasifiación mediante regresión logística

A continuación construimos un modelo de regresión logística con la lista de 39 genes seleccionados, lo que llamamos nuestro "predictor de clases", y realizamos una predicción de las muestras de test, para comprobar su precisión.

Obtenemos la siguiente matriz de confusión, en la que podemos observar que no hay falsos positivos ni falsos negativos; por lo tanto, tenemos una precisión del 100 %:

```
          Reference
Prediction  0  1
         0 20  0
         1  0 14
```

```
Accuracy
       1
```

Resultados

Golub *et al.* en su artículo exponen la fuerte correlación de muchos de los genes con la distinción de clases ALL/AML, de tal forma que, según ellos, casi cualquier predictor basado en entre 10 y 200 genes obtiene un 100 % de precisión.

Hemos podido comprobar la veracidad de estas afirmaciones, mediante la repetida ejecución de nuestro modelo, el cuál obtiene soluciones muy diferentes en cada ejecución, pero que, al estar seleccionando genes entre los 250 más correlacionados, casi siempre dan lugar a un predictor de precisión 100 %.

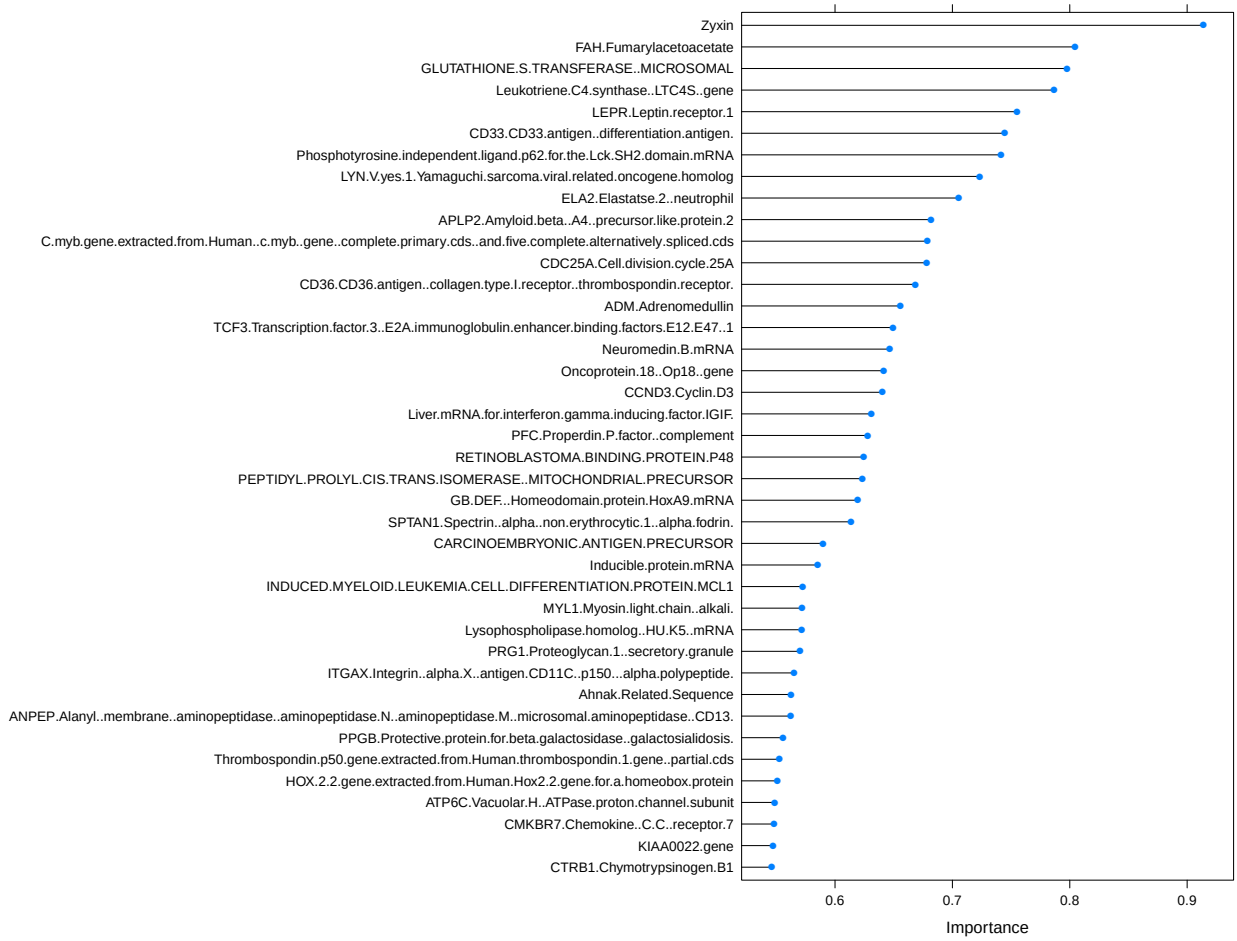
Se ha elegido el resultado mostrado anterioremente entre todos los obtenidos, por haber seleccionados muchos de los genes cuya relación con la distinción de clases de leucemia aguda ya está probada, así como otros que tienen funciones que en cierta medida se pueden ver relacionadas con los procesos tumorales a los que nos referimos en este estudio.

