ARTÍCULO CORTO

Predicción de la estructura secundaria de proteínas usando máquinas de soporte vectorial

Protein secondary structure prediction using support vector machines

D. J. Delgado[*, **], *H. Arguello*[*], *R. Torres*[**]

Abstract

Among the computational methods used for predicting secondary structure proteins highlights the use of support vector machines. This research shows the predicted secondary structure of protein from its primary amino acid sequence using Support Vector Machines. As inputs, in the proposed methodology, features are used from different structural motifs or text strings associated with the primary structure which represents the secondary structure, such as R-group and the probability that the amino acid at position adopts a central particular secondary structure. For feature extraction method is used coding of sequences in which each symbol in the primary structure is associated with each symbol in the secondary structure. The use of this encoding method reduces the dimensionality of the data of thousands of characteristics only 220 of these. The results obtained are comparable to those reported in the literature, taking about 70% accuracy. Furthermore, it is possible to reduce computational cost in the construction of classifiers because this work models the problem of multi classification as a group of binary classifiers.

Key words: coding methodology, support vector machines, prediction of protein secondary structure.

Resumen

Entre los métodos computacionales utilizados para la predicción de la estructura secundaria de proteínas, se destaca el uso de máquinas de soporte vectorial. Este trabajo de investigación presenta la predicción de la estructura secundaria de proteínas desde su secuencia primaria de aminoácidos usando Máquinas de Soporte Vectorial. Como entradas, en la metodología propuesta, se utilizan características de los diferentes motivos estructurales o cadenas de texto asociadas a la estructura primaria que representa la estructura secundaria, tales como el R-grupo y la probabilidad de que el aminoácido en la posición central adopte una determinada estructura secundaria. Para la extracción de características se utiliza un método de codificación de secuencias en el que cada símbolo en la estructura primaria se relaciona con cada símbolo en la estructura secundaria. El uso de este método de codificación permite reducir la dimensionalidad de los datos de miles de características a sólo 220 de estas. Los resultados obtenidos son comparables a los registrados en la literatura, teniendo cerca de un 70% de precisión. Además, se logra reducir los costos computacionales en la construcción de los clasificadores debido a que este trabajo modela el problema de multi-clasificación como un grupo de clasificadores binarios.

Palabras Clave: *máquinas de soporte vectorial, metodología de codificación, predicción de la estructura secundaria de proteínas.*

Recibido: mayo 18 de 2011 **Aprobado:** junio 29 de 2012

* Grupo de Investigación en Ingeniería Biomédica GIIB, Universidad Industrial de Santander, Colombia.

** Grupo de Investigación en Bioquímica y Microbiología GIBIM, Universidad Industrial de Santander, Colombia djdelgad@uis.edu.co, harguello@uis.edu.co, rtorres@uis.edu.co

Introducción

Las proteínas son macromoléculas poliméricas constituidas por cadenas lineales de 20 diferentes aminoácidos (aa), a estas cadenas se les denomina estructuras primarias, estas cadenas de aa generan tres grandes grupos estructurales al interior de las proteínas: las hélices o estructuras α*(H)*, las láminas o estructuras β*(E)* y las conformaciones coil (C), ver figura 1. A estas conformaciones se les denomina estructuras secundarias. En este trabajo se propone una metodología de predicción para la estructura secundaria de proteínas que reduzca el número de características medidas para la creación de máquinas de aprendizaje que infieran el contenido estructural de una proteína.

Diversos han sido los métodos computacionales utilizados para predecir la estructura secundaria de una proteína desde su secuencia primaria de aminoácidos; estos métodos incluyen: aquellos basados en la composición de los aminoácidos (Chou, 1980; Nakashima *et al*, 1985; Muskal and Kim, 1992), redes neuronales (Rost and Sander, 1993b), modelos ocultos Markov (HMM) (Hubbard and Park, 1995) y máquinas de soporte vectorial (MSV).

Hoy en día las MSV están siendo usadas para la solución de problemas en bioinformática, más que otras herramientas. La predicción de la estructura secundaria de proteínas no es la excepción (Hua and Sun, 2001). Predecir la estructura secundaria de una proteína puede analizarse como un típico problema de reconocimiento o clasificación de patrones, en el cual cada aa en la estructura primaria, debe ser clasificado en uno de los tres diferentes grupos estructurales: hélices-α, láminas-β o coil.

Este problema de clasificación se puede abordar mediante la clasificación de los patrones que generan las representaciones textuales de la estructura primaria y secundaria, ver figura 2. Dichas representaciones son generadas por dos alfabetos, el alfabeto que representa la estructura primaria el cual contiene los símbolos que representan a los 20 aminoácidos de los cuales se componen la mayor parte de las proteínas, este alfabeto se denominará Σ = {A,R,N,D,C,E,Q,G,H,I,L,M,F, P,S,T,W,Y,V}.

El segundo alfabeto, el cual representa simbólicamente los diferentes motivos estructurales que pueden tener los diferentes aminoácidos se denominará Γ = {E,H,C}. Donde *E* representa a las láminas-β, *H* a las hélices-α y *C* a las estructuras coil. Con estos conjuntos de símbolos se puede representar textualmente tanto a la estructura primaria como la estructura secundaria.

