Construyendo modelos de exones basados en árboles de decisión paralelos a los ejes

Constructing exon models based on axis-parallel decision trees

Oscar Bedoya, M. Sc.
Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Cali, Colombia
oscar.bedoya@correounivalle.edu.co

Recibido para revisión 19 de agosto de 2011, aceptado 18 de noviembre de 2011, versión final 1 de diciembre de 2011

***Resumen* — Una de las tareas que actualmente enfrentan los bioinformáticos es la construcción de modelos de exones que superen los niveles de exactitud de los ya existentes. Un modelo de exones permite clasificar una secuencia de ADN no caracterizada, ya sea como región exónica o bien como región no exónica. A pesar de los avances que se han logrado al incorporar técnicas como cadenas ocultas de Markov, redes neuronales, matrices de pesos, entre otras, este es aun un problema por explorar. En este artículo se presenta un nuevo modelo de exones construido con base en la aplicación de árboles de decisión paralelos a los ejes, que logra niveles de especificidad mayores que cualquier modelo existente, además de que permite conocer información relevante en la tarea de predicción por ser una de las técnicas que produce modelos de fácil interpretación para el experto.**

***Palabras clave* — Modelo de Exones, Predicción de Genes, Clasificación, Árboles de Decisión.**

***Abstract* — Constructing exon models is a central problem in bioinformatics. An exon model allows classifying a non characterized DNA sequence, either as being part of an exon or not. In spite of the advances that have been obtained when incorporating techniques such as Hidden Markov models, Neural networks and Weighted matrix, constructing exon models is still a problem that need to be explored. In this article a new exon model is proposed. This is constructed based on the application of axes parallel decision trees. After testing the model, it reaches specificity greater than any existing model, besides it allows knowing powerful information that is easy to understand by experts.**

***Keywords* — Exon Model, Gene Prediction, Classification, Decision Trees.**

I. INTRODUCCIÓN

El problema de la predicción de genes consiste en determinar si una secuencia de ADN no caracterizada presenta un exón o no. Esta es una tarea fundamental para los bioinformáticos debido a que son las regiones exónicas las que finalmente codifican para proteínas. Actualmente se continúa explorando la construcción de un modelo que logre identificar aquellos exones que no han podido ser descubiertos. Para la construcción de modelos de exones se han utilizado gran diversidad de técnicas [1,2,3,4,5,6,7]. En su mayoría, se utilizan modelos de Markov, tal como GenScan [8] y VEIL [9], aunque se pueden utilizar otras técnicas, tal como lo hace Grail [10] incorporando redes neuronales.

La predicción de exones es aun un problema abierto. A pesar de que existen herramientas para la predicción basadas en diferentes técnicas, es posible que para una secuencia de ADN no caracterizada, una herramienta como GenScan la clasifique como exón mientras que otra herramienta como MORGAN [11] la deseche. Uno de los enfoques que ha tomado fuerza para la construcción de modelos de exones consiste en la utilización de múltiples fuentes de evidencia [12]. Este enfoque consiste en tomar una decisión de clasificación utilizando varios criterios que ayuden a decidir si existe un exón o no. Esta estrategia difiere de la mayoría de las herramientas actuales en las que se toma la decisión con base en un solo criterio. Utilizar múltiples fuentes de evidencia se puede volver un proceso computacionalmente costoso. Si se integraran, por ejemplo, cinco criterios, la herramienta resultante tendría el inconveniente de que su tiempo de cómputo sería muy elevado, pues tendría que realizar las operaciones que los cinco criterios requieren para tomar una decisión. Se debe entonces realizar una mezcla de criterios que permita, por un lado, aumentar la exactitud de los modelos existentes, pero por otro, generar un modelo que no sea computacionalmente costoso.

En este artículo se propone la integración de criterios para la predicción de exones utilizando árboles de decisión paralelos a los ejes. Un árbol de decisión permite construir modelos de caja blanca, estos son modelos donde el experto puede ver los criterios encontrados para tomar las decisiones de clasificación. En un árbol de decisión, se tienen pruebas en cada nodo, los resultados de estas pruebas generan ramas, siendo cada una, una alternativa diferente del resultado de la prueba. Las hojas en los árboles representan las clases, por lo que el árbol se debe

Revista Avances en Sistemas e Informática, Vol.8 No. 3, diciembre de 2011 - Medellín. ISSN 1657-7663

recorrer de arriba a abajo, realizando inicialmente la prueba en el nodo raíz y siguiendo las ramas que indique el resultado de cada prueba hasta alcanzar un nodo hoja.

Los árboles paralelos a los ejes son un tipo de árboles de decisión que tienen pruebas unitarias en cada nodo. Existen también árboles oblicuos, en los que las pruebas son combinaciones lineales de los atributos. Las técnicas de árboles de decisión paralelos a los ejes resultan ser una técnica más eficiente que los árboles oblicuos ya que el problema de encontrar una combinación lineal es un problema NP mientras que al tener pruebas unitarias se trata tan solo de un problema polinomial. Así mismo, los árboles paralelos a los ejes se destacan por que son una de las técnicas de aprendizaje de máquinas más eficientes tanto para la construcción de modelos como para el uso del los mismos.

