Clasificación no lineal para datos de conteo: aplicación a datos de expresión de secuenciación de RNA

Diana Carolina Rodríguez Castro

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Estadística.
Bogotá, Colombia
2017

Clasificación no lineal para datos de conteo: aplicación a datos de expresión de secuenciación de RNA

Diana Carolina Rodríguez Castro

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:
Magister en Estadística

Director(a):
Doctora, Biologa Liliana López Kleine

Línea de Investigación:
Estadística genómica
Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Estadística
Bogotá, Colombia
2017

A mi familia

Lo importante en la ciencia no es tanto obtener nuevos datos, sino descubrir nuevas formas de pensar sobre ellos

William Lawrence Bragg

Agradecimientos

Agradezco a mi mis padres, hermanos, y a la profesora Liliana que supervisó el desarrollo de la presente tesis y a la Universidad Nacional por brindarme estos años de formación.

Resumen

Uno de los principales objetivos en el estudio de datos genómicos es la clasificación de genes basados en su coexpresión. Aquí se propone un kernel (medida de similitud) basado en la distancia Poisson y otra basada en información mútua para clasificar genes usando el método no lineal máquina de soporte vectorial en datos de conteo RNAseq, sus resultados se comparan con el método lineal discriminante Poisson.
Palabras clave: (Discriminante Poisson, distancia Poisson, kernel, máquina de soporte vectorial,Información Mutua, RNAseq).

Abstract

One of the main objectives in the study of genomic data is the classification of genes based on their coexpression. Here we propose a kernel (measure of similarity) based on Poisson distance and another based on mutual information to classify genes using the non-linear method vector support machine in RNAseq count data, their results are compared with the Poisson discriminant linear method **Keywords: (Poisson discriminant, Poisson distance, kernel, support vector machine, mutual information, RNAseq)**

Contenido

Agradecimientos vii

Resumen ix

1. Introducción 1

2. Capítulo 2. Antecedentes 4

2.1. Utilidad de la estadística genómica . . . 4
2.1.1. Datos de expresión génica . . . 5
2.1.2. Notación y caractéristicas . . . 6
2.2. Objetivos en el tratamiento estadístico de Datos de RNAseq . . . 7
2.2.1. Predicción funcional de genes . . . 8
2.2.2. Anotación funcional de genes . . . 9
2.3. Antecedentes en el tratamiento estadístico de los Datos de RNAseq . . . 10
2.3.1. Distribuciones supuestas para Datos RNA-seq . . . 10
2.3.2. Técnicas de Normalización de Datos RNA-seq . . . 12
2.3.3. Regresión Logística para predicción funcional . . . 12
2.3.4. Análisis Discriminante Lineal para predicción funcional . . . 13
2.4. Métodos sin normalización de datos RNAseq para predicción funcional . . . 14
2.4.1. Discriminante Poisson . . . 14
2.4.2. Reconocimiento de Patrones y métodos kernel . . . 15
2.4.3. Tipos de Kernel . . . 17
2.4.4. El Truco Kernel . . . 17
2.4.5. Máquinas de soporte vectorial. SVM . . . 18
2.5. Medidas de Similitud y distancias . . . 24
2.5.1. Razón de verosimilitud . . . 27

3. Capítulo 3. Materiales Y Métodos 29

3.1. Matrices simuladas . . . 29
3.2. Datos reales . . . 32
3.3. Métodos de Clasificación . . . 34
3.3.1. Discriminante Lineal Poisson . . . 34
3.3.2. Máquinas de soporte vectorial . . . 35

4. Capítulo 4. Resultados **39**
4.0.1. Resultados en Datos Simulados 43
4.0.2. Resultados en Datos Reales 48

5. Capítulo 5. Discusión **52**

6. Conclusiones y recomendaciones **54**
6.1. Conclusiones 54
6.2. Recomendaciones 55

A. Anexo: Gráficas gen vs gen **56**

B. Anexo: Código en R **59**

Bibliografía **64**

Lista de símbolos

Abreviaturas

Abreviatura	Término
SVM	Support vector machines (Máquinas de soporte vectorial)
KKT	Karush-Kuhn-Tucker
IM	Información Mutua

1. Introducción

La estadística genómica intenta comprender el funcionamiento de los sistemas biológicos mediante la aplicación de técnicas y modelos estadísticos en el tratamiento de datos genómicos, uno de los datos genómicos más informativos son los datos que miden la actividad de los genes (expresión génica en diversas condiciones). La técnica antes utilizada para medir el nivel de expresión de los genes conocida como microarreglos genera matrices de datos continuos, de los cuales se supone normalidad, y se usan métodos estadísticos adaptados a este supuesto.

La técnica de microarreglos tiene algunos problemas experimentales que se ven reflejados en datos poco precisos [3] y en el no descubrimiento de nuevas transcripciones. De manera que nuevas técnicas de secuenciación han sido desarrolladas obteniendo una mejor cuantificación de la expresión y coexpresión génica en diferentes condiciones biológicas.

Las nuevas técnicas de medición de la expresión génica permiten el secuenciamiento masivo de DNA copia obtenido a partir de RNA mensajero (RNAseq) y hace posible obtener perfiles de expresión. Se denomina secuenciación directa del RNA mensajero o RNAseq, sin embargo, esta técnica cambia la naturaleza de los datos objeto de análisis, ya que produce datos de conteo, de manera que no es correcto usar las técnicas convencionales para clasificación aplicadas a datos de microarreglos (de naturaleza continua y distribución normal).

La clasificación en estadística genómica se ha hecho con técnicas que suponen normalidad en los datos, como análisis discriminante lineal de Fisher [23] y PAM, siendo este un procedimiento de clasificación de genes basado en el método del centroide [48].

Además, se han realizado análisis basados en medidas de similitud lineales, como el coeficiente de correlación de Pearson Mansson y algunas modificaciones [45, 24] o creados para datos continuos,como la distancia euclidiana [2] o el coeficiente de correlación intraclase [46] y en los trabajos que se realiza clasificación con máquina de soporte vectorial se usa el kernel gaussiano [16, 12, 2, 51] . Sin embargo, estas medidas de similitud pueden no ser adecuadas para los datos RNAseq por ser de naturaleza de conteo y porque los genes presentan relaciones no lineales, como lo indican algunos estudios como el de López-Kleine, et. al. (2012) [27]

En general, para la clasificación de genes usando datos de RNAseq se realizan correcciones de sesgo y normalizaciones para finalmente usar las técnicas convencionales que suponen

normalidad en los datos. Sin embargo, estas correcciones introducen ruido y nuevos sesgos a los datos. Una de las transformaciones más conocidas en la normalización de datos RNAseq es llamada RPKM que significa lecturas por kilobase de gen por millon de lecturas encontradas, en donde se divide el total de lecturas de la muestra por la longitud del gen y por la profundidad de secuenciación (total de conteos en la muestra), para eliminar los sesgos, y posteriormente se normaliza mediante una transformación logarítmica [10, Cap. 10]. Sin embargo, existen varios enfoques para la normalización de estos datos, y se ha comprobado que su elección influye en los resultados del modelo [10, 14, 6]. De manera, que parece más adecuado usar técnicas que permitan usar los datos de conteo sin modificar.

Otras técnicas usadas para el análisis de RNAseq son EdgeR, que usa modelo Poisson sobredisperso para explicar tanto la variabilidad biológica como técnica, un método Bayesiano para modelar el grado de sobredispersión a través de transcripciones, y finalmente se evalúa la expresión diferencial para cada gen utilizando una prueba exacta análoga a la prueba de Fisher, pero adaptada para datos sobredispersos [42] y DESeq que propone un test para la expresión diferencial mediante el supuesto de la distribución binomial negativa.

Sin embargo, éstas técnicas no sirven para predicción funcional, solamente para detectar genes diferencialmente expresados comparando dos condiciones biológicas. En cuanto a la clasificación de datos de conteo, se han planteado técnicas estadísticas, como el Discriminante Lineal Poisson que basado en la regla de Bayes, encuentra una regla para clasificar suponiendo que los datos provienen de una distribución Poisson. Este método tiene tres (3) aspectos a mejorar, uno, el supuesto de independencia, otro la linealidad del clasificador y finalmente que no se comporta bien en datos con sobredispersión [9]

Por otro lado, existen métodos de clasificación no lineales, flexibles y con pocos supuestos, que se enmarcan dentro de los métodos de aprendizaje de máquina. Dentro de ellos, las máquinas de soporte vectorial que dependen básicamente de la función kernel que se elije para representar los datos, siendo esta en esencia una matríz de similitud. En los trabajos de clasificación de RNAseq hechos hasta el momento, se ha usado el kernel Gaussiano [16, 12, 2, 51, 9].

Existen entonces aportes en cuanto al análisis de datos de conteo y en cuanto a los clasificadores no lineales, pero de manera independiente. De manera que aquí se intenta mejorar la clasificación funcional de genes mediante la combinación de un método no lineal y una medida de similitud adaptada a datos de conteo crudos, sin el realizamiento previo de una normalización ya que como se ha probado infiere en los resultados obtenidos [10].

En efecto, se usa como clasificador no lineal las máquinas de soporte vectorial junto con dos kernel propuestos que parecen más adecuados para datos de conteo que el kernel Gaussiano.

Uno de los kernel propuestos está basado en la Información Mutua que es una medida ampliamente aceptada en datos de expresión génica por su capacidad de detección de relaciones no lineales y el segundo kernel está basado en la distancia Poisson que es una distancia análoga a la distancia euclidiana pero para datos provenientes de distribuciones Poisson. Estos resultados a su vez se comparan con el método lineal Discriminante Poisson.

Para la comparación de métodos se realizan simulaciones de datos de conteo provenientes de distribuciones binomiales negativas generando grupos de genes con relaciones lineales y no lineales, midiendo su poder predictivo mediante el porcentaje de genes mal clasificados en los datos de prueba obteniendo mejores resultados con el kernel basado en la distribución Poisson.

A su vez, se prueba con datos reales provenientes de muestras de yuca para dos tipos de clasificaciones, Pfam que corresponden a genes de distintas familias de proteínas, y PANTHER (Protein Analysis Through Evolutionary Relationships), que realiza una clasificación de genes mediante información completa sobre la evolución y la función de genes. Donde nuevamente las mejores clasificaciones, comparadas por el porcentaje de genes mal clasificados en los datos de prueba se obtienen para la máquina de soporte vectorial con el kernel basado en la distancia Poisson.

2. Capítulo 2. Antecedentes

2.1. Utilidad de la estadística genómica

La estadística genómica tiene como fin la comprensión de los sistemas biológicos mediante la aplicación de técnicas y modelos en el tratamiento de datos genómicos. Estos datos pueden ser datos obtenidos directamente sobre la secuencia u obtenidos a través de técnicas de laboratorio. En esta última categoría se encuentran los datos que miden la expresión de los genes (cantidad de RNA mensajero) bajo alguna condición específica y se han denominado de manera general experimentos de transcipción o de expresión génica.

La expresión génica es el proceso mediante el cual se transforma la información codificada en el ADN y ARN en proteínas que son las encargadas de realizar diversas tareas biológicas. Diferentes genes codifican proteínas responsables de diferentes tareas, y por tanto se activan dependiendo del proceso biológico que lleve a cabo en el organismo, no todos los genes se expresan en el mismo momento.

Cada gen es poseedor de una información génica necesaria para desempeñar un trabajo biológico, que se manifiesta mediante su producto funcional, las proteínas, que no son más que una cadena de aminoácidos.

El proceso de expresión génica, consta de dos fases, una de trascripción, donde la secuencia de ADN de un gen se copia para obtener una molécula de ARN, que se convierte en ARN mensajero y una de traducción, donde la información se traduce en proteína. Los genes contenidos en el ADN y que contienen información génica cifrada son expresados de acuerdo a estímulos, mediante la transcripción, donde se copia en RNA mensajero la información del gen. Cuando se lleva a cabo un proceso celular específico se produce RNA mensajero en diferentes cantidades de los genes que se relacionan a dicho proceso. De manera que la cantidad de moléculas de RNA mensajero producidas, se relaciona directamente con el nivel de participación de los genes en dichos procesos celulares.

Como dice García (2016) en su tesis, el nivel de expresiín de un gen es la cantidad de ARN que se transcribe, determinando la cantidad de proteínas que serán producidas por el gen. En todas las células, la expresión de cada gen individual está regulada, es decir, no se producen todas las proteínas posibles continuamente, sino que la célula ajusta el ritmo de la transcrip-

ción y traducción de cada gen independientemente, de acuerdo con sus necesidades puntuales.

Como la expresión de un gen está directamente relacionada con la cantidad de proteína que se produce, estudiando los niveles de expresión génica medimos cuánto está activado un gen en particular durante el proceso de la transcripción del ADN en ARN y la traducción de éste en proteínas [13]

Para el entendimiento de estos procesos se han generado técnicas que aportan información sobre cómo se activan y desactivan los genes y sobre otros aspectos de su expresión que se traduce en el estudio de transcriptomas en una variedad de condiciones y estados de enfermedad.

2.1.1. Datos de expresión génica

Existen dos clases de datos que miden la expresión génica que provienen de dos técnicas diferentes para cuantificar la expresión de los genes:

- **Datos de Microarreglos** Genera datos de naturaleza continua. Utiliza hibridación en sondas
- **Datos de RNAseq** Genera datos de conteo. Mide la cantidad de RNA mensajero en una muestra.

Se han desarrollado técnicas para el tratamiento de los datos de expresión génica generados por la técnica de microarreglos, pero dicha técnica tiene algunos problemas como la necesidad de algún conocimiento previo de la muestra en estudio, impidiendo el descubrimiento de nuevas transcripciones, además de tener otros problemas experimentales que generan datos poco precisos [3]. De manera que nuevas técnicas de secuenciación han sido desarrolladas obteniendo una mejor cuantificación del RNA mensajero y han permitido obtener información a gran escala acerca de la expresión y coexpresión génica en diferentes muestras. Estas técnicas se denominan secuenciación directa del RNA mensajero o RNAseq [26]

El RNAseq es una herramienta transcriptómica que permite el secuenciamiento masivo del DNA copia obtenido a partir del RNA mensajero y hace posible obtener perfiles de expresión génica determinando los genes que se activan bajo determinadas circunstancias y los que actúan simultáneamente o mediante algún tipo de relación. Esta tecnología permite la secuenciación de RNA, mediante el siguiente procedimiento [34]:

a.) Se aisla RNA y se fragmenta con una longitud promedio de 200 nucleotidos.
b.) El RNA fragmentado es convertido en DNAc (DNA complementario.)
c.) El DNAc es secuenciado.

El resultado son fragmentos, que se ensamban dentro del genoma, el número de fragmentos que corresponden a determinado gen nos indica su nivel expresión, por esta razón se generan los conteos. Esta técnica tiene ventajas comparada con la técnica de microarreglos, porque no sufre de algunos problemas como hibridación cruzada y por lo tanto sus datos tienen menos ruido.

Sin embargo, esta técnica cambia la manera de análisis de datos de transcripción génica en comparación con los microarreglos, ya que las técnicas clásicas no pueden ser usadas sin modificaciones, por tratarse de datos de conteo con más características (genes) (de 10 a miles) que observaciones (condiciones en las que se realiza la medición de RNA-mensajero) (de 10 a cientos).

2.1.2. Notación y caractéristicas

Los datos de resultados de microarreglos se escriben en una matriz x_{nxp}, siendo n el número de muestras y p el número de genes tomados como características de los individuos en los métodos estadísticos aplicados, donde x_{ij} es una variable continua producto de una medida de intensidad que refleja el nivel de expresión de cada gen (j), donde usualmente $n < p$.

Los resultados obtenidos en los experimentos de RNA-seq son escritos bajo la misma forma matricial, la diferencia fundamental radica en el tipo de variable resultado, en este caso x_{ij} es una variable de conteo positiva y entera, que nos indica el número de veces que se encontró en la muestra RNAm de determinado gen (j).

El estudio de los datos RNA-seq tiene varios objetivos dentro de los que se rescatan dos, la predicc0́n funcional que se obtiene agrupando los genes con características similares y la comparación de individuos de grupos distintos para el reconocimiento de expresiones diferenciales (en genes) entre observaciones de tratamiento, de control y de posterior clasificación de nuevos individuos.

Usualmente, la mayoría de los genes tienen pocos conteos, por lo que su distribución es sesgada a la derecha, y las medidas de similitud o disimilitud usadas comúnmente en métodos de clasificación no funcionan bien para estos datos. [10]

2.2. Objetivos en el tratamiento estadístico de Datos de RNAseq

La genómica trata de entender las correlaciones, estructuras y los patrones en los datos génomicos, hay una necesidad por volver información los datos disponibles con el objetivo de comprender mejor los sistemas vivos. Usualmente, se tienen tres clases de preguntas en el análisis de datos genómicos, particularmente de expresión génica:

- **Análisis de expresión diferencial**. Comprende la identificación de genes que están diferencialmente expresados entre clases (activación/represión), es decir, cuál de ellos está por encima o por debajo de la media de las muestras que pertenecen a ciertas clases. El diseño experimental determinará que métodos estadísticos son adecuados: comparación de dos clases, más de dos grupos, series temporales, correlación, análisis de supervivencia [13]. Este tipo de análisis reconoce genes asociados a una fución biológica midiendo la expresión entre muestras con diferentes condiciones, por ejemplo para reconocer los genes involucrados en cierta enfermedad se tienen muestras de pacientes con y sin la enfermedad, y se evalúan los genes que están sobreexpresados en las muestras con la enfermedad, de manera que estos genes están ligados a la función biológica causal de dicha enfermedad.

- **Análisis de clúster** Agrupa las muestras o unidades biológicas que presentan un patrón de expresión común, de forma no supervisada, es decir sin conocimiento previo de la clase.

  En estos estudios se realizan numerosas normalizaciones de los datos con el fin de evitar sesgos en las muestras, la primera correccción se hace por longitud de secuenciación, ya que todas las muestras no tienen el mismo número de lecturas y por tanto en análisis de expresión diferencial las muestras con mayor número de lecturas tienen genes que pueden parecer sobreexpresados aunque no lo estén.

  También se realiza corrección por longitud de gen, ya que hay genes más cortos, con menor número de nucleótidos que otros lo que es importante para clasificar correctamente los niveles de expresión génica dentro de la muestra para explicar el hecho de que los genes más largos acumulan más lecturas. Sin embargo esta corrección no es necesaria cuando se comparan los cambios en la expresión génica dentro del mismo gen a través de las muestras [7].

Se hacen dos tipos de clasificaciones (supervisadas y no supervisadas) usando datos de expresión génica, sobre las muestras y sobre los genes. Sobre las muestras se realizan clasificaciones cuando se conocen datos de expresión en el trascriptoma de pacientes bajo dos condiciones (enferno-no enfermo) y se busca encontrar un método de clasificación que me permita identificar los genes que se sobreexpresan o se subexpresan en determinada condición, de manera que nuevos individuos de los cuales no se conoce su condición, sean clasificados en alguna de las ya preestablecidas.