Metodología

Las máquinas de soporte vectorial

Para abordar el problema de clasificación de patrones asociado a la predicción de la estructura secundaria de una proteína se necesita de una herramienta matemática que permita clasificar las características extraídas de las cadenas de caracteres que representan las es-

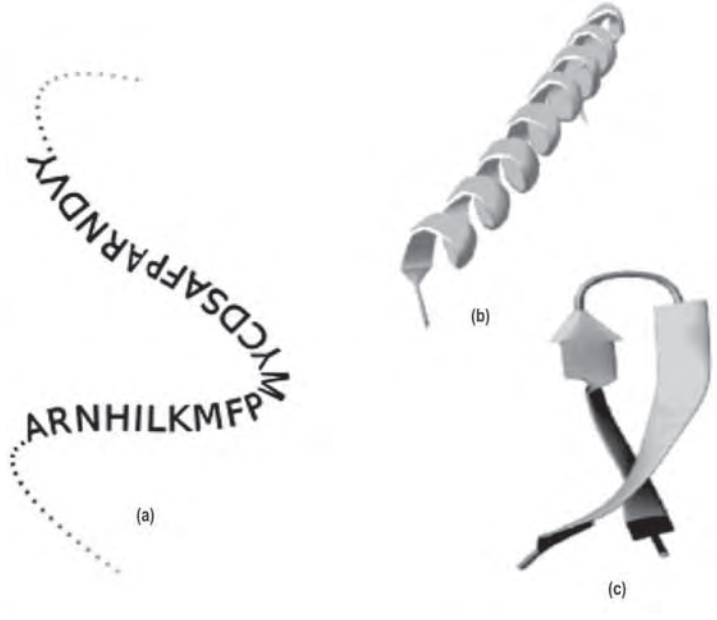

Figura 1: Representaciones estructurales de una proteína. (a) Estructura primaria, (b) y (c) estructura secundaria, (b) son las Hélices-α y (c) son las láminas-β.

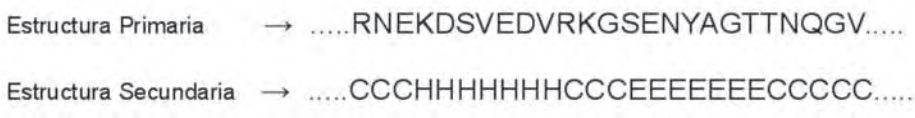

Figura 2: Representación simplificada para la estructura secundaria de una proteína.

tructuras primarias y que se asocian con su respectiva estructura secundaria. En este trabajo la herramienta usada fueron las máquinas de soporte vectorial (MSV) las cuales son un método efectivo en el área de reconocimiento de patrones en general. Una tarea de clasificación o reconocimiento de patrones generalmente necesita de un conjunto de datos para entrenamiento y otro para realizar las pruebas.

Cada instancia en el conjunto de entrenamiento tiene un valor objetivo (etiqueta de clase) y varios atributos (características). La meta de una MSV es generar un modelo que sea capaz de predecir correctamente los valores objetivo de alguna instancia perteneciente al grupo de pruebas sin conocer cómo está etiquetado, para luego poder extrapolar dicho modelo a cualquier individuo perteneciente al universo del cual se tomaron los ejemplos de entrenamiento.

Para generar un modelo de clasificación se parte de un conjunto de entrenamiento constituido por parejas (x_i, y_i) $i = 1,2,...,l$, donde $x_i \in \mathbb{R}^n$ e $y \in \{-1, +1\}^l$ y l es el número de ejemplos de entrenamiento. Una MSV (Vapnik, 1995; Cortes and Vapnik, 1995) requiere de la solución del siguiente problema de optimización, ver ecuación 1.

$$\begin{aligned} &\min_{W,b,\xi} \frac{1}{2} W^T W + C\sum_{i=1}^{l} \xi_i \\ &sujeto\,a: y_i(W^T \varnothing(x_i) + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0 \end{aligned} \qquad (1)$$

Donde $W^T\phi(x_i) + b$ representa el hiperplano de separación, C controla el equilibrio entre la complejidad de la máquina y el número de puntos no separables por un hiperplano, ξ_i mide la desviación de un punto x_i del punto de separación $W^T\phi(x_i) + b$. Los vectores x_i son mapeados a un espacio dimensional mayor por la función ϕ. Las MSV buscan un hiperplano que realice una separación lineal que tenga un margen de separación máximo entre los grupos a clasificar. $C > 0$ permite el balance entre maximizar el margen y minimizar el error. Además, $K(X_i, X_j) \equiv \phi(X_i)^T \phi(X_j)$ es llamada la función kernel.

La base de datos

Para elaborar el algoritmo de predicción de la estructura secundaria de proteínas es necesario contar con una buena colección de secuencias de péptidos (cadenas de texto que representan la estructura primaria), los cuales se deben usar para poder enseñarle a una máquina de aprendizaje las características que deben tener las diferentes combinaciones de segmentos de proteínas que pueden formar los diferentes motivos estructurales (cadena de texto asociada a la estructura primaria que representa la estructura secundaria). Así como también se deben usar otros péptidos para probar el correcto aprendizaje de las máquinas elaboradas. Para este trabajo se utilizaron dos conjuntos de secuencias de proteínas, uno para poder proporcionar conocimiento y otro para evaluar el conocimiento adquirido. Estos dos conjuntos son los denominados CB513 y el RS126 donde RS216 $\subseteq$ CB513.

La base de datos CB513 (Cuff and Barton, 1999) consta de 513 secuencias de proteínas en donde todas ellas tienen una longitud mayor a 30 residuos. Esta base de datos fue usada en este trabajo como el conjunto de entrenamiento para las máquinas de aprendizaje desarrolladas excluyendo las 126 secuencias incluidas en ella pertenecientes a la RS216. Mientras que de las RS126 (Rost and Sander, 1993a) se seleccionaron 126 secuencias de proteínas con una homología menor del 25%. Estas bases de datos continúan siendo hoy en día referentes de comparación para el rendimiento de los diferentes algoritmos, pues contienen información no redundante del espacio de proteínas (Sui *et al.*, 2011; Qu *et al.*, 2011; Chatterjee *et al.*, 2011), por lo que se usaron en este trabajo para validar los modelos que se construyeron.

Codificación de las secuencias

Con las dos bases de datos se obtienen los individuos que permitirán tanto implementar como verificar las máquinas de aprendizaje sin embargo, la información presente en estas bases de datos (secuencias de estructuras primarias y secundarias) no se pueden usar directamente y hay que transformar dicha información. Para poder extraer información proveniente de una secuencia de aminoácidos, es necesario convertir dicha cadena en información numérica, vectores que describan el contenido de una secuencia, de un segmento de secuencia o incluso de un aminoácido en particular. Existen diversas formas de codificar la infor-

mación presente en la estructura primaria de una proteína como diversos son también los problemas en los que se aplican estas codificaciones de la estructura primaria.

