PREDICCIÓN DE LA ESTRUCTURA 3D DE PROTEÍNAS USANDO TÉCNICAS BASADAS EN INTELIGENCIA ARTIFICIAL.

Darío José Delgado Quintero
Ingeniero de Sistemas

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Programa de Maestría en Ingeniería de Sistemas e Informática
Bucaramanga,2011

PREDICCIÓN DE LA ESTRUCTURA 3D DE PROTEÍNAS USANDO TÉCNICAS BASADAS EN INTELIGENCIA ARTIFICIAL.

Darío José Delgado Quintero

Trabajo de grado presentado para optar el título de Magister en Ingeniería de Sistemas e Informática

Directores del Proyecto:
Henry Arguello Fuentes, *Mpe.*
Rodrigo Gonzalo Torres Sáez, *PhD*

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Programa de Maestría en Ingeniería de Sistemas e Informática
Bucaramanga,2011

Porque todo tiene su tiempo...

Agradecimientos

A mis directores Henry Arguello y Rodrigo Torres, por su adecuada orientación.

A mis compañeros del GIIB, GIBIM y GIFTEX por hacer de este trabajo de investigación algo divertido.

A mis amigos y compañeros de maestría en Electrónica y Sistemas por hacer de este tiempo algo enriquecedor.

A Paola por ser una verdadera amiga y compañera de travesía.

A Francy por escuchar.

A mis padres por su apoyo incondicional.

Glosario

- **ADN:** Nombre común dado al ácido desoxirribonucleico. Es uno de los 2 principales tipos de ácidos nucleicos. Contiene la información genética usada en el desarrollo y el funcionamiento de los organismos vivos conocidos y de algunos virus, siendo el responsable de su transmisión hereditaria.

- **ARN:** Nombre común dado al ácido ribonucleico. Es uno de los dos principales tipos de ácidos nucleicos. En los organismos celulares desempeña diversas funciones. Es la molécula que dirige las etapas intermedias de la síntesis proteica; el ADN no puede actuar solo, y se vale del ARN para transferir esta información vital durante la síntesis de proteínas (producción de las proteínas que necesita la célula para sus actividades y su desarrollo).

- **Aminoácido:** Un aminoácido, como su nombre indica, es una molécula orgánica con un grupo amino ($-NH_2$) y un grupo carboxilo (-COOH; ácido).

- **Enlace peptídico:** Es un enlace amido covalente formado entre un grupo amino ($-NH_2$) de un aminoácido y el grupo carboxilo (-COOH) de otro aminoácido.

- **Enzimas:** Se llaman enzimas a las sustancias de naturaleza proteica que catalizan reacciones químicas, siempre que sea termodinámicamente posibles.

- **Fenotipo:** Conjunto de caracteres visibles que un organismo presenta como resultado de la interacción de su genotipo y el ambiente.

- **Genoma:** Es todo el material genético contenido en las células de un organismo en particular.

- **Genotipo:** Es el contenido genético (el genoma específico) de un individuo.

- **Péptido:** Son un tipo de moléculas formadas por la unión de varios aminoácidos mediante enlaces peptídicos (Por ejemplo proteínas).

- **Polipéptido:** Es el nombre utilizado para designar un Péptido de tamaño suficientemente grande.

- **PDB:** El PDB (Protein Data BanK) es un repositorio para datos estructurales en 3D de determinadas proteínas, Estas estructuras se han encontrado mediante metodologías experimentales de resonancia magnética nuclear (RMN) y difracción de rayos X (DRX).
- **Proteínas huérfanas:** Proteínas con baja homología estructural.
- **SWISSPROT:** Base de datos para almacenar información biológica de las secuencias de las proteínas, y software de análisis bioinformático y herramientas proteomicas
- **UNIPROT:** (Universal protein) Es el recurso de proteínas universal, un repositorio central de datos sobre proteínas.

Índice general

Índice de figuras

Índice de cuadros

Resumen

TITULO: Predicción de la estructura 3D de Proteínas usando técnicas basadas en inteligencia artificial[1]

AUTOR: Darío José Delgado Quintero. [2]

PALABRAS CLAVE: Algoritmo Genético, Estructuras proteicas, Métodos computacionales, Maquinas de soporte vectorial,

La predicción de la estructura 3D de proteínas, es uno de los problemas estudiados más importantes de la biología molecular. Metodologías experimentales como la difracción de rayos X y la resonancia magnética nuclear RMN son utilizadas para inferir las estructuras de las proteínas, sin embargo no son viables de utilizar de forma generalizada debido a costos en tiempo, dinero.

Debido a la explosión de información genética a partir del éxito en proyecto del genoma, y a la importancia de conocer la funcionalidad de dicha información. Lo cual se puede buscar mediante la obtención de la información estructural de las proteínas. Se torna de vital importancia desarrollar técnicas que permitan descifran dicha información y reducir el volumen de información genética sin estudiar.

A partir de esta problemática, las técnicas computacionales se han venido acomodando como aquellas que ayudaran a reducir la brecha entre la cantidad de datos genéticos disponibles y la obtención de información estructural de las proteínas.

En el presente trabajo de investigación se abordan los conceptos fundamentales para la predicción de la estructura 3D de proteínas empleando metodologías basadas en inteligencia artificial, mediante las cuales se intenta aproximar la información del contenido estructural como su estructura 3D mediante el uso de máquinas de aprendizaje y algoritmos de optimización evolutivos.

[1] Trabajo de grado

[2] Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Henry Arguello. Codirector: Rodrigo Torres.

Abstract

TITLE:Protein 3D structure prediction using artificial intelligence techniques [3]

AUTHOR: Darío José Delgado Quintero[4]

KEY WORDS: Computational methods, genetic algorithm, Learning machine, Protein structures.

The protein structure prediction is one of the most important problems in molecular biology. Actually is considered like the Sangrail in molecular biology, per years the scientist try to solve this problem without find good solutions. Experimental methodologies as X-ray diffraction, Magnetic Nuclear Resonance MNR are used to find the protein structures. However, these techniques can not be used in a general way because are expensive in money and time.

The growing level of genetic information with the success of the genome project, and the importance to know the information encrypted in this data. This information is possible to know in the structural conformation of proteins. Is important develop new techniques to analyze the huge volume of genetic information unstudied.

According with the problematic, the computational techniques are the candidates to reduce the gap between the genetic information available and the structural information. The computational techniques, are not the most accurate but are fast and cheap. For this motive this techniques are consider like promising in this area.

In this research work is possible to see the fundamental concepts to protein structural prediction using methodologies based on artificial intelligence. Methodologies by which, we try to proximate the structural information from proteins, using techniques like the Genetic algorithms and support vector machines.

[3] Research work

[4] Faculty of Physical-Mechanical Engineerings. Systems engineering and informatics department. Advisor: Henry Arguello. Co-advisor: Rodrigo Torres

1. Introducción

1.1. Aminoácidos y proteínas

Desde un punto de vista estructural, los elementos que constituyen a las proteínas se encuentran distribuidos en bloques o unidades estructurales que se llaman aminoácidos, que unidos entre si[5] integran una estructura polimérica [1]. El análisis de un gran número de proteínas de casi todas las fuentes han mostrado que todas estas están compuestas de 20 aminoácidos estándar [2] listados en el **Cuadro 1**.

1.2. Niveles estructurales de las proteínas

En su estado natural o estado nativo, cada tipo de molécula tiene una forma o estructura tridimensional característica. Se llama conformación de la proteína, a la distribución espacial de los polipéptidos en esta, es decir, la forma como los polipéptidos se doblan en el espacio [1]. Las propiedades de una proteína en gran medida están determinadas por su estructura tridimensional [2].

[5] En una proteína los aminoácidos pueden combinarse en cualquier orden y pueden repetirse de cualquier manera, lo cual determina una secuencia especifica.

Cuadro 1: Nombres y abreviaciones de los aminoácidos

No	Una letra	Tres letras	Nombre
1	A	Ala	Alanina
2	C	Sys	Cisteína
3	D	Asp	Ácido Aspartico
4	E	Glu	Ácido Glutámico
5	F	Phe	Fenilanina
6	G	Gly	Glicina
7	H	His	Histidina
8	I	Ile	Isoleucina
9	K	Lys	Lisina
10	L	Lue	Leucina
11	M	Met	Metionina
12	N	Asn	Asparagina
13	P	Pro	Prolina
14	Q	Gln	Glutamina
15	R	Arg	Arginina
16	S	Ser	Serina
17	T	Thr	Treonina
18	V	Var	Valina
19	W	Trp	Triptófano
20	Y	Tyr	Tirosina

Tabla extraída de [3]

Los 20 aminoácidos que se encuentran comúnmente en las proteínas están unidos por enlaces peptídicos. La secuencia lineal de los aminoácidos unidos contienen la información necesaria para generar una molécula proteica con una estructura tridimensional particular. La complejidad de una estructura proteica se puede analizar de manera simplificada si se toman en cuenta 4 niveles fundamentales de organización en las macromoléculas, los cuales se denominan: estructura primaria, secundaria, terciaria y cuaternaria.

El primer nivel estructural que se puede delimitar en una proteína, está constituido por el número y la variedad de aminoácidos que entran en su composición, como por el orden (llamado también secuencia) que se disponen estos a lo largo de la cadena polipéptidica.

El segundo nivel estructural se refiere a la relación espacial que guarda un aminoácido respecto al que le sigue y al que le antecede en la cadena polipéptidica. En algunos casos el polipéptido entero, o algunas zonas de este se mantienen extendidas (Laminas-β), mientras que en otros casos se enrollan en forma helicoidal (Hélices-α). A este segundo nivel se le llama estructura secundaria[6] [1]. Ver img 1[7].

