Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Tipo de artículo: Artículo original
Temática: Ingeniería y gestión de software
Recibido: 28/01/2014 | Aceptado: 13/05/2015

Algoritmos de aprendizaje automático para clasificación de Splice Sites en secuencias genómicas

Machine Learning algorithms for Splice Sites classification in genomic sequences

Heidy Díaz-Barrios[1], Yania Alemán-Rivas[2], Leidys Cabrera-Hernández[3], Alejandro Morales-Hernández[3], María del Carmen Chávez-Cárdenas[3], Gladys María Casas-Cardoso[3]

[1]AMPP Placetas. 2da del Norte # 46 e/ 3 y 4 del Este. Placetas, VC, Cuba. heidyd@uclv.edu.cu

[2] DMPF Placetas. Paseo Martí # 17A e/ Carretera Central y 1ra del Norte. Placetas, VC, Cuba. yaniaa@uclv.edu.cu

[3]Departamento de Computación, Centro de Estudios de Informática (CEI), Facultad Matemática, Física y Computación (MFC), Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba. leidysc@uclv.edu.cu, alejandromoralesh@uclv.edu.cu, mchavez@uclv.edu.cu, gcasas@uclv.edu.cu

Resumen

Las técnicas de clasificación se utilizan frecuentemente en la solución de diferentes problemas de la Bioinformática. Las secuencias de ADN de la mayoría de los genes se transcriben en ARN mensajero que se traducen en proteínas. El ADN contiene en los genes segmentos codificantes (exones) y no codificantes (intrones). Durante el proceso de transcripción los intrones son "cortados", mecanismo conocido como *splicing* que coloca a los exones de un gen consecutivamente, listos para traducirse en la secuencia de aminoácidos que conforman la proteína. En los *splice sites*, el principio del intrón es conocido como *donor* (par AG), y el final es conocido como *acceptor* (par GT). El presente trabajo aborda la predicción de sitios de *splicing*. Se utilizan técnicas de aprendizaje automatizado necesarias en la descripción de dominios biológicos y dos bases de datos de secuencias de nucleótidos, para clasificar verdaderos y falsos *splice sites* con 7000 casos cada una, 6000 falsos y 1000 verdaderos. Se prueba y compara una serie de algoritmos utilizando WEKA (*Waikato Enviroment for Knowledge Analysis*) para encontrar los mejores clasificadores. Para hacer la selección del mejor clasificador se aplican las medidas más conocidas basadas en la matriz de confusión: exactitud, razón de verdaderos positivos, curvas ROC, etc. Como resultados del estudio se concluye que los métodos bayesianos maximizaron el número de verdaderos positivos y el área bajo la curva, por lo que es la propuesta a utilizar para realizar la clasificación de sitios de *splicing*.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Palabras clave: acceptor, aprendizaje automatizado, clasificadores, donnor, *splicing*.

Abstract

The classification techniques are been used frequently in the solution of different Bioinformatic problems. The ADN sequences in the majority of the gene make a transcript to ARN messenger, whom have led to proteins. The ADN contain in the genes encode segments (exones), and unencode segments (introns). During the process of transcription the introns are cut, that mechanism is call splicing, it put the axons of the gene, one consecutive the other, and ready to lead to the sequence of amino acid to make the protein up. In the splice sites, the beginning of the introns is call donor (AG par), and the end is call acceptor (GT par). A few of these combinations are really splice sites. The present work is about the prediction of splicing. It is used the techniques of machine learning necessary to descript biology domains and two database of nucleates sequences to classify true or false splice sites, with 7000 cases, 6000 false and 1000 true. It is about to proof and compare a series of algorithms using WEKA (Waikato Enviroment for Knowledge Analysis) to find the best classifiers. To make the selection of the best classification it is applied the knowlest measure based in the Matrix of Confusion: accuracy, rate of True Positive (TP), area under the curve of Receiver Operator Curve (ROC), etc. As result of the study it is conclude that the Bayesian methods maximize the number of true positive and the area under the curve, which are the nominations to use to classify splice sites.
Keywords: *acceptor, classifiers, donnor, machine learning, splicing.*

Introducción

La Bioinformática constituye el campo de conocimientos multidisciplinario entre la biología, la informática y la matemática que debe abordar problemas que habían quedado sin solucionar a través de la historia, como es la necesidad de desarrollar nuevos algoritmos para el tratamiento de problemas de análisis de secuencias y localización de genes dentro del genoma de un cierto organismo (Chávez Cárdenas, 2008).