Vamos ahora a llevar a cabo otros métodos de selección de variables, con los cuales podemos comparar los resultados obtenidos por nuestro algoritmo genético. Teniendo en cuenta lo expuesto anteriormente sobre la fuerte correlación de los genes con la distinción de clases, es de esperar que la variabilidad de los resultados sea elevada.

Estimación de la importancia de las variables

Utilizando el paquete *randomForest*, podemos estimar, mediante un modelo que utiliza el método "knn" (K-Nearest Neighbor) controlado por validación cruzada, la importancia de las variables, a modo de ranking, en relación con tipo de leucemia.

A continuación mostramos una gráfica que representa este ranking, mostrando las 40 primeras:



Como podemos observar, entre los 40 genes con más importancia según este método, se encuentran algunos de los seleccionados por nuestro algoritmo genético como pueden ser: LEPR, E2A, Op18, CCND3.

Por el contrario, podríamos remarcar que uno de los genes más comentados por Golub et al. por su importancia, el Homeobox A9, no se encuentra ni entre 50 primeros, y que además, el gen cuya

importancia se estima mayor mediante este método y con una cierta diferencia respecto al resto, el Zyxin, no ha sido seleccionado en cambio por nuestro algoritmo genético.

Comparación con Backwards Feature Selection

Vamos a utilizar ahora un algoritmo RFE (Recursive Feature Selection) de la librería caret para realizar la selección de variables y poder comparar los resultados obtenidos con los del modelo GA.

El algoritmo utiliza una función de control de validación cruzada (10-folds) para el cálculo de la precisión.