- **Predicción funcional**. Por otro lado está la predicción funcional de genes que es el objetivo seguido aquí y comprende la identificación de la tarea realizada por un gen, en donde se clasifican dependiendo de sus interacciones bajo determinados procesos biológicos, que para muchos genes presentan patrones complejos, para lo cual se pueden usar métodos con medidas de similitud que capturen la no linealidad de los patrones de coexpresión, en el caso de aprendizaje supervisado y no supervisado.

  Dentro de los estudios que se realizan se encuentran, los análisis de aprendizaje supervisado cuando se tiene información sobre la clase a las que corresponde el objeto a clasificar, se construye un predictor supervisado de clases a partir del entrenamiento de un grupo de clases, para poder encontrar la clase de un nuevo individuo.

  La pregunta que deseamos resolver es, dado un grupo de genes, de los cuales conocemos su nivel de expresión, medido por la técnica RNAseq (datos de conteo) en un grupo de muestras y la clase a la cual pertenece, determinar la clase de nuevos genes de los cuales sólo conocemos su nivel de expresión.

2.2.1. Predicción funcional de genes

Uno de los principales objetivos en el estudio de datos genómicos es la clasificación de genes, según su coexpresión. La agrupación de genes con base en su perfil de expresión se denomina predicción funcional en estadística genómica y está basada en la hipótesis biológica de que los genes que tienen una actividad coordinada, participan en funciones similares. De esta manera, al agrupar genes cuyos productos tienen funciones conocidas con genes desconocidos, se puede inferir la función. Lo anterior es de suma importancia, ya que en la mayoría de los genomas secuenciados de procariotas, aproximadamente la mitad de los productos génicos tienen una función desconocida y esta cantidad es mucho mayor en plantas y animales. [21]

La técnica de RNAseq posibilita el conocimiento de nuevos transcritos, de los cuales se desconoce la categoría a la cual pertenece, la predicción funcional de genes busca asignar los nuevos genes encontrados a una de las categorías conocidas en función de su expresión bajo

determinadas condiciones.

El último paso estándar en un estudio transcriptómico es a menudo esta caracterización de las funciones moleculares o vías en las que están implicados los genes expresados. Los dos enfoques principales de la caracterización funcional que se desarrollaron son: (a) comparar una lista de genes expresados diferencialmente contra el resto del genoma para las funciones sobre representadas, y (b) el análisis de enriquecimiento de conjuntos de genes [7]. Que se hace mediante métodos de clasificación o de agrupación teniendo en cuenta las clasificaciones conocidas o la similitud en los perfiles de expresión.

Los sesgos como la longitud del gen complican la aplicación de los métodos para los datos de recuento y, por tanto, han propuesto herramientas específicas para datos RNAseq[7]. Siendo uno de los objetivos la no insersión de nuevos sesgos o de ruido mediante las transformaciones aplicadas.

El análisis funcional requiere la disponibilidad de suficientes datos funcionales de anotación para el transcriptoma en estudio. Recursos como la ontología de genes, Pfam, PANTHER contienen datos de anotación para la mayoría de las especies modelo [7]. Que corresponden a grupos dentro de los cuales se pueden asignar los nuevos transcritos.

En la predicción funcional, como dice Viera (2011), las anotaciones de productos génicos ya caracterizados, referidas a localización, función y procesos en los que están involucrados, pueden transferirse a partir de la similitud de secuencias a nuevas proteínas. Otros métodos se basan en el comportamiento de los genes en estudios de expresión génica, siendo este el que interesa en la presente tesis o en las redes de interacción proteína-proteína. Se presume entonces que los genes no caracterizados tienen funciones biológicas similares a los genes con los cuales se agrupan.

2.2.2. Anotación funcional de genes

Existen una serie de bases de datos con información sobre los genes, en las cuales se registran anotaciones funcionales que son funciones asociadas a genes, la forma de determinar la función de una proteína es anotarla con distintos términos en diferentes vocabularios, ya que puede cumplir distintas funciones dependiendo de su nivel de expresión. Estas anotaciones nos indica el grupo al cual pertenece un gen y son usadas en la predicción funcional. Algunas de las clasificaciones que se tienen son:

- **Gene Ontology (GO)** Son términos descriptivos que permiten asociar a cada producto génico un contexto molecular, celular y de proceso biológico. Contiene anotaciones en tres (3) categorías, funciones moleculares, que son tareas que llevan a cabo los

productos génicos; procesos biológicos, que son grandes procesos llevados a cabo por conjuntos ordenados de funciones moleculares y la localización del componente celular, que se refiere a la ubicación del producto génico [49]

Las onotologías, que facilitan el análisis de datos, por ejemplo, lo que se puede ver en el rol que juega actualmente la GO en el análisis estadístico de los datos de genómica funcional, en la anotación automática de las funciones de los productos génicos sobre la base de la similitud de secuencia, estructural o de perfiles de co-expresión. [49]

- **Pfam** Corresponden a familias de proteínas. Pfam-A es una base de datos manualmente gestionada, donde cada entrada almacena un alineamiento múltiple de secuencias de proteínas y un modelo oculto de Márkov ue cubre buena parte de dominios proteicos y familias comunes. [18]

- **PANTHER(Protein Analysis Through Evolutionary Relationships)** contiene los conjuntos completos de genes de codificación de proteínas para 104 organismos, y realiza una clasificación de genes mediante información completa sobre la evolución y la función de genes.

Esas bases de datos con clasificaciones son importantes para agrupar genes de familias conocidas con genes de familias desconocidas y mediante métodos de clasificación asignar estos últimos a un grupo, ya que se espera que genes que pertenezcan a la misma familia tengan el mismo perfil de expresión.

2.3. Antecedentes en el tratamiento estadístico de los Datos de RNAseq

2.3.1. Distribuciones supuestas para Datos RNA-seq

Algunas de las Distribuciones ampliamente aceptadas para los datos de conteo son la Distribución Poisson y la Distribución Binomial Negativa, por lo cual parece adecuado suponer que los datos RNAseq provienen de estas dos distribuciones cuando los datos no son modificados mediante las técnicas de normalización.

Distribución Poisson

Se trata de un modelo discreto, donde se dice que la variable aleatoria discreta X, cuyos valores posibles son $0, 1, 2...etc$, tienen distribución Poisson con parámetro λ y su función de

probabilidad es:

$$P(X = x) = \frac{e^{\lambda}\lambda^{x}}{x!}, \lambda > 0, \quad x = 0, 1, 2, 3... \tag{2-1}$$

donde:
λ: Promedio de ocurrencias en un intervalo (tiempo, volumen, área...)
x: Número de ocurrencias.

La definición de la función generatriz de Momentos, se define por:

$$\varphi(t) = E[e^{tx}] = \sum_{x=0}^{\infty} e^{tx}\frac{e^{-\lambda}\lambda^{x}}{x!}$$

$$\varphi(t) = e^{\lambda(e^{t}-1)}$$

De donde surge la media y la varianza:

$$\mu = E[x] = \lambda$$

$$\sigma^2 = \lambda$$

Una propiedad importante de la distribución de Poisson, es que la suma de n variables de Poisson independientes, $Ps(\lambda_1) + Ps(\lambda_2) + ... + Ps(\lambda_n)$, es también una variable Poisson siendo el valor de su parámetro λ, la suma de los parámetros de las variables individuales $Ps(\lambda_1, \lambda_2 + ... + \lambda_n)$[30].

Esta distribución puede presentar sobredisersión cuando la varianza es mayor que su valor esperado, es decir la varianza es mayor que la media.

Distribución Binomial Negativa

Es una distribución para datos discretos, donde la variable x se puede ver como el número de experimentos realizados hasta obtener el r-ésimo éxito, y tiene parámetros r (número de éxitos) y p siendo su función de probabilidad

$$f(k) = p(X = k) = \binom{k-1}{r-1} p^{r}(1-p)^{k-r} \tag{2-2}$$

La media y la varianza asociada a esta distribución es:

$$\mu = r\frac{1}{p}$$

$$\sigma^2 = r\frac{1-p}{p^2}$$

Con esta distribución se pueden tener valores distintos para la media y la varianza, solucionando el problema de sobresispersión que se presenta con la distribución Poisson.

2.3.2. Técnicas de Normalización de Datos RNA-seq

El tratamiento usual que se le da a los datos RNAseq en la predicción funcional de genes, es realizar una corrección de sesgos y normalización de los datos, para posteriormente realizar la clasificación mediante técnicas convencionales basadas en el supuesto de normalización de datos. Lo que induce a la introducción de nuevos sesgos y ruidos al modelo, de manera que se prefiere usar los datos crudos o con las correcciones de sesgo necesarias y sin suponer normalidad.

Los datos de RNA-seq presentan sesgos inter-muestras, el mayor de ellos dado por la diferencia en la profundidad de secuenciación (número de lecturas en el experimento), de manera que un mayor número de conteos puede no representar un alto grado de expresión de los genes y que las diferencias entre los conteos pueden deberse más a la profundidad de secuenciación que al nivel de expresión de los genes.

Para su corrección se han usado diferentes factores de corrección que permiten la comparación de los niveles de expresión entre muestras, y se han generado un gran número de enfoques para su normalización, comprobando que su elección influye en los resultados del modelo [10].

2.3.3. Regresión Logística para predicción funcional

Es un modelo de regresión lineal simple o múltiple pero donde la variable dependiente es binaria o dicotómica, las covariables que intentan explicar esta variable pueden ser cualitativas o cuantitativas. En este modelo, la probabilidad que la observación i pertenezca a la clase 1 o 2 es:

$$\begin{aligned} Pr(y_i = 1|x_i) &= \frac{1}{1+e^{\alpha+\beta^T x_i}} \\ Pr(y_i = 1|x_i) &= \frac{e^{\alpha+\beta^T x_i}}{1+e^{\alpha+\beta^T x_i}} \end{aligned} \tag{2-3}$$

donde α es un escalar no conocido y β es un vector no conocido de longitud p. La correspondiente log-verosimilitud queda de la forma:

$$l(\alpha,\beta) = \sum_{i=1}^{n}\{(y_i-1)\alpha + (y_i-1)\beta^T x_i - log(1+e^{\alpha+\beta^T x_i})\} \tag{2-4}$$

La estimación de los parámetros se hace usando mínimos cuadrados iterativamente reponderados [43]. Y la regresión logística nos lleva a estimar la probabilidad que una observación de prueba pertenezca a una clase en particular es:

$$\hat{Pr}(y^* = 1|x^*) = \frac{1}{1 + e^{\hat{\alpha}} + \hat{\beta}^T x^*} \tag{2-5}$$

Para clasificar se elige un punto de corte t, tal que se clasifica en 1 si $\hat{Pr}(y^* = 1|x^*) > t$ y en la clase 2 en otro clase. Por lo general este punto de corte, se define en t=0.5.

2.3.4. Análisis Discriminante Lineal para predicción funcional

El análisis discriminante lineal es uno de los métodos más usados para clasificación y se motiva por la regla de Bayes y en el supuesto de normalidad de los datos.

Sean x y y variables aleatorias de datos de expresión y su clase respectivamente y $\pi_k = P(Y = k)$ la probabildad a priori de que una observación pertenezca a la clase k,y $\sum_{k=1}^{k} \pi_k = 1$ y $p_k(x) = f(X = x|Y = k)$ es la función de densidad de que una observación pertenezca a la clase k. Usando la regla de Bayes, la probabilidad a posteriori de que una observación pertenezca a la clase k es:

$$P(Y = k|X = x) = \frac{\pi^k p_k(x)}{\sum_{i=1}^{k} \pi_i p_i(x)} \tag{2-6}$$

Se supone que cada observación de la clase k son iid de una distribución gaussiana de dimensión p $N(\mu_k, \Sigma)$, con matríz común de covarianza. De manera que sustituyendo la función de densidad gaussiana en la regla de Bayes obtenemos la regla de decisión:

$$\delta_k(x^*) = x^{*T}\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_{-1}(x - \mu_k + log\pi_k) \tag{2-7}$$

Se asigna a la clase cuyo valor sea mayor. Los valores de los parámetros son desconocidos y se estiman usualmente como:

$$\hat{\pi_k} = \frac{n_k}{n}$$

Existe una extensión llamado análisis discriminante cuadrático, para grupos de covarianza distintas ya que si se emplea la función lineal, las observaciones tienen a ser clasificadas en los grupos que tienen varianzas altas. [11]

2.4. Métodos sin normalización de datos RNAseq para predicción funcional

2.4.1. Discriminante Poisson

Se han hecho ajustes a los modelos tradicionales para el tratamiento de datos RNA-seq, dado que los datos RNA son datos de conteo y no es apropiado usar metodos tradicionales como análisis discriminante de Fisher ya que parte del supuesto de normalidad de los datos. De manera que D. Witten propuso el método análisis discriminante de Poisson, que se explica adelante [9]

Para hallar la regla discriminante, se asume que los datos surgen de el modelo:

$$x_{ij}|y_i = k \sim Poisson(N_{ij}d_{kj}), N_{ij} = s_i g_j \tag{2-8}$$

donde:

- $\sum_{i=1}^{n} s_i = 1$ para no tener problemas de identificabilidad.
- y_i Indica la clase de la observación i, $y_i \epsilon [1...k]$
- $d_{1j}, ..., d_{kj}$ término que permite expresión diferencial entre clases. Si es igual a cero no existe expresión diferencial

Para ajustar el modelo, se usa estimación de máxima verosimilitud para $\hat{N_{ij}} = \frac{x_i x_{.j}}{x_{..}}$ [1]. Por la condición de identificabilidad $\hat{s_i} = \frac{x_{i.}}{x_{..}}$ y $\hat{g_j}) = x_{.j}$.

Finalmente define:

$$d_{kj} = \frac{x_{C_{kj}} + \beta}{\sum_{i \epsilon Ck} \hat{N_{ij}} + \beta} \tag{2-9}$$

Este modelo permite variabilidad por el número de lecturas por muestra (s_i) y en el total de número de lecturas por gen (g_j)

El método de clasificación supone que queremos clasificar unas observaciones de prueba $x^* = (x_1^*...x_1^p)^T$ en base a unos datos de entrenamiento $(x_i, y_i)_{i=1}^n$, siendo y^* la clase no conocida. Este método se basa en la regla de Bayes:

$$P(y^* = k|x^*)\alpha f_k(x^*)\pi_k \tag{2-10}$$

donde f_k es la densidad de una observacón en la clase k y π_k es la probabilidad a priori que una observación pertenezca a la clase k. Se supone distribución Poisson para hallar la regla de clasificación.

De manera que se supone que los datos surgen del modelo presentado previamente $x_j^*|y^* \sim Poisson(s^* g_j d_{kj})$. Donde primero se estiman los s_i para los datos de entrenamiento, después estimamos g_j y d_{kj} en los datos de entrenamiento y finalmente, los s_i^* para los datos de prueba.

Para estimar π_k lo hacen mediante igualdad de probabilidad entre clases o con la probabilidad de pertenencia a caa clase dada por los datos de entrenamiento.

Finalmente D. Witten encuentra que:

$$logP(y^* = \hat{k}|x^*) = log\hat{f}_k(x^*) + log\hat{\pi}_k + c \tag{2-11}$$

$$logP(y^* = \hat{k}|x^*) = \sum_{j=1}^{p} x_j^* log\hat{d_{kj}} - \hat{s^*} \sum_{j=1}^{p} \hat{g_j}\hat{d_{kj}} + log\hat{\pi}_k + c \tag{2-12}$$

Sólo el primer término involucra las medidas x^* de los datos de prueba. Por lo tanto la regla de clasificación es asignar la observación de prueba a la clase cuyo valor sea más alto.

El modelo desarrollado plantedo por D. Witten plantea la clasificación para muestras, sin embargo funciona para clasificación entre genes mientras sean datos de conteo provenientea de la distribución Poisson, para lo cual la matriz usada es pxn donde p es el número de genes y n el número de muestras.

2.4.2. Reconocimiento de Patrones y métodos kernel

Una regularidad existente entre datos es un patrón que nos ayuda a agrupar nuevos individuos de los cuales se desconoce la clase en alguna de las clases existentes, hay patrones fácilmente reconocibles por algunos métodos y otros más complejos, que no son fáciles de identificar. De manera que el éxito de un método de clasificación depende de que tan complejos sean los patrones que puede reconocer.

A raíz de el auge de las máquinas capaces de hacer calculos rápidamente surgen técnicas de aprendizaje de máquina, capaces de detectar patrones de naturaleza no lineal y por lo

tanto que pueden no ser reconocidos por técnicas analíticas. Una carácteristica deseable en estos algoritmos es su robustez, es decir que no se vean afectados por datos atípicos. Uno de los enfoques enmarcados en el aprendizaje de máquina, es el aprendizaje estadisítico con funciones kernel.

Un algoritmo basado en funciones kernel consiste en:

- **Un conjunto de elementos**: llamado Input Space (Espacio de entrada) X que puede no ser un espacio vectorial, al que se va a definir una función que dará un valor a cada pareja de elementos y para definir la función no es necesario que x sea un espacio vectorial. Lo cual hace que sea de amplia aplicación.

- **Una función de embedding** $\phi : X \to F$ que transforma los datos de entrada, para obtener clasificadores no lineales es necesario realizar transformaciones no lineales. La transformación no lineal implica una correspondencia hacia un espacio de mayor dimensión, llamado espacio de características (Feature Space), posiblemente infinita. Este espacio debe estar dotado por de un producto punto, Espacio de Hilbert [38]

$$x : R^n \to \phi(x) : H$$

Al medir similitudes, como producto punto en el espacio de características se plantean modelos geométricos, que facilitan estudiar el algoritmo de aprendizaje usando álgebra lineal. Por otra parte, la selección de la función de embedding nos permite elegir una gran variedad de algoritmos de aprendizaje [31]

Los kernel conocidos son funciones que proporcionan directamete $\phi(x), \phi(z)\rangle$ sin necesidad de hacer explicitamente el embedding, ni el producto escalar.

Un kernel es una función de variable real $k : XxX \to R$ con las siguientes propiedades:

- $K(x) \in [0, \infty), x \in [-1, 1]$
- $K(x) = 0, x \in [-1, 1]$
- $K(x) = K(-x)$
- $\int_{-1}^{1} K(x)dx = 1$
- $\int_{-1}^{1} xK(x)dx = 0$
- $\int_{-1}^{1} x^2K(x)dx \in R^+$

Los Kernel representan objetos, como pares de comparaciones. Es decir que en lugar de usar una función $\phi : X \to F$ para representar cada objeto $x \in X$ por $\phi(x)$ (representación vectorial), una función de comparación de valor real $k : \chi x \chi \to R$ es usada y el conjunto de datos es representado por una matríz nxn de pares de comparaciones $k_{ij} = k(x_i, x_j)$ sin importar la naturaleza de los objetos analizados. [38]

Un kernel definido positivo puede ser visto como una medida de similitud, ya que $k(x, x')$ es más grande cuando x y x' son similares.

Además, toda función simétrica $k : X$ x $X \to R$ finitamente semidefinida positiva es un kernel válido. Efectivamente, dada una función simétrica finitamente semidefinida positiva $k : X$ x $X \to R$ es posible definir un espacio de Hilbert F_k, dotado de un producto interno $\langle\rangle : F_k \quad x \quad F_k \to R$ y una función $\phi : X \to F_k$ tales que $k(x, z) = \langle \phi(x), \phi(z) \rangle \forall x, z \epsilon X$ [17].