El ácido desoxirribonucleico, frecuentemente abreviado como ADN (y también DNA, del inglés (*deoxyribonucleic acid*), forma parte de todas las células. Para que la información que contiene el ADN pueda ser utilizada por la maquinaria celular, debe copiarse en primer lugar en nucleótidos más cortos llamados ARN. Las moléculas de ARN se copian exactamente del ADN mediante un proceso denominado transcripción (Galperin, 2007). Así, las secuencias de ADN de la mayoría de los genes se transcriben en ARN mensajero que a su vez se traducen en las proteínas. En los procariotas (organismos menos desarrollados) el ARN mensajero es una copia del ADN. Sin embargo, en los eucariotas, el ADN contiene en los genes segmentos codificantes (exones) y no codificantes (intrones) y estos últimos se "cortan" durante el proceso de transcripción a RNA mensajero. A este mecanismo se le conoce como *splicing*, consiste en colocar a los exones de un gen consecutivamente, y así estarán listos para traducirse en la secuencia de

aminoácidos que conforman la proteína (Foley, y otros, 2004). La detección de intrones y exones constituye una de las formas para abordar el problema de la localización de los genes.

Para la predicción de sitios de *splicing* en regiones genómicas codificantes para proteínas se utilizan las técnicas de aprendizaje automatizado, las que son necesarias en la descripción de dominios biológicos. Estos dominios son: genómica, proteómica, micro-arreglos (antes citados como matrices de ADN o micro *arrays*), sistemas biológicos, evolución y minería de texto. La identificación de sitios de *splicing* o corte de intrones, que separan zonas codificantes y no codificantes se aborda desde varios puntos de vista. Se conoce en primer lugar que todas las secuencias que representan un gen comienzan con un codón de inicio y finalizan con uno de los tres codones de terminación, pero la presencia de tales codones no siempre indica el inicio y el final del gen. (Ricardo, y otros, 2007b)

En los *splice sites,* el principio del intrón se conoce como *donor*, mientras que el que lo finaliza se conoce como *acceptor*. Los "*donors*" se caracterizan por la presencia del par de nucleótidos "GT" al inicio del intrón, los "*acceptors*" se identifican por el par "AG" al final del intrón. El inicio y el fin del intrón se marcan por los *splice sites*. Entonces se podría intentar reconocer *donors* y *acceptors* a través de estos dinucleótidos y con ellos los intrones. Estos dinucleótidos abundan en el genoma y sólo un pequeño por ciento de estas combinaciones son *splice sites* reales de ahí la limitación de este enfoque. (Saeys, 2004)

Si se tienen secuencias con el par "GT" de las cuales se conozca si son verdaderos o falsos *donors* se puede intentar "aprender" a clasificarlos utilizando la información de las bases nucleotídicas de su entorno y otro tanto podría hacerse a partir de secuencias con el par "AG" de las cuales se conozca si son verdaderos o falsos *acceptors*. Así el problema original se descompone en dos problemas de clasificación.

Las bases de datos de *splice sites* para humanos fue construida en la Universidad de Ghent, Bélgica, a partir de obtener ARN mensajero desde la base de datos pública EMBL (Base de datos de secuencias nucleotídicas). (EMBL, 2009).

El objetivo de este estudio es clasificar verdaderos y falsos *splice sites:* identificación de *donors* y *acceptors*, con los diferentes métodos que ofrecen la herramientas de Aprendizaje Automático Weka (Witten, et al., 2000), (Serrano, Tomecková, & Zvárová, 2012) y encontrar aquellos que clasifican la mayor cantidad de casos como verdaderos según diferentes parámetros.

A continuación se muestran los resultados estadísticos obtenidos después de probar un número considerable de algoritmos en el "entorno para análisis del conocimiento de la Universidad de Waikato" Weka (*Waikato Environment for Knowledge Analysis*), y se explica cómo interpretarlos.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Materiales y Métodos

Para cumplir con el objetivo planteado se cuenta con dos bases de datos de secuencias de nucleótidos, las bases de datos para este trabajo se conformaron con 7000 casos cada una, 6000 falsos y 1000verdaderos, tal como sugiere la proporción aproximada real de verdaderos y falsos *splice sites* en los genomas.

Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión (Tabla 1) que se obtiene cuando se prueba el clasificador en el conjunto de datos de entrenamiento.

Tabla 1. Matriz de confusión de un problema de dos clases

Matriz de Confusión		*Clase verdadera*	
		Pos	*Neg*
Clase Predicha	*pos*	*VP*	*FP*
	neg	*FN*	*VN*
Total columna		*P*	*N*

En la Tabla 1 las siglas *VP* y *VN* representan los elementos bien clasificados de la clase positiva y negativa respectivamente y *FP* y *FN* identifican los elementos negativos y positivos mal clasificados respectivamente. Basados en estas medidas, se calcula el error, la exactitud, la razón de *VP* (*rVP= VP/P*) o sensibilidad, la razón de *FP* (*FP=FP/N*), la razón de los *VN* (*rVN=VN/N*) o especificidad y la razón de los falsos negativos (*FN=FN/P*).Otra forma de evaluar el rendimiento de un clasificador es por las curvas ROC (*Receiver Operator Curve, Curva de operación del receptor*) (Fawcett, 2004). En esta curva se representa el valor de razón de VP contra la razón de FP, mediante la variación del umbral de decisión. El umbral de decisión es aquel que decide si una instancia *x*, a partir del vector de salida del clasificador, pertenece o no a cada una de las clases. Usualmente, en el caso de dos clases se toma como umbral por defecto 0.5; pero esto no es siempre lo más conveniente. Se usa el área bajo esta curva, denominada AUC (*Area Under the Curve*, área bajo la curva ROC) como un indicador de la calidad del clasificador. En tanto dicha área esté más cercana a la unidad, el comportamiento del clasificador está más cercano al clasificador perfecto (100% de VP con un 0% de FP). (Chávez Cárdenas, 2008).