II. CONCEPTOS PRELIMINARES

A. Clasificación

La clasificación de datos es un proceso que involucra dos etapas: construcción y utilización de un modelo. En la primera, se construye un modelo que describe un conjunto predeterminado de clases, analizando los atributos de las instancias de entrenamiento. Se asume que cada instancia pertenece a una clase predefinida, la cual está determinada por uno de los atributos, llamado etiqueta de clase. En el contexto de la clasificación, las tuplas se llaman también ejemplos u objetos. Las tuplas que se analizan para construir el modelo, se denominan ejemplos de entrenamiento y forman colectivamente el conjunto de entrenamiento. Por otra parte, las tuplas que se utilizan para probar el modelo, se conocen como ejemplos de prueba. El aprendizaje del modelo es supervisado, puesto que se conoce la clase a la cual pertenece cada ejemplo de entrenamiento. Esto contrasta con el aprendizaje no supervisado, en donde las etiquetas de clase de cada ejemplo de entrenamiento y el número de clases, no se conoce, necesariamente, con anticipación.

Por lo general, el modelo aprendido tras el proceso de clasificación se representa por medio de reglas de clasificación, árboles de decisión o de una formulación matemática. Por ejemplo, dada una base de datos con información de los créditos de clientes, las reglas de clasificación se pueden encontrar para identificar clientes con un pobre o un excelente crédito. Las reglas se pueden usar para categorizar ejemplos de datos futuros, así como para proporcionar un mejor entendimiento del contenido de la base de datos.

En la segunda etapa, el modelo se usa para clasificar ejemplos nuevos. De esta manera, es posible estimar su exactitud predictiva. Existen muchos métodos para estimar este valor y en general, hacen referencia al porcentaje de ejemplos del conjunto de prueba, que son correctamente clasificados por el modelo. Para cada ejemplo de prueba, la etiqueta de clase conocida se compara con la clase predicha por el modelo para este ejemplo. Si la exactitud de un modelo se considera aceptable, éste se puede usar para clasificar futuras tuplas u objetos cuya etiqueta de clase no se conoce.

Uno de los métodos para evaluar la exactitud se basa en tablas de contingencia, tal como se muestra en la tabla 1. Una tabla de contingencia permite contrastar, por una parte, el juicio del experto, quien conoce de antemano las etiquetas de clase correctas para los casos sobre los que se realizarán las pruebas, y por otra, la decisión del clasificador.

Tabla 1. Tabla de contingencia

Etiqueta de clase c_i		*Decisión del experto*	
		SI	NO
Decisión del clasificador	SI	TP_i	FP_i
	NO	FN_i	TN_i

En una tabla de contingencia se muestra un consolidado con la siguiente información por cada clase c_i:

- Total de casos de prueba clasificados incorrectamente en la clase c_i, valor conocido como falsos positivos (FP).
- Total de casos de prueba clasificados correctamente como no pertenecientes a la clase c_i, o lo que es mismo, casos verdaderos negativos (TN).
- Total de casos de prueba clasificados correctamente en la clase c_i, es decir, verdaderos positivos (TP).
- Total de casos de prueba clasificados incorrectamente como no pertenecientes a la clase c_i, estos son, los casos falsos negativos (FN)

Utilizando la tabla de contingencia es posible obtener una medida para evaluar la exactitud α de los algoritmos de clasificación sobre cada clase c_i tal como se muestra en (1):

$$\alpha_i = \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i} \quad (1)$$

El valor α se calcula para cada clase que el modelo deba distinguir. Cada valor α permite conocer qué tan bien representada está cada clase en el modelo, es decir, indica la exactitud resultante del modelo construido para cada clase. La importancia de este parámetro está en que permite conocer qué tan acertados han sido los atributos seleccionados con respecto a la tarea de modelar cada clase.

Además del conjunto el entrenamiento es necesario llevar a cabo la selección de datos para realizar la validación y la prueba del modelo. El conjunto de validación se compone de los datos usados para optimizar los parámetros en el clasificador [13] y finalmente, el conjunto de prueba se forma con las instancias sobre las cuales se evalúa la exactitud.

Existen estrategias que orientan al experto en la forma como se debería dividir el conjunto de ejemplos para el entrenamiento, validación y prueba del clasificador. Una de las estrategias ampliamente utilizada se conoce como validación cruzada con k pliegues (k-fold cross validation) [13], que consiste en dividir el conjunto de datos en k partes y calcular k veces la exactitud del clasificador. Para cada valor de k, se considera como conjunto de entrenamiento todos los datos que no pertenecen al grupo k. La fase de prueba se realiza utilizando los datos del grupo k. Este proceso se realiza k veces, de tal manera que se obtienen k estimaciones de la exactitud. Finalmente, se calcula el promedio usando los k valores.

1) Árboles de decisión paralelos a los ejes

Un árbol de decisión paralelo a los ejes es una estructura en la cual cada nodo interno denota una prueba sobre un solo atributo, cada rama representa una salida de la prueba y los nodos hoja representan clases. Cuando se consideran atributos numéricos, las pruebas son de la forma *atributo*$\leq$ *valor*, lo que genera una partición paralela con respecto al atributo seleccionado del conjunto de entrenamiento. Para atributos categóricos, las pruebas serán de la forma *atributo*$\in b$, donde b es un subconjunto de valores posibles que puede tomar el atributo seleccionado. Para clasificar un ejemplo desconocido, los valores de los atributos del ejemplo se prueban en el árbol de decisión. Se traza un camino desde la raíz hasta el nodo hoja que indica la clase a la cual pertenece el ejemplo.