2.4.3. Tipos de Kernel

Dentro de los kernel más conocidos se encuentran:

- **Kernel lineal.** Corresponde al producto interno en el espacio de entrada.

$$k(x, x') = \langle x, x' \rangle$$

- **Kernel Polinomial.** representa la expansión a todas las combinaciones de monomios de orden d:

$$k(x, x') = \langle x, x' \rangle^d$$

- **Kernel Gaussiano.** está contenido dentro de las funciones de base radial:

$$k(x, x') = exp[\frac{\|x - x'\|^2}{2\sigma^2}]$$

2.4.4. El Truco Kernel

Dado un algoritmo formulado en términos de un kernel k definido positivo, se puede contruír un algoritmo alternativo reemplazando k por otro kernel definido positivo. [37]

El truco Kernel es basado en la propiedad de los kernel de ser pensados como productos internos e implica que cualquier algoritmo para datos vectoriales puede ser implícitamente realizado en el espacio de la características asociado con cualquier kernel, al reemplazar cada producto punto por otro kernel. Lo que genera una transformación de los métodos lineales como el análisis lineal discriminante o el análisis de componentes principales, a métodos no lineales simplemente reemplazando el producto punto clásico por un kernel. De esta manera,

se consigue la no linealidad manteniendo el algoritmo igual.

Es decir, es suficiente la información recogida por los productos escalares calculados para todas las parejas de elementos del espacio. Aunque resulta evidente que prescindir de las coordenadas reales de los elementos en el espacio y limitarse a la información recogida en el conjunto de productos escalares supone una pérdida de información (por ejemplo, se pierde la información relativa a la orientación del conjunto de datos en el espacio o la relativa a la alineación de los elementos con las variables originales), en muchas ocasiones esta pérdida no es relevante para alcanzar el objetivo de detección de patrones [17]

2.4.5. Máquinas de soporte vectorial. SVM

El método SVM es un método kernel supervisado, con óptimos globales por tratarse de un problema de optimización convexa. Este método es conocido como *"hiperplano óptimo de decisión"* dado que clasifica los elementos del conjunto de entrenamiento mediante un hiperplano que maximiza la distancia a los datos de cada categoría. A las muestras más cercanas al hiperplano óptimo se le conoce como vectores soporte [8]

La formulación de las máquinas de vectores se basa en el principio de minimización estructural del riesgo que ha demostrado ser superior al principio de minimización del riesgo empírico. La solución no depende de la estructura del planteamiento del problema. [31]

El hiperplano óptimo se busca en el espacio de características, es decir en el espacio al que fueron transformados los datos mediante el kernel, de manera que se encuentran hiperplanos de separación lineales en ese espacio.

Esta técnica busca minimizar el error en la separación de los objetos dados y maximizar el margen se separación. El hiperplano separador de margen máximo ha demostrado tener una buena capacidad de generalización, evitando en gran medida el problema del sobreajuste.

Para maximizar el margen, es decir la distancia entre los vectores soporte y el hiperplano óptimo, se formula el siguiente problema de optimización:

$$minimiza_{w \in H, b \in R} \tau(w) = \frac{1}{2}\|w^2\|$$

sujeto a

$$y_i(\langle w, x_i \rangle) + b) \geq 1 \text{ para toda } i = 1, .., m$$

donde:

- H: Espacio de producto punto
- w: Margen de separación del hiperplano y los vectores soporte.
- $\langle w, x_i \rangle) + b$: Hiperplano óptimo de separación.

La restricción lleva a que la clasificación de los individuos sea correcta.

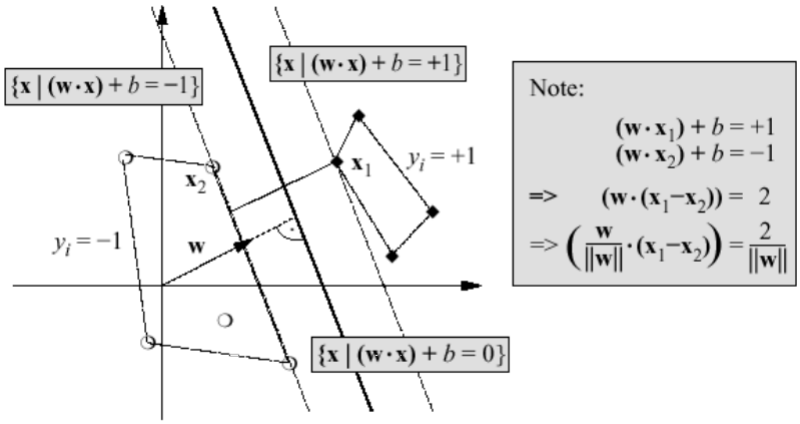


Figura 2-1.: Máquina de soporte vectorial [37].

Desde un punto de vista algorítmico, el problema de optimización del margen geométrico representa un problema de optimización cuadrático con restricciones lineales que puede ser resuelto mediante técnicas estándar de programación cuadrática. La propiedad de convexidad exigida para su resolución garantizan una solución única [44]

Existen tres (3) tipos de clasificadores en las máquinas de soporte vectorial, los de de margen duro (separables linealmente), de margen blando, que contienen ruido y son cuasi-separables linealmente y los que no son separables linealmente para lo cual se usa una máquina de soporte vectorial kernelizada.

Máquina de soporte vectorial para clasificación lineal

Supongamos que tenemos un conjunto de entrenamiento $x^i, i = 1, ..., N$ agrupados en clases $y_i = -1, 1$, y que existe un hiperplano que separa ambas clases sin error:

$$D(x) = (w_1 x_1 + ... + w_d x_d) + b = \langle w, x \rangle + b \tag{2-13}$$

Los puntos x sobre el hiperplano satisfacen $w^T x + b = 0$ donde el vector w es normal al hiperplano, entonces $\frac{|b|}{||w||}$ es la distacia perpendicular del hiperplano al origen siendo $|.|$ el operador valor absoluto y $\|.\|$ el operador norma.

Si d_1 y d_2 es la distancia más corta al hiperplano de separación a una muestra u otra más cercana. Se define el margen del hiperplano como $d_1 + d_2$. De manera que en este caso, el algoritmo busca el hiperplano con mayor margen.

En este caso se supone que todos los datos de entrenamiento satisfacen:

$$w^T x_i + b \geq +1, \quad para \quad y_i = +1 \tag{2-14}$$

$$w^T x_i + b \leq -1, \quad para \quad y_i = -1 \tag{2-15}$$

esto es:

$$y_i(w^T x_i + b) \geq 0 \forall i \tag{2-16}$$

En las ecuaciones 2-14 y 2-15 cuando se cumple la igualdad, lo puntos están sobre el hiperplano $H_1 : w^T x_i + b = 1$ con norma w y distancia al origen $|\frac{1-b|}{\|w\|}$. Para el hiperplano H_2 la distancia al origen es $|\frac{-1-b|}{\|w\|}$. Por lo tanto $d_1 = d_2 = \frac{1}{\|w\|}$ y el margen es $\frac{2}{\|w\|}$. Entre los dos hiperplanos no existen puntos de entrenamiento. Se maximiza el margen minimizando la función de costo $\frac{1}{2}\|w\|^2$ con la restricción 2-16 que garantiza que cada punto esté del lado del hiperplano que corresponde a su clase.

Es decir que el problema para el caso lineal queda planteado como:

$$minimiza_{w \in H, b \in R} \tau(w) = \frac{1}{2}\|w^2\|$$

sujeto a

$$y_i(\langle w, x_i \rangle) + b) \geq 1 \text{ para toda } i = 1, .., n$$

Para resolverlo, se usa la teoría de la optimización que establece que si un problema primal tiene una función a optimizar y restricciones con funciones estrictamente convexas, existe un problema dual que permite resolver el primal.

De manera que planteamos el problema dual, mediante las condiciones de Karush-Kuhn-Tucker (KKT), esto sustituye las restricciones iniciales pr restricciones sobre multiplicadores de Lagrange más fáciles de manejar. Además reformula el problema, de manera que los datos de entrenamiento aparecen en forma de producto punto, lo que es favorable para la posterior transformación del algoritmo a uno no lineal mediante el truco kernel.

Se introducen N multiplicadores de Lagrange $(\alpha_1, ..., \alpha_N)$, uno para cada restricción del problema primal. Para restricciones de la forma $\zeta_i \geq 0$ las ecuaciones que las definen se multiplican por multiplicadores de Lagrange positivos y se restan de la función objetivo para formar el lagrangiano. En caso de restricciones de la forma $\zeta_i = 0$ los multiplicadores de Lagrange no tienen restricciones [31].

Primero, se construye un problema de optimización sin restricciones utilizando la función Lagrangiana:

$$L(w, b, \alpha) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{n} \alpha_i [y_i(\langle w, x_i\rangle) - 1] \tag{2-17}$$

siendo $\alpha_i \geq 0$ los multiplicadores de Lagrange.

Ahora, se aplica las condiciones KKT, obteniendo:

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad i = 1, ..., n \tag{2-18}$$

que expresa los parámetros de w y b en términos de α_i. Y

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad i = 1, ..., n \tag{2-19}$$

que establecen restricciones para α_i. Y:

$$\alpha_i[1 - y_i(\langle w*, x_i\rangle + b*)] = 0, \quad i = 1, ..., n \tag{2-20}$$

Finalmente, la forma dual queda de la siguiente forma:

$$\begin{aligned} max \quad L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle \\ s.a \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, i \equiv= 1, ..., n \end{aligned} \tag{2-21}$$

Al igual que el problema primal, este problema se resuelve mediante programación cuadrática, pero el costo computacional es menor ya que la magnitud del problema dual depende de las muestras mientras que el del primal depende de la dimensionalidad. [31]

Con la solución del problema dual, obtenemos la solución del primal susituyendo el resultado en la ecuación 2-18 y a su vez este resultado en 2-13:

De la ecuación 2-20 resultante de aplicar las restricciones de KKT, se observa que si $\alpha_i > 0$ el segundo factor de la parte izquierda deberá ser cero y por tanto:

$$y_i(\langle w*, x_i\rangle + b*) = 1 \tag{2-22}$$

es decir, el correspondiente ejemplo (x_i, y_i) satisface la restricción de igualdad del problema dual y son llamados vectores soporte.

El hiperplano de separación óptimo será una combinación lineal de sólo los vectores soporte del conjunto de ejemplos donde α_i, ya que los demás tendrán asociado un $\alpha_j = 0$.

El valor de b* se calcula despejando b* de la ecuación anterior:

$$b* = y_{vs} - \langle w*, x_{vs}\rangle \tag{2-23}$$

siendo (x_{vs}, y_{vs}) la pareja de cualquier vector soporte

Máquina de soporte vectorial para clasificación no lineal

Para este tipo de clasificación se definen espacios transformados de alta dimensionalidad y cómo buscar hiperplanos de separación óptimos en dichos espacios transformados. A cada uno de estos espacios se le denomina espacio de características, para diferenciarlo del espacio de ejemplos de entrada (espacio-x).

Sea $\phi : X \longrightarrow H$ la función de embedding que hace corresponder cada vector de entrada x con un punto en el espacio de características H, donde $\phi(x) = [\phi_1(x), ..., \phi_m(x)\,]\, y\phi_i(x)$ es una función no lineal. Se construye un hiperplano de separación lineal en este nuevo espacio. La frontera de decisión lineal obtenida en el espacio de características se transformará en una frontera de decisión no lineal en el espacio original de entradas (figura 2-2)

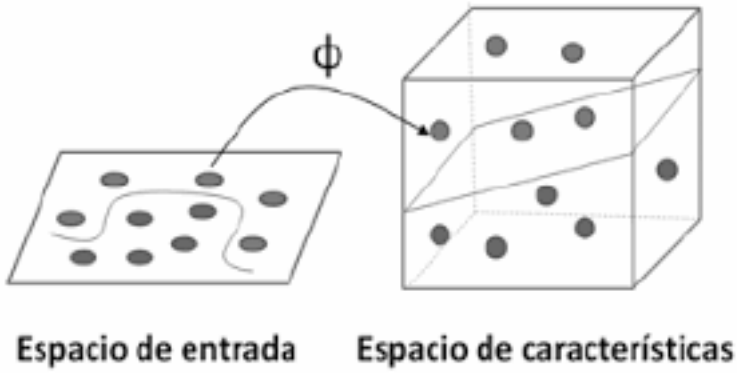


Figura 2-2.: Transformación en el espacio de características ([41])

De manera que la función de decisión en el espacio de características viene dada por:

$$D(x) = (w_1\phi_1(x)...w_m\phi_m(x)) = \langle w, \phi(x)\rangle \tag{2-24}$$

y en su forma dual, la función de decisión se obtiene transformando la expresión del caso lineal:

$$D(x) = \sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i) \tag{2-25}$$

siendo $K(x, x_i)$ una función kernel. Y usando el truco kernel, se sustituye el producto punto por una función kernel, haciendo el algoritmo no lineal. De manera que el problema es de nuevo encontrar los parámetros $\alpha_i^*, i = 1, ..., n$ que optimizan el problema dual, ahora expresado como:

$$\begin{aligned} max \quad L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.a \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, i \equiv= 1, ..., n \end{aligned} \tag{2-26}$$

El valor C es un parámetro de regularización, para el cual no hay manera óptima de obtenerlo.

2.5. Medidas de Similitud y distancias

En el aprendizaje se quieren generalizar los datos no vistos y asignarlos a algún grupo previamente conocido, para lo que son de vital importancia las medidas de similitud entre objetos, es decir, es importante encontrar una función que dados dos valores x, x', devuelva un valor que caracterice su similitud. Una medida de similitud es el producto punto entre dos datos, y se define como:

$$\langle x, x' \rangle = \sum_{i=1}^{N} (x_i)(x_i')$$

donde x_i es la i-ésima entrada del vector.

El Producto Punto es el coseno del ángulo (θ) entre los vectores x, x' por el producto de sus módulos:

$$xx' = |x||x'|cos\theta$$

El Producto Punto nos permite llevar a cabo toda la construcción en términos geométricos y formularlo en función de cosenos, longitudes y distancias.

Información Mutua

La Información mutua es una medida de similitud que reconoce relaciones lineales y no lineales. Su estimación depende de las funciones de densidad marginal y se enmarca en el campo de la teoría de la información mutua, siendo una medida interesante a la hora de evaluar asociación de datos génicos.

La Información mutua entre dos variables x e y se define como:

$$I(x,y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p(x,y) log \frac{p(x,y)}{p(x), p(y)}$$

donde $p(x)$ es la probabilidad de $X = x$, $p(y)$ es la probabilidad de $Y = y$ y $p(x,y)$ es la probabilidad conjunta de que $X = x$ e $Y = y$.

En el caso continuo, la información Mutua entre dos variables X e Y, se define como:

$$I(X,Y) = \int_{Y} {}_{X} f_{xy}(x,y) log \frac{f_{xy}(x,y)}{f_x(x), f_y(y)} dxdy$$

siendo $f_{xy}(x,y)$ la función de densidad conjunta de X e Y, y $f_x(x)$ y $f_y(y)$ las funciones de densidad marginales de x e y respectivamente.

La Información mutua mide la cantidad de información que comparten ambas variables, si su valor es cero, las variables no comparten información, es decir el conocimiento de una de las variables no brinda información sobre la otra, y si su valor es mayor que cero, existe asociación entre x e y, esto es, el conocimiento de una de las variables nos aporta información sobre la otra.[15]

La Información mutua entre dos variables se puede interpretar como la reducción de la incertidumbre en la variable x una vez conocida la variable y. La Información mutua de una variable consigo misma es igual a su entropía. Siendo la entropía el valor medio de la información de una variable, y se define como:

$$H_x = -\sum_{i=1}^{N} p(x = i) log[p(x = i)]$$

Se puede considerar una medida de dispersión de la densidad $p(x)$ Es máxima cuando la distribución es uniforme y mínima cuando la $p(x) = 0$. La entropía diferencial (caso contínuo) es máxima en el caso de la densidad gaussiana.

La información mutua se puede ver como una entropía relativa (ver gráfica) o distancia de Kullback Leibler, la cual dadas os funciones de probabilidad $p(x)$, $q(x)$, se define como:

$$D(p|1) = \sum_{x \epsilon X} p(x) log \frac{p(x)}{q(x)}$$

donde:

- $0 * log(0/q) = 0$
- $p * log(p/0) = \propto$

No se trata de una distancia real ya que no es simétrica, pero suele verse como una distancia entre distribuciones, ya que es no negativa y es cero si $p = q$.

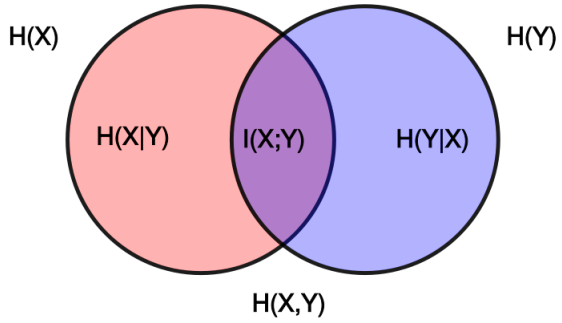


Figura 2-3.: Relación entre entropía e información mutua (Fuente. Wikipedia)

Distancia Poisson

Basada en la idea que la distancia euclidiana puede encontrarse como consecuencia de desarrollar una prueba de hipótesis en un modelo gaussiano y es equivalente a la razón de log-verosimilitud para los datos se desarrolla la distancia Poisson. [9]

Para lo cual se considera el modelo:

$$x_{ij} \sim Poisson(N_{ij}d_{ij})$$
$$x_{i'j} \sim Poisson(N_{i'j}d_{i'j})$$
$$N_{ij} = s_i g_j$$
$$N_{ij} = s_{i'} g_j$$

Se supone independencia. Y se tiene el siguiente contraste de hipótesis:

$$H_0 = d_{ij} = d_{i'j} = 1, j = 1, ...p$$
$$H_a = d_{ij}, d_{i'j} \geq 1$$

El resultado del test de verosimilitud se usa como distancia entre x_i y $x_{i'}$. Aunque en lugar de estimar por máxima verosimilitud d_{ij} y $d_{i'j}$ bajo la hipótesis alternativa como se hace en el test estándar, se calcula un test de razón de verosimilitud modificado donde se evalúa la log- verosimilitud bajo H_a usando los estimados:

$$\hat{d_{ij}} = \frac{x_{ij} + \beta}{\hat{N_{ij}} + \beta}$$
$$\hat{d_{i'j}} = \frac{x_{i'j} + \beta}{\hat{N_{i'j}} + \beta}$$

El resultado de la razón de verosimilitud modificada obtenido por D. Witten es:

$$\sum_{j=1}^{p} (\hat{N_{ij}} + \hat{N_{i'j}} - \hat{N_{ij}}\hat{d_{ij}} - \hat{N_{i'j}}\hat{d_{i'j}} + x_{ij}log(\hat{d_{ij}}) + x_{i'j}log(\hat{d_{i'j}}))$$

Dos observaciones iguales tienen distancia igual a cero (0) y todos los valores son no negativos. Esta distancia puede ser calculada para hallar la matriz de distancias Poisson nxn, es decir la matriz de distancias entre las muestras. Y también para calcular la matriz de distancias entre genes pxp en cuyo caso se revesan los roles de muestras y genes en las ecuaciones pesentadas.