En la resolución de este problema se emplearán algoritmos de aprendizaje automatizado, pues son los usados para cuando hay presencia de gran cantidad de datos, patrones ruidosos y la ausencia de teorías generales determinísticas.

Este estudio se realizó como continuación del trabajo Modelos de Redes Bayesianas en el estudio de secuencia genómicas y otros problemas biomédicos, de la doctora María del Carmen Chávez Cárdenas, en el cual se desarrollaron algoritmos basados en Redes Bayesianas que mejoraron los resultados existentes hasta ese momento.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

La investigación permitió identificar los clasificadores de mejores resultados en bases de datos con un número considerable de atributos, como las que usualmente se trabajan en Bioinformática (Chávez Cárdenas, 2008), para que sirvan de apoyo en la implementación de nuevos algoritmos de clasificación que mejoren los resultados alcanzados.

Herramienta WEKA

Para probar y comparar una serie de algoritmos de clasificación se usó una herramienta, desarrollada en la Universidad de Waikato, Nueva Zelanda. Este sistema está escrito en Java. (Witten, y otros, 2000)

En Weka se aplicaron métodos de aprendizaje a las bases de datos *donors* y *acceptors*, y se analizaron las salidas para extraer información sobre los datos.

Según los clasificadores utilizados se describe el funcionamiento de los algoritmos probados con este estudio en la Tabla 2.

Tabla 2. Funcionamiento algunos algoritmos de clasificación (Witten, y otros, 2000)

	Algoritmo	**Función**
Bayes	*AODE*	Promediado, estimadores de una dependencia.
	BayesNet	Aprender redes Bayesianas.
	NaiveBayes	Clasificador probabilístico estándar *NaiveBayes*.
	NaiveBayesSimple	Implementación de *NaiveBayes* simple
	NaiveBayesUpdateable	Clasificador *NaiveBayes* incremental el cual aprende una instancia a la vez.
Árboles	*ADTree*	Construye árboles de decisión alternativos.
	DecisionStump	Construye árboles de decisión de un nivel.
	Id3	Algoritmo árbol de decisión basado en "divide y vencerás".
	J48	Aprende árbol de decisión C4.5 (C4.5 implementados, revisión 8).
	LMT	Construye árboles de modelo logístico
	NBTree	Construye un árbol de decisión con clasificadores *NaiveBayes* en las hojas.
	RandomForest	Construcción de árboles aleatorios.
	RandomTree	Construir un árbol que considera un número aleatorio de características dadas en cada nodo.
	REPTree	Aprendizaje de árbol rápido que usa la poda en la reducción de errores
	UserClassifier	Deja a los usuarios construir ellos mismos el árbol de decisión.
Reglas	*ConjuntiveRule*	Aprende regla conjuntiva simple.
	DecisionTable	Construye una tabla de decisión simple del clasificador mayoritario.
	JRip	Algoritmo *RIPPER* (poda incremental reducida para producir reducción de error) para rapidez, regla de inducción eficaz
	Nnge	Método vecino más cercano de generación de reglas usando ejemplos generalizados no anidados.
	OneR	Clasificador 1R.

	Part	Obtiene reglas a partir de árboles de decisión construidos usando J4.8.
	Prism	Algoritmo de cobertura simple para reglas.
	Ridor	Regla de aprendizaje ondular hacia abajo.
	ZeroR	Predice la clase mayoritaria (si es nominal) o el valor promedio (si es numérico).
Funciones	*Logistic*	Construye modelos de regresión logística lineal.
	MultilayerPerceptron	Red neuronal de propagación hacia atrás
	RBFNetwork	Implementa una red de función radial básica.
	SimpleLogistic	Construye modelos de regresión logística lineal con selección de atributo incorporado.
	SMO	Algoritmo de optimización mínimo secuencial para soporte de clasificación de vectores.
	VotedPerceptron	Algoritmo *perceptron* votado.
	Winnow	Perceptron motivado a error con actualizaciones múltiples.
Perezosos	*IB1*	Aprendizaje basado en instancia un vecino más cercano básico.
	IBk	Clasificador k vecino más cercano.
	KStar	Vecino más cercano con función de distancia generalizado.
	LBR	Clasificador de Reglas Bayesianas Perezosas.
	LWL	Algoritmo general para aprendizaje localmente pesado.
Meta	*AdaBoostM1*	Aumentar usando el método AdaBoostM1
	Bagging	Un clasificador bolsa (bag), trabaja por regresión también.
	MultiBoostAB	CombinaKboostin y bagging usando el método *MultiBoosting*
	MultiClassClassifier	Usa un clasificador de dos clases para conjuntos de datos multiclases.
	Stacking	Combina varios clasificadores usando el método apilado (*stacking*).
	StackingC	Versión más eficiente de *stacking*.
	Vote	Combina clasificadores usando promedio de estimados de probabilidad o predicciones numéricas.