El tercer nivel estructural se refiere a la relación espacial que guardan entre si las diferentes zonas o áreas de cada cadena polipéptidica que forman una proteína. A este nivel estructural se le llama estructura terciaria [4], ver img 2 [8]. Cuando una proteína tiene más de una cadena polipéptidica, es posible que interactúen entre ellas , lo cual determina la estructura cuaternaria [5].

[6] Existe otro motivo estructural denominado Coil el cual son combinaciones de $Helices - \alpha$ y $Laminas - \beta$

[7] Imagen extraída de http://www.web.virginia.edu/Heidi/chapter5/prostru/sheets/sheets.html (Mayo 3 de 2011)

[8] Imagen: Lcanthamoeba Castellanii Profilin ib (1acf), extraida del protein data bank

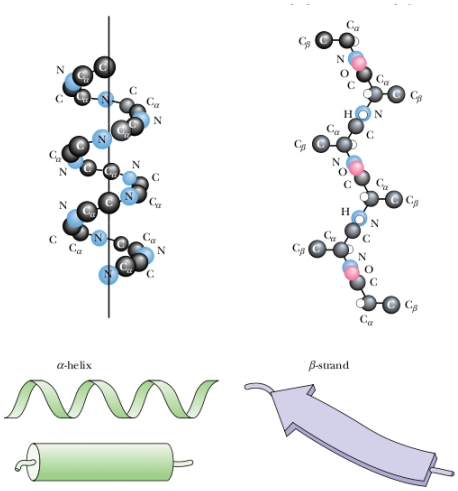

Figura 1: Estructura secundaria de una proteína

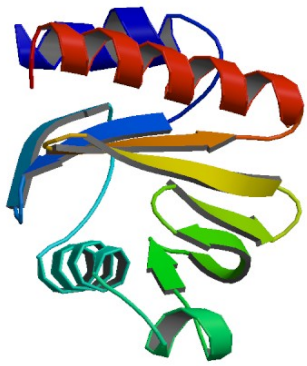

Figura 2: Estructura terciaria de una proteína

2. Clasificación de patrones, contenido estructural y estructura secundaria de las proteínas

Como se mencionó anteriormente, las proteínas son macromoléculas poliméricas constituidas por cadenas lineales de 20 diferentes aminoácidos (AA). A estas se les denomina estructuras primarias. Estas cadenas de Aminoácidos al interior de las proteínas generan dos grandes grupos estructurales: las Hélices o estructuras α (H), las Láminas o estructuras β (E). las cuales son denominadas estructuras secundarias (Fig 3). Una definición más formal para los diferentes arreglos estructurales presentes en una proteína se encuentra en el DSSP (*Diccionario de la estructura secundaria de las proteínas*) [6]. Allí se menciona que cada AA se puede asociar a uno de los ocho diferentes tipos de estructuras: "H" (Hélices-α), "G" (3-hélice, o 3_{10}-hélice) , "I" (5-hélice o π-hélice), "B" (Residuos aislados de puentes-β) , "E" (Láminas extendidas), "T" ("Hydrogen bond turn"), "S" ("bend") y "_" (Cualquier otra estructura). Sin embargo, usualmente sólo se usan tres grupos estructurales: (H, que también incluye los tipos H y G), las láminas (E, que también incluye los tipos E y B) y las conformaciones "Coil" (C, que incluye los demás tipos estructurales) [7] Dichas estructuras se pueden representar mediante cadenas de caracteres los cuales simbolizan los diferentes motivos estructurales al interior de una proteína (Fig 3).

De acuerdo con su contenido estructural, las proteínas se agrupan en 4 grandes conjuntos. En los estudios realizados por Levitt y Chothia [8], éstas se pueden considerar como: proteínas-α, las cuales tienen un contenido muy reducido de laminas-β (menos del 5 %), o se pueden catalogar

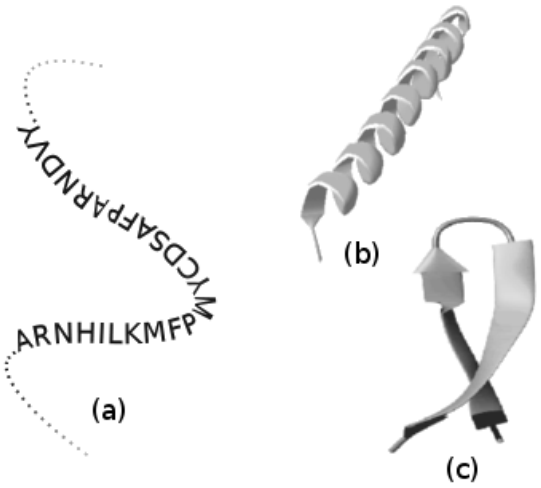

Figura 3: Representaciones estructurales de una proteína; (a)Estructura primaria ,(b) y (c) estructura secundaria, (b) Hélices-α y (c) láminas-β..

Estructura Primaria → .....RNEKDSVEDVRKGSENYAGTTNQGV.....

Estructura Secundaria → .....CCCHHHHHHHCCCEEEEEEECCCCC.....

Figura 4: Representación simplificada para la estructura secundaria de una proteína.

como proteínas-β, las cuales tienen un número reducido de hélices-α (menos del 5 %) [9]. El tercer y cuarto grupo estructural son las proteínas de tipo α/β y $\alpha+\beta$, las cuales tienen en su interior los dos tipos de estructuras, tanto las hélices-α como las laminas-β. Estas dos ultimas clases en estudios realizados por Michie y otros [10] reducen los dos grupos estructurales a uno soló denominado proteína-αβ, reduciendo los 4 grupos de proteínas a 3.

De acuerdo con lo anteriormente expuesto, predecir la estructura secundaria de una proteína puede analizarse como un típico problema de reconocimiento o clasificación de patrones, en el cual para cada AA en la estructura primaria hay que clasificar en uno de los tres tipos estructurales que estos pueden adoptar: hélices-α (H), laminas-β (E) o Coil (C).

De igual forma, predecir el contenido estructural de una proteína se puede considerar como un problema de clasificación de patrones. En lugar de identificar a que motivo estructural pertenece cada AA de una proteína en particular, se identifíca que contenido estructural puede tener una secuencia de aminoácidos completa, los cuales pueden ser clasificados de acuerdo con Levitt y Chothia [8] proteínas-α, proteínas-β, proteínas-α/β y proteínas-$\alpha+\beta$, y de acuerdo Michie y otros [10] en proteínas-α, proteínas-β y proteínas-$\alpha\beta$.

En este trabajo, se busca predecir tanto el contenido estructural de una proteína como su estructura secundaria, empleando para esto maquinas de aprendizaje que permitan inferir patrones que predigan tanto el contenido estructural general como el de cada aminoácido.

Este problema de clasificación se puede abordar mediante la clasificación de los patrones que generan las representaciones textuales de la estructura primaria y secundaria, (fig 4). 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 denominrá $\Sigma = \{A,R,N,D,C,E,Q,G,H,I,L,M,F,P,S,T\ ,W,Y,V\}$.

El segundo alfabeto, el cual representa simbólicamente los diferentes motivos estructurales que pueden tener los diferentes aminoácidos se denominará $\Gamma = \{E,H,C\}$. 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, así como inferir el contenido estructural de una proteína.

En este trabajo, se emplearon métodos computacionales, los cuales a partir de un conjunto de proteínas con sus respectivos motivos estructurales

puedan aprender a reconocer, que secuencias de aminoácidos son las más propensas a producir un determinado motivo estructural. Para este trabajo se usaron Máquinas de Soporte Vectorial (MSV) [11] [12] las cuales en trabajos como los realizados por [13], [14], [15] y [16] , han mostrado que su efectividad es mayor que otras máquinas de aprendizaje empleadas en este tipo de problemas.

2.1.Las Máquinas de Soporte Vectorial

Para abordar el problema de clasificación de patrones asociado a la predicción de la estructura secundaria de una proteína y el contenido estructural de ésta, se necesita de una herramienta matemática que permita clasificar las características extraídas de las cadenas de caracteres que representan las estructuras primarias y que se asocian con su respectiva estructura secundaria o a su contenido estructural. En este trabajo las herramientas usadas 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 de las máquinas y otro para realizar las pruebas de éstas.

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 pertenecientes al grupo de pruebas sin conocer como 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) con $i = 1, 2, ..., l$, donde $x_i \in \mathcal{R}^n$, e $y \in \{-1, +1\}^l$ y l es el número de ejemplos de entrenamiento.
Una MSV [11] [12] requiere de la solución del siguiente problema de optimización, Ver ecuación 1.

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

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 y 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.

2.2.Clasificación del contenido estructural de una proteína

2.2.1. La base de datos

Para la elaboración de modelos de clasificación basados en MSV es indispensable contar con un conjunto de datos con el cual las MSV puedan aprender y realizar una correcta clasificación. Es muy importante la selección del conjunto de datos, en este caso proteínas, para realizar el entrenamiento y el desarrollo de métodos de predicción basados en máquinas de aprendizaje. En los trabajos realizados por [17], [18], [19], [20] se realizó una selección de proteínas teniendo en cuenta la clasificación de contenido realizada por [8], en donde también se resalta la importancia de tener un buen conjunto de entrenamiento y se muestran las bases de datos utilizadas para tal fin. En este trabajo se utilizó la base de datos denominada $25pdb$ la cual se menciona en [18] [19] [20]. Esta base de datos es especial para entrenar máquinas de aprendizaje con proteínas que tengan un bajo porcentaje de homología, pues todas las proteínas contenidas en

$25pdb$ tienen menos de 25 % de homología en la secuencia. Las características de esta base de datos de acuerdo al SCOP [21] se pueden ver en el Cuadro 2.