Para este trabajo se implementó una metodología para codificar las secuencias que permitiera generar vectores codificados de una dimensionalidad baja y que a su vez trataran de reducir la correlación de los vectores codificados para las diferentes clases existentes. Para ello se emplearon algunos conceptos expuestos por Yang and Wang 2003 y Ruan *et al.*, 2005.

Se necesita relacionar cada símbolo en la estructura primaria con cada símbolo en la estructura secundaria sin embargo, también se debe tener en cuenta a sus vecinos. Para poder extraer dicha información se realiza un ventaneo sobre la secuencia, el resultado de dicho ventaneo es una colección de N-gramas pertenecientes a una misma estructura primaria.

El N-grama

Para obtener información de cada motivo estructural presente en la secuencia de una proteína es necesario recorrer de f1orma adecuada dicha cadena. En Yang and Wang 2003 se muestra una metodología denominada N-grama, la cual se emplea en este trabajo para extraer los segmentos de secuencias pertenecientes a la estructura primaria. Estos segmentos deben ser posteriormente codificados en vectores de características. Para extraer estos segmentos de secuencia se debe tener en cuenta lo siguiente:

$O = \{O_1, O_2, \ldots, O_n\}$ es la estructura primaria de una proteína la cual está compuesta por una cadena de caracteres $O_i \in \Sigma$ y n es la longitud de la secuencia.

$S = \{S_1, S_2, \ldots, S_n\}$ es la estructura secundaria la cual está compuesta por otra cadena de caracteres $S_1 \in \Gamma$ de la misma longitud que O.

$C_S = \{(cs_{i,1}, cs_{f,1}), \ldots, (cs_{i,w}, cs_{f,w})\}$ es el conjunto de parejas (cs_i, cs_f) que denotan los puntos de inicio *(i)* y fin *(f)* de cada una de las subsecuencias de aminoácidos que tienen asociado un mismo símbolo $S_1 \in S$ al interior de una misma proteína siendo W el número de sub segmentos en ésta. Ver figura 3.

A partir de la cadena O y la cadena S se extraen las posiciones de inicio y fin de cada uno de los segmentos de estructura que pertenecen a un mismo motivo estructural. Cada uno de estos segmentos es una secuencia perteneciente a un motivo estructural al cual se desea codificar. El N-grama hace referencia a segmentos de N caracteres consecutivos $O_i \in \Sigma$ donde el caracter en el centro de esta subcadena es aquel al cual se desea codificar. La forma como se deben extraer dichos N-gramas de las diferentes subsecuencias se puede ver en el algoritmo 1.

Es evidente que aquellos segmentos que se encuentran al inicio y al final de la secuencia O corresponden a posiciones fuera del rango de las estructuras primaria y secundaria, estas posiciones en el N-grama deben ser reemplazadas por algún símbolo que permita su posterior codificación.

Algoritmo 1: Extracción de los N-gramas de una subsecuencia

Entrada:

- Sea $m = [(Cs_f, Cs_i) + 1]$ la longitud de una subsecuencia
- Sea $l_i = Cs_i - \left(\frac{n}{2}\right)$ el punto de inicio en O del último N-grama para una secuencia dada
- Sea l_f el punto de inicio de la última subsecuencia

Los N-gramas que se extraen de una subsecuencia dada son:

$\{[O_{l_i}, O_{l_i+1}, \ldots, O_{l_i+n}], \ldots, [O_{l_f}, O_{l_f+1}, \ldots, O_{l_f+n}]\}$ para la estructura primaria O

$\{[S_{l_i}, S_{l_i+1}, \ldots, S_{l_i+n}], \ldots, [S_{l_f}, S_{l_f+1}, \ldots, S_{l_f+n}]\}$ para la estructura secundaria S

```
O  = RNEKDSVEDRKGSENYAGTTNGGV
S  = CCCHHHHHHCCCEEEEEEEECCCC
Cs = {(1,3), (4,10), (11,13), (14,21), (22,25)}
```

Figura 3: Representación de la información contenida en la estructura primaria y secundaria para la extracción de datos pertenecientes a la secuencia.

Codificación de las subsecuencias

Los segmentos de aminoácidos que se obtienen (los N-gramas) deben ser convertidos en vectores de características que permitan plantear un algoritmo de clasificación. Las metodologías empleadas en este trabajo plantean la codificación de las secuencias con base en el VCM y las propiedades de grupo de los aminoácidos, las cuales permitirán descorrelacionar la información que se obtiene. Para convertir en vectores de características los N-gramas extraídos de una secuencia, el procedimiento a seguir es el siguiente: Primero se halla el VCM modificado (VCMM) para un N-grama dado, ver procedimiento para calcular el VCMM en el algoritmo 2, hay que tener en cuenta que se deben hacer ciertas modificaciones sobre el cálculo de dicho vector, dichas modificaciones radican en el cambio del alfabeto sobre el que se realizan los cálculos, el nuevo alfabeto debe contemplar las posiciones nulas del principio y fin de la secuencia *O* en la extracción de los N-gramas (Ruan *et al.,* 2005; Ganapathiraju *et al.,* 2004; Yang and Wang, 2003).