A partir de los resultados obtenidos por cada uno de los algoritmos se enfatiza en los resultados que maximizan la razón de los verdaderos positivos, los que el valor de la curva ROC es más cercano a 1 y los de mayor exactitud, porque estos son los que se acercan al clasificador perfecto, es decir, los que tienen menor cantidad de errores al clasificar los verdaderos *donors* y *acceptors*. (Chávez Cárdenas, 2008).

Igualmente se utilizó una herramienta creada que combina los resultados de los clasificadores individuales con los resultados de medidas de diversidad, las cuales han sido creadas por distintos autores para detectar a los clasificadores más diversos entre sí. Esto lo hace usando Algoritmos Genéticos, y obtiene un multiclasificador, que a la vez combina los clasificadores más diversos posible y maximiza la exactitud respecto a la de los clasificadores individuales en la medida de lo posible. (Morales Hernández, 2014)

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Resultados y discusión

Algoritmos bayesianos

Una red bayesiana es un modelo gráfico probabilístico que representa un conjunto de variables y sus dependencias probabilísticas. Puede calcular la distribución de probabilidad para cualquier subconjunto de variables de la red, dado los valores o distribuciones de las variables restantes. (Mitchell, 1997)

Este tipo de clasificador no es muy sensible a los cambios de sus parámetros, ya que se basa en información de toda la base, lo cual hace que pequeños cambios en la base no sean necesariamente significativos (Chávez Cárdenas, 2008).

Tabla 3. Resultados utilizando Redes Bayesianas en la base de datos *Acceptors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
AODE	0.727	0.96	**0.792**	0.95
BayesNet	0.72	0.958	0.791	0.949
HiddenNaiveBayes (HNB)	0.752	**0.962**	0.787	0.957
NaiveBayes	0.718	0.958	0.791	0.948
NaiveBayesSimple	0.718	0.958	0.791	0.948
NaiveBayesUpdateable	0.718	0.958	0.791	0.948
WAODE	**0.743**	0.961	0.78	0.955

Tabla 4. Resultados utilizando Redes Bayesianas en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
AODE	0.75	0.961	0.737	0.959
BayesNet	0.747	0.956	0.711	0.96
HiddenNaiveBayes (HNB)	0.765	**0.97**	**0.796**	0.959
NaiveBayes	0.747	0.956	0.711	0.96
NaiveBayesSimple	0.747	0.956	0.711	0.96
NaiveBayesUpdateable	0.747	0.956	0.711	0.96
WAODE	**0.778**	0.966	0.751	0.964

Los métodos bayesianos fueron altamente balanceados en cuanto a los parámetros medidos, de todos se obtuvieron importantes resultados por lo que constituyen buenos clasificadores en las bases de datos utilizadas y permiten su aplicación para obtener la mejor clasificación. Se destacaron los métodos WAODE y HNB en ambas bases con los mejores valores de área bajo la curva y razón de verdaderos positivos

Algoritmos de árboles de decisión

Este esquema de aprendizaje automatizado se deriva del pensamiento divide y vencerás. Un árbol de decisión clasifica las instancias ordenándolas de la raíz a las hojas. Cada nodo interior del árbol especifica una prueba de algún atributo y las hojas son las clases en las cuales se clasifican las instancias, cada rama descendiente de un nodo interior

corresponde a un valor posible del atributo probado en ese nodo. Así, cada rama, de la raíz a un nodo hoja, corresponde a una conjunción de atributos y el árbol en sí, a una disyunción de estas conjunciones. (Witten, et al., 2000)

Entre las ventajas más sobresalientes de los árboles de decisión se encuentra que provee una estructura sumamente efectiva dentro de la cual se puede estimar, cuáles son las opciones e investigar las posibles consecuencias de seleccionar cada una de ellas (Autores, 2012).

Tabla 5. Resultados utilizando Árboles de Decisión en la base de datos *Acceptors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
AlternatingDecisionTree (ADTree)	0.711	**0.931**	**0.591**	0.96
DecisionStump	0	0.726	0	1
Id3	0.492	0.73	0.555	0.904
J48	0.591	0.727	0.548	0.937
RandomForest (10 trees, 8 random features)	**0.771**	0.847	0.111	0.995
RandomForest (5 trees, 8 random features)	0.613	0.785	0.241	0.975
RandomTree	0.219	0.544	0.219	0.87
REPTree	0.674	0.877	0.565	0.955
SimpleCart	0.687	0.878	0.559	0.958