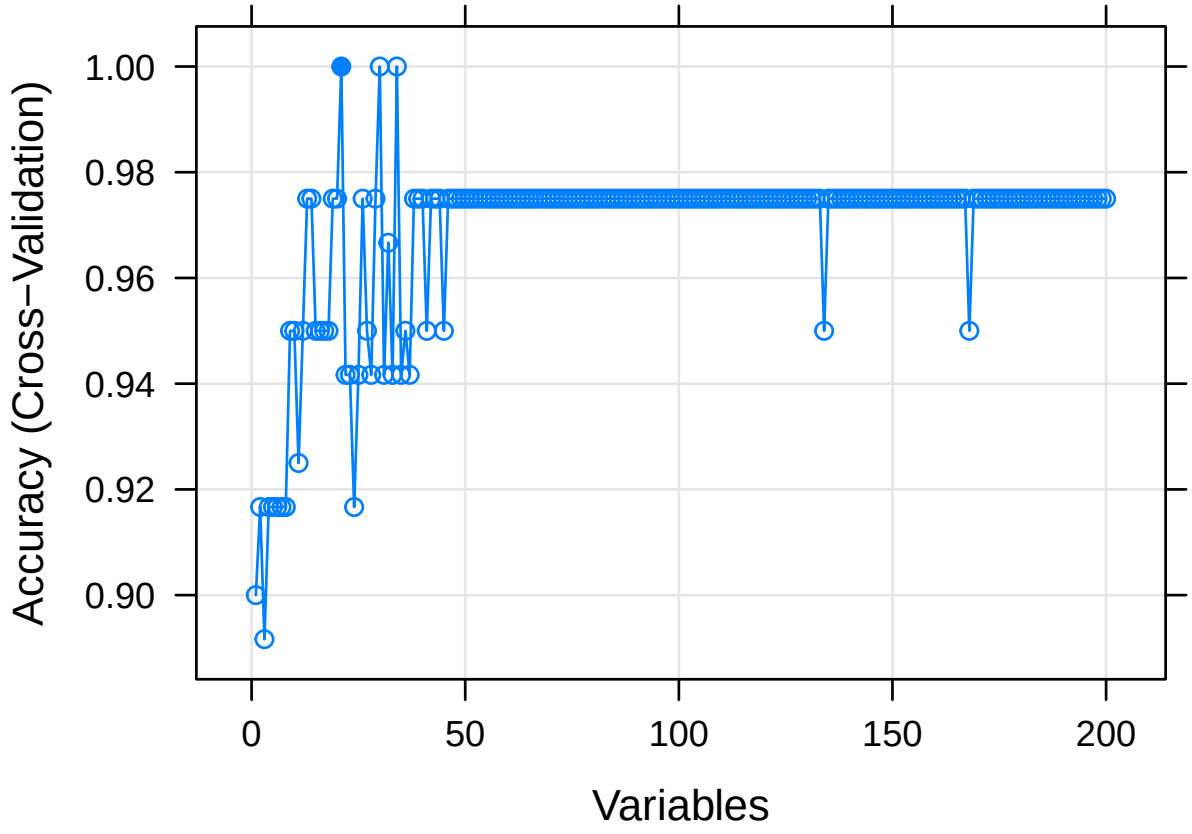
A continuación mostramos la lista de los genes seleccionados:

```
 [1] "Zyxin"
 [2] "FAH.Fumarylacetoacetate"
 [3] "Leukotriene.C4.synthase..LTC4S..gene"
 [4] "LEPR.Leptin.receptor.1"
 [5] "CD33.CD33.antigen..differentiation.antigen."
 [6] "GLUTATHIONE.S.TRANSFERASE..MICROSOMAL"
 [7] "APLP2.Amyloid.beta..A4..precursor.like.protein.2"
 [8] "LYN.V.yes.1.Yamaguchi.sarcoma.viral.related.oncogene.homolog"
 [9] "TCF3.Transcription.factor.3..E2A.immunoglobulin.enhancer.binding.factors.E12.E47..1"
[10] "CDC25A.Cell.division.cycle.25A"
[11] "C.myb.gene.extracted.from.Human..c.myb..gene..complete.primary.cds..and.five.complete.al
[12] "CD36.CD36.antigen..collagen.type.I.receptor..thrombospondin.receptor."
[13] "PFC.Properdin.P.factor..complement"
[14] "Phosphotyrosine.independent.ligand.p62.for.the.Lck.SH2.domain.mRNA"
[15] "Neuromedin.B.mRNA"
[16] "ADM.Adrenomedullin"
[17] "CCND3.Cyclin.D3"
[18] "SPTAN1.Spectrin..alpha..non.erythrocytic.1..alpha.fodrin."
[19] "Liver.mRNA.for.interferon.gamma.inducing.factor.IGIF."
[20] "Oncoprotein.18..Op18..gene"
[21] "RETINOBLASTOMA.BINDING.PROTEIN.P48"
```

Como podemos observar, se han seleccionado 21 genes, entre los cuales podemos encontrar, de nuevo, algunos que coindicen con los seleccionados por el algoritmo genético como LEPR, E2A, CCND3, y Op18. Además nos encontramos también con el gen Zyxin el cual no se encuentra dentro de nuestra lista de genes del predictor.

Los resultados de este método parecen ser bastante acordes con la estimación de importancia de las variables realizada anteriormente con "Random Forest".

A continuación podemos ver la gráfica de la evolución de la precisión en este modelo, con respecto al número de variables seleccionadas, de forma que el valor máximo de precisión se alcanza con 21 variables, como ya se ha mencionado anteriormente:



Predicción de clases

Si construimos un modelo de regresión logística utilizando estos 21 genes seleccionados mediante RFE, y realizamos una predicción sobre los datos de test obtenemos la siguiente matriz de confusión y el valor de precisión mostrado a continuación:

```
          Reference
Prediction  0  1
         0 20  0
         1  0 14
```

```
Accuracy
       1
```

Comparación con Secuential Forward Selection

Para realizar la comparación del modelo GA con la selección secuencial hacia delante vamos a implementar una función SFS que funciona de la siguiente forma:

1. La primera variable es seleccionada construyendo un modelo de regresión logística y eligiendo la variable única que mejor predicción realiza.