Algoritmo 2: Vector Composición de Momento Modificado (VCMM)

Entrada: N-gramas *Ng*

- Sea A = {A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,U,V,W,Y,*} los diferentes símbolos que pueden estar en un N-grama
- Sea *N* la longitud de las cadenas *Ng*
- Sea A_i el *i – ésimo* AA, cuando los AA se ordenan como en A
- Para un $W > 0$ donde $w \in \mathbb{Z}$, se define $(X_1^w, X_2^w, \cdots, X_{20}^w)$ como el VCMM de order *w*

$$X_i^w \frac{\sum_{j=1}^{w_i} n_{i,j}^w}{\prod_{d=0}^{w}(N-d)} \quad \text{para } i = 1, 2, \cdots, 20$$

- w_i es el número total de veces que aparece el *i – ésimo* AA en *Ng*
- $n_{i,j}$ es la *j – ésima* posición del *i – ésimo* AA en *O*

El algoritmo 2, muestra el cálculo del VCMM. Para este trabajo se utilizaron los vectores de orden cero y uno $V_{cmm} = \left(X_i^0, X_i^1\right)$ con los cuales se realiza una primera etapa de la codificación de un N-grama. Dado un V_{cmm} perteneciente a un N-grama el cual representa un segmento de secuencia en una proteína, se desea dar importancia al caracter central en el N-grama, para ello se le incorporará información estadística perteneciente a dicho caracter. También se desea descorrelacionar los N-gramas de acuerdo a dicho caracter para lo cual se emplearán las propiedades físicoquímicas de los diferentes aa. Este enfoque de codificación considera las probabilidades de que cada caracter en Σ pueda adoptar un determinado tipo estructural Γ dados los diferentes grupos biológicos a los que puede pertenecer cada aminoácido (Nelson and Cox, 2000), ver tabla 1.

Tabla 1: Clasificación de los aminoácidos de acuerdo a sus propiedades químicas.

R-grupos	Codificación	Aminoácidos
No polares, Alifáticos C_1	[1,0,0,0,0]	A, V, L, I, M
Aromáticos C_2	[0,1,0,0,0]	F, Y, G
Polares, No cargados C_3	[0,0,1,0,0]	G, S, P, T, C, N, Q
Cargados positivos C_4	[0,0,0,1,0]	K, H, R
Cargados positivos C_5	[0,0,0,0,1]	D, E

La información estadística que se puede agregar a la codificación es la probabilidad de que un aminoácido pueda adoptar una estructura Γ dada una de las clasificaciones físicoquímicas en las que se pueden clasificar cada aminoácido. Estas clasificaciones se codificarán y se les llama los R-grupos, los cuales también forman parte de la codificación, ver tabla 1. Estas probabilidades se pueden encontrar de la siguiente manera: dado un conjunto de entrenamiento Δ y un conjunto de grupos $C = \{C_1, C_2, C_3, C_4, C_5\}$ en los que se puedan clasificar los aminoácidos, la probabilidad de que un residuo $aa_i \in AA$ en C_j para $j = 1, 2,\cdots,5$ sea una hélice (E), una lámina (H) o una conformación coil (C) es, ver ecuación 2:

$$P\left(\frac{aa_i}{c_j}\right)_\Gamma = \frac{1}{N_\Gamma}\sum_{aa_i \in c_j} N_{\Gamma_j} \qquad (2)$$

Donde $P\left(aa_i/c_j\right)_\Gamma$ para $i = 1, 2, \cdots,20$ es la probabilidad de que el residuo aa_i dado un grupo C_j esté en Γ, es decir la probabilidad de que el residuo aa_i sea una hélice $P\left(O_c/c_j\right)_H$, una lámina $P\left(O_c/c_j\right)_E$, o Coil $P\left(O_c/c_j\right)_C$, en un conjunto de entrenamiento Δ. N_Γ es el número total de residuos de cada una de las diferentes conformaciones H, E y C que hay en Δ, y N_Γ es el número en el que el residuo aa_i que pertenece a el grupo C_j adopta una conformación Γ.

En este trabajo se usa el producto de Kronecker (Zwillinger 1996) entre las probabilidades encontradas, los vectores de codificación que obtienen de los R-grupos y los VCMM que se calculan a partir de los N-gramas para descorrelacionar los vectores de características (Yang and Wang, 2003). Obteniendo de esta forma la codificación de los segmentos de los aminoácidos para ser usados más adelante. La codificación de las secuencias se puede ver en la ecuación 3.

$$v = P\left(aa_i/c_j\right)_\Gamma \circ c_j \otimes V_{cmm} \qquad (3)$$

En donde $P\left(aa_i/c_j\right)_\Gamma$ es la probabilidad de que el aminoácido en la posición central de un N-grama adopte una determinada estructura secundaria dado uno de los diferentes c_j grupos en los que se pueden agrupar los diferentes residuos. C_j es la codificación del aminoácido en la posición central del N-grama (el R-grupo) y V_{cmm} es el VCMM que se calcula del N-grama, el operador (o) representa el producto elemento a elemento entre dos vectores, y el operador (⊗) representa el producto de Kronecker.

Una vez codificados los N-gramas, lo que se busca es encontrar funciones F_s, ver ecuación 4, que permitan asociar vectores V con una de las diferentes estructuras $\Gamma \in \{C,E,H\}$.

$$F_s(v) \rightarrow \Gamma \qquad (4)$$

Planteamiento del problema

Sea *VCS* el conjunto de posibles vectores codificados pertenecientes a N-gramas extraídos de secuencias de proteínas empleadas como ejemplos de entrenamiento. Sea Γ el conjunto finito de clases en las que se pueden clasificar los ejemplos *VCS* y k el tamaño de Γ (k =3, C, E, H). Formalmente el algoritmo de aprendizaje A (para este trabajo MSV) toma un conjunto de ejemplos de entrenamiento $((v_1, y_1), (v_2, y_2),...,(v_m, y_m))$ como entradas, donde $y_1 \in \Gamma$ son las etiquetas asignadas a los ejemplos de entrenamiento $v_1 \in VCS$. El objetivo del algoritmo de aprendizaje es generar una hipótesis $f : V \times \Gamma \rightarrow \mathbb{R}$ donde f pertenece al espacio de hipótesis F.

El algoritmo de clasificación a utilizar son las MSV, las cuales son clasificadores binarios y el problema de clasificación que se tiene cuenta con más de dos clases, para problemas binarios (k = 2 clases) los ejemplos son etiquetados como -1 y +1, por conveniencia. Lo que se busca es generar una hipótesis $f : V \rightarrow \{-1, +1\}$. Por tanto se debe adecuar un problema de multiclasificación en términos de problemas de clasificación binaria.