Cuando se construyen árboles de decisión, muchas de las ramas pueden reflejar ruido o valores fuera de rango existente en el conjunto de entrenamiento. Eliminar estas ramas se conoce como poda en árboles de decisión y se lleva a cabo con el propósito de mejorar la exactitud de la clasificación.

Uno de los algoritmos más usados para la construcción de árboles de decisión es C4.5 [14]. Se trata de un algoritmo voraz que construye árboles de decisión de arriba hacia abajo de manera recursiva. El atributo que hace parte de la prueba en cada nodo se selecciona según la ganancia de información, esto es, el atributo que minimice la información necesaria para clasificar los ejemplos en las particiones resultantes. La descripción completa del algoritmo se puede encontrar en [14]. La principal limitación de los árboles paralelos a los ejes está en que se pueden generar árboles muy profundos, comparados con los construidos con otras técnicas tales como los árboles oblicuos. La profundidad de un árbol determina el tiempo que toma el modelo para establecer una decisión de clasificación. Su profundidad se deriva de que en cada nodo solo puede realizar una prueba sobre un solo atributo, por lo que particionar un conjunto de entrenamiento tomará mayor cantidad de nodos que si se utilizaran, por ejemplo, pruebas mutivaluadas.

La característica principal de los árboles paralelos a los ejes es que son modelos de caja blanca en los cuales se pueden ver directamente la frecuencia de aparición de cada atributo. Además, le permite al experto conocer el atributo con mayor poder de clasificación, es decir, aquel que se localice en el nodo raíz. Obtener esta información a partir de un árbol oblicuo puede resultar complejo ya que las pruebas incluyen una mezcla de atributos.

A continuación se muestra un ejemplo de la construcción de un árbol de decisión para resolver un problema común en bioinformática, la clasificación de regiones promotoras en organismos eucariotes. Se muestra el seguimiento del algoritmo C4.5.

La clasificación de regiones promotoras se define como el problema que toma la entrada S, una secuencia de ADN, con el fin de determinar si S corresponde, o no, con una región promotora en un gen procariote. Considere los datos que se muestran en la tabla 2, estos corresponden con 10 registros que forman el conjunto de entrenamiento.

La tabla 2 muestra los valores para los 10 registros considerando tres atributos: Curvatura, Termo-estabilidad y Motivo en la región -10, además de la etiqueta de clase. La curvatura ha sido discretizada tal como lo sugiere el algoritmo.

Tabla 2: Conjunto de entrenamiento

	Curvatura	Termoestabilidad	Motivo en la región -10	Clase
1	<0.25	Media	Si	Si
2	<0.25	Media	Si	Si
3	>0.25	Alta	Si	No
4	<0.25	Baja	No	No
5	<0.25	Baja	Si	Si
6	>0.25	Alta	Si	No
7	>0.25	Baja	Si	No
8	<0.25	Baja	Si	Si
9	>0.25	Alta	Si	No
10	<0.25	Media	No	No

Inicialmente, se calcula la información necesaria esperada I para clasificar los ejemplos:

$I(s_1,s_2)=I(4,6)=-(4/10)\log_2(4/10)-(6/10)\log_2(6/10)=0.9709$

donde s_1 y s_2 corresponden con las cantidades en la clase 1 (Promotor) y en la clase 2 (No promotor).

Se calcula ahora la entropía E de cada atributo.

E(Curvatura)=0.6426

E(Termoestabilidad)=0.7578

E(Motivo)=0.7578

Se debe calcular la ganancia de información G para cada atributo y se selecciona como atributo de prueba aquel con la mayor ganancia.

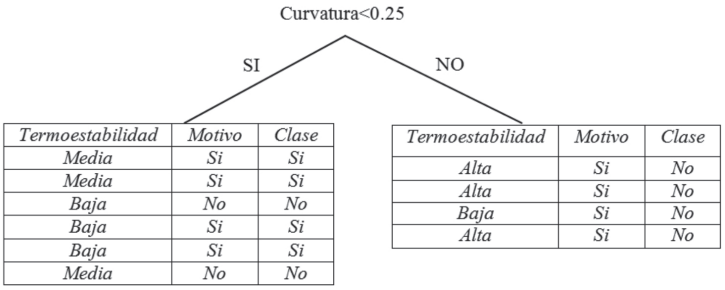
$G(Curvatura)=I(s_1,s_2)-E(Curvatura)=0.3283$

G(Termoestabilidad)=I(s1,s2)-E(Termoestabilidad)=0.2131

$G(Motivo)=I(s_1,s_2)-E(Motivo)=0.288$

Revista Avances en Sistemas e Informática, Vol.8 No. 3, diciembre de 2011 - Medellín. ISSN 1657-7663

En este caso particular, se selecciona la Curvatura como atributo de prueba por tener la mayor ganancia de información, esto genera la división del conjunto de entrenamiento que se muestra en la figura 1.