2.5.1. Razón de verosimilitud

Prueba de Hipótesis

Supongase que se tiene una muestra aleatoria $X = (x_1, ..., x_n)$ con cierta densidad $f(x, w)$ y con una distribución $F(x, w)$ donde $w \epsilon \Omega \subset R^n, n \geq 1$ y Ω es el espacio de parámetros. Sea $\Omega = \Omega_0 \cup \Omega_1$ donde Ω_0 y Ω_1 son dos conjuntos disjuntos, es decir

$\Omega_1 = \Omega - \Omega_0$. Dadas las realizaciones de X queremos decidir si w está en Ω_0 o en Ω_1.[28]

La hipótesis de que w pertenece a Ω_0 se le conoce como hipótesis nula y se denota por H_0 mientras que la hipótesis de que w pertenece a Ω_1 se le llama hipótesis alternativa y se denota por H_1.

Para determinar si w esta en Ω_0 y Ω_1 se emplea una regla basada en X, la cual es una funcion ϕ que toma los valores 0 y 1, es decir, si $\phi(X) = 0$ entonces decimos que $\theta \epsilon \Omega_0$ y si $\phi(X) = 1$ entonces $w \epsilon \Omega_0$. Esta función depende de X mediante otra función cuyo valor llamado estadístico de prueba, se denota por $T = T(X)$ [28].

Prueba de Razón de Verosimilitud Generalizada

Un procedimiento para un juego de hipótesis es la razón de verosimilitud generalizada, el cual utiliza el lema Neyman-Pearson, que describe el criterio óptimo para distinguir dos hipótesis, junto con la estimación de máxima verosimilitud [28].

El estadístico de Razón de Verosimilitud Generalizada (RVG) se de- fine como el cociente de:

$$\lambda(X) = \frac{sup_{w \epsilon \Omega_0} f(X, w)}{sup_{w \epsilon \Omega_1} f(X, w)} = \frac{sup_{w \epsilon \Omega_0} L(w)}{sup_{w \epsilon \Omega_1} L(w)} \tag{2-27}$$

donde $L(w)$ es la función de verosimilitud.

Para usar el estadístico λ es necesario conocer su distribución cuando H_0 es verdadera. Algunas veces es muy difícil obtener la distribución de λ, sin embargo una alternativa es utilizar su distribución asintótica [28].

3. Capítulo 3. Materiales Y Métodos

3.1. Matrices simuladas

En este capítulo se presenta el procedimiento, las herramientas estadísticas y de software usado en el desarrollo de la presente tesis(ver gráfica 3-1).

Las simulaciones se hacen con el fin de representar una matriz proveniente de datos RNAseq, es decir con datos de conteo con mil (1000) genes y cien(100) muestras, donde los genes presentan coexpresión, es decir, en la cual existen grupos de genes que tienen algún tipo de relación y pertenecen a grupos diferentes.

Inicialmente, se realiza una simulación para una matriz con dos grupos, donde existen relaciones lineales entre los genes de cada grupo, en la segunda simulación, se genera una matriz donde se distorsiona la linealidad de uno de los grupos, y en la tercera matriz se generan dos grupos de genes, uno con relaciones lineales y otro con relaciones no lineales (sinusoidales)(Ver Apéndice 1). Cada grupo tiene 500 genes. Esto con el fin de posteriormente poner a prueba los métodos de clasificación lineal y no lineales con las distintas medidas de similitud propuestas y compararlas con las existentes.

Para la obtención de los datos de conteo se usa la función CountDataSet del paquete PoiClaClu [50] de R [47], que genera k grupos de distribuciones binomiales negativas, según un parámetro de dispersión determinado y la función rnbinm que genera datos de conteo provenientes de una distribución binomial negativa según el parámetro de dispersión y la probabilidad de éxito de cada ensayo

Para obtener el primer grupo de genes con relaciones lineales entre ellos, se simularon dos subgrupos, cada uno de 250 genes, el primero con la función CountDataSet con el parámetro de dispersión (param) proveniente de una distribución uniforme de 80 a 120, con k=1, es decir que los datos provienen del mismo grupo y tienen relaciones lineales entre ellos, el segundo subgrupo son los primeros elevados a una potencia proveniente de una distribución uniforme de (1.2 a 1.5) más la parte entera de un ruido provenien-

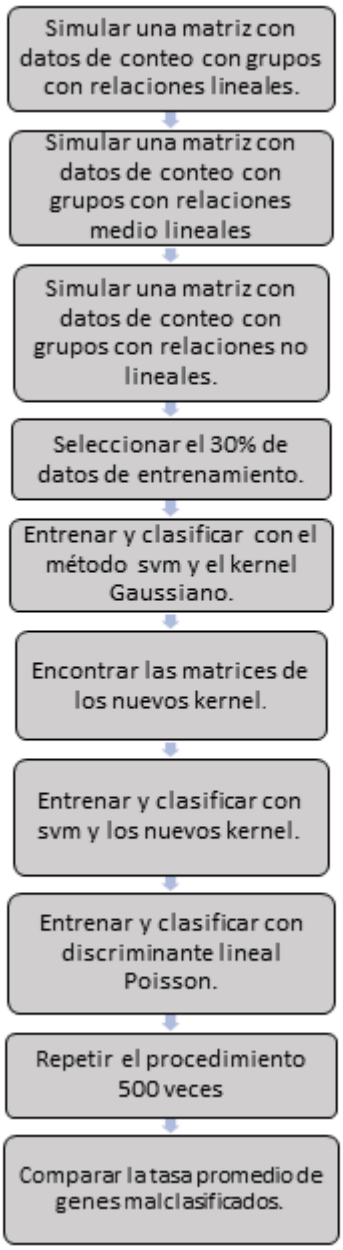


Figura 3-1.: Diagrama de flujo proceso datos simulados

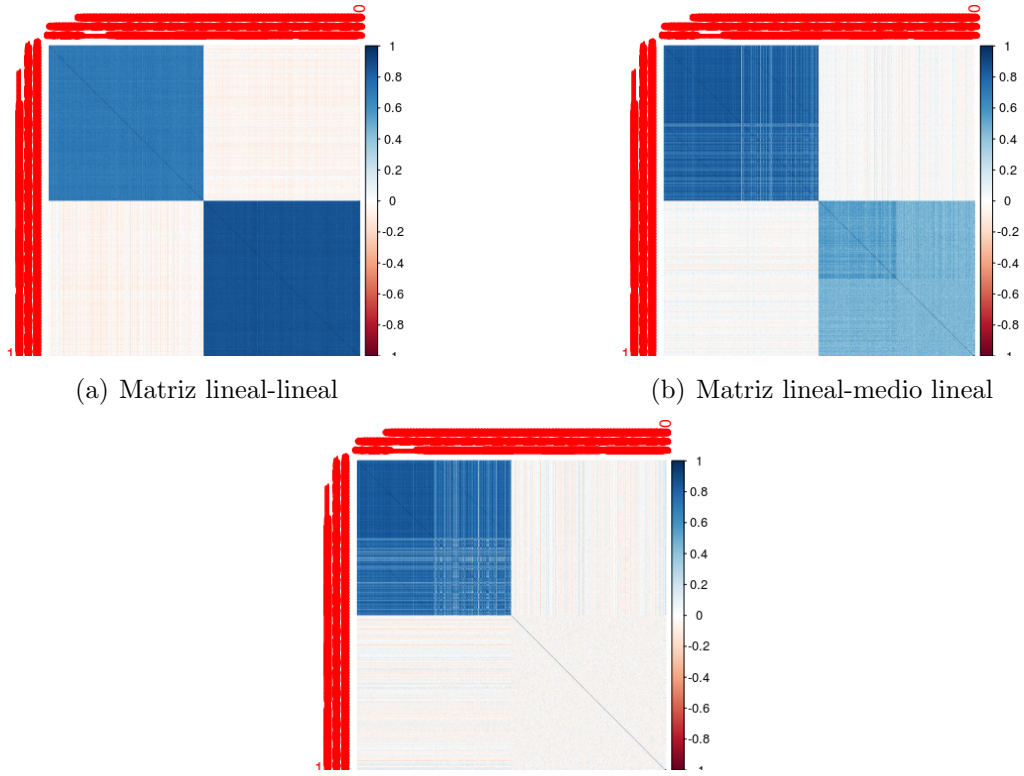
te de una distribución uniforme con límites inferior y superior 20 y 250 respectivamente.

Para el segundo grupo con relaciones lineales se simulan dos subgrupos con la función CountDataSet cada uno con k=1,con parámetros provenientes de distribuciónes uniformes, la primera de 40 a 60, y la segunda de 2 a 22, los dos subgrupos también con relaciones lineales entre ellos.

De manera que los datos son provenientes de distribuciones binomiales negativas con distintos parámetros, manteniendo entre cada grupo de genes relaciones lineales. Los genes del primer y segundo grupo no están correlacionados. Se verifican las correlaciones de la matriz mediante el correlograma de la matrices de correlación de Spearman, mediante la función Corrplot de la libreria con el mismo nombre [50] (Ver gráfica 3-2).

Para la segunda matriz, se distorsiona la relación lineal del primer grupo de genes, aplicando una relación sinusoidal a las primeras 20 muestras y adicionando la parte entera de un ruido aleatorio proveniente de una distribución uniforme entre 60 y 180 a las últimas 20. Los genes de grupos diferentes no están correlacionados

Para la tercera matriz, al primer grupo de datos proveniente de una distribución binomial negativa se le aplica una relación sinusoidal. De manera que tenemos un grupo de genes con relaciones lineales y otro grupo con relaciones sinusoidales (Ver apendice 1).



(a) Matriz lineal-lineal

(b) Matriz lineal-medio lineal

(c) Matriz lineal-no lineal

Figura 3-2.: Correlogramas de las correlaciones de Spearman

Se seleccionan aleatoriamente el 30 % de los genes de cada grupo con el fin de asegurar que la muestra de entrenamiento tenga datos de los dos grupos. El 70 % restante se

usa para probar los clasificadores. El porcentaje se elige pensando en la realidad de las matrices de transcritos.

El porcentaje de genes mal clasificados es realizada en los datos de prueba, y se realiza de la siguiente manera:

$$\%\text{genes mal clasificados} = \frac{\text{No. de genes mal clasificados en los datos de prueba*100}}{\text{No total de genes de prueba}} \tag{3-1}$$

3.2. Datos reales

El procedimiento que se sigue para tratar los datos reales se muestra mediante un diagrama de flujo (Ver gráfica 3-3). Se utilizan datos de RNAseq que aún no han sido publicados y que fueron obtenidos por el grupo de investigación de fitopatología vegetal del profesor Camilo López del Departamento de biología de la Universidad Nacional de Colombia.

Las muestras corresponden a tallos de yuca cultivadas en condiciones de invernadero (30/20 +/- 2^oC de temperatura de día/noche, 12h de fotoperiodo y 70 % de humedad relativa). Unas de ellas inoculadas con 10nm de $MgCl_2$ (muestras de control) y otras con 10 ul de $MgCl_2$ (Muestras de tratamiento)

Las muestras se midieron el día del procedimiento de inoculación (día cero), uno, tres y cinco días después, cada vez para tres muestras biológicas. De manera que las mediciones realizadas son:

- $R1{,}00.MgCl/R1{,}00.Bac$: Muestra de control/tratamiento medida en el día cero del procedimiento para la replica biológica uno.
- $R2{,}00.MgCl/R2_00.Bac$: Muestra de control/tratamiento medida en el día cero del procedimiento para la replica biológica dos.
- $R3{,}00.MgCl/R3_00.Bac$: Muestra de control/tratamiento medida en el día cero del procedimiento para la replica biológica tres.
- $R1.d1.MgCl/R1_d1.Bac$: Muestra de control/tratamiento medida en el día uno del procedimiento para la replica biológica uno.
- $R2.d1.MgCl/R2_d1.Bac$: Muestra de control/tratamiento medida en el día uno del procedimiento para la replica biológica dos.

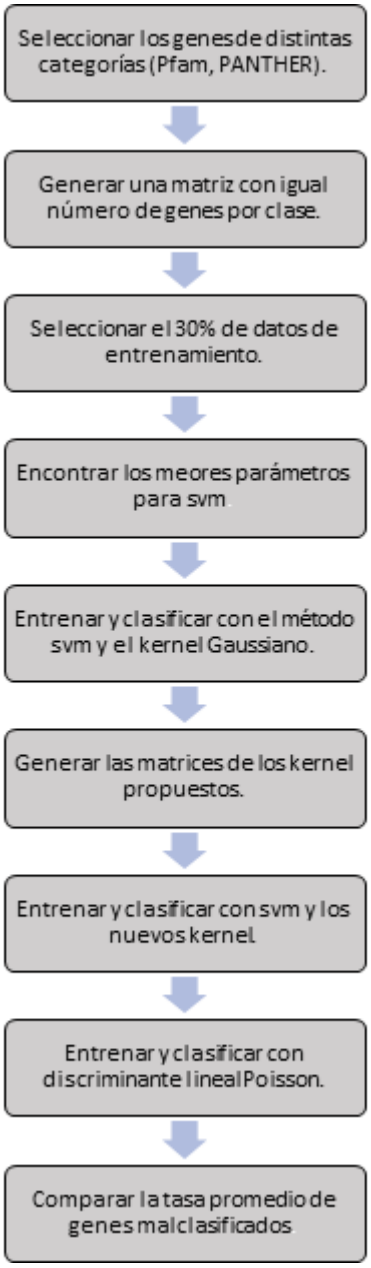


Figura 3-3.: Diagrama de flujo proceso Datos reales

- $R3.d1.MgCl/R3_d1.Bac$: Muestra de control/tratamiento medida en el día uno del procedimiento para la replica biológica tres.
- $R1.d2.MgCl/R1_d2.Bac$: Muestra de control/tratamiento medida en el día tres del procedimiento para la replica biológica uno.
- $R2.d2.MgCl/R2_d2.Bac$: Muestra de control/tratamiento medida en el día tres del procedimiento para la replica biológica dos.

- $R3.d2.MgCl/R3_d2.Bac$: Muestra de control/tratamiento medida en el día tres del procedimiento para la replica biológica tres.
- $R1.d2.MgCl/R1_d2.Bac$: Muestra de control/tratamiento medida en el día cinco del procedimiento para la replica biológica uno.
- $R2.d2.MgCl/R2_d20{,}5.Bac$: Muestra de control/tratamiento medida en el día cinco del procedimiento para la replica biológica dos.
- $R3.d2.MgCl/R3_d2.Bac$: Muestra de control/tratamiento medida en el día cinco del procedimiento para la replica biológica tres.

De la matríz de datos completa se seleccionaron los genes que pertenecen a cuatro (4) categorías Pfam que corresponden a genes de distintas familias de proteínas. (PF00067, PF00069, PF00249, PF01535), obteniendo una matriz de 1766x24. Cada grupo tiene 341, 548, 288 y 589 genes respectivamente, de los cuales se seleccionaron aleatoriamente el 30 % de los genes de cada grupo para entrenamiento y el 70 % restante para prueba.

Después para tener una matriz con el mismo número de genes en cada grupo, se selecciona de cada uno una muestra aleatoria de 250 genes, obteniendo una matriz de 1000 genes por 24 muestra.

También se seleccionan 4 tipos de genes, los correspondientes a las categorías PTHR24015, PTHR24420, PTHR10641 y PTHR24298 con 367, 161, 168 y 169 genes respectivamente, según la clasificación PANTHER (Protein Analysis Through Evolutionary Relationships), que realiza una clasificación de genes mediante información completa sobre la evolución y la función de genes, obteniendo una matriz de 865 genes x 24 muestras, con un desbalance en la cantidad de genes por grupo, para ver el comportamiento de los métodos en este caso.

3.3. Métodos de Clasificación

3.3.1. Discriminante Lineal Poisson

Nuestro método de referencia lineal es el análisis discriminante Poisson desarrollado por Daniela Witten [9]. En este método se crea una regla de clasificación lineal mediante la regla de Bayes, suponiendo que los datos provienen de una distribución Poisson. El Discriminante Poisson se lleva a cabo mediante la función d.poisson del paquete PoiClaClu [50]

El método fue desarrollado en el marco del analisis de datos RNA seq dado que las transformaciones para normalizar los datos para su posterior clasificación mediante métodos tradicionales (discriminante lineal de Fisher...) presentan distintos problemas y agregan ruido a los datos. Algunos trabajos indican que el intento de corrección de sesgo por longitud introduce un sesgo en las varianzas por gen, en particular para genes con baja expresión [36].

De manera que este método de clasificación se ajusta mejor a la distribución de los datos de RNA seq. Sin embargo, este modelo no funciona bien para distribuciones con sobredispersión y supone independencia entre los datos. Lo que puede no ser cierto ni cuando se clasifican muestras, ya que las muestras pueden presentar algún tipo de relación, ni cuando se clasificangenes, ya que los genes pueden presentar coregulación. Además de tratarse de un método lineal.

3.3.2. Máquinas de soporte vectorial

Las máquinas de soporte vectorial han sido ampliamente utizadas en el reconocimiento de patrones por su flexibilidad, eficiencia computacional y capacidad para manejar grandes volúmenes de datos. Se han usado en la clasificación con información génica oteniendo mejores resultados que con otros métodos. [16, 12, 2, 29]

Distintos trabajos evidencian que la selección del kernel y un buen ajuste de los parámetros, es crucial en la obtención de una buena clasificación. [5, 35] Razón por la cual se utiliza el kernel Gaussino y dos nuevos kernel propuestos, basados en Información Mutua y en la distancia Poisson.

Máquina de soporte vectorial con kernel gaussiano

El primer kernel implementado es el kernel gaussiano, por su flexbilidad y su reducido número de parámetros comparado con otros kernel, como el polinomial. Su principal problema es la selección del parámetro, para el cual se han desarrollado varios métodos, el más conocido mediante validación cruzada, que hace una búsqueda exhaustiva dentro de un grupo de valores posibles, con el problema de altos tiempos de procesamiento. [52]

Existen otros métodos basados en la maximización de la distancia entre clases, reduciendo el tiempo de procesamiento en relación con las técnicas de validación cruzada, pero basados en la distancia euclidiana [52, 53] que pueden no ser adecuados para

nuestros datos provenientes de una distribución binomial negativa.

Para elegir el parámetro gamma y el parámetro C, costo de mala clasificación, se combinan los valores de 0.5 a 5, con intervalos de 0.5 para gamma y de 0.5 a 10 con intervalos de 0.5 para el parámetro de costo. Luego de obtener un valor, se toman nuevos valores alrededor de éste con intervalos de 0.01. Para lo cual, se utiliza la función tune.svm del paquete e1071 [32] que nos arroja la combinación de parámetro gamma y costo C que minimiza el error de clasificación en los datos de entrenamiento.

Luego de conocer el parámetro óptimo en una prueba piloto se elige un conjunto de valores cercanos al encontrado para realizar multiples clasificaciones con el fin de observar los efectos de un cambio de parámetro en el kernel Gaussiano.