Descripción de la solución

Un problema de multi-clasificación se puede reducir a múltiples problemas de clasificación binarios los cuales se pueden resolver separadamente. Existen diversas formas de reducir un problema de multi-clasificación en problemas de clasificación binaria (Trevor and Tibshirani, 1998; Dietterich and Bakiri, 1994), uno de ellos indica que a cada clase $k \in \Gamma$ se puede asociar con una fila de una matriz de codificación $M \in \{-1,0,1\}^{k \times l}$ la cual relaciona los diferentes clasificadores binarios f_s que se pueden conformar mediante combinaciones de las clases Γ en las cuales se desea clasificar, en esta matriz se muestran las respuestas que se esperan de cada clasificador binario cuando los datos provienen de una clase en particular, ver tabla 2. Donde l representa el número de clasificadores binarios f_s que se crearon empleando un algoritmo de aprendizaje, para $S = 1,2,\cdots,l$, además l también representa el número de clasificadores en los que se puede descomponer el problema de multiclasificación. Los clasificadores binarios S se pueden desarrollar teniendo en cuenta el enfoque de emparejamiento total (Allwein *et al.,* 2000) de las k clases. Para este problema en particular se tie-

ne $l=\left(\frac{k}{2}\right)$ clasificadores ($f_1 = E|H$, $f_2 = E|C$, $f_3 = C|H$). Los clasificadores f_s son entrenados para cada columna de la matriz *M*, es decir cada columna de la matriz de codificación contempla un problema de clasificación binaria donde las etiquetas (v_i, $M(y_i.s)$) indican cuales son los ejemplos de entrenamiento para cada clasificador f_s. Los datos donde (v_i, $M(y_i.s)$) = 0 no se contemplan para el entrenamiento, pues son aquellos datos que no corresponden al clasificador binario en cuestión.

Tabla 2: Matriz de codificación M

	f_1	f_2	f_3
E	1	1	0
C	0	-1	1
H	-1	0	1

Para el entrenamiento de los diferentes clasificadores binarios, los cuales se muestran en las columnas de la matriz de codificación *M* , se debe entrenar una MSV las cuales tienen dos parámetros que se deben ajustar, C y γ, donde γ es el parámetro libre de la función Kernel que se usó, en este caso la función de base radial (RBF), ver ecuación 5.

$$K(v_i, v_j) = \exp\left(\gamma\left\|v_i - v_j\right\|^2\right), \gamma \rangle\ 0 \qquad (5)$$

Se tomó como función Kernel la RBF debido a que en diversos trabajos esta función es la que mejores resultados ha ofrecido (Shoyaib *et al.*, 2007; Chen *et al.*, 2006; Hua and Sun, 2001; Cai *et al.*, 2001). Como se tienen dos parámetros libres, el problema es encontrar qué valores deben asumir estos dos parámetros para encontrar el mejor clasificador. El objetivo es identificar (*C*,γ) tales que el clasificador sea capaz de predecir adecuadamente los datos de prueba, es decir aquellos que no se utilizan para generar el modelo. Se recomienda una combinación de los diferentes parámetros *C* y γ, para ello se toma un intervalo de estos dos parámetros *C* y *g* donde $C_{inicial}$ y C_{final} así como $\gamma_{inicial}$ y γ_{final} y denotan los límites entre los cuales se desea probar las MSV, Δ*C* y Δγ es el paso que se toma para construir los intervalos y *m* es el número de muestras que se desea tomar.

$$\Delta C = \frac{C_{inicial} - C_{final}}{m} \qquad (6)$$

$$\Delta\gamma = \frac{\gamma_{inicial} - \gamma_{final}}{m} \qquad (7)$$

Lo que se busca es encontrar la combinación (C_i,γ_j) que genere la MSV que tenga el mejor rendimiento $Q_{i,j}$ (ver tabla 3). Realizado esto con las MSV que se deben entrenar con los datos que proporciona la matriz M, se tendrán los diferentes clasificadores binarios que se van a utilizar.

Tabla 3: Malla para seleccionar los parámetros libres en una MSV

-	γ_1	γ_2	⋯	γ_m
C_1	$\overline{Q}_{1,1}$	$\overline{Q}_{1,2}$	⋯	$\overline{Q}_{1,m}$
C_1	$\overline{Q}_{2,1}$	$\overline{Q}_{2,2}$	⋯	$\overline{Q}_{2,m}$
⋮	⋮	⋮	⋱	⋮
C_m	$\overline{Q}_{m,1}$	$\overline{Q}_{m,2}$	⋯	$\overline{Q}_{m,m}$

El problema de multi-clasificación, se necesita que, para un ejemplo V_{CS}, se pueda saber a qué clase *K* pertenece. Para ello se utiliza el enfoque denominado códigos de corrección de errores de salida (por sus siglas en inglés ECOC) (Dietterich and Bakiri, 1994). Tomando *M(k)* como una fila de la matriz de codificación y sea *f(V$_i$)* el vector de las predicciones que se obtienen de los clasificadores f_s para un vector *V*.

$$f(V) = (f_1(V), f_2(V), f_3(V)) \qquad (8)$$

La forma de encontrar la clase $k \in \Gamma$ de cualquier vector *f(V)* es encontrando la fila de *M* que minimice la distancia *d(M(k), f(V))* para alguna distancia *d*. Para medir estas distancias y encontrar las clases a las cuales se le puede asociar un dato *V*, se puede realizar mediante una función de pérdida *L*, ver ecuación 9, la cual mide el margen de pérdida cuando un clasificador f_s es evaluado con un ejemplo V_i respecto a *M(y$_i$, s)*. La función *L* se evalúa sobre sobre las diferentes filas de la matriz *M*.