	α	β	α/β	$\alpha+\beta$
No de Secuencias	443	443	346	441

Cuadro 2: Clasificación del 25pdb de acuerdo a el SCOP.

2.2.2. Codificación de las secuencias

Para poder extraer información proveniente de una secuencia de aminoácidos es necesario convertir dicha cadena en datos numéricos o vectores que describan el contenido de una secuencia. Existen diversas formas de codificar la información presente en la estructura primaria de una proteína. Se pueden tambien realizar diversas mediciones sobre esta secuencia, para clasificar el contenido estructural de una proteína. Entre ellas se destacan: La longitud de la secuencia de la proteína, peso molecular, punto isoelectrico [22] [23], Vector de composición [9] [17] , Vector de composición de momento [9], propiedades de grupo [24]. Para este trabajo se seleccionó la metodología de codificación denominada: Vector de composición de momentos (VCM), el cual resulta sencillo y rápido de calcular. A demás, permite extraer información del contenido de AA en la secuencia, así como la ubicación de los aminoácidos en la misma. El VCM se puede calcular dada una secuencia de aminoácidos, Ver Algoritmo 1.

El algoritmo 1 muestra el cálculo del VCM. Para este trabajo se utilizaron los vectores de orden cero y uno (x_i^0 y x_i^1), con los cuales lo que se busca es encontrar funciones f_s, ver ecuación 2, que permitan asociar vectores x con con uno de los diferentes tipos de contenido estructural $\lambda \in \{\alpha, \beta, \alpha/\beta, \alpha+\beta\}$.

$$f_s(x_1^0, x_2^0, ..., x_{20}^0, x_1^1, x_2^1, ..., x_{20}^1) \rightarrow \lambda \tag{2}$$

Algoritmo 1 Vector composición de momento VCM

Entrada: Secuencia de aminoácidos $O = \{o_1, o_2, ..., o_N\}$.

- Sea Σ={A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y} los diferentes aminoácidos contenidos en una secuencia perteneciente a una proteína.
- Sea N la longitud de la cadena O
- Sea A_i el $i-esimo$ AA, cuando los AA se ordenan como se muestra en Σ
- Para un orden $w > 0$ donde $w \in Z$

$$x_i^w = \frac{\Sigma_{j=1}^{W_i} n_{i,j}^w}{\Pi_{d=0}^{w}(N-d)} \text{ para } i = 1, 2, ..., 20$$

- Donde $n_{i,j}$ es la $j-esima$ posición del $i-esimo$ AA en O
- Wi es el número total de veces que aparece el $i-esimo\ AA$ en O

2.2.3. Descripción del problema

Sea X el conjunto de posibles vectores codificados pertenecientes a secuencias de proteínas empleadas como ejemplos. Sea λ El conjunto finito de clases en las que se pueden clasificar los ejemplos X y sea k el tamaño de λ (k=4, proteínas-α, proteínas-β,proteínas-α/β, proteínas-$\alpha+\beta$). Formalmente el algoritmo de entrenamiento (para este trabajo MSV) toma un conjunto de ejemplos de entrenamiento (x_1,y_1), ... ,(x_m,y_m) como entradas, donde $y_i \in \lambda$ son las etiquetas asignadas a los ejemplos $x_i \in X$. Usualmente, la meta del clasificador es generar una hipótesis $f : X \times \lambda \to \Re$ donde f pertenece al espacio de hipótesis F.

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

	f_1	f_2	f_3	f_4	f_5	f_6
α	1	1	1	0	0	0
β	-1	0	0	1	1	0
α/β	0	-1	0	-1	0	1
$\alpha+\beta$	0	0	-1	0	-1	-1

Cuadro 3: Matriz de codificación M

2.2.3. Descripción de la solución

Un problema de multi clasificación se puede reducir a múltiples problemas de clasificación binaria los cuales se pueden resolver separadamente. Existen diversas formas de reducir un problema de multi clasificación en problemas de clasificación binaria [25] [26]. Uno de ellos indica que a cada clase $k \in \lambda$ 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 λ con los k clases en las cuales se desea clasificar. En esta matriz se muestran las respuestas que se esperarían de cada clasificador binario cuando los datos provienen de una clase en particular. Ver Cuadro 3. donde l representa el número de clasificadores binarios f_s que se crearon empleando un algoritmo de aprendizaje (En este caso MSV), para $s = 1,2,...,l$. Además, l también representa el número de clasificadores en los que se puede descomponer el problema de multi clasificación. Los clasificadores binarios s se pueden desarrollar teniendo en cuenta el enfoque de emparejamiento total [27] de las k clases . Para este problema en particular se habla de $l = \binom{k}{2}$ clasificadores ($f_1 = \alpha|\beta$, $f_2 = \alpha|\alpha/\beta$, $f_3 = \alpha|\alpha+\beta$, $f_4 = \beta|\alpha/\beta$, $f_5 = \beta|\alpha+\beta$, $f_6 = \alpha/\beta|\alpha+\beta$). Las MSV son entrenadas 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 $(x_i, M(y_i,s))$ indican cuales son los ejemplos de entrenamiento para cada clasificador f_s, donde aquellos datos $(x_i, M(y_i,s)) = 0$ no se contemplan para el entrenamiento, pues son aquellos datos que no corresponden al clasificador binario en cuestión.

Para el entrenamiento de los diferentes clasificadores binarios, los cua-

les se muestran en las columnas de la matriz de codificación M, y para cada uno de los clasificadores, se debe entrenar una MSV la cuale tienen dos parámetros que se deben ajustar , C y γ, donde γ es el parámetro libre de la función Kernel usada.

En este caso la función kernel RBF (Función kernel de base radial), ver ecuación 3.

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2), \gamma > 0. \tag{3}$$

Se tomó como función kernel la RBF, debido a que en diversos trabajos esta función es la que mejores resultados ha ofrecido [28], [13], [14], [29]. Como se tienen dos parámetros libres, si se usa una función kernel RBF a la hora de encontrar un modelo de clasificación basado en MSV: C, que corresponde al modelo de las MSV, y γ, que corresponde a la función kernel, el problema es encontrar que valores deben asumir estos dos parámetros para que encontrar el mejor clasificador. La meta es identificar (C, γ) tales que el clasificador sea capaz de predecir adecuadamente los datos de test, es decir, aquellos que no se utilizan para generar el modelo. La forma como se escogen los datos para entrenar y probar se discutirá mas adelante. Chin-Wei et al [30] recomiendan una malla con diferentes parámetros C y γ donde se debe tener en cuenta como se van a definir dichos valores. Para ello se tomó un intervalo de estos dos parámetros, C y γ, donde $C \in [C_1, C_2, \cdots, C_m]$, y $\gamma \in [\gamma_1, \gamma_2, \cdots, \gamma_m]$, donde m es el número de muestras que se tomaron por cada parámetro. El cálculo de dichos intervalos se realiza teniendo en cuenta la ecuación 4 y 5, donde, $C_{inicial}$ y C_{final}, así como $\gamma_{inicial}$ y γ_{final}, denotan los limites entre los cuales se desea probar las MSV y ΔC $\Delta\gamma$ el paso que se toma para construir los intervalos.

$$\Delta C = \frac{C_{inicial} - C_{final}}{m} \tag{4}$$

$$\Delta\gamma = \frac{\gamma_{inicial} - \gamma_{final}}{m} \tag{5}$$

Lo que se busca es encontrar la pareja (C_i, γ_i) que genere la MSV que tenga el mejor rendimiento $Q_{i,j}$ Ver Ecuación 10 y Ver Cuadro 4. Al reali-

	γ_1	γ_2	.....	γ_m
C_1	$Q_{1,1}$	$Q_{1,2}$	.....	$Q_{1,m}$
C_2	$Q_{2,1}$	$Q_{2,2}$	.....	$Q_{2,m}$
$\vdots$	$\vdots$	$\vdots$	.....	$\vdots$
C_m	$Q_{m,1}$	$Q_{m,2}$	.....	$Q_{m,m}$

Cuadro 4: Malla para seleccionar los parámetros libres en una MSV

zar esto con 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.

Retornando al problema de multi clasificación, se necesita que, para un ejemplo x se pueda saber a que clase k pertenece. Para ello se utilizó los códigos de corrección de errores de salida (por sus siglas en inglés ECOC) [26]. Se toma $M(k)$ como una fila de la matriz de codificación, y $f(x)$ como el vector de las predicciones que se obtienen de los clasificadores f_s , ver ecuación 6, para una instancia x, se tiene que.

$$f(x) = (f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x)) \tag{6}$$

La forma de determinar a que clase $k \in \lambda$ de cualquier vector $f(x)$, es encontrando la fila de M que minimice la distancia $d(M(k), f(x))$ para alguna distancia d. Para medir estas distancias y encontrar las clases a cuales se le puede asociar un dato x, se puede realizar mediante una función de perdida L, ver Ecuación 7. Esta mide el margen de pérdida cuando un clasificador f_s es evaluado con un ejemplo x_i respecto a $M(y_i, s)$. La función L se evalúa sobre los diferentes filas de la matriz M.

$$L(M(y_i, s), f_s(x_i)) = \sum_{j=1}^{l} (x_j - M_{i,j})^2 \tag{7}$$

La idea es escoger la clase k que más coincida con las predicciones realizadas por los diferentes clasificadores f_s. Para ello se calculan mediante el uso de la función de perdida L las distancias entre el vector $f(x)$ y las filas de la matriz M, Ver Ecuación 8, con las cuales se quiere buscar a que clase

k pertenece el vector x. Esta clase k es la que tenga la distancia mínima, Ver Ecuación 9. Esta ecuación permite encontrar la fila de la matriz M que al compararla con el vector f tiene la menor distancia, y por tanto es la clase en la cual se va a clasificar un vector x. Este enfoque es denominado decodificación basada en pérdida [27].

$$d_L(M(k), f(x)) = \sum_{s=1}^{l} L(M(k,s) f_s(x)) \tag{8}$$

La clase k predicha $\hat{y} \in \{1, 2, ..., k\}$ es:

$$\hat{y} = \arg\min_{k} d_L(M(k), f(x)) \tag{9}$$

2.2.4. Resultados y discusión

2.2.4.1. Entrenamiento y pruebas

Para verificar el rendimiento de los clasificadores, el test de Jackknife es una técnica altamente usada, y además, es considerada como una de las más rigurosas [31]. En esta técnica, cada proteína en la base de datos es apartada del resto para ser usada como ejemplo de test, utilizando demás proteínas como ejemplos de entrenamiento, haciendo esto con cada una de las proteínas en la base de datos.