En la clasificación mediante éste método se usa la función svm del paquete e1071 [32]. Para visualizar los valores de las matrices kernel usamos la función ksvm del paquete kernlab [25] y nuevamente la función corrplot. Para ver código ir al Apéndice 2.

Máquina de soporte vectorial con kenel basado en distancia Poisson

El segundo kernel es el producto punto de la matriz de distancia Poisson, con lo cual se obtiene una matriz semidefinida positiva, requisito suficiente para que una matriz sea considerada un kernel, que es una medida de similitud sugerida aquí. Este kernel presenta dos ventajas relevantes, la primera, tiene en cuenta la naturaleza discreta de los datos y la segunda, permite trabajar con los datos de conteo directamente, sin realizar ninguna corrección de sesgo innecesaria o normalización disminuyendo el ruido introducido al modelo. Dentro del cálculo de la distancia Poisson se tiene en cuenta la profundidad de secuenciación (ver marco teórico)

La distancia Poisson es la razón de log verosimilitud basado en el supuesto que los datos provienen de una distribución Poisson. Para su cálculo se desarrolla una prueba de hipótesis con la hipótesis nula que el parámetro $d_{ij} = 0$, siendo este el parámetro el que mide la diferencia de expresión en los grupos [9].

La distancia Poisson es calculada por la función PoissonDistance del paquete PoiClaClu [50], posteriormente se kerneliza mediante el producto punto y se clasifica con el método máquina de soporte vectorial usando la función svm del paquete e1071 [32] (Ver código en Apéndice 2). Para visualizar los valores de las matrices kernel usamos

la función ksvm del paquete kernelab [25].

Para visualizar las similitudes de las diferentes matrices kernel se normalizaron dividiendo por el valor máximo en cada matriz, con el único fin de visualizarlas mediente la función corrplot del paquete que lleva el mismo nombre [50].

Para elegir el parámetro C, costo de mala clasificación, se prueban los valores de 0.5 a 10 con intervalos de 0.5. Luego de obtener un valor, se toman nuevos valores alrededor de éste con intervalos de 0.01 usando la función tune.svm del paquete e1071 [32] que nos arroja el parámetro que minimiza el error de clasificación en los datos de entrenamiento.

Máquina de soporte vectorial con kenel basado en Información Mutua

El tercer kernel es el producto punto de la raíz cuadrada de la matriz de Información Mutua, con lo cual se obtiene una matriz semidefinida positiva, requisito suficiente para que una matriz sea considerada un kernel, que es una medida de similitud sugerida aquí, dado que la Información Mutua es una medida de asociación entre variables que detecta relaciones lineales y no lineales, siendo adecuada para sistemas biológicos complejos, funcionalmente diversos y a menudo multifuncionales que son susceptibles de interactuar de manera no lineal [19].

La información mutua y sus modificaciones han sido ampliamente utilizadas como medidas de similitud en estudios de clasificación con información génica, obteniendo resultados favorables en relación con otras medidas de similitud. [16, 40]

Para el cálculo de la información mutua se discretizan los datos mediante la función discretize del paquete infotheo [33] en intervalos de igual frecuencia, y se usa la regla de Sturges para determinar el número de intervalos: $nb = 1 + log_2(p)$ siendo p el número de genes [39]. Para hallar el coeficiente de información mutua, se usa la función mi.empirical del paquete entropy [22] de R (Ver Apéndice 2), que después es kenelizada aplicando el kernel lineal (producto punto) para ser usada en la clasificación de genes mediante la función svm del paquete e1071 [32] (Ver código en Apéndice 2).

Al igual que con el kernel Gaussiano y kernel basado en la distancia Poisson, para elegir el parámetro C, costo de mala clasificación, se prueban los valores de 0.5 a 10. Luego de obtener un valor, se toman nuevos valores alrededor de éste con intervalos de 0.01 usando la función tune.svm del paquete e1071 [32] que nos arroja el parámetro

que minimiza el error de clasificación en los datos de entrenamiento.

4. Capítulo 4. Resultados

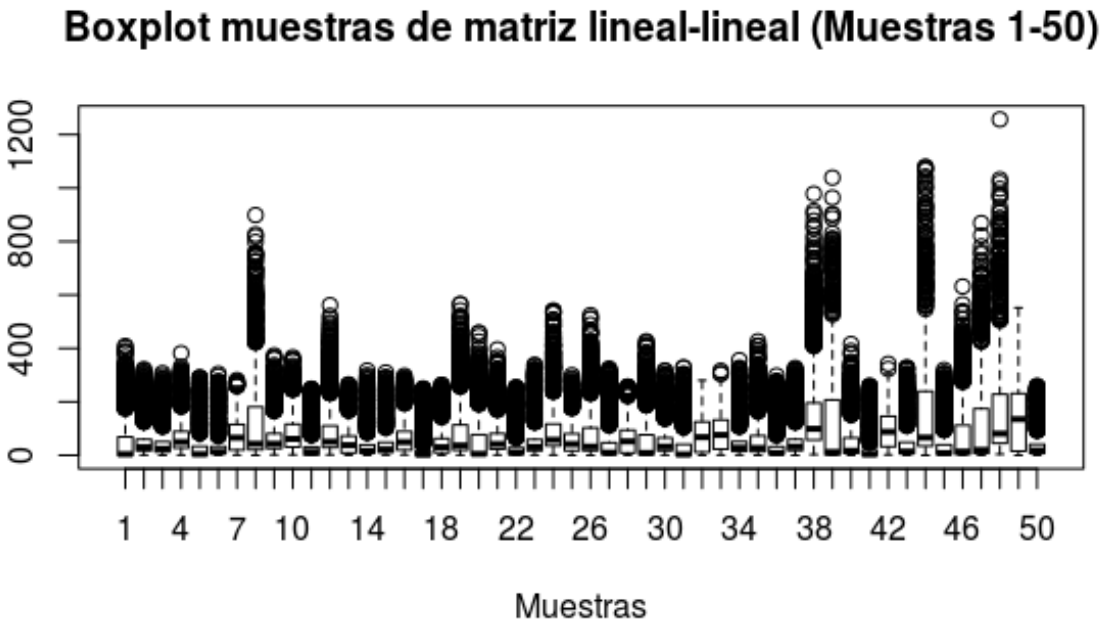
En el presente capítulo se presentan los resultados obtenidos en datos reales, provenientes de datos de yuca y en datos simulados. Se tienen tres (3) tipos de matrices simuladas (ver materiales y métodos) cada una con dos grupos de genes, en la primera matriz los genes de cada grupo presentan relaciones lineales entre ellos, la segunda matriz tiene un grupo con relaciones lineales entre genes y otro donde se inserta ruido a la relación lineal inicial (matriz lineal-medio lineal), la tercera matriz tiene un grupo con relaciones lineales y el otro con relaciones no lineales entre genes, puntualmente de tipo sinusoidal (matriz lineal-no lineal).

Todas las matrices contienen datos de conteo, provenientes de distribuciones binomiales negativas como se asume en la distribución de los conteos RNAseq. Para visualizar la estructura de los datos simulados se muestran los boxplot de las 100 muestras (Ver gráficas de 4-1 a 4-3) y un ejemplo de un histograma de una muestra de cada caso (Ver gráfica 4-4).

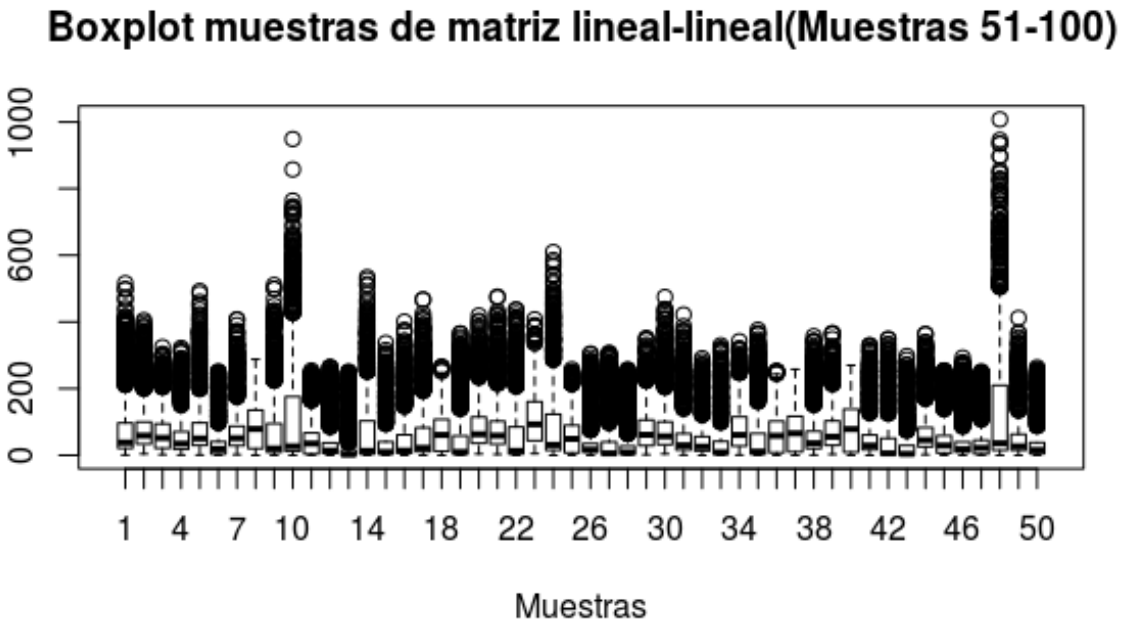
Los métodos que se prueban son dos, uno de naturaleza lineal, Análisis Discriminante Poisson y otro de naturaleza no lineal, máquinas de soporte vectorial, este último se prueba con distintos tipos de kernel, uno ya existente, el kernel Gaussiano, por ser el más usado debido a sus propiedades y dos propuestos en este trabajo, uno basado en información mútua y otro basado en la distancia Poisson.

El kernel basado en información mutua es el producto punto de la información mutua entre genes y el basado en la distancia Poisson es el producto punto de la distancia Poisson entre genes, siendo esta la distancia análoga a la distancia Euclidiana para datos de conteo provenientes de una distribución Poisson. [9]

Teniendo en cuenta que la distancia Euclidiana es consecuencia de desarrollar una prueba de hipótesis sobre las medias suponiendo normalidad de los datos y es equivalente a la razón de log-verosimilitud, aquí se encuentra la razón de verosimilitud de una prueba de hipótesis basada en el parámetro $d_{i'j}$ que define la diferencia de expresión entre genes (Ver marco teórico). [9]

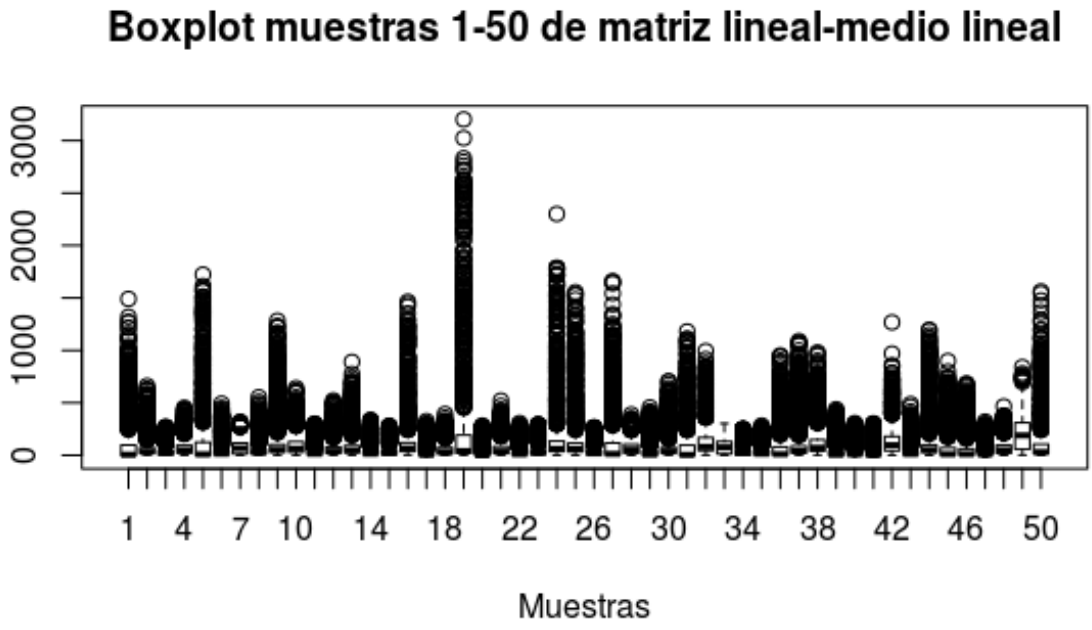


(a) Boxplot muestras 1 a 50

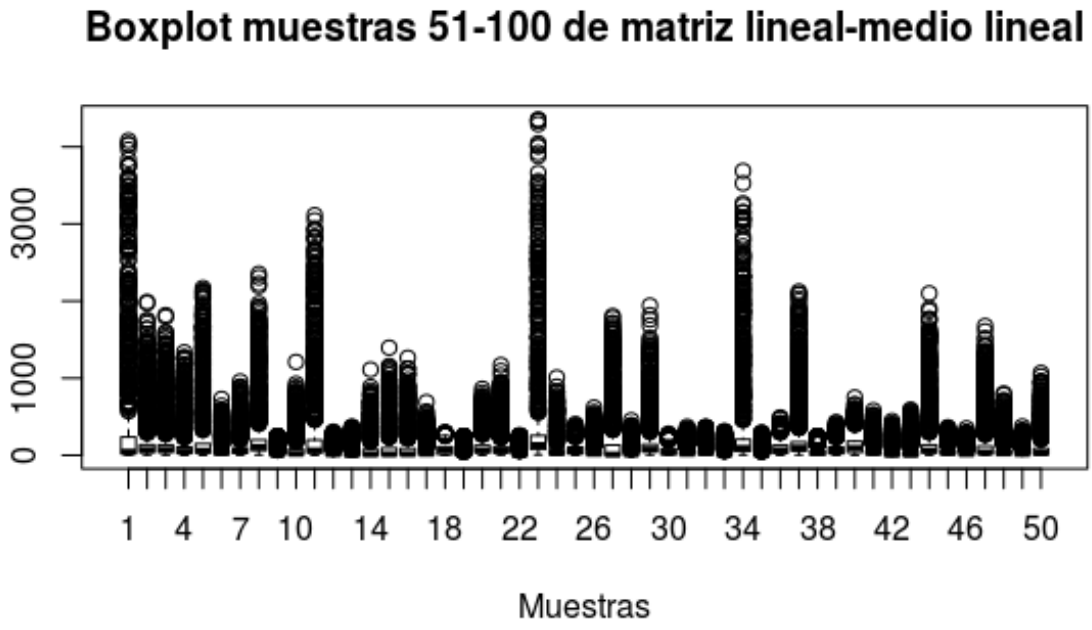


(b) Boxplot muestras 51 a 100

Figura 4-1.: Boxplot de las muestras en la matriz con relaciones lineal-lineal

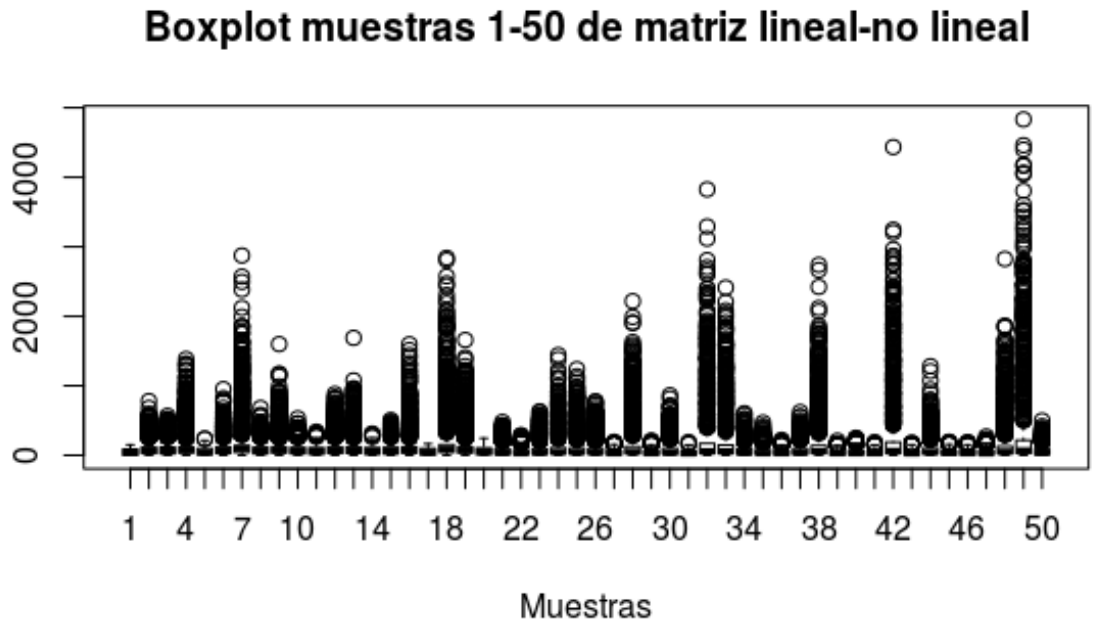


(a) Boxplot muestras 1 a 50

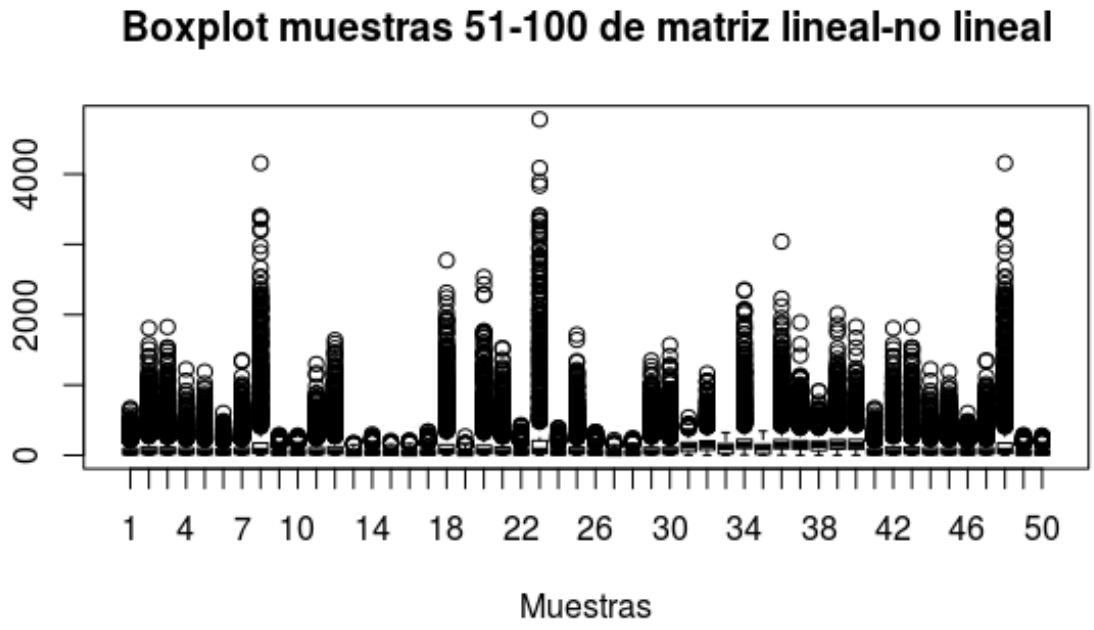


(b) Boxplot muestras 51 a 100

Figura 4-2.: Boxplot de las muestras en la matriz con relaciones lineal-medio lineal

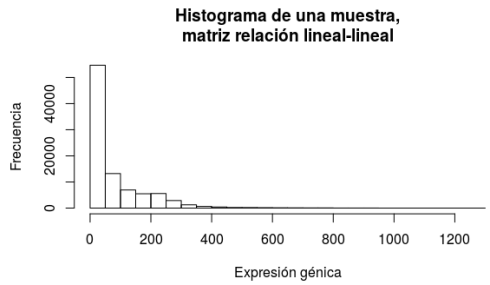


(a) Boxplot muestras 1 a 50



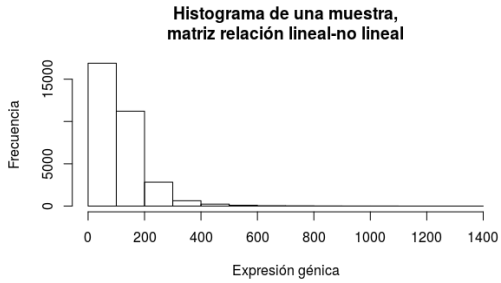
(b) Boxplot muestras 51 a 100

Figura 4-3.: Boxplot de las muestras en la matriz con relaciones lineal-no lineal



(a) Matriz lineal-lineal

(b) Matriz lineal-medio lineal



(c) Matriz lineal- no lineal

Figura 4-4.: Ejemplo de histograma de una de las muestras de cada matriz simulada

4.0.1. Resultados en Datos Simulados

Se evalúan los resultados del promedio de genes mal clasificados en 500 simulaciones, en cada matrÃz con los tres (3) tipos de relaciones entre genes.