$$L(M(y_i, s), f_s(v_i)) = \sum_{j=1}^{l} (v_j - M_{i,j})^2 \qquad (9)$$

Se selecciona la clase *k* que más coincida con las predicciones realizadas por los diferentes clasificadores f_s,

para ello mediante el uso de la función de pérdida *L* se calculan las distancias entre el vector *f(V)* y las filas de la matriz M, con las cuales se quiere buscar a qué clase K pertenece el vector V, esta clase K es la que tenga la mínima de las distancias $\hat{y}$ ver ecuación 10. En donde $\hat{y}$ permite inferir con cuál de las filas de la matriz *M* tiene una mayor similitud el vector de resultados del clasificador *f*, con lo cual se puede también inferir en qué clase se va a clasificar el vector *V*. Este enfoque es denominado decodificación basada en pérdida (Allwein *et al.,* 2000).

$$\hat{y} = \arg\min_{k} d_L\,(M(k), f(v)) \qquad (10)$$

Implantación de la solución

Dados los 3 clasificadores f_s y la estrategia para combinarlos con el fin de generar un multiclasificador, lo que se busca es crear una metodología que permita clasificar subsecuencias de caracteres que representan aminoácidos, donde a dichas subsecuencias no se les conoce su estructura secundaria. Los pasos descritos en apartados anteriores, dicen que se deben determinar los *N-gramas* contenidos al interior de la secuencia que representa a una proteína, (ver algoritmo 2), se debe a partir de dicha subsecuencia y más concretamente del caracter que se encuentra justo en el centro de ésta, clasificar este aminoácido en uno de los grupos mostrados en la tabla 1, de donde se obtiene el R-grupo y, además de este mismo aminoácido, interesa encontrar la probabilidad $P\left(aa/c_i\right)_\Gamma$ que adopte.

Dichas probabilidades en la etapa de entrenamiento son fáciles de calcular debido a que se conoce a qué tipo de estructura pertenece un determinado aminoácido. Sin embargo, para realizar la predicción sólo se cuenta con la estructura primaria de la proteína. Por lo cual se plantea una forma de calcular dichas probabilidades con el siguiente enfoque:

Sea $\hat{O} \in O$ un N-grama, sea O_c el caracter que se ubica en la parte central de $\hat{O}$, lo que se busca es que con base en la información que se pueda extraer de $\hat{O}$ predecir a qué tipo de estructura secundaria Γ pertenece O_c. De acuerdo al enfoque mostrado en este trabajo se debe calcular la probabilidad $P\left(aa/c_i\right)_\Gamma$, donde O_c es el caracter central, O_i es la clasificación que se puede realizar sobre los caracteres Σ, ver tabla I. Sin embargo Γ_i no se conoce. Para ello se supone que O_c puede adoptar cualquiera de las estructuras secundarias Γ, se van a calcular $P\left(O_c/c_i\right)_E$, $P\left(O_c/c_i\right)_H$ y $P\left(O_c/c_i\right)_\ell$, para poder hacer uso de los clasificadores f_s, ver algoritmo 3, con lo cual se puede inferir qué tipo de estructura secundaria puede tomar O_c.

Algoritmo 3: Adecuación de los datos para predecir la estructura secundaria de un AA

Entrada: $P\left(O_c/c_i\right)_E$, $P\left(O_c/c_i\right)_H$ y $P\left(O_c/c_i\right)_C$

- Calcular $v_{CMM}, \hat{O}, o_c$,
- $\boldsymbol{V_c}$ usando la probabilidad $P\left(O_c/c_i\right)_C$
- $\boldsymbol{V_E}$ usando la probabilidad $P\left(O_c/c_i\right)_E$
- $\boldsymbol{V_H}$ usando la probabilidad $P\left(O_c/c_i\right)_H$
- Evaluar $f_1 = f_s(V_C)$,
- $f_2 = f_s(V_E)$
- $f_3 = f_s(V_H)$
- Hallar la clase Γ mediante votación utilizando f_1, f_2, f_3

Resultados

Entrenamiento y pruebas

En la validación de los modelos elaborados para predecir la estructura secundaria de una proteína se utilizó la base de datos denominada RS125, para la cual se codificaron todas sus secuencias y se evaluaron en las MSV entrenadas con la base de datos CB513. Para medir la capacidad de estas máquinas en la interpretación de los patrones que se encuentran en la estructura primaria de las proteínas y así poder comparar

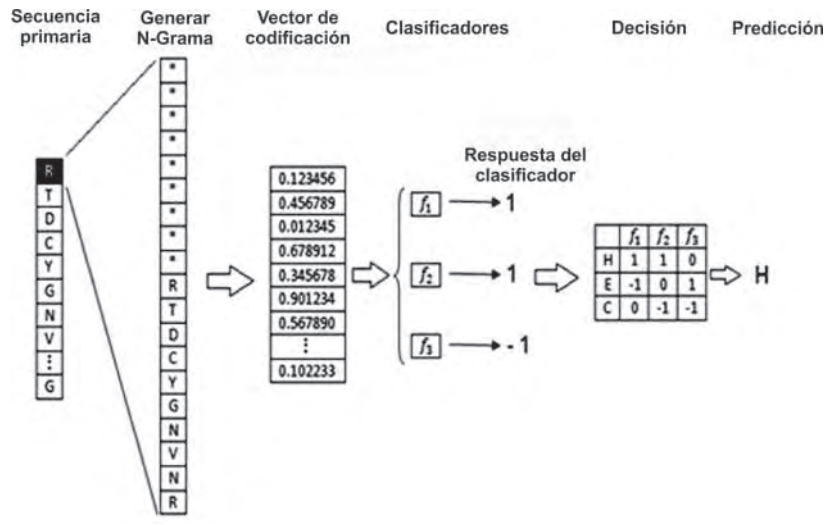

Figura 4: Esquema del proceso de predicción

su efectividad con los enfoques elaborados en otros trabajos de investigación similares.