Otra forma de medir el rendimiento de los clasificadores es empleando una metod- ología de validación cruzada (VC) y usando para ello 10 conjuntos para evaluar la precisión de este método. La elección de los conjuntos se realiza de forma aleatoria, donde 9 de ellos se toman para entrenar y uno para realizar pruebas, Ver figura 10. Este proceso se repite con cada uno de los conjuntos que se generan tomando un nuevo conjunto de entrenamiento y un nuevo conjunto de test.

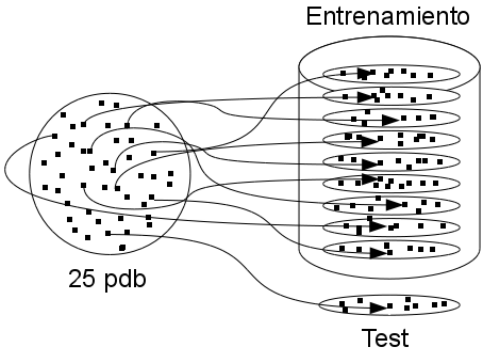

Figura 5: 10 conjuntos para realizar la validación cruzada.

2.2.4.2. Medidas de rendimiento

Generalmente, se usa la medida de rendimiento para cada algoritmo de clasificación y se define como.

$$Q = \frac{1}{N} \sum_{\lambda=1}^{k} P(\lambda). \tag{10}$$

Donde la función $P(\lambda)$ calcula el número de aciertos en las diferentes clases λ, y N es el número de ejemplos para el test. De acuerdo con el procedimiento para el método de VC, la base de datos se divide en 10 subconjuntos de igual tamaño. A su vez, se toma cada subconjunto como un subconjunto de test con el cual se espera evaluar el modelo de predicción. El resto de subconjuntos se usan para construir el clasificador. Para el caso del test de Jackknife se toma una secuencia para realizar el test y se usa el resto de secuencias para entrenar, y el mismo procedimiento se realiza con cada una de las secuencias en la base de datos. La media y la desviación estándar de la precisión de todos modelos que se realicen indicarán el rendimiento de la predicción, y se definen como sigue, Ver ecuación 7:

$$\bar{Q} = \sum_{i=1}^{\Omega} \frac{Q_i}{\Omega}, \; S = \sqrt{\sum_{i=1}^{\Omega} \frac{(Q_i - \bar{Q})^2}{(\Omega - 1)}} \tag{11}$$

Clasificador	$\bar{Q}(\%)$	**S**	**Sens**	**Espc**	**CCM**
$\alpha\|\beta$	81.73	8.48	0.819	0.818	0.637
$\alpha\|\alpha/\beta$	80.31	8.97	0.862	0.742	0.609
$\alpha\|\alpha+\beta$	73.09	8.16	0.76	0.69	0.452
$\beta\|\alpha/\beta$	77.47	7.36	0.78	0.75	0.541
$\beta\|\alpha+\beta$	65.54	17.45	0.68	0.61	0.2908
$\alpha/\beta\|\alpha+\beta$	67.50	10.50	0.641	0.72	0.354

Cuadro 5: Resultados de los diferentes clasificadores binarios

Test	α	β	α/β	$\alpha+\beta$	Total	CCM_{α}	CCM_{β}	$CCM_{\alpha/\beta}$	$CCM_{\alpha+\beta}$
VC	67.34	64.68	58.03	18.99	53.97	0.52	0.44	0.53	0.13
Jackknife	68.04	63.82	59.29	19.05	54.30	0.51	0.43	0.49	0.11

Cuadro 6: Resultados obtenidos con los test de Validación Cruzada (VC) y Jackknife

Método	α	β	α/β	$\alpha+\beta$	**Total**	**Referencias**
MSV con kernel polinomial	50.1	49.4	28.8	29.5	34.2	[18]
regresión logística multinomial	56.2	44.5	41.3	18.8	40.2	[32]
Arboles aleatorios	58.7	47.0	35.5	24.7	41.8	[33]
Tri-péptidos Información discrepancia	45.8	48.5	51.7	32.5	44.7	[34]
Arboles de decisión y logitBoost	56.9	51.5	45.4	30.2	46.0	[35]
Di-péptidos Información discrepancia	59.6	54.2	47.1	23.5	47.0	[36]
MSV con kernel polinomial	61.2	53.5	57.2	27.7	49.5	[33]
Clasificadores Binarios	68.0	63.8	59.3	19.0	54.3	**Este trabajo**
Regresión logística multinomial	69.91	61.6	60.1	38.3	57.1	[18]
Tri-péptidos específicos	60.6	60.7	67.9	44.3	58.6	[20]
MSV con Kernel RBF	69.7	62.1	67.1	39.3	59.5	[19]
MSV con kernel polinomial	77.4	66.4	61.3	45.4	62.7	[37]

Cuadro 7: Comparación de los resultados obtenidos en este trabajo con otros métodos que emplearon la base de datos 25pdb y como metodología de evaluación el test de Jackknife.

Método	α	β	α/β	$\alpha+\beta$	**Total**	**Referencias**
Regresión logística multinomial	56.9	44.2	42.2	17.7	40.2	[18]
Arboles aleatorios	53.5	51.0	37.6	22.0	41.2	[33]
Arboles de decisión y logitBoost	51.09	53.7	46.5	32.4	46.1	[35]
MSV con kernel Gaussiano	67.9	59.1	58.1	27.7	53.0	[29]
Clasificadores Binarios	67.3	64.6	58.3	18.9	53.9	**Este trabajo**
Regresión logística multinomial	69.1	60.05	59.5	38.1	56.7	[18]
Regresión logística multinomial	69.9	65.3	66.5	38.4	60.0	[19]
MSV kernel polinomial	77.7	66.8	60.7	45.4	62.8	[32]
SCPRED	92.8	80.6	74.3	71.4	80.1	[38]

Cuadro 8: Comparación de los resultados obtenidos en este trabajo con otros métodos que emplearon la base de datos 25pdb y como metodología de evaluación el test de VC.

Donde Q_i es la precisión de la $i-esima$ evaluación y Ω es el número de validaciones cruzadas que se van a realizar. La precisión total o la media de la precisión, la sensibilidad ($Sens^i$), la especificidad ($Espc^i$) y el coeficiente de correlación de Matthews (CCM^i), de cada uno de los conjuntos que se evalúan, son otras formas de evaluar la precisión. Donde la sensibilidad y la especificidad , Ver Ecuación 12, son mediciones probabilísticas sobre los clasificadores que se crearon. La sensibilidad mide la proporción de verdaderos positivos (proteínas correctamente clasificadas) que estén correctamente identificados como tales. La especificidad mide la proporción de aspectos negativos que han sido identificados correctamente. Para le ecuación 12 y 13 se tiene que VP representa a las proteínas que son correctamente clasificadas en una clase determinada. FN representa a las proteínas que sin pertenecer a una clase se clasifican como no pertenecientes a ellas. VN representa aquellas proteínas que perteneciendo a una clase son identificadas como no miembros y FP son aquellas proteínas que siendo no miembros de una clase son identificadas como miembros de ella. Adicionalmente, el coeficiente de correlación de Mathews, Ver Ecuación 13, 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.

$$Sens^i = \frac{VP_i}{(VP_i + FP_i)}, \; Espc^i = \frac{VN_i}{(VN_i + FN_i)} \tag{12}$$

$$CCM^i = \frac{VP_i VN_i - FP_i FN_i}{\sqrt{(VP_i + FP_i)(VP_i + FN_i)(VN_i + FP_i)(VN_i + FN_i)}} \tag{13}$$

2.2.4.3. Resultados obtenidos

Para la creación del clasificador de contenido estructural se crearon 6 diferentes clasificadores binarios, los cuales combinan las diferentes k clases en las que se debe clasificar una secuencia de aminoácidos. Para las pruebas de VC se calcularon los promedios de la precisión , la varianza, la sensibilidad, especificidad y el coeficiente de correlación de Matthews.Todos estos valores dan una idea del rendimiento de cada uno de los

clasificadores binarios que se crearon para luego generar el clasificador de contenido estructural. Para cada uno de los k clasificadores se obtuvieron los siguientes resultados, Ver cuadro 5.

Con cada uno de los clasificadores binarios se construye el multiclasificador con el cual se encuentra el contenido estructural de una proteína. Al entrenar los clasificadores se calcularon las medidas de rendimiento que el clasificador de contenido estructural tiene para cada una de las *k* clases $(\alpha, \beta, \alpha/\beta, \alpha+\beta)$, así como el rendimiento total que este alcanzó. El cálculo de las medidas de rendimiento se realizaron mediante la utilización de los test de Jackknife y validación cruzada. Los resultados que se obtuvieron se pueden ver en el cuadro 6.