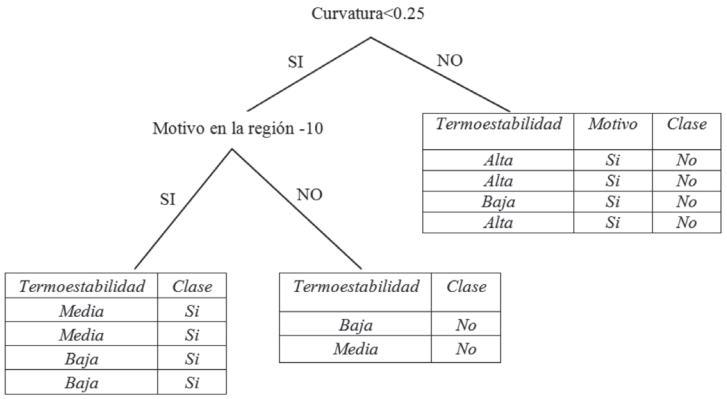


Termoestabilidad	Motivo	Clase
Media	Si	Si
Media	Si	Si
Baja	No	No
Baja	Si	Si
Baja	Si	Si
Media	No	No

Termoestabilidad	Motivo	Clase
Alta	Si	No
Alta	Si	No
Baja	Si	No
Alta	Si	No

Figura 1: Curvatura como nodo raíz

Como la clase de la división generada en la rama derecha del nodo de prueba tiene todos sus ejemplos de la clase NO, se termina la construcción del árbol sobre esta rama. Se continúa el proceso de construcción en la rama izquierda. Como se puede verificar, se deberá seleccionar como atributo de prueba a la característica Motivo en la región -10, lo que genera el árbol mostrado en la figura 2. La construcción del árbol de decisión termina debido a que los ejemplos en las hojas pertenecen a una misma clase.



Termoestabilidad	Motivo	Clase
Alta	Si	No
Alta	Si	No
Baja	Si	No
Alta	Si	No

Termoestabilidad	Clase
Media	Si
Media	Si
Baja	Si
Baja	Si

Termoestabilidad	Clase
Baja	No
Media	No

Figura 2: Construcción del árbol

B. Biología molecular

1) El ADN

El ADN (ácido desoxirribonucleico) se forma por dos cadenas de nucleótidos que se unen en espiral para dar origen a lo que conoce como la doble hélice. Los nucleótidos están constituidos por un ácido fosfórico, azúcar desoxirribosa y una base nitrogenada. Dentro de estos componentes, la base nitrogenada permite distinguir los nucleótidos de acuerdo a cuatro clases, estas son: adenina y guanina, que se conocen como las bases púricas, y citosina y timina, también llamadas bases pirimidínicas.

Los elementos en las dos hebras del ADN se relacionan por medio de puentes de hidrógeno, con la característica de que se conserva un apareamiento específico entre las bases que se unen por cada puente: una purina se une a una pirimidina, la adenina se une a la timina y la guanina se une a la citosina. La importancia que tiene el ADN se enfoca en la conservación y regulación de la información de la célula y la transmisión de esta información al replicarse.

2) Estructura del ADN

El ADN presenta una estructura compleja, es por esto que se ha clasificado de acuerdo a cuatro niveles, estos son: primario, secundario, terciario y cuaternario.

La estructura primaria hace referencia a la secuencia de nucleótidos. Se utiliza la letra inicial de las bases nitrogenadas para indicar cada nucleótido, por lo que una secuencia de ADN en su estructura primaria se formará únicamente por la combinación de las letras A, T, G y C. Además, como se tienen dos hebras pero se conoce que son complementarias, se tienen dos formas de representar la misma secuencia de ADN, esto se hace de acuerdo ala hebra sobre la cual se describan sus nucleótidos, si es en la hebra positiva, se colocan los nucleótidos en la dirección 5' a 3', esto es, la información se indica desde el extremo que tiene el ácido fosfórico hacia el extermino del átomo de carbono de la desoxirribosa. En caso de considerar la hebra que va de 3' a 5' se estará considerando la hebra negativa.

En la estructura secundaria se considera al ADN como el conjunto formado por las hebras complementarias, es decir,

la doble hélice. Por su parte, en la estructura terciaria se tiene en cuenta el superenrrollamiento que sufre el ADN para que físicamente pueda estar en el interior de la célula. En la estructura cuaternaria el ADN finalmente se presenta como cromosomas.

3) Detección y modelado de genes

El modelado de genes consiste en caracterizar la estructura de los genes. Debido a que se trata de un problema tan general, éste se ha dividido de acuerdo al tipo de organismo, procariotas y eucariotas. Para caracterizar la estructura de los genes se intenta precisar la posición de las señales que lo caracterizan, así como determinar cuáles son los elementos que se presentan comúnmente en cada tipo de gen.

III. METODOLOGÍA

En esta sección se presenta cada uno de los pasos que hicieron parte de la metodología seguida en la investigación realizada y que permitieron construir los modelos, realizar las pruebas y comparar los resultados obtenidos con los modelos de exones existentes.

A. Selección de datos

El éxito en la construcción de un modelo se debe, en gran medida, en la selección del conjunto de entrenamiento. En el caso particular de la predicción de exones, se debe proveer al conjunto de entrenamiento con ejemplos que pertenezcan tanto a la clase que representa los exones, etiquetada Exón, como la que representa a todas aquellas secuencias que no son exones, etiquetada como NoExón. Para la clase Exón se tiene la base de datos de exones-introns (http://www.meduohio.edu/bioinfo/eid/) y para la clase NoExón, se presentan varias alternativas, estas son: utilizar introns, obtenidos de la base de datos de exones-introns, utilizar promotores, obtenidos de una base de datos de promotores en organismos eucariotas (http://www.epd.isb-sib.ch/) , utilizar tanto introns como promotores o incluso cualquier otra secuencia que represente un componente diferente a un exón, entre las que se encuentran las regiones intergénicas y las colas polyA.