Tabla 6. Resultados utilizando Árboles de Decisión en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
AlternatingDecisionTree (ADTree)	**0.827**	**0.967**	0.683	0.976
DecisionStump	0	0.706	0	1
Id3	0.695	0.841	0.735	0.946
J48	0.759	0.842	0.745	0.961
RandomForest (10 trees, 8 random features)	0.823	0.834	0.116	0.996
RandomForest (5 trees, 8 random features)	0.563	0.759	0.211	0.973
RandomTree	0.179	0.522	0.183	0.861
REPTree	0.768	0.929	**0.771**	0.961
SimpleCart	0.794	0.932	0.763	0.967

Varios de estos algoritmos de árboles de decisión no funcionaron con las bases de datos del estudio, puesto que no se construye el modelo de aprendizaje y la herramienta deja de funcionar. Los algoritmos con los que sucede esta situación son: *NBTree, BFTree, LMT, UserClassifier*.

El algoritmo *ADTree* resultó el mejor método para la base de datos *Acceptors* según el área bajo la curva ROC y la razón de verdaderos positivos, mientras que en la de *Donors* fue por la exactitud y el área bajo la curva. La mayor razón de verdaderos positivos la obtuvo el método *REPTree* en ambas bases. Los clasificadores basados en árboles de decisión no brindaron resultados significativos puesto que los parámetros medidos fueron bajos.

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Algoritmos basados en Reglas

Son una alternativa popular de los árboles de decisión. El antecedente o predicción de una regla es una serie de pruebas como las que se hacen en el nodo en árboles de decisión. El consecuente o conclusión da la clase o clases que aplica a instancias cubiertas por esa regla o tal vez da una probabilidad de distribución acerca de las clases.

Una regla es generada por cada hoja. El antecedente de cada regla incluye la condición de cada nodo en el camino desde la raíz hasta la hoja y el consecuente de la regla es la clase asignada por la hoja.

Tabla 7. Resultados utilizando Reglas en la base de datos *Acceptor*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
ConjuntiveRule	0	0.5	0	1
JRip	0.614	**0.809**	**0.659**	0.931
OneR	0	0.5	0	1
PART	0.622	0.808	0.626	0.937
Prism	0.777	0.71	0.506	0.979
Ridor	**0.76**	0.695	0.411	0.978
ZeroR	0	0.5	0	1

Tabla 8: Resultados utilizando Reglas en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
ConjuntiveRule	0	0.5	0	1
JRip	0.753	**0.886**	**0.807**	0.956
OneR	0	0.5	0	1
PART	0.69	0.824	0.685	0.949
Prism	**0.84**	0.752	0.597	0.983
Ridor	0.773	0.839	0.713	0.965
ZeroR	0	0.5	0	1

El algoritmo *DecisionTable* al igual que el *LibSVM*, presenta un problema evaluando el clasificador, las clases no se encuentran dentro del *CLASSPATH*.

Estos algoritmos obtienen resultados poco significativos, con varios métodos que clasificaron erróneamente en todos los casos. Sin embargo se puede destacar el *JRip* como el de mejores resultados en este grupo.

Algoritmos Funciones (Regresión Logística)

La regresión logística es un instrumento estadístico de análisis multivariado, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia se ha puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes,

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

que pueden ser cuantitativas o categóricas. El propósito del análisis consiste en predecir la probabilidad de que ocurra cierto "evento"(Le Cessie, y otros, 1992).

Si los datos se pueden separar en dos grupos usando un hiperplano, que separa las instancias pertinentes de las diferentes clases, se dice que es linealmente separable y para esto se usan algoritmos *Perceptron* (Saeys, 2004)

Tabla 9. Resultados utilizando Regresión Logística en la base de datos *Acceptors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
Logistic	0.767	**0.96**	0.713	0.964
RBFNetwork	**0.792**	0.948	0.694	0.97
SMO	0.756	0.833	0.703	0.962
VotedPerceptron (936 perceptrons)	0.743	0.792	0.56	0.968
MultilayerPerceptron	0.746	0.958	**0.756**	0.957
Winnow	0.318	0.63	0.404	0.856

Tabla 10. Resultados utilizando Regresión Logística en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
Logistic	0.782	**0.962**	0.742	0.966
RBFNetwork	**0.788**	0.956	0.675	0.97
SMO	0.778	0.851	0.737	0.965
VotedPerceptron (870 perceptrons)	0.779	0.825	0.61	0.971
MultilayerPerceptron	0.755	0.96	**0.769**	0.959
Winnow	0.244	0.562	0.256	0.868

El algoritmo LibSVM presenta un problema evaluando el clasificador, las clases *libsvm* no se encuentran dentro del CLASSPATH.

Estos métodos demostraron ser lentos y sus resultados son buenos para algunos de ellos, destacándose el *MultilayerPerceptron* y el *Logistic* por sus valores de área bajo la curva y verdaderos positivos.