2. Utilizando esta primera variable, probamos todos los posibles pares seleccionando una más y elegimos el par que mejor predicción realice utilizando la regresión logística de nuevo.

3. Utilizando el par ya seleccionado, formamos el trío de mayor precisión añadiendo una variable más, de la misma forma que en el paso anterior.

4. Este procedimiento continúa hasta que se alcanza la precisión máxima o un número predefinido de variables seleccionadas.

A continuación mostramos el código de la función:

```
> sfsFunction <- function(dataSet){
+
+   genes <- colnames(dataSet)[-251]
+   usedGenes <- vector()
+   best.acc <- 0
+   total.acc <- 0
+
+   for(i in 1:length(genes)){
+     forml <- as.formula(paste("Class~", genes[i], sep=""))
+     regLog <- glm(forml, data=dataSet, family=binomial("logit"))
+     pred <- predict(regLog, dataSet, type="response")
+     pred.th <- pred
+     pred.th[pred.th<0.5]<-0
+     pred.th[pred.th>=0.5]<-1
+     confMatrix <- confusionMatrix(pred.th, dataSet$Class)
+     accuracy <- confMatrix$overall[1]
+     if(accuracy > best.acc){
+       best.acc <- accuracy
+       best.gene <- genes[i]
+     }
+   }
+   usedGenes <- c(usedGenes, best.gene)
+   genes <- genes[-which(genes==best.gene)]
+
+   while(length(genes)>0) {
+     best.acc <- 0
+     used <- paste(usedGenes, collapse="+")
+     for(i in 1:length(genes)){
+       forml <- as.formula(paste("Class~", paste(used, genes[i], sep="+"), sep=""))
+       regLog <- glm(forml, data=dataSet, family=binomial("logit"))
+       pred <- predict(regLog, dataSet, type="response")
+       pred.th <- pred
+       pred.th[pred.th<0.5]<-0
+       pred.th[pred.th>=0.5]<-1
+       confMatrix <- confusionMatrix(pred.th, dataSet$Class)
+       accuracy <- confMatrix$overall[1]
+       if(accuracy > best.acc){
+         best.acc <- accuracy
+         best.gene <- genes[i]
+       }
+     }
+     usedGenes <- c(usedGenes, best.gene)
```

```
+     genes <- genes[-which(genes==best.gene)]
+     if(best.acc > total.acc){
+       total.acc <- best.acc
+       best.subset <- usedGenes
+     }else{
+       break
+     }
+   }
+   res.list <- list(total.acc = total.acc, best.subset=best.subset)
+   return(res.list)
+ }
```

Utilizando esta función, la ejecutamos con los 250 genes y obtenemos la siguiente selección de variables:

```
$total.acc
Accuracy
       1

$best.subset
[1] "Zyxin"
[2] "Leukotriene.C4.synthase..LTC4S..gene"
```

Podemos ver que el número de genes seleccionados por la función SFS es muy bajo, tan solo 2 genes, ya que alcanza muy pronto una precisión = 1 utilizando el, ya nombrado varias veces, gen *Zyxin* y otro más.

Predicción de clases

Intentamos ahora realizar una predicción tan sólo utilizando estos 2 genes seleccionados, sobre las muestras del conjunto de datos de test. Lo haremos como anteriormente con un modelo de regresión logística, con el cual obtenemos la siguiente matriz de precisión y el consecuente valor de precisión:

```
          Reference
Prediction  0  1
         0 18  1
         1  2 13


 Accuracy
0.9117647
```