B. Selección de atributos

Los atributos representan los criterios que tendrá el modelo para tomar la decisión de clasificación. Como en este proyecto se lleva a cabo la integración de criterios, se han considerado tanto criterios ab-initio como basados en homologías. Los criterios ab-initio se basan en modelar cada señal característica de un gen de acuerdo a su estructura, contenido y longitud. Por otra parte, los criterios basados en homologías intentan predecir la presencia de un gen, al medir la homología que tiene la secuencia con proteínas, es decir, su similitud a nivel de bases. Se debe tener en cuenta que las proteínas son el resultado de las regiones codificantes, o exones, por lo que la presencia de un gen en una secuencia no caracterizada será más probable si ésta tiene homología con proteínas previamente identificadas.

Los atributos ab-initio considerados para la construcción de los modelos fueron: frecuencia de hexámeros, asimetría de Fickett, HMM1 (puntaje asignado por el modelo de Markov de la herramienta MORGAN), HMM2 (puntaje asignado por el modelo de Markov de GeneID) y HMM3 (puntaje asignado por el modelo de Markov de GenScan). Por otra parte, los atributos basados en homologías fueron HSP (High Scoring Segment Pairs) y PCP (Protein coding potential). Puede encontrar una descripción de estos atributos en [11].

C. Construcción de los modelos

Para la construcción de modelos de exones aplicando la técnica de árboles de decisión paralelos a los ejes se utilizó el algoritmo CRUISE [15]. Como implementación de este algoritmo, se utilizó la herramienta disponible en http://www.stat.wisc.edu/~loh/cruise.html. La entrada de datos a tal herramienta requiere listar los valores de cada uno de los siete atributos, que en este caso son numéricos, seguidos de la etiqueta de clase, de tipo categórica que tendrá dos posibles valores Exón y NoExón. Para probar la sensibilidad de la técnica con respecto al conjunto de entrenamiento, se utilizaron tres conjuntos de prueba, estos se describen en la tabla 3.

Tabla 3: Conjuntos de prueba

Conjunto de prueba	Descripción
Prueba1	Los ejemplos de entrenamiento de la clase NoExón, contienen solo promotores
Prueba2	Los ejemplos de entrenamiento de la clase NoExón contienen solo introns
Prueba3	Los ejemplos de entrenamiento de la clase NoExón, contienen tanto promotores como introns

El modelo construido utilizando el conjunto Prueba1 se muestra en la figura 3. Se tienen 900 ejemplos de entrenamiento de cada clase. La prueba en el nodo raíz es la Asimetría de Fickett. Los valores umbrales con los cuales se compara el atributo en cada nodo son determinados por el algoritmo. La rama izquierda que se deriva del nodo raíz tiene otra prueba que se evalúa sobre el atributo HMM3. En este nodo, si se toma el hijo izquierdo, se llega a una hoja, la cual está etiquetada con la clase NoExón. Este camino desde la raíz hasta el nodo hoja, se puede ver como la regla:

Si la asimetría de Fickett es menor o igual a -7.89859 y HMM3 es

menor o igual a 3,805, clasifique como NoExón.

Revista Avances en Sistemas e Informática, Vol.8 No. 3, diciembre de 2011 - Medellín. ISSN 1657-7663

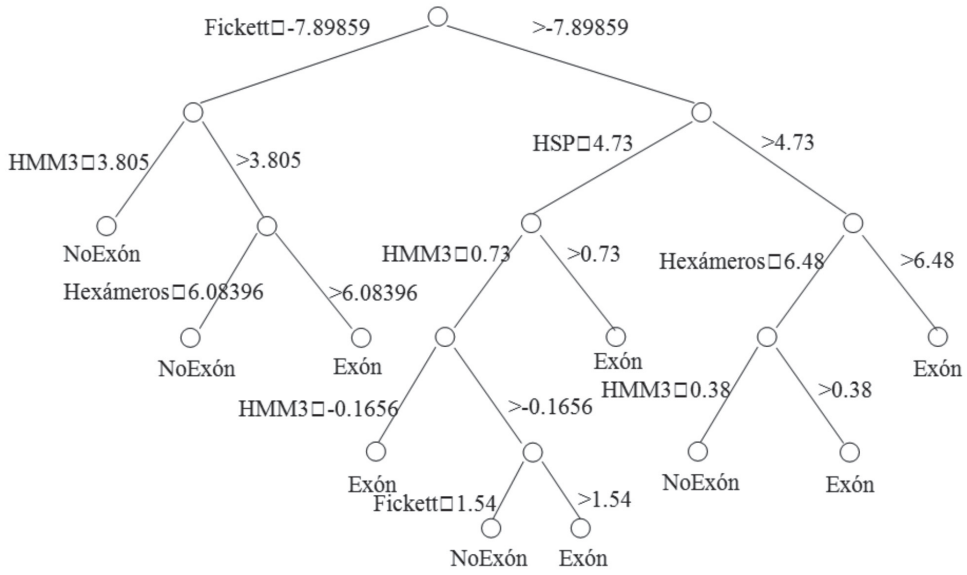


Figura 3: Modelo de exones 1

En el árbol se indica que esto sucedió en 829 ejemplos de entrenamiento y que además, hubo cinco instancias que presentaron los mismos valores para la asimetría y HMM3 pero que pertenecían a la clase Exón. Estos valores indican la cantidad de ejemplos de entrenamiento que llegan a cada hoja y sirven para conocer la pureza de las mismas, esto es, qué tan homogéneos son los ejemplos que se asignan a cada hoja.