Como se indica en materiales y métodos se selecciona el mejor parámetro C para las máquinas de soporte vectorial con valor de 0 a 5 con intervalos de 0.5 en una prueba piloto, para las tres matrices arrojo un C óptimo igual a uno, de manera que trabajamos con este C en todas las simulaciones.

El parámetro gamma se elige de una prueba piloto y se prueban varios valores gamma cercanos al óptimo (de 0.1 a 0.000001) para ver como se comporta el kernel con parámetros diferentes.

Se observa que tanto el método lineal como el no lineal con las tres medidas de similitud tienen buen desempeño cuando cada grupo de genes tiene relaciones lineales entre

sí, todos tienen tasas promedio de genes mal clasificados tendientes a cero.

Para las matrices que tienen un grupo con relaciones lineales y otro con relaciones medio-lineales, es decir cuando se le incorpora ruido a uno de los grupos, los tres grupos tienen un buen comportamiento, aunque la tasa promedio de genes mal clasificados es menor para la máquina de soporte vectorial con el kernel basado en distanciaón Poisson. La gran diferencia se evidencia en los datos con relaciones no lineales (sinusoidales), los cuales presentan un desempeño significativamente mejor con el kernel Poisson, ya que la tasa de genes mal clasificados es de alrededor 4 % mientras que para los demás métodos es superior al 35 % (ver tabla 4-1)

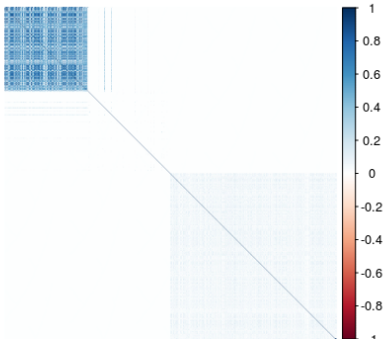
Método	Lineal-Lineal	Lineal- Medio Lineal	Lineal-No lineal
Discriminante Poisson	0	0.05285	0.3811
SVM Kernel Gaussiano	0.0771	0.01428	0.4499
SVM Kernel IM	0	0	0.4285
SVM Kernel Poisson	0	0	0.0372

Tabla 4-1.: Porcentaje de genes mal clasificados Datos Simulados 2 grupos

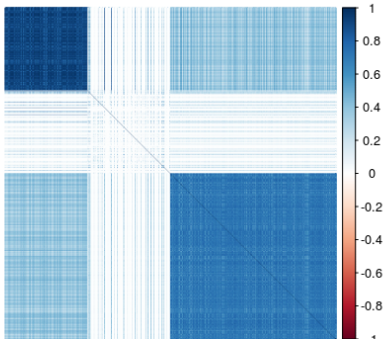
Los resultados varían significativamente entre métodos y entre kernel, cuando uno de los grupos tiene relación no lineal (sinusoidal) se ve que el kernel basado en la distancia Poisson funciona mucho mejor que los demás.

Se observa que la elección del parámetro del kernel Gaussiano hace que varíen significativamente los resultados (VER TABLA 4-1). Por ejemplo en la matriz con relación no lineal en uno de sus grupos, se tiene un porcentaje promedio de genes mal clasificados igual a cero para varios valores del parámetro (0.01, 0.001), pero al clasificar los datos de prueba con la máquina de soporte vectorial entrenada con estos mismos parámetros el porcentaje de genes mal clasificados aumenta. (Ver tabla 4-2)

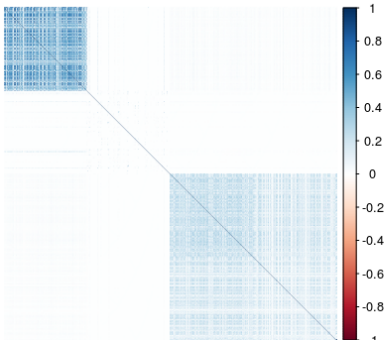
En la tabla 4.2 se observan los porcentajes de error evaluados en los datos de entrenamiento y en los datos de prueba para diferentes parámetros. Se ve claramente que el porcentaje de error en datos de entrenamiento (tercera columna tabla 4-2) es cero para la mayoría de casos, sin importar e valor del parámetro. Sin embargo no ocurre lo mismo cuando se evalúan los datos de prueba, en donde se encuentran errores de hasta 50 % usando los mismos parámetros.



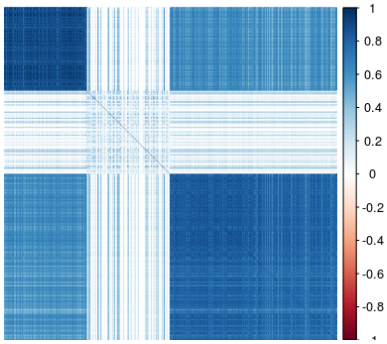
(a) Relación Lineal-Lineal($\alpha = 0{,}00001$)



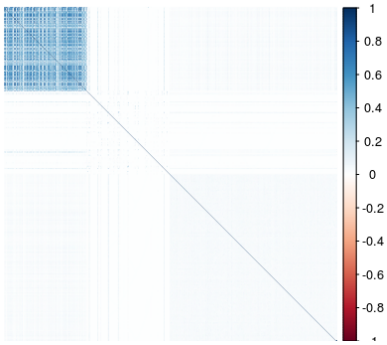
(b) Relación Lineal-Lineal($\alpha = 0{,}000001$)



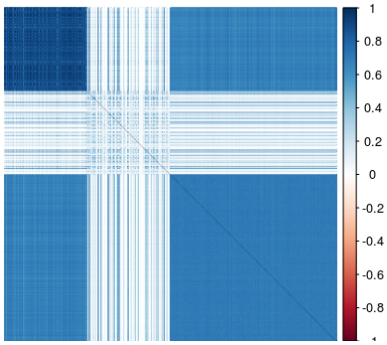
(c) Relación Lineal-medio Lineal($\alpha = 0{,}00001$)



(d) Relación Lineal-medio Lineal($\alpha = 0{,}000001$)



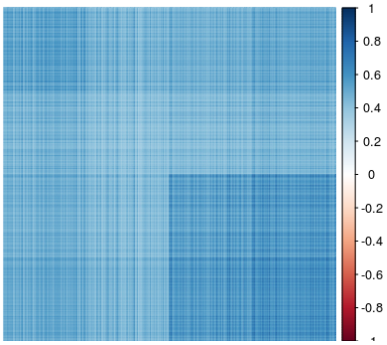
(e) Relación Lineal-no Lineal($\alpha = 0{,}01$)



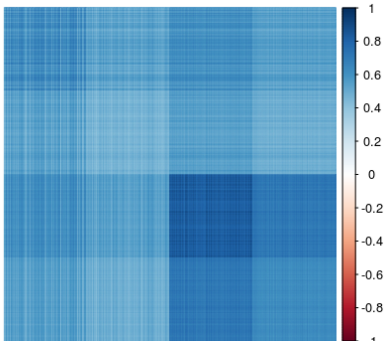
(f) Relación Lineal-no Lineal($\alpha = 0{,}001$)

Figura 4-5.: Correlograma de los kernel gaussianos con diferentes parámetros gamma para las matrices los tres tipos de relaciones entre sus genes

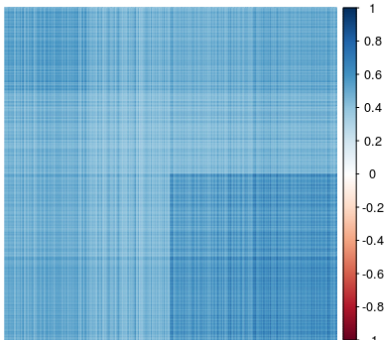
Al observar el correlograma del kernel basado en Información Mutua (Ver figura 4-6) no se diferencian claramente los grupos simulados, es decir el color no cambia significativamente en cada cuadrante, genes que pertenecen al mismo grupo y genes de diferentes grupos tienen similitudes dentro del mismo rango y lo que se espera es que las similitudes de los genes de diferente grupo tiendan a cero, mientras que las el mismo grupo tengan valores altos.



(a) Relación Lineal-Lineal



(b) Relación Lineal-medio Lineal



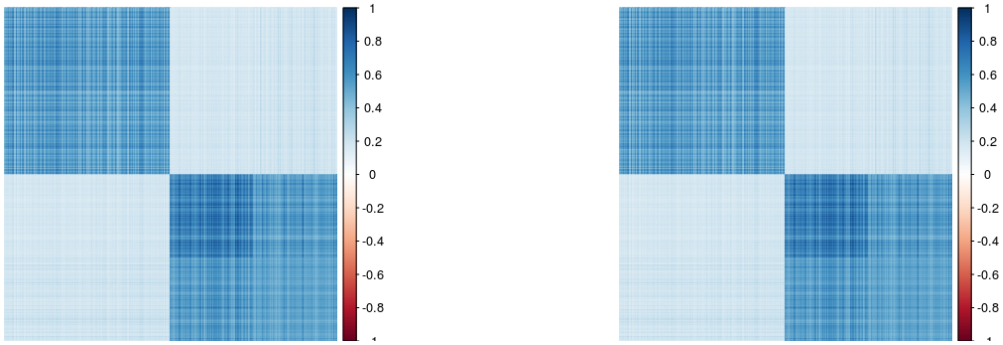
(c) Relación Lineal-no Lineal

Figura 4-6.: Correlogramas del Kernel basado en Información Mútua para las matrices con tres tipos de relaciones entre sus genes

Los correlogramas de los kernel Gaussianos se ven diferentes para distintos valores del parámetro, se evidencia que cuando se acerca a cero, las similitudes son mayores, de manera que el correlograma se ve más azul. Mientras que cuando el parámetro se acerca a uno, las similitudes son menores, y el correlograma se ve más blanco (Ver gráfica 4-5)

En los correlogramas con el kernel Gaussiano, se observa que los datos provenientes de la misma distribución presentan similitudes dentro del mismo rango, a pesar de tener relaciones de diferente tipo, es decir que más que el tipo de coexpresión que presentan los datos reconoce los datos que provienen de la distribución con los mismos parámetros. Lo que se ve reflejado por las franjas del mismo color que atraviesan los cuadrantes.

Otra cosa ocurre en los correlogramas del kernel basado en la distancia Poisson, cuyos cuadrantes 2 y 4 son azules indicando valores altos en la similitud, siendo estas las similitudes entre genes que pertenecen al mismo grupo, mientras que los cuadrantes 1 y 2 se ven blancos, indicando correlaciones bajas entre los genes de grupos diferentes.



(a) Relación Lineal-Lineal

(b) Relación Lineal-medio Lineal

(c) Relación Lineal-no Lineal

Figura 4-7.: Correlogramas del Kernel basado en Distancia Poisson para las matrices con tres tipos de relaciones entre sus genes

(ver figura 4-7). Adicionalmente no se visualiza como en el kernel Gaussiano diferencia entre los datos provenientes de diferentes distribuciones. De manera que esta medida de similitud es la que mejor identifica las relaciones entre datos(coexpresión entre genes),independiente de la distribución de la que provenga.

Finalmente, las diferencias entre el porcentaje de genes en las muestras de entrenamiento y de prueba en el kernel Gaussiano nos indica un sobreajuste de los datos de entrenamiento con algunos de los parámetros, por ejemplo, para las matrices con relaciones tipo lineal-lineal y los parámetros 0.1, 0.01, 0.001 y 0.0001 donde arroja porcentajes de genes mal clasificados en los datos de entrenamiento cercanos a cero y en los datos de prueba valores superiores al 17 %

	Parámetro	D.Entrenamiento	D. Prueba
Relación lin-lin	0.1	0	0.3585
	0.01	0	0.3571
	0.001	0	0.3371
	0.0001	0	0.1785
	0.00001	0	0.0771
	0.000001	0	0.0786
Relación lin-med. lin	0.1	0	0.4114
	0.01	0	0.3528
	0.001	0	0.2757
	0.0001	0	0.2071
	0.00001	0	0.05285
	0.000001	0	0.05285
Relación lin-no lin	0.1	0.1650	0.5000
	0.01	0	0.4507
	0.001	0	0.4499
	0.0001	0.083	0.5364
	0.00001	0.17	0.6657
	0.000001	0.17	0.6657

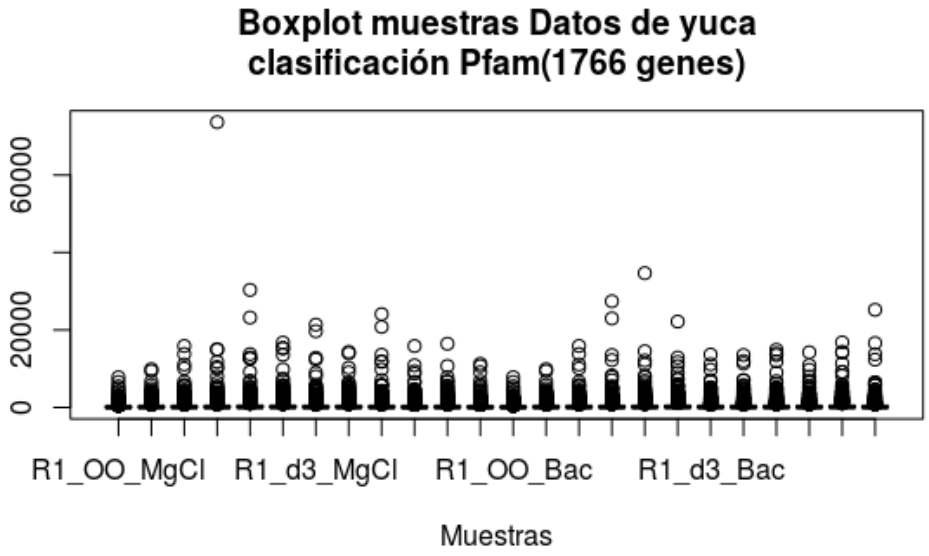
Tabla 4-2.: Porcentaje de genes mal clasificados según parámetro sigma de kernel Gaussiano

4.0.2. Resultados en Datos Reales

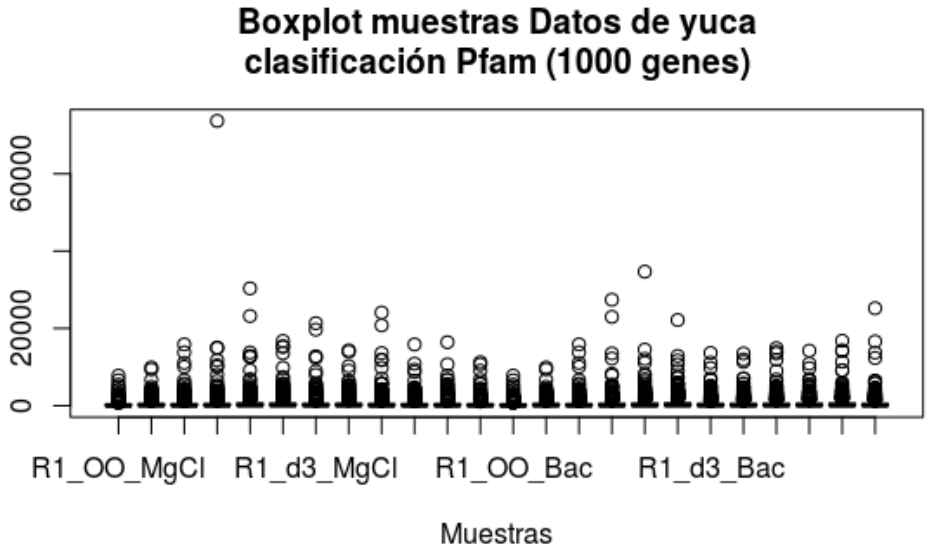
Tenemos dos matrices con datos reales para clasificación provenientes de datos de yuca (ver materiales y métodos), la primera matriz con cuatro (4) tipos de genes provenientes de la clasificación Pfam (matriz de 1527x24) y la segunda con cuatro (4) tipos de genes de la clasificación PANTHER (matriz de 865x24).Adicionalmente, de la misma matriz de clasificación Pfam, se seleccionan 250 genes aleatoriamente de cada grupo generando una matriz de 1000x24, con el mismo número de genes por grupo, para tener la misma estructura que con datos simulados.

Se verifica la estructura de los datos de las tres matrices, para lo cual se grafican los boxplot de las muestras y un ejemplo del histograma de una muestra de los datos de yuca. (Ver gráficas de 4-8 a 4-11)

Al comparar los boxplot y los histogramas de las muestras se verifica que tienen un comportamiento similar a los datos simulados. De manera que repetimos el procedimiento de clasificación que se realizó con los datos simulados con los dos métodos elegidos y los tres kernel.