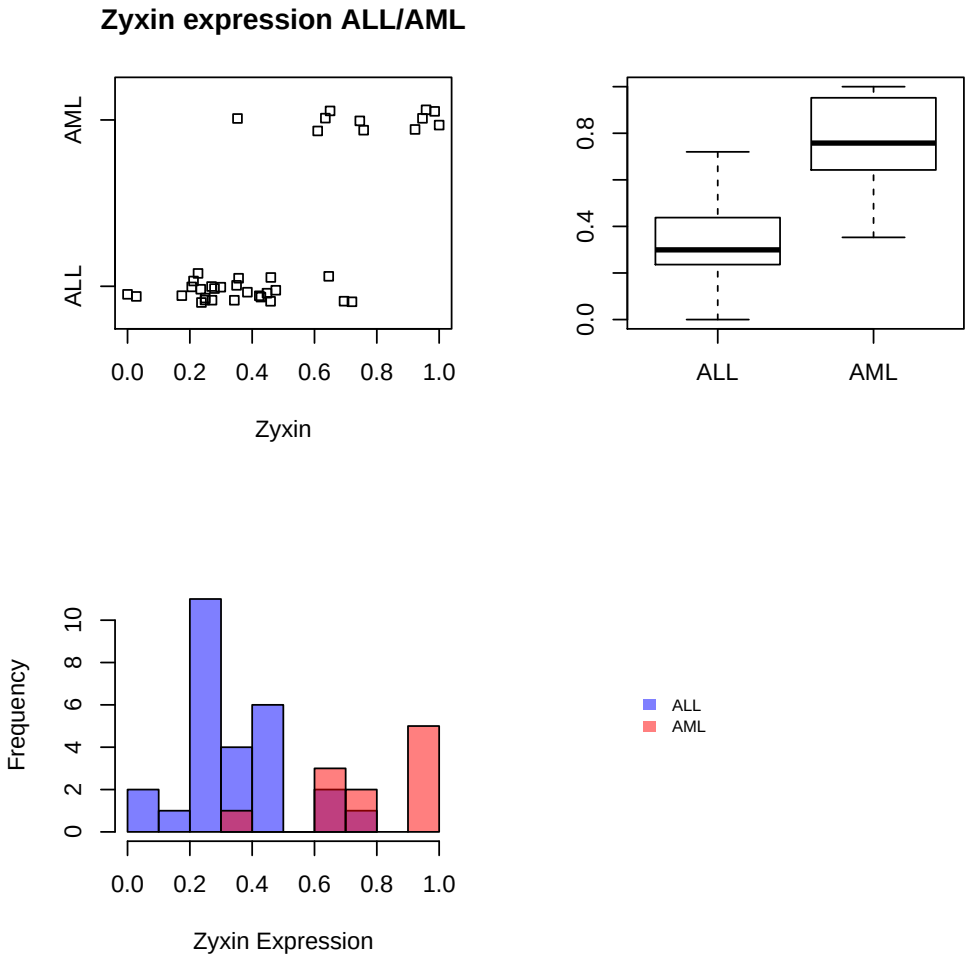
Conclusión

Hemos podido comprobar que los resultados del algoritmo genético son muy variables debido a la fuerte correlación de los genes con la distinción de clases ALL/AML. Al modificar los parámetros del algoritmo podemos variar notablemente los resultados, de tal forma que si elevamos demasiado el número de iteraciones obtenemos normalmente un predictor con muy pocos genes.

Lo que sí es bastante evidente es que podemos conseguir un buen predictor para la distinción de tipos de leucemia aguda, con el que alcanzar muy buenos valores de precisión en la predicción para los datos de test, el problema más complejo es decidir cual sería la mejor opción de entre todas la posibles, y además eficaces, obtenidas.

Una de las cuestiones llamativas es la ausencia del gen *Zyxin* en el predictor creado por nuestro modelo, el cual selecciona 39 genes; cuando parece bastante evidente la alta correlación de este gen con las clases ALL/AML según todos los métodos utilizados posteriormente para la comparación de resultados.

A modo de comprobación vamos a observar la expresión diferencial de este gen en concreto:



Como podemos observar existe también una muy fuerte correlación entre este gen y la clase. Esto podría explicar la obtención de un valor de precisión superior a 0.9 utilizando un predictor que tan sólo contiene este gen y otro más, como es el obtenido mediante la función SFS implementada.

Como última observación, lo que parece bastante evidente es la necesidad de un conjunto de datos bastante mayor, con un número mucho más elevado de muestras sobre todo para el conjunto de test, de tal forma que facilite la selección del mejor predictor de entre todos los posibles, lo cual parece una tarea dificil en este caso.