Los clasificadores desarrollados en este trabajo se compararon con otros clasificadores que emplearon la base de datos 25pdb para ser entrenados , la comparación se realizó mediante la utilización de los test de Jackknife y los test de validación cruzada. Los resultados de la comparación de métodos se puede observar en los cuadros 7 y 8.

Con lo referente al rendimiento de los modelos, al comparar el clasificador que se implementó en este trabajo el cual para el test de Jackknife alcanzó un rendimiento del 54.3 % se pudo observar que se alcanzaron rendimientos mejores que otros trabajos. Estos trabajos se diferenciaban unos de otros más que en la máquina de aprendizaje o técnica que se utilice para clasificar, en la forma como se lleva a cabo la codificación de la secuencia de aminoácidos, algunos de ellos como en el trabajo de [Kurgan L, 2008] pueden llegar a tener rendimientos muy buenos como se puede observar en el Cuadro 6.

2.2.5. Conclusiones

En este trabajo se desarrolló un clasificador de contenido estructural usando MSV, empleando para caracterizar la información de una proteína la metodología de codificación denominada VCM. Se logró mostrar paso a

paso como un modelo de clasificación basado en MSV puede ser utilizado en la creación de un multi-clasificador robusto para abordar el problema de la predicción del contenido estructural de una proteína. También se muestran las técnicas de validación que se deben usar en este tipo de problema en particular.

Con lo referente al rendimiento de los modelos se logro un 54.3 % empleando la prueba de Jackknife como referente de validación.

La forma como se desarrolló este trabajo permite la utilización de cualquier otra técnica de codificación de la secuencia, de modo tal se espera que al utilizar otros tipos más sofisticados de codificación bajo la misma metodología de clasificación se obtengan resultados mejores.

2.3.Predicción de la estructura secundaria de proteínas

2.3.1. La base de datos

Al igual que en el problema de predecir el contenido estructural de una proteína, predecir en detalle el motivo estructural que tiene cada aminoácido necesita de un conjunto de datos que permita entrenar las máquinas de aprendizaje a utilizar. 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). Estos 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). Se deben también usar otros péptidos para probar el correcto aprendizaje de las máquinas elaboradas y así verificar el correcto aprendizaje de éstas. 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 $RS126 \subseteq CB513$.

La base de datos CB513 [39] 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 *RS*126. Por otro lado, en las RS126 [40] se seleccionaron 126 secuencias de proteínas con una homología menor del 25 %, la cual se uso para validar los modelos que se construyeron.

2.3.2. 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 puede usar directamente.

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 información 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 dimencionalidad 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 [24] y [9].

En este caso, se necesita relacionar cada símbolo en la estructura primaria con cada símbolo en la estructura secundaria. Sin embargo, se debe

tener en cuenta tambien a sus vecinos. Para poder extraer dicha información, se realiza un barrido sobre la secuencia. El resultado de dicho barrido es una colección de N-gramas pertenecientes a una misma estructura primaria.

2.3.2.1. El N-grama

Para obtener información de cada motivo estructural presente en la secuencia de una proteína es necesario recorrer de forma adecuada dicha cadena.
En [24] 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:

- $O = \{o_1, o_2, \cdots, o_n\}$ es la estructura primaria de una proteína la cual está compuesta por una cadena de caracteres $o_i \in \Sigma$ y donde n es la longitud de la secuencia.
- $S = \{s_1, s_2, \cdots, s_n\}$ es la estructura secundaria, la cual está compuesta por otra cadena de caracteres $s_i \in \Gamma$ de la misma longitud que O.
- $C_s = \{(cs_{i,1}, cs_{f,1}), \cdots, (cs_{i,w}, cs_{f,w})\}$ 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 simbolo $s_i \in S$ al interior de una misma proteína siendo w el número de sub segmentos en ésta. Ver figura 6.

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 pertenenciente a un motivo estructural al cual se desea codificar.

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

Figura 6: Representación de la información contenida en la estructura primaria y secundaria para la extracción de datos pertenecientes ala secuencia.

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 2.

Algoritmo 2 Extracción de los N-gramas de una subsecuencia

Entrada: O, S, C_s.

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

-Sea l_f el punto de inicio de la ultima subsecuencia.

Los N-gramas que se extraen de una subsecuencia dada son:
$\{[o_{l_i}, o_{l_i+1} \cdots, o_{l_i+n}], \cdots, [o_{l_f}, o_{l_f+1}, \cdots, o_{l_f+n}]\}$ Para la estructura primaria O.
$\{[s_{l_i}, s_{l_i+1}, \cdots, s_{l_i+n}], \cdots, [s_{l_f}, s_{l_f+1}, \cdots, s_{l_f+n}]\}$ para la estructura secundaria S.

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 remplazaas por algún símbolo que permita su posterior codificación.

2.3.2.2. 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 clasificacón. Las metodologías empleadas en este trabajo plantean la codificación de las secuencias con base en el VCM y las propie-

dades 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 3) 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. [9], [7], [24].

Algoritmo 3 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,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 $i-esimo$ AA, cuando los AA se ordenan como en A
-Para un $W > 0$ donde $w \in \mathbb{Z}$, se define $(x_1^w, x_2^w, ..., x_{20}^w)$ como el VCMM de orden w

$$x_i^w = \frac{\Sigma_{j=1}^{w_i} n_{i,j}^W}{\Pi_{d=0}^{w}(N-d)} \text{ para } i = 1,2,...,20$$

-w_i es número total de veces que aparece el $i-esimo$ AA en Ng
- $n_{i,j}$ Es la $j-esima$ posición del $i-esimo$ AA en O

El algoritmo 3, muestra el cálculo del VCMM. Para este trabajo se utilizaron los vectores de orden cero y uno $V_{cmm} = (x_i^0, x_i^1)$, con los cuales se realiza una primera etapa de la codificación de un N-grama.

Dado un V_{cmm} pertenenciente a un N-grama, el cual representa un segmento de secuencia en una proteína, se desea dar importancia al carácter central en el N-grama. Para ello se le incorporara información estadística perteneciente a dicho caracter. También se desea descorrelacionar los N-gramas de acuerdo con dicho caracter para lo cual se emplearán las propiedades físico químicas de los diferentes AA. Este enfoque de codificación considera las probabilidades de que cada caracter en Σ pueda adoptar un

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 Negativos c_5	[0,0,0,0,1]	D,E

Cuadro 9: Clasificación de los Aminoácidos de acuerdo con sus propiedades químicas.

determinado tipo estructural Γ dado los diferentes grupos biológicos a los que pueden pertenecer cada aminoácido [41]. Ver cuadro 9.

La información estadística que se puede agregar a la codificación, es la probabiliad de que un aminoácido pueda adoptar una estructura Γ dado una de las clasificaciones físico químicas 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 cuadro 9. 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 que un residuo $aa_i \in AA$ en c_j para $j = 1, 2, ..., 5$ sea una Hélice (E), una Lámina (H) o una conformación Coil (C) es:

$$P(aa_i/c_j)_\Gamma = \frac{1}{N_\Gamma} \sum_{aa_i \in c_j} N_{\Gamma_i} \tag{14}$$

Donde $P(aa_i/c_j)_\Gamma$ para $i = 1, 2, 3, ..., 20$ es la probabiliad 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 $P(aa_i/c_j)_H$, una lámina $P(aa_i/c_j)_E$ ó Coil $P(aa_i/c_j)_C$ 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_i} 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 [42], para relacionar las probabilidades calculadas con la ecuación 14, los R-grupos, y los VCCM calculados a partir de los N-gramas. Además de utilizar el producto de Kronecker para relacionar la mediciones echas, también se utiliza para descorrelacionar los vectores de características [24] en los 5 grupos mostrados en el cuadro 9. Obteniendo de esta forma la codificación de de los segmentos de los aminoácidos para ser usados más adelante. La codificacón de las secuencias se puede ver en la ecuación 15.

$$v = P(aa_i/c_j)_\Gamma \circ c_j \otimes V_{cmm} \tag{15}$$

En donde $P(aa_i/c_j)_\Gamma$ es la probabilidad 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 puede 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 ($\circ$) representa el producto elemento a elemento entre dos vectores , y el operador $\otimes$ representa el producto de Kronecker.

Una vez codificados los N-gramas, lo que se busca es encontrar funciones F_s, Ver ecuación 16, que permitan asociar vectores v con una de las diferentes estructuras $\Gamma \in \{C, E, H\}$.

$$F_s(v) \rightarrow \Gamma \tag{16}$$

2.3.3. Planteamiento del problema

Sea V el conjunto de posibles vectores codificados pertenecientes a N-gramas extraidos de secuencias de proteínas empleadas como ejemplos de entrenamiento. Sea Γ el conjunto finito de clases en las que se pueden clasificar los ejemplos V y k el tamaño de Γ (k=3, C, E, H). Formalmente el algoritmo de aprendizaje (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 $y_i \in \Gamma$ son las etiquetas asignadas a los ejemplo de entrenamiento ($v_i \in V$). 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 multi clasificación en términos de problemas de clasificación binaria.

2.3.4. Descripción de la solución

Al igual que en el problema de la clasificación del contenido estructural de una proteína, la predicción de su estructura secundaria es un problema de multi clasificación, que debido a las características de la maquina de aprendizaje escogida se debe plantear en términos de un problema de clasificación binaria. Para utilizar las MSV en este problema se empleó la misma metodología que se usó para la predicción del contenido estructural, creando para ello nuevamente 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 cuadro 10. Donde l representa el número de clasificadores binarios f_s que se crearon empleando un algoritmo de aprendizaje, para $s = 1, 2, ..., l$. Además l, también representa el número de clasificadores en los que se puede descomponer el problema de multiclasificacón . Para este problema en particular se tiene $l = \binom{k}{2}$ 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.