(a) Muestras de Yuca clasificación Pfam (1766 genes)



(b) Muestras de Yuca clasificación Pfam (1000 genes)

Figura 4-8.: Boxplot muestras de yuca clasificación Pfam

Para seleccionar los parámetros en las máquinas de soporte vectorial, se selecciona el que minimiza el error de clasificación en los datos de entrenamiento. Se combinan valores del parámetro gamma (de 0.5 a 5, con intervalos de 0.5) y del parámetro de costo(de 0.5 a 10 con intervalos de 0.5), para el caso del kernel Gaussiano. Mientras

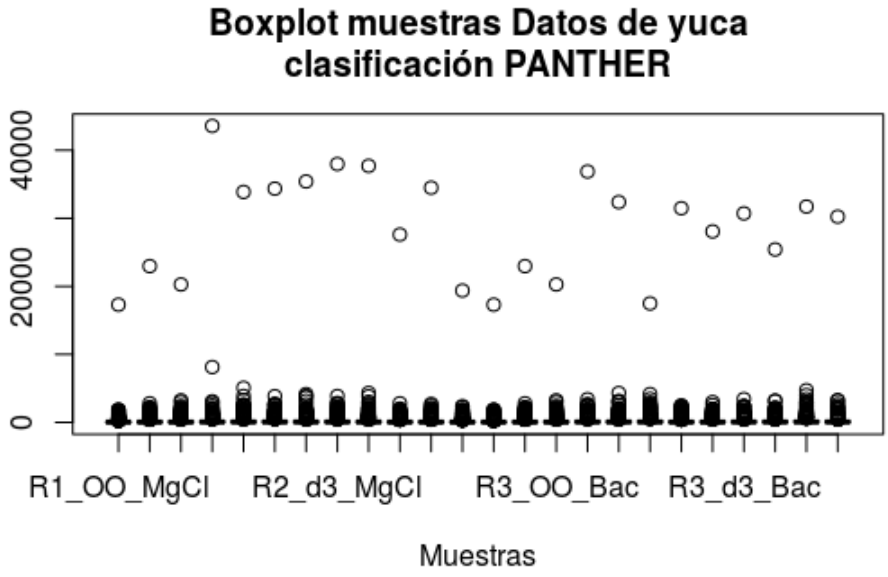


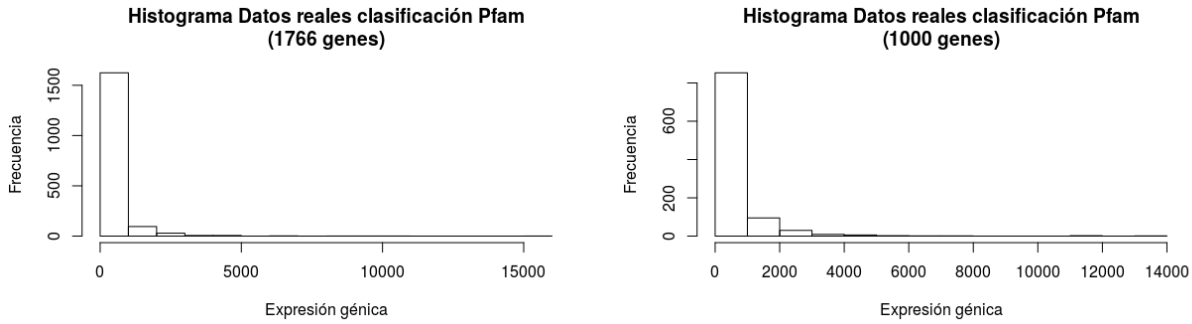
Figura 4-9.: Boxplot muestras de yuca clasificación PANTHER

	Pfam	**Pfam**	**PANTHER**
Método	Matriz 1766 X 24	Matriz 1000 x 24	Matriz PANTHER
K.Gauss. gamma	4.5	5	1
K.Gauss. C	2	5	3
K.IM	1	1	0.5
K.Poisson	0.5	1	0.02

Tabla 4-3.: Parámetros para los tres tipos de kernel en datos reales

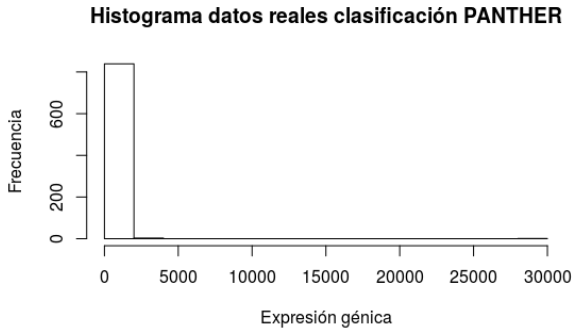
que para el kernel basado en información mutua y en la distancia Poisson, se selecciona entre los valores de 0 a 10 con intervalos de 0.5 encontrando el parámetro C. Después de obtener el mejor valor en los intevalos propuestos, se eligen intervalos más pequeños (Ver tabla 4-3).

Con los parámetros seleccionados se realiza la clasificación con los diferentes métodos obteniendo el porcentaje de genes mal clasificados en los datos de prueba (Ver tabla 4-4). Los resultados nuevamente apuntan a que el kernel basado en la distancia Poisson mejora el rendimiento de la máquina de soporte vectorial para datos RNAseq y a la vez presenta mejor clasificación que la realizada por el discriminante lineal Poisson para las tres matrices de datos reales.



(a) Histograma ejemplo muestra(1766 genes) (b) Histograma ejemplo muestra (1000 genes)

Figura 4-10.: Histogramas muestra de datos de yuca clasificación Pfam



(a) Histograma ejemplo muestra (1000 genes)

Figura 4-11.: Histogramas muestra de datos de yuca clasificación PANTHER

Método	Matriz 1766x24	Matriz 1000x24	Matriz PANTHER
Discr.Poisson	0.5283	0.5114	0.5144
SVM K.Gaussiano	0.4797	0.4957	0.5093
SVM K.IM	0.4314	0.4778	0.4556
SVM K.Poisson	0.3796	0.3900	0.3918

Tabla 4-4.: Porcentaje de genes mal clasificados en Datos de Yuca con categorías Pfam

5. Capítulo 5. Discusión

De acuerdo con los resultados en datos de conteo RNAseq provenientes de matrices simuladas y reales, donde la mejor clasificación se obtuvo por la máquina de soporte vectorial con el kernel basado en distancia Poisson, se deduce que este kernel reconoce patrones más complejos, de naturaleza no lineal entre genes basado en datos de conteo RNAseq.

El kernel basado en la distancia Poisson parece adecuado para datos de conteo ya que reconoce su naturaleza evitando la normalizacián que como se ha mencionado en distintos trabajos [10, 14] influye en los resultados.

Tal como ocurre con el trabajo de López-Kleine, et.al. (2012) [27] donde se comparan diversos métodos lineales y no lineales para datos de microarreglos, se obtienen mejores resultados en la predicción funcional de genes con los métodos de naturaleza no lineal. Los resultados son en el mismo sentido, ya que los tipos de relaciones entre genes son los mismos aunque sean medidos mediante técnicas diferentes, microarreglos o RNAseq y al adaptar los métodos a la naturaleza de los datos, esperamos resultados similares.

Al trabajar con datos RNAseq y métodos adaptados a la naturaleza de estos datos se observa que el poder predictivo de la máquina de soporte vectorial supera al análisis discriminante lineal Poisson, en el caso en el que se usa el kernel basado en la distancia Poisson.

La diferencia entre el porcentaje de genes clasificados erróneamente con los distintos tipos de kernel y con los distintos parámetros en los datos simulados evidencia como se ha escrito en diferentes artículos [5, 35], la importancia de la selección del kernel y de los parámetros en los resultados obtenidos con las máquinas de soporte vectorial.

El éxito en la clasificación con el método máquina de soporte vectorial está ligado a que las observaciones sean separables linealmente en el espacio de caracterstíicas que se encuentra en el espacio de Hilbert. De manera que la selección de un kernel adecuado a los datos se hace indispensable para conseguir una buena predicción. El kernel

propuesto basado en la distancia Poisson mejora significativamente el desempeño de la máquina en relación con el kernel Gaussiano usado en varios trabajos de análisis de RNAseq [16, 12, 2, 51], que parece no ser adecuado para datos de conteo con distribuciones Poisson o binomial negativa.

Aunque los resultados de la máquina de soporte vectorial con el kernel basado en información ḿutua también presenta mejores resultados que con el kernel Gaussiano, el porcentaje de error de clasificación es similar en ambos casos, es decir que no mejora sustancialmente el desempeño del método.

El Kernel basado en distancia Poisson parece reconocer mejor las relaciones entre genes que los otros kernel, ya que encuentra similitudes altas para los genes que pertenecen al mismo grupo y similitudes bajas para genes de diferentes grupos. Esta diferencia entre las similitudes no es tan clara en el kernel basado en Información mutua ni en el kernel Gaussiano.

La selección del parámetro gamma influye de gran manera en el desempeño de la máquina de soporte vectorial y puede generar un sobreajuste de los datos de entrenamiento, lo que lleva a porcentajes altos de error en la clasificación en los datos de prueba aunque en los datos de entrenamiento se tengan niveles de error cercanos a cero.

6. Conclusiones y recomendaciones

6.1. Conclusiones

La mejor clasificación para datos simulados y reales provenientes de datos de conteo, fue la obtenida por la máquina de soporte vectorial con el kernel basado en distancia Poisson, de manera que se concluye que este kernel reconoce patrones más complejos, de naturaleza no lineal entre genes y parece más adecuado como medida de similitud para el caso de los datos de esta naturaleza y mejora el desempeño en la clasificación de genes usando el método mencionado.

El desempeño de las máquinas de soporte vectorial depende en gran medida del tipo de kernel usado, de manera que debe elegirse un kernel teniendo en cuenta las características y la naturaleza de los datos en estudio. Para nuestro caso usar un kernel adaptado a datos de conteo mejoró significativamente los resultados.

Trabajar sobre los datos crudos, sin normalizar parece mejorar el desempeño de los clasificadores, ya que se ha probado que el método de normalización usado afecta los resultados [10, 14, 6]. Nuestro kernel basado en la distancia Poisson tiene en cuenta la profundidad de secuenciación en su cálculo, pero no realiza modificaciones sobre los datos, lo que evita la introducción de nuevos sesgos, como sucede cuando se normaliza, por ejemplo mediante el método RPKM [10]

El kernel basado en la distancia Poisson reconoce patrones de mayor complejidad, sin afectar sus resultados por datos provenientes de distribuciones con difrentes parámetros. De manera que esta medida de similitud es la que mejor identifica las relaciones entre datos(coexpresión entre genes), y parece adecuada en la predicción funcional de genes.

Aunque el kernel basado en Información Mútua mejora el desempeño de la máquina de soporte vectorial, no presenta valores sustancialmente diferentes en las similitudes de los genes que pertenecen a un grupo en relación a los que no pertenecen al mismo grupo. Adicionalmente el porcentaje de genes mal clasificados es mayor que con el kernel

basado en distancia Poisson. De manera que no se recomienda en el análisis funcional de genes.

Las máquinas de soporte vectorial tienen la ventaja que no tiene supuestos sobre los datos, cómo si los tienen los métodos paramétricos como el discriminante lineal Poisson, que supone independencia [9] (de muestras en nuestro caso) que pueden no ser realistas.

6.2. Recomendaciones

Como trabajo futuro, se puede definir una medida de similitud para la distribución binomial negativa mediante el procedimiento usado para la distancia Poisson, es decir hallar la razón de verosimilitud o un aproximado ligado a la prueba de hipótesis sobre diferencias entre observaciones.

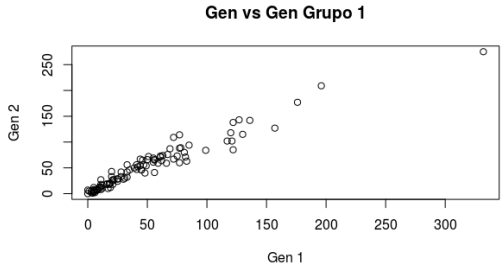
Con el fin de comprobar el poder del kernel propuesto, pueden utilizarse otros métodos de aprendizaje de máquina con otros enfoques como al análisis discriminante de Fisher con kernels.

También pueden simularse datos con otros tipos de relaciones no lineales y con mayor número de grupos para terminar de comprender el comportamiento del kernel propuesto bajo diferentes relaciones entre genes.

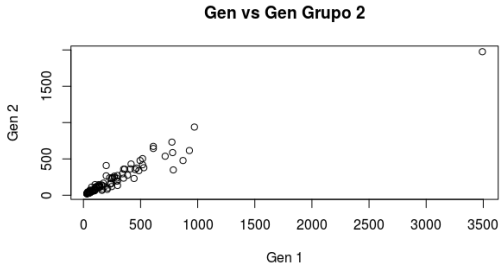
En futuros trabajos, se pueden realizar análisis basados en datos provenientes de multiples experimentos mejorando el poder pedictivo de los clasificadores mediante la incorporación de mayor información, para lo cual se hace necesario encontrar métodos que hagan posible su comparación, mediante la eliminación de sesgos debido a las diferencias en el diseño experimiental y las condiciones propias de cada experimiento.

A. Anexo: Gráficas gen vs gen

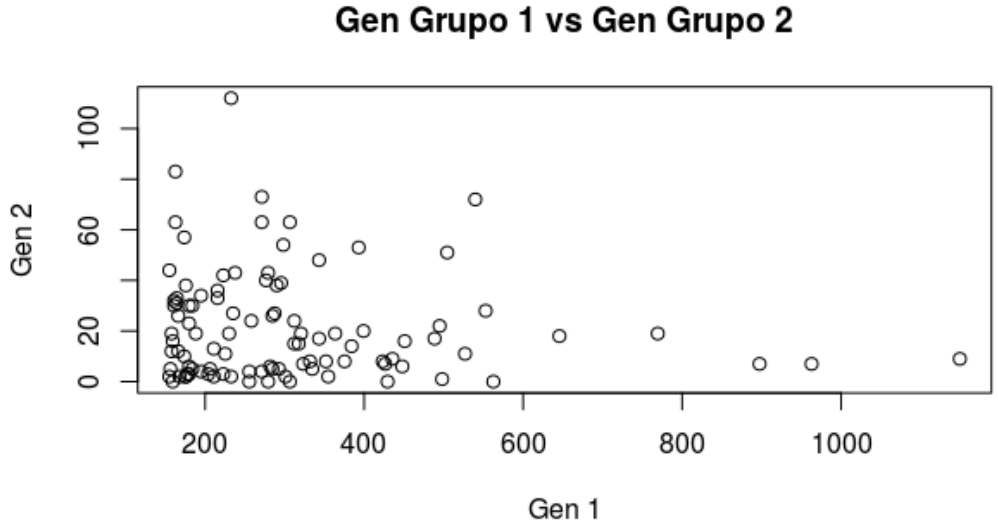
En este apéndice se muestran las relaciones gen vs gen de los grupos simulados en cada matriz y las relaciones de los genes que no pertenecen al mismo grupo



(a) Relación Gen vs Gen Grupo 1

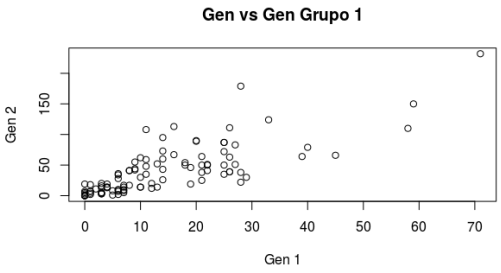


(b) Relación Gen vs Gen Grupo 2

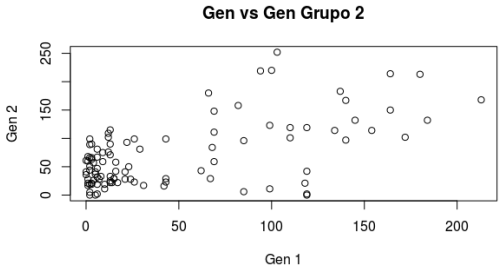


(c) Relación Gen Grupo 1 vs Gen Grupo 2

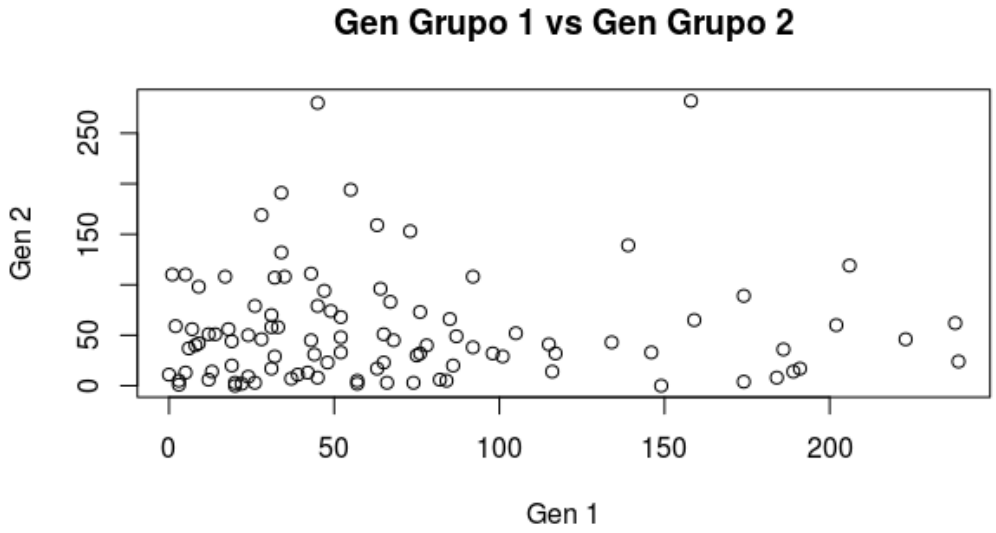
Figura A-1.: Ejemplo del tipo de relación en cada grupo de genes coexpresados. Matriz con relación lineal-lineal



(a) Relación Gen vs Gen Grupo 1

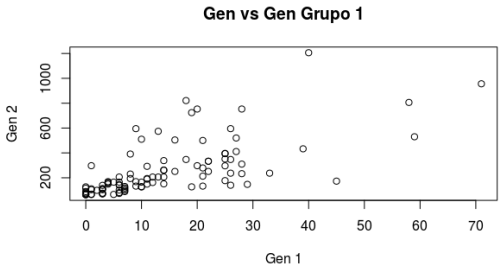


(b) Relación Gen vs Gen Grupo 2

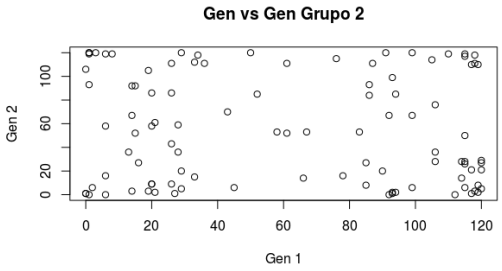


(c) Relación Gen Grupo 1 vs Gen Grupo 2

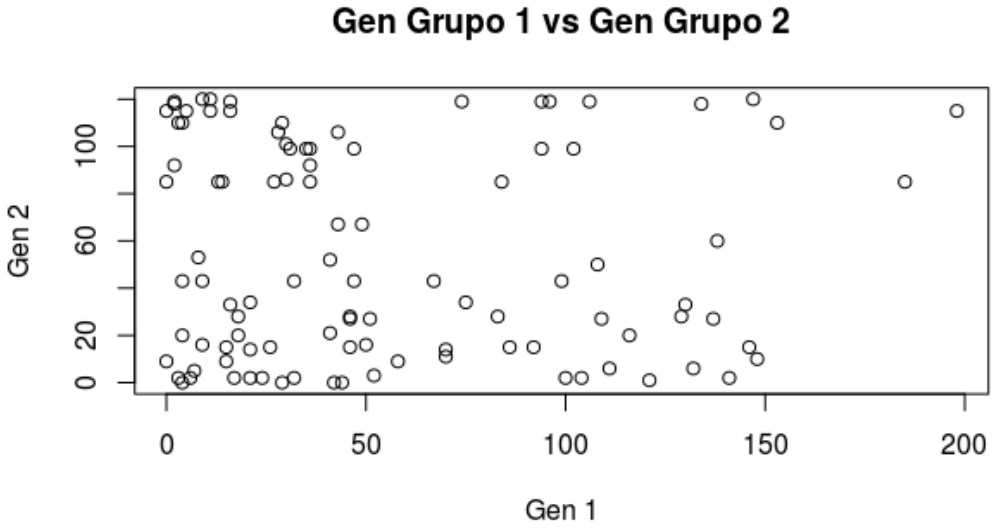
Figura A-2.: Ejemplo del tipo de relación en cada grupo de genes coexpresados. Matriz con relación lineal-medio lineal