La profundidad del árbol es cinco, lo cual indica que para tomar una decisión de clasificación, a lo sumo se tendrán que realizar cinco pruebas. Al examinar el árbol, se puede ver que existen algunos atributos que el algoritmo CRUISE no consideró, estos son: HMM1, HMM2 y PCP. Mientras que hay otros que se repiten, incluso en el mismo camino, Fickett y HMM3.

Se obtuvo un modelo diferente al considerar el conjunto Prueba2, éste se muestra en la figura 4. En este caso, el algoritmo considera en la raíz el atributo HSP. El árbol es ahora menos profundo, lo cual indica que se tendrá una decisión de clasificación más rápidamente. Además, tan solo se utilizan tres de los siete atributos seleccionados. Esto indica que el conjunto de entrenamiento es más fácil de dividir dadas las clases Exón y NoExón, cuando la etiqueta NoExón representa intrones. Este modelo muestra, además, que el modelo puede diferenciar más fácilmente un exón de un intrón que un exón de un promotor.

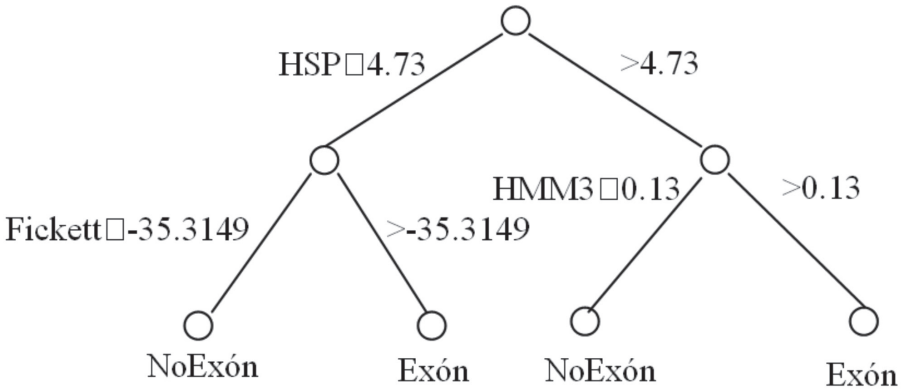


Figura 4: Modelo de exones 2

Finalmente, se construyó un modelo utilizando el conjunto Prueba3. Éste se muestra en la figura 5. En este caso, el algoritmo construye un modelo que le permite diferenciar promotores e intrones como ejemplos de la clase NoExón. El árbol obtenido tiene profundad cinco, lo que se debe principalmente a que es necesario distinguir, en el conjunto de entrenamiento, ejemplos de la clase NoExón que son promotores. Este árbol es más general que los obtenidos anteriormente, lo que significa que podrá ser utilizado eventualmente para distinguir exones tanto de promotores como de intrones.

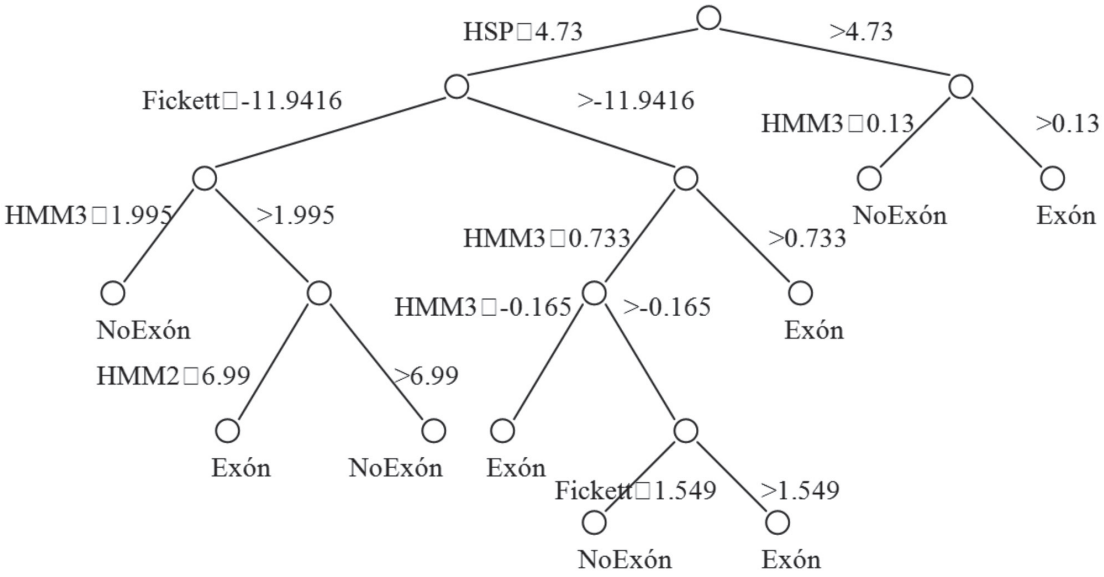


Figura 5: Modelo de exones 3

D. Selección del conjunto de prueba

Para realizar una comparación con los modelos de exones de las herramientas existentes de manera justa, se tomó el mismo conjunto de prueba que fue utilizado para probar las demás herramientas. Se trata del conjunto de datos de Burset y Guigó que contiene un conjunto de 570 secuencias de organismos eucariotas (http://www1.imim.es/datasets/genomics96/seqs/DNASequences.fasta).