Algoritmos Perezosos (*lazy*)

El razonamiento basado en casos se basa en el principio de usar experiencias viejas para resolver problemas nuevos. Muchos algoritmos usan este razonamiento para resolver los problemas y entre los más comunes están los de clasificación. Aunque todos los métodos de clasificación se basan en casos, existe un conjunto que se conoce como algoritmos basados en casos, o también como métodos de aprendizaje perezoso. (García, 2011)

Una nueva instancia se compara con el resto de la base de casos a través de una medida de similitud o de distancia. La clase de la nueva instancia será la misma que la del caso que más cercano esté a la nueva instancia. A este proceso se le conoce con el nombre de método del "vecino más cercano" (*nearest neighbor*) (García, 2011).

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

El tiempo que toma hacer una predicción es proporcional al número de instancias de entrenamiento. Una solución es adoptar la estrategia K-vecinos, donde k puede escogerse probando diferentes valores y escogiendo el mejor.

Tabla 11. Resultados utilizando Métodos de Aprendizaje Perezoso en la base de datos *Acceptors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
IB1	0.436	0.686	**0.473**	0.898
IBK=1	0.449	0.707	0.426	0.913
IBK=5	0.698	0.859	0.303	0.978
IBK=10	**0.835**	0.902	0.228	0.993
KStar	0.452	0.821	0.461	0.907
LocallyWeightedLearning (LWL)	0	0.948	0	1

Tabla 12. Resultados utilizando Métodos de Aprendizaje Perezoso en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
IB1	0.378	0.647	0.406	0.889
IBK=1	0.408	0.679	0.375	0.909
IBK=5	0.688	0.823	0.232	0.983
IBK=10	0.814	0.872	0.144	0.995
KStar	0.405	0.783	0.401	0.902
LocallyWeightedLearning (LWL)	0	0.936	0	1

Dentro de los algoritmos perezosos, el LBR trabaja para conjuntos de pruebas pequeños, puesto que cada instancia de prueba selecciona un conjunto de atributos para los cuales la supuesta independencia no debe ser hecha, los demás son tratados como independientes de cada una de las clases dadas y el conjunto de atributos seleccionado. Por esta razón, con las bases de datos que se utilizan, ese método responde muy lentamente sin que se puedan obtener sus resultados.

Estos algoritmos, no aportan resultados significativos para la clasificación en las bases de datos del estudio, en general existe un desbalance de los parámetros para considerar un método superior al resto. El algoritmo IBk demostró que a medida que se aumenta el valor de k, aumenta la exactitud y el área bajo la curva pero disminuyen los verdaderos positivos.

Algoritmos meta (multiclasificadores en Weka)

La combinación de clasificadores es en la actualidad un área activa de investigación en el aprendizaje automatizado y el reconocimiento de patrones. Se han publicado numerosos estudios teórico y empíricos que demuestran las ventajas del paradigma de combinación de clasificadores por encima de los modelos individuales. (Kunheva, y otros, 2002).

Existen varias formas en las cuales se pueden construir multiclasificadores. En todos los casos se basan en la selección de los clasificadores de base y la elección de la forma de combinar las salidas. (Bonet, 2008).

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

Entre los modelos más populares que combinan clasificadores están *Bagging, Boosting, Stacking*, métodos basados en rasgos.

Tabla 13. Resultados utilizando Multiclasificadores de Weka en la base de datos Acceptor

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
Bagging	0.719	0.928	0.569	0.963
Stacking	0	0.5	0	1
Vote	0	0.5	0	1
MultiClassClassifier	**0.767**	**0.96**	**0.713**	0.964
AdaBoostM1	0.632	0.9	0.516	0.95
MultiBoostAB	0	0.754	0	1
SrackingC	0	0.5	0	1

Tabla 14. Resultados utilizando Multiclasificadores de Weka en la base de datos *Donors*

Algoritmo	**Exactitud**	**Área bajo la curva ROC**	**rVP**	**rVN**
Bagging	0.719	0.928	0.569	0.963
Stacking	0	0.5	0	1
Vote	0	0.5	0	1
MultiClassClassifier	**0.767**	**0.96**	**0.713**	0.964
AdaBoostM1	0.632	0.9	0.516	0.95
MultiBoostAB	0	0.817	0	1
SrackingC	0	0.5	0	1

Los algoritmos meta, considerados multiclasificadores en la herramienta Weka, no constituyen buenos clasificadores para las bases de datos del estudio, siendo el *MultiClassClassifier* el único que mostró resultados a tener en cuenta.

Multiclasificación usando herramienta especializada

A pesar de que WEKA es un ambiente de simulación computacional que presenta un amplio soporte para la experimentación con varios métodos estadísticos y de Inteligencia Artificial, se consideró oportuno utilizar la implementación de la versión de Algoritmo Genético (AG) que propone la tesis de pregrado del estudiante Alejandro Morales Hernández (Morales Hernández, 2014).

Las potencialidades de este software se tienen en la herramienta desarrollada denominada *Splicing*, un ambiente que permite decidir qué clasificadores usar en la construcción de un sistema multiclasificador de forma fácil, relativamente rápida y segura.