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

Cuadro 10: Matriz de codificación M.

Se deben crear tantas MSV como columnas hay en M, para lo cual se debe tener en cuenta el mismo procedimiento de adecuación de sus parámetros libres que se mostró para la predicción del contenido estructural.

El problema de multi clasificación, necesita que, para un ejemplo v, se pueda saber a que clase k pertenece. Para ello se utiliza el enfoque denominado códigos de corrección de errores de salida (por sus siglas en inglés ECOC). Se toma $M(k)$ como una fila de la matriz de codificación y sea $f(v_i)$ como el vector de las predicciones que se obtienen de los clasificadores f_s para un vector v.

$$f(v) = (f_1(v), f_2(v), f_3(v)). \tag{17}$$

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 disancias 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 18, 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.

$$L(M(y_i, s), f_s(v_i)) = \sum_{j=1}^{l} (v_j - M_{i,j})^2. \tag{18}$$

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 perdida L, se calculan las distancias entre el vector $f(v)$ y las

filas de la matriz M, con las cuales se quiere buscar a que clase k pertenece el vector v. Esta clase k es la que tenga una distancia minima $\hat{y}$, ver ecuación 19. En donde $\hat{y}$, permite inferir con cuál de las filas de la matriz M se tiene una mayor similitud el vector de resultados del clasificador f, con lo cual se puede también inferir en que clase se va a clasificar el vector v. Este enfoque es denominado decodificación basada en pérdida.

$$\hat{y} = \arg\min_{k} d_L(M(k), f(v)) \tag{19}$$

2.3.5. Implementación 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. e conoce la información referente a las subsecuencias codificadas, ver algoritmo 2. Sin embargo, se debe también, a partir de dicha subsecuencia, y más concretamente del caracter que se encuenta justo en el centro de ésta, clasificar este aminoácido en uno los grupos mostrados en el cuadro 9, de donde se obtiene el R-grupo. Además de este mismo aminoácido, interesa encontrar la probabilidad $P(aa_i/c_j)_\Gamma$ que adopte de acuerdo al R-grupo al que pertenezca.

Dichas probabilidades en la etapa de entrenamiento son fáciles de calcular debido a que se conoce a que tipo de estructura pertenece un determinado aminoácido. Sin embargo, para realizar la predicción solo 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 carácter 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 que tipo de estructura secundaria Γ pertenece o_c. Deacuerdo con el enfoque mostrado en este trabajo, se debe calcular la pro-

babilidad $P(o_c/c_j)_\Gamma$, donde o_c es el carácter central, y c_i es la clasificación que se puede realizar sobre los caracteres Σ. Ver cuadro 9. Sin embargo Γ_i no se conoce. Para ello se supone que o_c puede adoptar cualquiera de las estructuras secundarias Γ. Por tanto se calculan $P(o_c/c_j)_E$, $P(o_c/c_j)_H$, $P(o_c/c_j)_C$, para poder hacer uso de los clasificadores f_s, ver algoritmo 4, con lo cual se puede inferir que tipo de estructura secundaria puede tomar o_c, ver figura 7.

Algoritmo 4 Adecuación de los datos para predecir la estructura secundaria de un AA

Entrada: V_{CMM_i},$\hat{O}$,o_c

-Calcular: $P(o_c/c_j)_E$, $P(o_c/c_j)_H$, $P(o_c/c_j)_C$,

- v_C usando la probabilidad $P(o_c/c_j)_C$

- v_E usando la probabilidad $P(o_c/c_j)_E$

- v_H usando la probabilidad $P(o_c/c_j)_H$

-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

2.3.6. Resultados y discusión

2.3.6.1. Entrenamiento y pruebas

En la validación de los modelos elaborados para predecir la estructura secuandaria 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 su efectividad con los enfoques elaborados en otros trabajos de investigación similares.

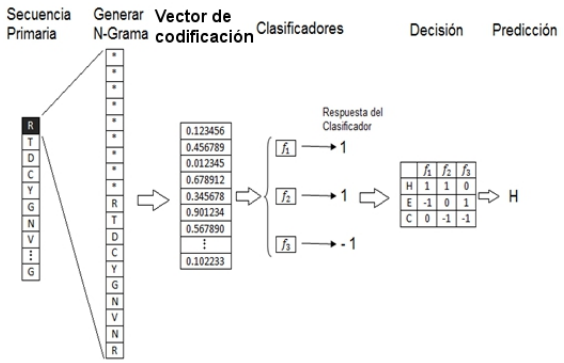

Figura 7: Esquema del proceso de predicción del contenido estructural que tiene un aminoácido en particular.

Rendimiento	Valor
Q	66.73 %
Q_H	75.70 %
Q_E	68.66 %
Q_C	58.97 %

Cuadro 11: Rendimiento alcanzado en el clasificador de acuerdo con las diferentes clases.

2.3.6.2. 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 con las medidas de rendimiento antes mencionadas.

Primero, se evaluó como fue el comportamiento global y también el de cada una de las clases a clasificar, ver cuadro 11.

Se evaluó también el rendimiento de cada uno de los clasificadores f_{s_i} para poder tener una noción más detallada del funcionamiento del clasificador general. Ver cuadro 12.

Además, 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 te entrenamiento y va-

Medidas 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

Cuadro 12: Rendimiento alcanzado en los diferentes clasificadores f_s.

Autor	**Rendimiento** Q
[43]	66 %
Este trabajo	**66.73** %
[40]	68 - 72 %
[44]	74.86 %

Cuadro 13: Tabla de comparación entre diferentes autores

lidación. Ver cuadro 13.

2.3.6.3. Discusión de los resultados

En el cuadro 11 se muestra el rendimiento global del clasificador, así como también el rendimiento que dicho clasificador alcanza 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 carácter *C*. Si se teiene en cuenta 12, se observa con más detalle el clasificador. Mostrandose el rendimiento de cada clasificador binario. En este cuadro se observa 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 cuadro 13.

2.3.7. Concluciones

Los resultados obtenidos 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.

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 cualesquiera dos conjuntos de datos, presentando variaciones en el rendimiento dependiendo de cómo se ajusten los parámetros libres presentes en el modelo. Sin embargo, el éxito de las MSV radican en gran medida en la manera como se codifiquen dichos datos y como dicha codificación asegure la menor correlación entre las clases precentes.

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. El uso de este tipo de herramientas es útil cuando se realizan las predicciones con diferentes herramientas de predicción y que con base en los resultados obtenidos por todos ellos se puede tomar una decisión acerca de cual podría ser el contenido estructural presente en una proteína.

3. Algoritmos Genéticos y la estructura 3D de las proteínas.

Computacionalmente hablando, el plegamiento de proteínas se puede ver como un problema de optimización, en el cual se busca encontrar una distribución espacial de las coordenadas que representan los átomos que componen dicha biomolécula. Se han empleado diferentes metodologías para dar solución a este problema desde un enfoque computacional. Sin embargo, realizar simulaciones del plegamiento de una proteína a partir de su secuencia primaria, empleando toda su información estructural eleva significativamente su complejidad, y compromete la viabilidad de la solución [45] [46].

Para poder abordar este problema se ha optado por versiones simplificadas de éste. Una de estas versiones simplificadas es la formulada por [47], el cual reduce la complejidad del problema. La metodogía propuesta por Dill, conocida como modelo Hidrofóbico-Polar (HP), aborda la problematica desde dos puntos diferentes, el primero de ellos es el de simplificar el cálculo de la energía, de acuerdo con [48]. Esta no se ha podido modelar adecuadamente. El segundo punto hace referencia al problema planteado por [49], en donde se muestra que uno de los principales inconvenientes es el tamaño del espacio de búsqueda de aquellas conformaciones que disipen la menor energía en el sistema.

En el modelo HP se trata de plegar una proteína bajo la premisa de que las relaciones de hidrofobicidad de los elementos presentes en la secuencia de aminoácidos tienen un efecto significativo en la energía que se libera, sin tener en cuenta algún otro factor para el cálculo energético en el proceso de plegamiento de la proteína. Se podría pensar que la simulación de

este sería relativamente fácil. Sin embargo, lo expuesto por Levintha sigue teniendo vigencia en este modelo. La introducción de un kernel geométrico que permite discretizar los grados de libertad en el movimiento que tiene una cadena de aminoácidos reduce significativamente el espacio de búsqueda de posibles soluciones. Sin embargo, el problema continúa teniendo una complejidad computacional alta. Se dice que este problema es de tipo $NP-Completo$ [50]. Esto implica, que para buscar un minímo de energía en el espacio de búsqueda sigue siendo inviable computacionalmente, pues realizar una búsqueda exhaustiva consumirá demasiado tiempo.

El hecho de que la complejidad del problema sea $NP-completo$ implica que el uso de estrategias convencionales de optimización no es una opción para su solución. Desde sus orígenes se han empleado metodologías alternas a éstas. Éstas han dado resultados aproximados que han ido mejorando a través del tiempo. Inicialmente se usaron metodologías basadas en dinámica molecular [51], las cuales tenían inconvenientes con el espacio de búsqueda quedando atrapado en mínimos locales. Se emplearon también métodos basados en simulaciones de Monte Carlo y Redes Neuronales [52] [53], y métodos deterministicos de minimización [54] los cuales permitían solucionar problemas de estancamiento en los algoritmos.
Sin embargo, las metodologías que han demostrado tener buenos resultados en este modelo han sido las basadas en algoritmos evolutivos como lo son los algoritmos genéticos [55] [56] , colonias de hormigas [57], [58], y enjambres de partículas [59] entre otros.