E. Pruebas y evaluación

La calidad de los modelos de exones se puede calcular por medio de varios parámetros. Sin embargo, se utilizan principalmente dos parámetros para la comparación con los modelos de exones existentes, estos son: sensitividad (Sn) (2) y especificidad (Sp) (3), definidos de la siguiente manera:

$$Sn = TE/AE \quad (2)$$

$$Sp = TE/PE \quad (3)$$

donde TE (True exon) es la cantidad de exones correctamente clasificados. AE (Annotated exon) es la cantidad de exones anotados existentes en el conjunto de prueba. PE (Predicted exon) corresponde a la cantidad total de exones que el modelo predijo en el conjunto de prueba.

La exactitud obtenida con cada uno de los conjuntos de prueba se presenta en la tabla 4. Los valores α_1 y α_2 indican la exactitud para las clases Exón y NoExón, respectivamente. La exactitud de la clase Exón se mantiene en todos los modelos ya que se utilizó el mismo conjunto de exones en los tres conjunto de entrenamiento. El modelo de la clase NoExón presenta niveles de exactitud diferentes, esto se debe a que se varía el conjunto de entrenamiento que representa la clase NoExón. La exactitud para la clase NoExón fue mayor cuando se consideraron tanto intrones como promotores en el conjunto de entrenamiento.

Tabla 4: Consolidado de los resultados

	Sn	Sp	α_1	α_2
Prueba1	0.4047	0.4412	0.7294	0.3614
Prueba2	0.5241	0.5694	0.7294	0.5871
Prueba3	0.7890	0.8280	0.7294	0.7094

IV. RESULTADOS Y DISCUSIÓN

En esta sección se discuten los resultados de las pruebas realizadas y se lleva a cabo un análisis comparativo con los modelos de exones que son utilizados actualmente.

A. Análisis de los modelos

De acuerdo con los resultados obtenidos, se puede establecer que:

- Los atributos HMM1 (puntaje asignado por el modelo de Markov de MORGAN) y PCP (Protein Coding Potential) son irrelevantes en el proceso de clasificación si se utilizan en un mismo conjunto con los atributos HMM3 (puntaje asignado por el modelo de Markov de GenScan), asimetría de Fickett, HSP (High Scoring Segment Pairs), HMM2 (puntaje asignado por el modelo de Markov de GeneID) y frecuencia de hexámeros. Esto indica que los atributos HMM1 y PCP podrían no ser tenidos en cuenta en futuros proyectos de integración de criterios, si se utilizan en conjunto con los otros atributos seleccionados en este proyecto.
- Los atributos HMM3 (puntaje asignado por el modelo de Markov de GenScan), asimetría de Fickett y HSP (High Scoring Segment Pairs), presentan mayor poder de clasificación de la clase Exón que los demás. Esto indica, que son criterios más acertados para la tarea de detectar exones

y que deberían ser utilizados en otras técnicas que integren criterios. Su frecuencia de aparición en los árboles obtenidos, reflejan su alta capacidad para clasificar la clase Exón.

- Con respecto al atributo HMM3, su poder de clasificación es mayor que el todos los atributos seleccionados. En cuatro de los ocho nodos internos del árbol construido utilizando el conjunto Prueba3 se consideró como atributo de prueba.
- Los valores α_1 y α_2, indican que los modelos obtenidos representan de manera más exacta a la clase Exón que a la NoExón.

B. Análisis comparativo

En esta sección se presenta, en la tabla 5, un análisis de los modelos de exones de las herramientas para la predicción de genes que han sido ampliamente usados, incluyendo el modelo que se construyó tomando el conjunto Prueba3 que resultó ser el más exacto entre los modelos propuestos.

Con relación a los criterios de exactitud, sensitividad y especificidad, el modelo construido con árboles paralelos a los ejes considerando el conjunto Prueba3 se ubica como un mejor modelo que el de herramientas como FGENEH, Genie, MORGAN e incluso GenScan, uno de los predictores de genes más usados en la actualidad. A nivel de especificidad, supera a los modelos de las herramientas existentes. Sin embargo, no supera la sensitividad de la herramienta GeneZilla. El resultado obtenido, en cuanto a los niveles de exactitud, es justificable debido a que para la construcción de los modelos se integraron siete atributos que evidencian la presencia de exones, algunos de los cuales se basan en homologías y otros en criterios ab-initio. La exactitud se debe entonces, por una parte, en el hecho de que se cuenta con más criterios de los que consideran las herramientas de predicción existentes para decidir si una secuencia dada es un exón. Por otra parte, se debe a que los árboles de decisión permiten modelar un conjunto de poca dimensionalidad, como fue el caso al considerar siete atributos, obteniendo valores altos de exactitud para cada clase modelada.