La herramienta *Splicing* permite seleccionar varios clasificadores individuales para combinar, la regla de combinación de sus salidas, las medidas de diversidad para determinar cuán diversos son estos clasificadores, la forma en que se va a evaluar el modelo de clasificación obtenido con el multiclasificador (*Cross-validation, percentage split*, etc.) y los parámetros requeridos para configurar el AG; todo esto en un ambiente amigable a usuarios menos especializados. La

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

meta es encontrar una exactitud del multiclasificador superior a la mayor exactitud de los clasificadores de forma individual (Morales Hernández, 2014).

En las siguientes tablas se muestran los resultados utilizando esta herramienta, combinando diferentes clasificadores de Weka.

Tabla 15. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Acceptors*

Clasificadores	**Exactitud**
NaiveBayes	0.9239
IB1	0.8269
J48	0.8735
JRip	0.8987
Logistic	0.9273
Exactitud Final	**0.9311**

Tabla 16. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Donors*

Clasificadores	**Exactitud**
NaiveBayes	0.9298
IB1	0.8151
J48	0.9227
JRip	0.9315
Logistic	0.9269
Exactitud Final	**0.9424**

Tabla 17. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Acceptors*

Clasificadores	**Exactitud**
BayesNet	0.9239
ADTree	0.9046
OneR	0.8634
KStar	0.8408
MultilayerPerceptron	0.9282
Exactitud Final	**0.9395**

Tabla 18. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Donors*

Clasificadores	**Exactitud**
BayesNet	0.9298
ADTree	0.9298
OneR	0.8634
KStar	0.8273
MultilayerPerceptron	0.9265
Exactitud Final	**0.9391**

Tabla 19. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Acceptors*

Clasificadores	**Exactitud**
AODE	0.9256
SimpleLogistic	0.9290
Id3	0.8815
Ridor	0.9
IBk=5	0.8496
Exactitud Final	**0.9374**

Tabla 20. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Donors*

Clasificadores	**Exactitud**
AODE	0.9303
SimpleLogistic	0.9345
Id3	0.8966
Ridor	0.9269
IBk=5	0.8798
Exactitud Final	**0.9424**

Tabla 21. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Acceptors*

Clasificadores	**Exactitud**

Tabla 22. Resultados utilizando sistemas multiclasificadores basados en Algoritmos Genéticos en la base de datos *Donors*

Clasificadores	**Exactitud**
BayesianLogisticRegression	0.9256

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 155-170

BayesianLogisticRegression	0.9013
SMO	0.9277
IBk=10	0.8895
DecisionTable	0.8634
NBTree	0.9223
Exactitud Final	**0.9307**

SMO	0.9252
IBk=10	0.8752
DecisionTable	0.8634
NBTree	0.9340
Ninguna combinación de clasificadores supera la mejor exactitud de los clasificadores individuales.	

Después de un análisis exhaustivo de los diferentes clasificadores y de cada algoritmo de manera individual y teniendo en cuenta los parámetros que se expresan en las tablas anteriores, se concluye que los clasificadores de redes bayesianas fueron los que mostraron mejores resultados en su conjunto y específicamente los métodos AODE y HNB resultaron superiores al resto de los métodos probados con los valores más altos de verdaderos positivos y área bajo la curva ROC.

Igualmente, la combinación de clasificadores para obtener el mejor multiclasificador con la herramienta *splicing*, mostró que en la base de datos *Acceptor*, la combinación de los métodos *BayesNet, ADTree, OneR, KStar y MultilayerPerceptron*, obtuvo la mayor exactitud del multiclasificador a pesar de que tomó la mayor cantidad de tiempo, siendo este tiempo menor utilizando *AODE, SimpleLogistic, Id3, Ridor e IBk* con k=5 y con una exactitud similar.

Para la base de datos *Donors* fue igual el valor de la exactitud con dos de las combinaciones y a la vez fue superior a los obtenidos en la base *Acceptors*. Las combinaciones con mejores resultados fueron *NaiveBayes, IB1, J48, JRip y Logistic* así como *AODE, SimpleLogistic, Id3, Ridor e IBk* con k=5.

Conclusiones

No existe un modelo clasificador mejor que otro de manera general, es por esto que han surgido varias medidas para evaluar la clasificación y comparar los modelos empleados para un problema determinado.

Al observar el comportamiento de todos los grupos de clasificadores, se concluye que los algoritmos que usan Redes Bayesianas fueron los de mejor comportamiento para la localización de genes en un genoma completo, o en una larga secuencia genómica, puesto que los resultaros fueron muy regulares a la hora de maximizar los verdaderos positivos en ambas bases de datos.

El tiempo es algo fundamental en los problemas de Bioinformática, pues casi siempre hay grandes volúmenes de información para procesar. Los modelos perezosos fueron los más afectados por este parámetro, además, los perezosos tuvieron malos resultados en cuanto a razón de verdaderos positivos esencialmente.