En este trabajo, se desarrolló un modelo de plegamiento de proteínas empleando algoritmos genéticos por ser la metodología que ha mostrado ser la más versátil para la solución de este problema. Para ello, se empleo un kernel octaedral el cual permite realizar una aproximación más real al plegamiento nativo de una proteína. Además, se incluyeron restricciones al interior de la función a optimizar las cuales permiten controlar más aún el simulado del plegamiento de una proteína.

3.1. El Modelo Hidrofóbico-Polar (HP)

El modelo HP es un modelo basado en una reprecentación espacial de una malla (en ingles *Lattice*). Son modelos simplificados de proteínas a los cuales se aplican las siguientes simplificaciones [60]: i) Cada elemento en la proteína es representado como un punto en los vértices de una malla. ii) Las posiciones de los elementos estructurales están restringidos a los puntos de la malla. iii) Las distancias entre los puntos siempre es la misma. iv) La función que describe la energía de plegamiento es simplificada.

Para este modelo, una proteína se representa como una cadena lineal de caracteres que representan aminoácidos. Cada aminoácido puede ser clasificado en uno de los siguientes dos grupos: H (Hidrofóbico o no polar) ó P (Hidrofílico o Polar)

Las conformaciones espaciales que pueda adoptar la proteína estarán embebidas en una malla bidimensional o tridimensional, la cual permite discretizar el espacio de posibles conformaciones que pueda adoptar una secuencia., reduciendo así los grados de libertad que pueda tener cada elemento estructural.

El modelo asume que la energía libre en una conformación espacial es computada entre vecinos no adyacentes, donde cada contacto de tipo H-H pose una cantidad de energía libre. Cualquier otro par de combinaciones entre elementos H o P no posee energía libre, ver figura 8. Esto está en concordancia con el literal *iv* de las simplificaciones del modelo expuestas por Backofen. Además, la función que describe la energía es sencilla.

3.1.1. La malla Octahedral

Para este trabajo se empleó un kernel octahedrico con el cual construir la malla sobre la que se proyectarían las conformaciones de las proteínas. Matemáticamente, una malla L es generada a partir de un conjunto finito de vectores $\vec{r}_1, \cdots, \vec{r}_k$ En este este trabajo $\vec{r}_i \in \Re^3$, L se define como el con-

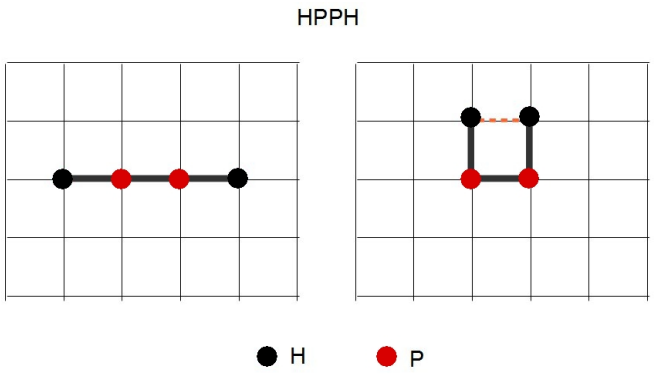

Figura 8: Esquema de dos conformaciones de cuatro aminoácidos dos de los cuales son Hidrofóbicos (Negros) y los otros dos dos son polares (Rojos), los cuales se encuentran sobre una malla 2D. La conformación de la derecha posee una cantidad de energía libre por encontrarse dos elementos Hidrofobicos como vecinos no adyacentes.

junto mínimo que contiene a $\vec{r}_1, \cdots, \vec{r}_k$, y es cerrado bajo la suma y la resta (Para cualesquiera $\vec{u}_1, \vec{u}_2 \in L$ $\vec{u}_1 + \vec{u}_2$ y $\vec{u}_1 - \vec{u}_2$ estan en L).

La malla usada en este trabajo es una llamada octahedrica, Esta tiene como unidad estructural un octahedro el cual tiene 14 caras y 12 vértices y fue desarrollada por Raghunathan y Jerniga [61]. Los vectores $\vec{r}_i$ que la generan se pueden encontrar en los trabajos de [62] [61]. Esta es una estructura de red cubica centrada en las caras (del término en ingles *face-centered cubic* FCC) en donde las conexiones entre las 12 esquinas y su centro tienen la misma longitud, ver figura 9.

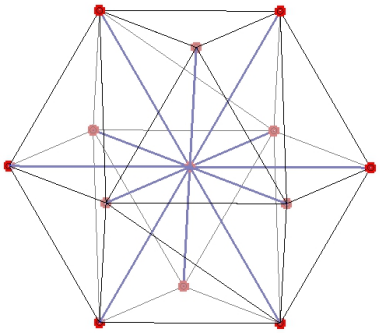

Figura 9: Unidad Octahedrica de la malla para la simulación del plegamiento de proteínas

Esta estructura permite determinar caminos a partir de un punto inicial hacia 12 posibles vectores, generando ángulos de torsión entre pares de elementos de 0°, 60°, 90°, 120° y 180°, al conectar tripletas de elementos se generan ángulos de torsión de 0°, 54,7°, 70,5°, 109,5°, 125,3° y 180°. De esta manera, se discretiza el espacio de posibles plegamientos.

Cuando se repite el conjunto de vectores de la figura 9, genera un arreglo periódico de puntos en el espacio creciendo de forma radial desde el centro, y formando así la estructura sobre la cual se puede plegar una proteína.

3.2. Planteamiento del problema

Se planteó en función del el modelo HP y la malla L, sobre la cual se desea realizar los plegamientos. En esta sección se mostrara el planteamiento matemático sobre el cual se plantea el problema a resolver. Sea $S = \{s_1, s_2, \ldots, s_n\}$ Una secuencia lineal de caracteres de la forma $\{H,P\}^*$. Sea C el conjunto de posibles conformaciones espaciales en 3D que puede adoptar la cadena S, se define a $c \in C$ como la función $c : [1,..,n] \to L$ tal que:

- $\forall\ 1 \leq i < n : (c[i]\ y\ c[i+1])$ son vecinos.
- $\forall\ 1 \leq i < j \leq n : (c[i] \neq c[j])$

Dada la secuencia $S \in \{H,P\}$ de longitud n y un plegamiento de la secuencia c en L, el cálculo de la energía libre de la conformación c se realiza de la siguiente manera.

$$E(c) = \sum_{1 \leq i+1 < j \leq n} \beta(s_i, s_j)\delta(r_i, r_j) \tag{20}$$

En donde la función β evalúa los tipos de interacciones entre los tipos de aminoácidos en la secuencia S, Ver ecuación 21. La función δ valida si

la interacción entre dos aminoácidos es una interacción de vecindad entre aminoácidos no adyacentes, Ver ecuación 22.

$$\beta(s_i, s_j) = \begin{cases} -1 \; Si \; s_i \wedge s_j = H \\ 0 \; De \; otra \; manera. \end{cases} \tag{21}$$

$$\delta(r_i, r_j) = \begin{cases} 1 \; Si \; \left\| r_i - r_j \right\| = 1 \\ 0 \; De \; otra \; manera. \end{cases} \tag{22}$$

En donde la ecuación 21, busca los aminoácidos no adyacentes que sean Hidrofóbicos. La ecuación 22 busca aquellos aminoácidos no adyacentes que sean vecinos, siendo $\left\| r_i - r_j \right\|$ la norma de la diferencia de la distancia que existe entre dos puntos en L. Si esa distancia es igual a 1, se dice que los dos puntos son vecinos.

Se dice que una proteína se pliega a la conformación con menor energía libre, por tanto, lo que se busca es minimizar la función de energía , ver ecuación 20. Encontrando de esta forma la conformación c que aproxime el plegamiento de ésta.

Siendo $E : C \to \Re$ la función que asocia un valor de energía a las posibles conformaciones espaciales $c \in C$ que pueda tener la cadena S, lo que se busca es encontrar $c^* \in C$ tal que $\forall c \in C : E(c) \geq E(c^*)$, Lo cual se puede ver mejor en la ecuación 23.

$$\begin{aligned} & \text{mín } E(c) \\ Sujeto\, a : \forall\, 1 \leq i < n \;\; & \|c_i - c_{i+1}\| = 1 \\ & \forall\, i \neq j : \; c_i \neq c_j \end{aligned} \tag{23}$$

En donde la primera restricción dice que la distancia entre dos aminoácidos adyacentes siempre debe ser la misma. Y la segunda restricción, que ningún aminoácido puede ocupar el mismo espacio en la malla que ya este ocupando otro aminoácido.

3.3. Planteamiento de la solución

Se mencionó anteriormente que el problema a resolver expuesto en la ecuación 23, es un problema de tipo $NP-Completo$. Cuando se conoce la complejidad computacional de un problema a solucionar, se tiene información valiosa acerca de cuál enfoque es el más adecuado para resolverlo, destacándose en la actualidad las técnicas heurísticas.

Desde un punto de vista computacional, todos los problemas $NP-Completos$ son igualmente complejos de resolver. Más que cualquier otra metodología, los Algoritmos Genéticos (AG) han sido empleados con éxito en la solución de problemas de este tipo más que otras metodologías.

El problema del plegamiento de proteínas empleando modelos HP se considera NP-completo en el sentido de que no se puede resolver determinísticamente en un tiempo polinómico. Luego, el éxito de los AG está en mapear un problema NP en uno que pueda ser resuelto en tiempo polinómico.