Tabla 5: Comparación entre modelos de exones

Modelo de exones	Sensitividad a nivel de exones	Especificidad a nivel de exones
GeneZilla	0.82	0.80
GenScan	0.78	0.81
FGENEH	0.61	0.64
GeneID	0.44	0.46
Genie	0.55	0.48
GenLang	0.51	0.52
GeneParser2	0.35	0.40
GRAIL2	0.36	0.43
SORFIND	0.42	0.47
MORGAN	0.58	0.51
Modelo propuesto	0.78900	0.82810

Los modelos obtenidos no fueron mejores que el modelo de GeneZilla, con relación a la sensitividad, debido a que esta herramienta ofrece un conjunto de alternativas para detectar exones basado en su mayoría en variantes de la técnica de modelos de Markov, entre las que se encuentran MDD (Maximal Dependence Descomposition), WAM (Weight Array Matrix), WMM (Weight Matrix Model), WWAM (Windowed WAM). Además, GeneZilla permite entrenar los modelos de Markov de acuerdo al tipo de organismo que se esté tratando. Lo anterior indica que el modelo de exones de GeneZilla es altamente configurable, esto es, permite adaptar el modelo al tipo de organismo y utilizar diferentes técnicas para entrenarlo. Por otra parte, los modelos aquí propuestos se basan en integración de criterios, pero no permiten la configurabilidad de acuerdo al organismo.

Los resultados obtenidos tras la comparación permiten establecer que sería posible mejorar la exactitud de los modelos propuestos si éstos fueran adaptables al tipo de organismo. Esto implica que los modelos se construyan con base en un conjunto de entrenamiento especificado por el usuario, tal como lo hace GeneZilla.

V. CONCLUSIONES Y TRABAJO FUTURO

Utilizar árboles de decisión paralelos a los ejes para la predicción de genes, teniendo en cuenta los atributos seleccionados, permite superar los niveles de especificidad de cualquiera de las herramientas de predicción existentes. Además de aumentar los niveles de exactitud, los árboles de decisión permitieron conocer información adicional, como cuáles son los atributos que hacen parte del modelo, cuáles tienen mayor frecuencia de aparición o mayor poder de clasificación, cuántas pruebas requiere el modelo para tomar una decisión, entre otra.

Otro de los aspectos importantes tiene que ver con el futuro mejoramiento de los modelos. Los modelos obtenidos podrían aumentar su exactitud si se considera un análisis adaptable al tipo de organismo, esto es, tener un conjunto de entrenamiento propio de cada organismo y repetir el procedimiento aquí llevado a cabo. Además, se podrían seleccionar atributos que ayuden a representar de manera más acertada los ejemplos de la clase NoExón para así intentar aumentar los niveles de exactitud de los modelos propuestos.

REFERENCIAS

[1] Zhang M. Identification of protein coding regions in the human genome by quadratic discriminant analysis. Proceedings of the National Academy of Sciences of the United States of America, vol. 94, No. 2, pp. 565-568. 1997.

[2] Salzberg S., Delcher A. Kasif S. and White O. Microbial gene identification using interpolated Markov models. Nucleic Acids Research, vol. 26, No. 2, pp. 544-548. 1998.

[3] Borodovsky M. and McIninch J. GeneMark: parallel gene recognition for both DNA strands. Computers & Chemistry, vol. 17, No. 19, pp. 123-133. 1993.

[4] Hsieh, S.J., Lin, C.Y., Liu, N.H., Chow, W.Y., Tang, C.Y. GeneAlign: A coding exon prediction tool based on phylogenetical comparisons. Nucleic Acids Research, vol. 34, pp. 280-284. 2006.

[5] Alekseyenko AV, Kim N, Lee CJ. Global analysis of exon creation versus loss and the role of alternative splicing in 17 vertebrate genomes. RNA, vol. 13, No. 5, pp. 661-670. 2007.

[6] Wu J. Improving the specificity of exon prediction using comparative genomics. BMC Genomics, vol. 9, pp. 113-125. 2008.

[7] Wu J, Haussler D. Coding exon detection using comparative sequences. Journal of Computational Biology, vol. 13, No. 6, pp. 1148-1164. 2006.

[8] Burge C. The New GENSCAN Web Server at MIT. Disponible en línea: http://genes.mit.edu/GENSCAN.html. 2005.

[9] Henderson J., Salzberg S. and Fasman K. Finding Genes in Human DNA with a Hidden Markov Model. Journal of Computational Biology, vol. 4, No. 2, pp. 127-141. 1997.

[10] Xu Y., Einstein J., Mural R., Shah M. and Uberbacher E. An Improved System for Exon Recognition and Gene Modeling in Human DNA Sequences. Proceedings International Conference on Intelligent Systems for Molecular Biology, vol. 2, pp. 376-384. 1994.

[11] Salzberg S., Delcher A., Fasman K and Henderson J. A Decision Tree System for Finding Genes in DNA. Journal of Computational Biology, vol. 5, No. 4, pp. 667-680. 1998.

[12] Pertea M., Allen J. and Salzberg S. Computational gene prediction using multiple sources of evidence. Genome Research, vol. 14, No. 1, pp. 142-148. 2004.

[13] Witten I., Frank E. Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations. San Francisco. Morgan Kaufmann. 2000.

[14] Quinlan J.R. C4.5: Programs for Machine Learning. San Francisco. Morgan Kaufmann Publisher. 1993.

[15] Hyunjoong K. and Loh W. Classification Trees With Unbiased Multiway Splits. Journal of the American Statistical Association, vol. 96, pp. 598-604. 2001.

Revista Avances en Sistemas e Informática, Vol.8 No. 3, diciembre de 2011 - Medellín. ISSN 1657-7663