En el uso de multiclasificadores en Weka, del grupo de los meta, para bases de datos de gran cantidad de atributos se debe tener cuidado, pues en este estudio los resultados de varios de ellos no fueron favorables comparados con otros grupos de clasificadores. No obstante, el *MultiClassClassifie*r tuvo un buen aprendizaje con estas bases de datos.

Con el uso de la herramienta de *Splicing*, los resultados fueron satisfactorios según la exactitud que muestra la combinación de los algoritmos en cada base de datos.

Se puede concluir, luego de un exhaustivo análisis, que el grupo de algoritmos bayesianos es el que mejor logra clasificar con todos sus métodos las bases de datos *Donors* y *Acceptors*. Esto se comprueba dado que en todos los casos logra maximizar el área bajo la curva ROC, lo que es un indicador de la calidad del clasificador. Se recomienda el uso de algoritmos que utilicen Redes Bayesianas para el aprendizaje automatizado en bases de datos del genoma humano con atributos discretos.

Referencias

AUTORES, C. D. V*entajas Y Desventajas Del Árbol De Decisón. Introducción A La Programación.* 2012. Disponible En: Http://Ipg3.Blogspot.Com/2012/02/Ventajas-Y-Desventajas-Del-Arbol-De.Html

BOANET, ISIS. Modelo Para La Clasificación De Secuencias En Problemas De La Bioinformática, Usando Técnicas De Inteligencia Artificial. Tesis En Opción Al Grado Científico De Doctor En Ciencias Técnicas. Universidad Central "Marta Abreu" De Las Villas, Santa Clara, 2008.

CHÁVEZ CÁRDENAS, MARÍA DEL CARMEN. *Modelos De Redes Bayesianas En El Estudio De Secuencias Genómicas Y Otros Problemas Biomédicos.* Tesis En Opción Al Grado Científico De Doctor En Ciencias Técnicas. Universidad Central "Marta Abreu" De Las Villas, Santa Clara, 2008. Págs. 80-88.

EMBL. Bases De Datos De Secuencias Nucleotídicas. Consultado En Septiembre De 2014. Disponible En: Http://Www.Ebi.Ac.Uk/Embl/Index.Html

FAWCETT, T. Roc Graph: Notes And Practical Consideration For Researchers Machine Learning , 2004. Consultado En Agosto De 2014. Disponible En: Https://Home.Comcast.Net/~Tom.Fawcett/Public_Html/Papers/Roc101.Pdf

FOLEY, R. A. Y LEWIN, R. *Principles Oh Human Evolution,* 2004. *Segunda Edición.* S.L. : Backwell Publishing, Review From Times Education Supplement, University Of Durham.

GALPERIN, M. Y. *The Molecular Biology Database*, 2008. *Nucleic Acids Research*, 2007 - Oxford Univ Press.

GARCÍA, M. M. *Modelo De Un Sistema De Razonamiento Basado En Casos Para El Análisis En La Gestión De Riesgos,* 2011. *Serie Científica De La Universidad De Las Ciencias Informáticas, No. 11, Vol. 4. Disponible En: Http://Publicaciones.Uci.Cu/*

KUNHEVA Y SHIP. *Relationship Between Combination Methods And Measures Of Diversity In Combining Classifiers,*2002. *Information Fusion* 3 (2), 135-148

LE CESSIE, S y VAN HOUWELINGEn, J. *Ridge Estimators In Logistic Regression,* 1992. *Applications Statistics*. 41 No. 1, Pag. 191-201

MITCHELL, T. M. *Machine Learning*. Mcgraw-Hill Science/Engineer, 1997. 421 Pags.

MORALES HERNÁNDEZ, ALEJANDRO. Construcción De Sistemas Multiclasificadores Usando Algoritmos Genéticos Y Medidas De Diversidad, 2014. Tesis En Opción Al Título De Licenciado En Ciencia De La Computación. Universidad Central "Marta Abreu" De Las Villas, Santa Clara. Págs. 14-16.

RICARDO, GRAU, y OTROS. *Boolean Algebraic Structures Of The Genetic Code. Possibilities Of Applications*, , 2007. *Proceeding Kdecb'06 Proceedings Of The 1st International Conference On Knowledge Discovery And Emergent Complexity In Bioinformatics.* Springer-Verlag Berlin, Heidelberg ©, *Pages* 10-21

SERRANO, J., TOMECKOVÁ, M., & ZVÁROVÁ, J. (2012). Métodos De Aprendizaje Automático Para El Descubrimiento De Conociemiento En Datos Médicos. *European Journal For Biomedical Informatics*. Disponibe En: Http://Www.Ejbi.Org/En/Ejbi/Article/41-Es-Metodos-De-Aprendizaje-Automatico-Para-El-Descubrimiento-De-Conocimiento-En-Datos-Medicos-Sobre-Arterosclerosis.Html

WITTEN, IAN H. Y EIBE, FRANK. Weka Machine Learning Algorithms In Java, 2000. *Data Mining: Practical Machine Learning Tools And Techniques With Java Implementations.* 10, Págs. 404-417.