Medidas de rendimiento

La medida de rendimiento que se utilizó para evaluar los modelos elaborados para cada algoritmo de clasificación es la que se usa usualmente y se define como.

$$Q = \frac{1}{N}\sum_{\Gamma=1}^{k} P(\Gamma) \qquad (11)$$

Donde la función P(Γ) calcula el número de aciertos en las diferentes clases Γ y N es el número de ejemplos para prueba (125). También se evaluaron por separado cada uno de los clasificadores f_s para ello se empleó el coeficiente de correlación de Mathews (CCM) (Baldi *et al.,* 2000), la sensibilidad (*Sens*) y la especificidad (*Espc*) de cada una de las máquinas creadas, ver ecuaciones 12 y 13. Donde la sensibilidad y la especificidad son mediciones probabilísticas sobre los clasificadores que se crearon. La sensibilidad mide la proporción de verdaderos positivos (aminoácidos correctamente clasificados), la especificidad mide la proporción de aspectos negativos que han sido identificados correctamente. Para las ecuaciones 12, 13 y 14 se tiene que *VP* representa a los aminoácidos que son correctamente clasificados en una clase determinada, *FN* representa a los aminoácidos que sin pertenecer a una clase se clasifican como no pertenecientes a ellas, *VN* representa aquellos aminoácidos que perteneciendo a una clase son identificados como no miembros y *FP* son aquellos aminoácidos que siendo no miembros de una clase son identificados como miembros de ella. Adicionalmente el coeficiente de correlación de Mathews proporciona una medida de la calidad de los clasificadores binarios que se crearon, un CCM de 1 indica que se construyó un clasificador binario eficiente y un *CCM* de 0 indica que el clasificador fue deficiente.

$$Sens^i = \frac{VP_i}{(VP_i + FP_i)} \qquad (12)$$

$$Espc^i = \frac{VN_i}{VN_i + FN_i)} \qquad (13)$$

$$CCM^i = \frac{VP_iVN_i - FP_iFN_i}{\sqrt{(VP_i + FP_i)(VP_i + FN_i)(VN_i + FP_i)(VN_i + FN_i)}} \qquad (14)$$

Resultados obtenidos

Luego de montar la infraestructura de los diferentes clasificadores y haberlos configurado como un solo clasificador, se procedió a evaluar su rendimiento de acuerdo a las medidas de rendimiento antes mencionadas. Primero se evaluó cómo fue el comportamiento global y también el de cada una de las clases a clasificar, ver tabla 4.

Tabla 4: Rendimiento alcanzado en el clasificador de acuerdo a las diferentes clases.

Rendimiento	Valor %
Q	66.73
Q_H	75.70
Q_E	68.66
Q_C	58.97

Se evaluó también el rendimiento de cada uno de los clasificadores f_{si} para poder tener una noción más detallada del funcionamiento del clasificador general. Ver tabla 5.

Tabla 5: Rendimiento alcanzado en los diferentes clasificadores f_s.

Medida de Rendimiento	f_{HE}	f_{HC}	f_{EC}
Sens	0.78	0.84	0.72
Espc	0.63	0.55	0.96
CCM	0.69	0.49	0.72

Se realizó la comparación del rendimiento global del clasificador con algunos trabajos realizados por otros autores, los cuales emplearon las mismas bases de datos para realizar los procesos de entrenamiento y validación. Ver tabla 6.

Tabla 6: Tabla de comparación entre diferentes autores

Autor/Método	Rendimiento *Q* %
(Garnier *et al.,* 1978)	66
Este trabajo	66.73
(Rost and Sander, 1993a)	68-72
(Zhang *et al.,* 2005)	74.86
Jpred	71.9
Bharipred	65.6

Discusión de resultados

Los resultados obtenid105os por el modelo de clasificación construido dejan entrever el grado de dificultad que existe para dar solución al problema de la predicción de la estructura secundaria de proteínas, empleando cadenas de texto para inferir los diferentes motivos estructurales.

En la tabla 4 se muestra el rendimiento global del clasificador, así como también se muestra el rendimiento que dicho clasificador obtiene con cada uno de los motivos estructurales a clasificar. En esta tabla se evidencia que la clase que mayor dificultad presenta es aquella etiquetada con el caracter *C*. Si se observa la tabla 5, el cual muestra con más detalle el clasificador. Esto debido a que se muestra cómo es el rendimiento de cada clasificador binario. Se observa en la tabla un comportamiento especial en el clasificador f_{HC} en el cual el *CCM* y la *Espc* arrojan valores que sugieren que para los grupos etiquetados con los caracteres *H* y *C* existe un desbalanceo de información, lo que lleva a que se presente este comportamiento.

El método de codificación empleado permite generar vectores de características de una dimensionalidad baja en comparación a otros métodos existentes actualmente, alcanzando valores de rendimiento similares a aquellas metodologías que por su gran dimensionalidad en sus vectores de características aseguran la descorrelación entre las diferentes clases pero hacen del proceso de clasificación una tarea más difícil, ver tabla 6.

No es posible realizar una comparación del costo computacional o la cantidad de recursos que se consumen en la clasificación debido a que otros autores no incluyen este tipo de información, pero además de obtener resultados comparables a los obtenidos por otros autores como se menciona anteriormente, el hecho de modelar el problema como un conjunto de clasificadores binarios y utilizar un vector de características de menor dimensionalidad produce reducción del costo computacional.

Los resultados obtenidos muestran un porcentaje menor de rendimiento sin embargo, cabe resaltar que en los conjuntos de validación, la base de datos RS126 está contenida dentro de la base de datos CB513. Los estudios muestran que se realiza el entrenamiento de las diferentes máquinas de aprendizaje empleadas en la literatura con el conjunto de datos CB513 y la validación con el conjunto RS126. Lo cual muestra que las máquinas en cierta forma conocen los datos con los cuales serán evaluadas. En este trabajo se excluyó el conjunto de datos RS126 del conjunto de secuencias CB513, por lo cual aseguramos que los resultados obtenidos son producto de la generalización producida por el entrenamiento de los datos y no por la memorización de estos.

Conclusiones

Los resultados obtenidos con las máquinas de soporte vectorial para la predicción de la estructura secundaria de una proteína ratifican la capacidad de esta herramienta para llevar a cabo minería de datos o predicción, en este caso el éxito de las predicciones estuvo cercano al 65%, lo cual para este tipo de problema se considera un rendimiento aceptable.