(a) Relación Gen vs Gen Grupo 1



(b) Relación Gen vs Gen Grupo 2



(c) Relación Gen Grupo 1 vs Gen Grupo 2

Figura A-3.: Ejemplo del tipo de relación en cada grupo de genes coexpresados. Matriz con relación lineal-no lineal

B. Anexo: Código en R

CÓDIGO PARA GENERACIÓN DE MATRICES SIMULADAS

Generar Matriz lineal-lineal

```
library("PoiClaClu")
##Simulacion Primer grupo de genes con relaciones lineales
dat0<- CountDataSet(n=250,p=100,sdsignal=0.03,K=1,param=runif(1, 80,120))
dat2= (dat0$x)^runif(1, 1.2, 1.5) + round(runif(250, 20, 250),0)

##Simulacion segundo grupo de genes con relaciones lineales
set.seed(10)
dat1<- CountDataSet(n=250,p=100,sdsignal=1.2,K=1,param=runif(1,40,60))
set.seed(10)
dat1_a<-CountDataSet(n=250,p=100,sdsignal=0.9,K=1,param=runif(1,2,22))

#Matriz con dos grupos de relaciones lineales
ad=rbind(dat0$x, dat2$x, dat2$x)
```

Matriz lineal-medio lineal

```
##Simulacion genes grupo 1 con ruido
(con relaciones medio lineales)

library("PoiClaClu")
set.seed(10)
dat1<-CountDataSet(n=250,p=100,sdsignal=1.2,K=1,param=runif(1,40,60))
set.seed(10)
dat1_a<- CountDataSet(n=250,p=100,sdsignal=0.9,K=1,param=runif(1,2,22))
dat=rbind(dat1$x, dat1_a$x)
```

```
#generacion de ruido para 500 genes
ruido1<-matrix(nrow = 500, ncol = 20)
for(i in 1:nrow(ruido1))
{ruido1[i, 1:20] <-round(runif(20, 60, 180),0)}
d1=60*sin(dat[1:500, 1:20])+60
d1=round(d1,0)
d2=dat[1:500, 81:100]+ruido1

#primer grupo de genes, con relacion medio lineal
datos1 = cbind(d1, dat[1:500, 21:80], d2)

####Genes grupo 2 con relaciones lineales

set.seed(10)
dat0<- CountDataSet(n=250,p=100,sdsignal=0.03,K=1,
param=runif(1,80,120))
dat2= (dat0$x)^runif(1, 1.2, 1.5) + round(runif(250, 20, 250),0)
##segunda matriz con un grupo de genes con de relacion lineal
y un segundo grupo de genes con relacion medio-lineal
ad=rbind(datos1, dat0$x, dat2)
```

Matriz lineal-no lineal

```
#Simulacion grupo 1 de genes (relaciones no lineales)

library("PoiClaClu")
set.seed(10)
dat0<- CountDataSet(n=250,p=100,sdsignal=0.03,K=1,param=runif(1,80,120))
dat2= (dat0$x)^runif(1, 1.2, 1.5) + round(runif(250, 20, 250),0)
dat=rbind(dat1$x, dat2) #generacion grupo de genes con relaciones lineales

##Simulacion grupo 2 de genes (relaciones no lineales)

np=500
gen1_100<-matrix(nrow = np, ncol = 100)
for(i in 1:250) {gen1_100[i, 1:100] <-rnbinom(100, mu=25, size=1)}
for(i in 251:500) {gen1_100[i, 1:100] <-rnbinom(100, mu=45, size=3)}
gen25_100=60*sin(gen1_100)+60
```

```
gen25_100=round(gen25_100,0) #grupo de genes con relaciones no lineales
ad=rbind(dat, gen25_100) #matriz relacion lineal-no lineal
```

Código para generar kernel

```
##Funcion para generar la matriz de IM despues de discretizar
IMutua<-function(m1)
{
  IMutua<-matrix(nrow = 1000, ncol = 1000)
  for (j in 1:ncol(IMutua))
    for(i in 1:nrow(IMutua)) {{IMutua[i, j] <- mi.empirical
    (data.frame(m1[j,1:100],m1[i,1:100]))}}
  return(IMutua)
}

library(entropy) #Paquete para calcular a matriz de IM

library("PoiClaClu") #Paquete para calcular distancia Poisson
y clasificar mediante Discriminante Lineal

library("e1071") #Paquete para la clasificacion con svm

ad: Matriz de datos n:genes x p:muestras

IM1=IMutua(ad) #Calcula matriz de IM
RIM=(IM1)^(1/2) #Saca la raiz cuadrada de IM

d <- PoissonDistance(m1,type="mle") #calculo distancia Poisson
dp=as.matrix(d$dd) #Matriz de distancia Poisson

grupofin=rep(c(1,2), c(500, 500)) #Clase de cada gen
```

Código para seleccionar datos de entrenamiento y de prueba

```
ind30=round(runif(300, 1,1000),0)
```

```
tindex <- ind30 #seleccion datos de prueba

xtrain <- m1[tindex,] #30% de D.Entrenamiento
xtest <- m1[-tindex,] #70% de D.Prueba
xtest_RIM <- RIM[-tindex,] #70% de D.Prueba (Raiz IM)
xtrain_RIM <- RIM[tindex,] #30% de D.Entrenamiento (Raiz IM)
xtest_dp <- dp[-tindex,] #70% de D.Prueba (Dist. Poison)
xtrain_dp <- dp[tindex,] #30% de D.Entrenamiento (Dist. Poisson)
ytest <- grupofin[-tindex] #70% Clase D.Prueba
ytrain <- grupofin[tindex] #30% clase D.Entrenamiento
```

Código para realizar las clasificaciones con los metodos propuestos

```
##Clasificacion con svm kernel kaussiano

library("e1071") #libreria para realizar svm
p="Valor del parametro" # donde p=0.1-0.01-0.001-0.0001-0.00001
svp.01 <- svm (xtrain, ytrain, kernel = "radial" ,
type="C-classification", gamma=p) #clasificacion mediante
svm con k. gaussiano
ypred.01 <- predict(svp.01, xtrain) #prediccion en D.Entrenamiento
svm_kg.01=table(ytrain, ypred.01) #matriz de confusion D.Entrenamiento

fdr_kg.1=(sum(svm_kg.01)-sum(diag(svm_kg.01)))/(sum(svm_kg.01))
fdr_kg.1 #porcentaje de genes mal clasificados en D.Entrenamiento

ypred01t <- predict(svp1, xtest) #prediccion datos prueba
svm_kg.01.t=table(ytest,ypred01t) #matriz de confusion D.Prueba
fdr_svm_kg.1.t=(sum(svm_kg.01.t)-sum(diag(svm_kg.01.t)))/
(sum(svm_kg.01.t))
fdr_svm_kg.1.t #porcentaje de genes mal clasificados en d.rueba

#Clasificacion con svm kernel IM

svp2 <- svm (xtrain_RIM, ytrain, kernel = "linear",
type="C-classification") #Clasificacion K. lineal con Raiz de IM
ypred2 = predict(svp2,xtest_RIM) #Prediccion D. Prueba K. lineal
con Raiz de IM
```

```
svm_kRIM=table(ytest,ypred2) #matriz de confusion
fdr_svm.kRIM=(sum(svm_kRIM)-sum(diag(svm_kRIM)))/(sum(svm_kRIM))
fdr_svm.kRIM  #porcentaje de genes mal clasificados
en datos de prueba

##Clasificacion con svm kernel Poisson

svp3 <- svm (xtrain_dp, ytrain, kernel = "linear", ,
type="C-classification") #Clasificacion K. lineal con Dist.Poisson
ypred3 = predict(svp3,xtest_dp) #Prediccion D.Prueba  K. lineal con
Dist.Poisson
svm_k.dP=table(ytest,ypred3) #matriz de confusion
fdr_svm_k.dP=(sum(svm_k.dP)-sum(diag(svm_k.dP)))/(sum(svm_k.dP))
fdr_svm_k.dP # porcentaje de genes mal clasificados en D.Prueba

##Clasificacion con Discriminante Poisson

d.poisson <- Classify.cv(xtrain,ytrain) #Realiza Disc.Lineal
de Poisson
out <- Classify(xtrain,ytrain, xte=xtest, rho=cv.out$bestrho)
#Clasifica los datos de prueba
table_dP=(table(out$ytehat,ytest)) #matriz de confusion
fdr_dP=(sum(table_dP)-sum(diag(table_dP)))/(sum(table_dP))
fdr_dP #porcentaje de genes mal clasificados en datos de prueba
```

Bibliografía

[1] Agresti, A: *Categorical Data Analisis*. Hoboken, NJ : Wiley, 2002

[2] Alshamian. Hala, Alohali Y.: mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling. En: *Biomed Res Int* 15 (2015)

[3] Arianne. Richard, Peters. James Biasci. Daniele Flint. Shaun M Lee. James C McKinney. Eoin F Siegel. Richard M. ; Kenneth, Smith.: Comparison of gene expression microarray data with count-based RNA measurements informs microarray interpretation. En: *BMC Genomics* 15 (2014)

[4] Azuaje. Francisco J, Zheng. Huiru Léonard. Fréderique Magali Rolland-Turner Lu Zhang Yvan Devaux ynd Wagner Daniel R: Predictive integration of gene functional similarity and co-expression defines treatment response of endothelial progenitor cells. En: *BMC Systems Biology* 5 (2011)

[5] B. Chen, H. L. ; Bao, Z.: A kernel optimization method based on the localized kernel fisher criterion. En: *IEICE TRANS. INF. SYST* E93 (2010), p. 3352–3358

[6] Bullard James, Hansen Kasper y Dudoit S.: Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. En: *BMC Bioinformatics* 11 (2010), p. 94–105

[7] Conesa Ana, Tarazona Sonia Gomez-Cabrero David Cervera Alejandra-McPherson Andrew Wojciech Szczesniak Michal Gaffney Daniel J Elo Laura Zhang Xuegong y Ali M.: A survey of best practices for RNA-seq data analysis. En: *Genome Biology* 17 (2016)

[8] Cortes. C, V: Support vector networks. En: *Machine Learning* 20 (1995), p. 273–297

[9] Daniela, M W.: Classification and clustering of sequencing data using a Poisson model. En: *The Annals of Applied Statistics* 5 (2011), p. 2493–2518

[10] Datta. S, D: *Statistical Analysis of Next Generation Sequencing Data*. primera. New York : Springer, 2014

[11] Díaz, Luis G.: *Análisis de Datos Multivariados*. Bogotá, Colombia : Universidad Nacional de Colombia, 2016

[12] Firoozbakht. F, Porter. L Rueda. L.: Breast cancer subtype identification using machine learning techniques. En: *4th International Conference on Computational Advances in Bio and Medical Sciences* (2014)

[13] Francisco, García: Métodos de análisis de enriquecimiento funcional en estudios genómicos / Universidad de Valencia. 2016. – Informe de Investigación. – 191–204 p.

[14] Franck Rapaport, Yupu Liang Mono Pirun Azra Krek Paul Zumbo Christopher E Mason Nicholas D Socci y Doron B.: Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. En: *Genome Biology* 14 (2013)

[15] Fuente, Juan José de la: *La información mutua como medida de asociación y su utilidad en análisis genéticos*, Universitat Politéica de Catalunya, Tesis de Grado, 2011

[16] Gharibi. A, Vart. A Mohebian M.: Identification of gene signatures for classifying of breast cancer subtypes using protein interaction database and support vector machines. En: *5th International Conference on Computer and Knowledge Engineering* (2015), p. 195–200

[17] Gibaja, Juan J.: *Aprendizaje estadístico con funciones kernel*, Universidad Nacional de Colombia, Tesis de Grado, 2010

[18] Goonesekere Nalin, OConnor K.: The challenge of annotating protein sequences: The tale of eight domains of unknown function in Pfam. En: *Computational Biology and Chemistry* 34 (2010), p. 210–214

[19] H, Kitano.: Computational systems biology. En: *Nature* 420 (2002), p. 206–216

[20] Hannemann. Juliane, Halfwerk. Johannes BG Bas Kreike Peterse. Johannes y Van J M.: Classification of ductal carcinoma in situ by gene expression profiling. En: *Breast Cancer Research* 8 (2006)

[21] Hanson. Andrew D, Anne Waller. Jeffrey C Valerie de Crecy-Lagard: Unknown proteins and orphan enzymes: the missing half of the engineering parts list and how to find it. En: *Biochemical Journal* 425 (2010)

[22] Hausser, Jean ; Strimmer, Korbinian: *entropy: Estimation of Entropy, Mutual Information and Related Quantities*, 2014. – R package version 1.2.1

[23] Huang. Desheng, He. Miao 4. ; Baosen, Desheng.: Comparison of linear discriminant analysis methods for the classification of cancer based on gene expression data. En: *Journal of experimental Clinical Cancer Research* 28 (2009)

[24] Johanna Hardin, Leanne Hicks y Brian V.: A robust measure of correlation between two genes on a microarray. En: *BMC Bioinformatics* 8 (2007)

[25] Karatzoglou, Alexandros ; Smola, Alex ; Hornik, Kurt ; Zeileis, Achim: kernlab – An S4 Package for Kernel Methods in R. En: *Journal of Statistical Software* 11 (2004), Nr. 9, p. 1–20

[26] Soto. J y López, 2012: RNAseq: Herramienta tránscriptómica útil para el estudio de interacciones planta-patógeno. En: *Fotosanidad* 15 (2012), Nr. 2, p. 101–113

[27] López-Kleine Liliana, Tejedor H. Fabio Gordillo Luz A.: Virulence factor prediction in Streptococcus pyogenes using classification and clustering based on microarray data. En: *Appl Microbiol Biotechnol* 93 (2012)

[28] Maníquez, Cid Luis y Valdez M.: Test de Razón de verolimilitud generalizada / Congreso Latinoamericano de Sociedades de Estadística. Octubre, 2012. – Informe de Investigación. – 17 p.

[29] Mark Burton, Qihua Tan y Torben A. K.: Gene Expression Profiles for Predicting Metastasis in Breast Cancer: A Cross-Study Comparison of Classification Methods. En: *The Scientific World Journal* 2012 (2012)

[30] Martínez Gómez. Mónica, Marí Benlloch. M.: La distribución Poisson / Universidad Politécnica de Valencia. 2014. – Informe de Investigación. – 9 p.

[31] Martínez Ruedas, Cristina: *Detector Multiusuario para DS-CDMA basado en SVM*, Universidad de Sevilla, Tesis de Grado, 2006

[32] Meyer, David ; Dimitriadou, Evgenia ; Hornik, Kurt ; Weingessel, Andreas ; Leisch, Friedrich: *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2015. – R package version 1.6-7

[33] Meyer, Patrick E.: *infotheo: Information-Theoretic Measures*, 2014. – R package version 1.2.0

[34] Morozova. O, M ; Marra, M.A: Application of new sequencing technologies for trascriptme analysis. En: *Annu. Rev. Genomics Hum Genet* 10 (2009), p. 135–151

[35] O. Chapelle, O. B. ; Mukherjee, S.: Choosing multiple parameters for support vector machines. En: *Mach. Learn* 46 (2002), p. 131–159

[36] Robinson MD, Oshlack A.: A scaling normalization method for differential expression analysis RNAseq data. En: *Genome Biol* 11 (2010), p. R25

[37] Scholkopf. B, Smola. A.: *Learning with Kernels. Support vector machines, regularization, optimization and beyond.* primera. MIT : Massachusetts Institute of Technology, 2002

[38] Scholkopf. B, Vert. J.: *Kernel methods in computational biology.* primera. MIT : Massachusetts Institute of Technology, 2004

[39] Scholkopf. B, Vert. J.: *Multidimensional Density Estimation*. 23. Amsterdam, Elsevier : Handbook of Statistics, data mining and computational statistics, 2004

[40] Sehhati. M, Rabbani. H Pourhossein. M.: Stable Gene Signature Selection for Prediction of Breast Cancer Recurrence Using Joint Mutual Information. En: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12 (2015), p. 1440–1448

[41] y Shawe-Taylor J, Cristianini N.: *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, MA : MIT Press, 2002

[42] Smyth GK, Verbyla A.: A conditional approach to residual maximum likelihood estimation in generalized linear models. En: *J. R. Stat. Soc. B.* 58 (1996), p. 565â“572

[43] Somnath Datta, Dan N.: *Statistical Analysis of Next Generation Sequencing Data*. Louisville, USA : Springer, 2014

[44] Súarez, Enrique C.: Tutorial sobre Máquinas de Vectores Soporte (SVM) / Universidad Nacional de Educación a Distancia (UNED). 2014. – Informe de Investigación. – 25 p.

[45] Sven Bergmann, Jan Ihmels y Naama B.: Similarities and Differences in Genome-Wide Expression Data of Six Organisms. En: *Plos Biology* 2 (2004)

[46] Tan. Qihua, Li. Shuxia Christiansen Lene Kruse Torben A y Christensen K.: Differential and correlation analyses of microarray gene expression data in the CEPH Utah families. En: *Genomics* 92 (2013), p. 94–100

[47] Team, R C.: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2016

[48] Tibshirani. R, Narasimhan B Chu G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. En: *Proceedings of the National Academy of Science of USA* 10 (2002)

[49] Viera, Castellanos Miguel Sautié Mara í A. Zardón Navarro Martínez Ortiz Carlos Hernández Cáceres José Luis: Un acercamiento a la ontologáa de genes y sus aplicaciones. En: *Revista Cubana de Informática médica* 13 (2011)

[50] Witten, Daniela: *PoiClaClu: Classification and clustering of sequencing data based on a Poisson model*, 2013. – R package version 1.0.2

[51] Xiaoping Cheng, Yue Zhang Bo Xu y Weifeng S.: Optimal combination of feature selection and classification via local hyperplane based learning strategy. En: *BMC Bioinformatics* 16 (2015)

[52] Yang. Xu, Yang X.: Optimal Gaussian Kernel Parameter Selection for SVM Classifier. En: *Pattern Recognition* 41 (2009), p. 1098–1109

[53] Yin. Shen, Yin. J.: Tuning kernel parameters for SVM based on expected square distance ratio. En: *Information Sciences* 370-371 (2016), p. 92–102