La herramienta asegura que es capaz de encontrar los hiperplanos de separación óptimos para dos conjuntos de datos cualesquiera, presentando variaciones en

el rendimiento dependiendo de cómo se ajusten los parámetros libres presentes en el modelo sin embargo, el éxito de las MSV radica en gran medida en la manera como se codifiquen dichos datos y en cómo dicha codificación asegure la menor correlación entre las clases presentes.

El uso del método de codificación de las secuencias permitió generar vectores de características de una dimensionalidad baja y la reducción de correlación entre las diferentes clases, lo que en todo problema de clasificación conlleva a una reducción del costo computacional.

El rendimiento global obtenido es comparable con los resultados obtenidos por otros autores, pero se destaca que reduce la dificultad del proceso de clasificación al crear un vector de características de muy baja dimensión, pasando de miles de características a 220 de estas.

Los rendimientos promedio de la mayoría de soluciones en esta área de investigación dejan entrever que es una problemática no resuelta aún y que los resultados obtenidos por un solo modelo de clasificación no son confiables. Por tanto, el uso de este tipo de herramientas es útil cuando se realizan las predicciones con diferentes herramientas y que con base en los resultados obtenidos por todos ellos, se puede tomar una decisión acerca de cuál podría ser el contenido estructural presente en una proteína.

Para mejorar los resultados obtenidos, se propone mejorar la selección de los parámetros libres en las máquinas de aprendizaje, así como la incorporación de información inherente al contexto biológico que pueda incrementar la precisión del método. Se propone además, que en el conjunto de características obtenido a partir de la codificación propuesta, se apliquen métodos como el análisis de componentes principales (PCA) en pro de reducir la información redundante y aún más la información presente en los vectores de codificación y así disminuir los tiempos de cómputo.

Referencias bibliográficas

Allwein E.L., Schapire R.E., and Singer Y. 2000. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*. 1:113–141.

Baldi P., Brunak S., Chauvin Y., Andersen C.A.F. and Nielsen H. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*. 16:412–424.

Cai Y.D., Liu X.J., Xu X.B. and Zhou, G.P. 2001. Support vector machines for predicting protein structural class. *BMC Bioinformatics*. 2:3.

Chatterjee P., Basu, S., Kundu, M., Nasipuri, M., and Plewczynski, D. 2011. PSP_MCSVM: brainstorming consensus prediction of protein secondary structures using two-stage multiclass support vector machines. *Journal of Molecular Modeling*. 17(9):2191–2201.

Chen C., Tian Y., Zou X., Cai P., and Mo J. 2006. Prediction of protein secondary structure content using support vector machine. *Talanta*. 71(5): 2069–2073.

Chou P.Y. 1980. Amino acid composition of four classes of proteins. Second Chemical Congress of the North American Continent, Las Vegas, Nevada.

Cortes C. and Vapnik V. 1995. Support-vector networks. *Machine Learning*. 20(3):273–297.

Cuff J.A., Barton G.J. 1999. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins: Structure, Function, and Bioinformatics. 34(4): 508-519.

Dietterich T.G. and Bakiri G. 1994. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*. 2(1): 263–286.

Ganapathiraju M.K., Klein-Seetharaman J., Balakrishnan N. and Reddy, R. 2004. Characterization of protein secondary structure. IEEE *Signal Processing Magazine*. 21(3): 78–87.

Garnier J., Osguthorpe D.J. and Robson, B. 1978. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*. 120(1): 97–120.

Hua S. and Sun Z. 2001. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology*. 308(2): 397–407.

Hubbard T.J., Park J. 1995. Fold recognition and ab initio structure predictions using hidden markov models and beta-strand pair potentials. *Proteins*. 23(8): 398–402.

Muskal S.M. and Kim, S.H. 1992. Predicting protein secondary structure content: A tandem neural network approach. *Journal of Molecular Biology*. 225(3): 713–727.

Nakashima H., Nishikawa K. and Ooi T. 1985. The folding type of a protein is relevant to the amino acid composition. Oxford Journals Life Sciences. *The Journal of Biochemistry*. 99(1): 153–162.

Nelson, D. and Cox, M. 2000. Lehninger principles of biochemestry. W.H. Freeman and company. New York. 1152.

Qu W., Yang B., Jiang W. and Wang L. 2011. HYBP_PSSP: a hybrid back propagation method for predicting protein secondary structure. *Neural computing and applications*. 21(2):337-349.

Rost B. and Sander C. 1993a. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. Proceedings of the National Academy of Sciences of the United States of America. *Biophysics*. 90:7558–7562.

Rost B. and Sander C. 1993b. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*. 232(2): 584–599.

Ruan J., Wang K., Yang J., Kurgan L.A. and Cios K.J. 2005. Highly accurate and consistent method for prediction of helix and strand content from primary protein sequences. *Artificial Intelligence in Medicine*. 35(1-2): 19–35.

Shoyaib M., Baker S., Jabid T., Anwar F. and Khan H. 2007. Protein secondary structure prediction with high accuracy using support vector machine. 10th International Conference on Computer and Information Technology. 1–4.

Sui H., Qu W., Yan B., and Wang L. 2011. Improved protein secondary structure prediction using an intelligent HSVM method with a new encoding scheme. International *Journal of Advancements in Computing Technology*. 3(3):239-250.

Trevor H. and Tibshirani. 1998. Classification by pairwise coupling. *The Annals of Statistics*. 26:451–471.

Vapnik, V.N. 1995. The nature of statistical learning theory. Springer-Verlag New York. New York.

Yang X. and Wang B. 2003. Weave amino acid sequences for protein secondary structure prediction. Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. 80–87.

Zhang G.Z., Huang D.S., Zhu, Y.P., and Li, Y.X. 2005. Improving protein secondary structure prediction by using the residue conformational classes. *Pattern Recognition Letters*. 26(15): 2346–2352.

Zwillinger, D. 1996. Standard mathematical tables and formulae. University of California. 30th Edition. CRC Press. p 812.