Los AG en general parten de una población inicial de individuos. Para este trabajo, se parte de una población inicial de plegamientos c_i, la cual dará paso a nuevas generaciones, a través de la ejecución de un proceso de reproducción y muerte similar al que se da en la naturaleza. Cada individuo en la población trae consigo información de sus características inherentes al problema, así como también las características de sus ancestros, y parte de esta información será pasada a su descendencia. Esto sucede debido a, que al igual que en la naturaleza, los AG realizan procesos de Selección, Cruzamiento y mutación de los individuos con el fin de ir mejorando generación tras generación sus características como población.

Al finalizar un AG, no se puede asegurar que se ha encontrado una solución definitiva al problema que se trata de resolver. Sin embargo, lo que sí se puede asegurar es que se ha encontrado una población de individuos con características mejores que aquellos individuos con los cuales se inicio

dicho algoritmo, cuya población generalmente puede ser tomada como una aproximación a la solución real del problema.

3.4. Adaptación del problema a un AG

Un algoritmo genético genérico, empleado para resolver cualquier problema de tipo $NP-Completo$, se podría representar por algunas acciones simples: La generación de una población inicial de tamaño μ, la evaluación de dicha población en una función de aptitud f, la selección, cruce y mutación de λ hijos y el posterior reemplazo de la nueva población en la población inicial, y el desarrollo de este siclo hasta que se cumplan las expectativas de rendimiento estipuladas.

Para poder generar una población de posibles plegamientos c, es necesario generar los individuos que representen dicha información espacial. Dicha información hace referencia a las coordenadas que pueda tener una conformación en el enmallado. Emplear coordenadas cartesianas [63] puede complicar el desarrollo del AG, por lo cual se debe poder expresar esta información de una manera alterna, Una de las alternativas es generar coordenadas absolutas [64] [65] para los posibles movimientos de cada uno de los elementos estructurales que representan a la proteína, aprovechando las características propias de la malla. Se sabe que la unidad estructural es un octaedro FCC, el cual se genera con 12 vectores r_i y se enumeran en la figura 3.

Las coordenadas absolutas, permiten localizar un elemento en el enmallado con relación al elemento inmediatamente anterior. Por tanto, una conformación c puede ser definida como una secuencia de movimientos a partir de un punto definido en la malla.

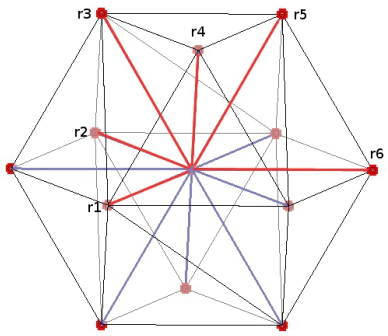

Figura 10: Vectores generadores de la malla Octahedral, se muestran 6 de estos vectores. Los otros 6 son los negativos de los que se muestran en la gráfica.

3.4.1. Diseño de los cromosomas

Suponga una secuencia $S \in \{H,P\}$ con n elementos, una posible conformación para esa secuencia es generada por un conjunto de $n-1$ doblamientos. Las coordenadas absolutas para este trabajo son los vectores r_i que generan el enmallado L. Se tienen por tanto 12 coordenadas absolutas $C_{oor} = \{r_1, r_2, r_3, r_4, r_5, r_6, -r_1, -r_2, -r_3, -r_4, -r_5, -r_6\}$

En el diseño de los cromosomas, para la generación del algoritmo genético, se emplearon coordenadas absolutas para representar una conformación c que puede adoptar una secuencia S sobre el enmallado L. Se cuenta con 12 posibles movimientos absolutos que puede adoptar cada elemento.

Un elemento de la población inicial, sera entonces la secuencia $c = \{r_1, r_2, r_3, r_4, r_5, r_6, -r_1, -r_2, -r_3, -r_4, -r_5, -r_6\}^{(n-1)}$ que representa un plegamiento de la secuencia S, ver figura 11.

Adaptar el modelo HP a un AG, implica tener en cuenta las restricciones propias del problema a resolver, ver ecuación 23, y las restricciones propias inherentes al problema biológico que subyace a esté.

En la definición de la malla y el núcleo octahedral a utilizar se definieron los posibles ángulos de torsión que puede generar éste. Sin embargo, solo

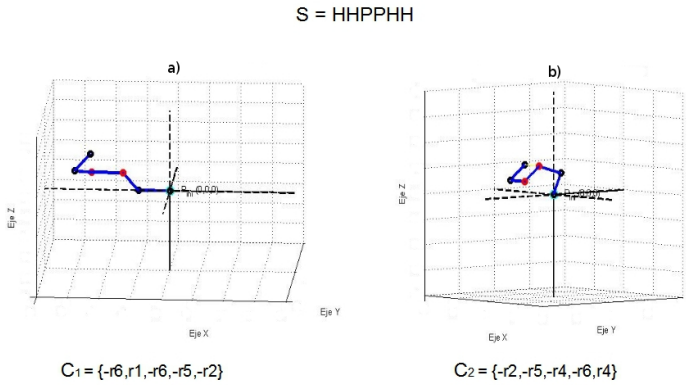

Figura 11: Para una misma secuencia $S \in \{H,P\}$, se generan dos conformaciones c_1 y c_2 a partir de coordenadas absolutas, con origen en el punto $P(0,0,0)$ (las imágenes han sido rotadas para su mejor visualización).

Vector Antecesor	120°	90°
r_1	$-r_2, r_4, -r_5, -r_6$	$r_3, -r_3$
r_2	$-r_1, r_3, -r_4, -r_6$	$r_5, -r_5$
r_3	$r_2, r_4, r_5, -r_6$	$r_1, -r_1$
r_4	$r_1, -r_2, r_3, r_5$	$r_6, -r_6$
r_5	$-r_1, r_3, r_4, r_6$	$r_2, -r_2$
r_6	$-r_1, -r_2, -r_3, r_5$	$r_4, -r_4$

Cuadro 14: Restricciones para los ángulos de torsión entre un par de elementos en el enmallado, para las contrapartes negativas de los 6 vectores mostrados en las tablas, son las mismas restricciones pero con signos contrarios.

ángulos de 90° y 120° son permitidos por ser biológicamente autorizados debido a la restricción estérica, Para ser coherentes con estas restricciones, se deben restringir los posibles movimientos que tenga un elemento teniendo en cuenta su antecesor, ver cuadro 14

Los individuos o cromosomas para el AG serán de esta manera elementos que dependerán de la longitud de la secuencia *S*, y los genes que componen dicho cromosoma, serán cada una de las coordenadas absolutas que representa su plegamiento en la malla, ver figura 12.

S = HHP ..... HPH

-	r_3	r_5	.....	$-r_6$	r_1	$-r_5$
i_0	i_1	i_2	.....	i_{n-2}	i_{n-1}	i_n

Figura 12: Representación de un cromosoma para una conformación c de longitud $n-1$ que representa el plegamiento de una secuencia S empleando coordenadas absolutas.

3.4.2. Diseño de la función de aptitud

No todas las conformaciones c que se pueden generar a partir de las coordenadas r_i son conformaciones permitidas para el problema a resolver. Si se observa en la definición del problema, la ecuación 23 tiene dos restricciones sobre las conformaciones. Estas señalan que todo elemento vecino debe estar a la misma distancia y ningún elemento sobre la malla puede ocupar el mismo espacio que otro elemento. Estas dos restricciones tienen que incluirse en el desarrollo del AG. Para ello se empleó la función de aptitud.

La función de aptitud, es la encargada en este trabajo de hacer cumplir las restricciones y de minimizar la energía del plegamiento. Sin embargo, para utilizar el AG se debe plantear este problema en términos de un problema de maximización. Para ello se planteó la siguiente función de aptitud, ver ecuación 24.

$$f(c) = \begin{cases} \frac{|E(c)|}{\beta} \; Si\, \Gamma(c) = 0 \\ -\alpha\Gamma(c) \; De\, otra\, manera. \end{cases} \tag{24}$$

En donde $|E(c)|$ representa el valor absoluto de la energía libre para un plegamiento c, ver ecuación 20. El cual crece a medida que la energía disminuye. β se denomina parámetro de compactación, el cual se calcula con base a la distancias máxima y promedio de los diferentes nodos en la malla y su promedio, ver algoritmo 5. Esta valor hace crecer la función de aptitud cuando es un valor pequeño, es decir cuando la configuración de la

proteína es compacta, y tiene el efecto contrario cuando el plegamiento se encuentra disperso por la malla.

Algoritmo 5 Cálculo del parámetro de compactación β

Entrada: dada una conformación c

- Convertir las coordenadas absolutas c en coordenadas cartesianas χ_c
- Calcular las distancias que hay entre todos los elementos de la conformación.

$\vec{d}$ Son las distancias $\|\chi_c(i) - \chi_c(j)\| \; \forall i \neq j$

- Encontrar el argumento máximo del vector de distancias $\vec{d}$ calculado.

$D_{max} = \arg\min \vec{d}$

- Calcular las distancias promedio.

$\bar{D} = \frac{1}{longitud(\vec{d})} \sum_{i=1}^{longitud(\vec{d})} \vec{d}$

Salida: $\beta = \frac{D_{max} - \bar{D}}{2}$

La función $\Gamma(c)$ representa las penalizaciones existentes en la configuración c, brinda información acerca del número de solapamientos de los elementos en la configuración, ver ecuación 25.

$$\Gamma(c) = \sum_{i \neq j}^{n} \xi(\|r_i - \chi_c(j)\|) \tag{25}$$

En donde la función $\xi(\|r_i - \chi_c(j)\|)$ determina si existe solapamiento entre cualesquiera dos elementos en la conformación c, ver ecuación 26.

$$\xi(\|r_i - \chi_c(j)\|) = \begin{cases} 1 \; Si \; \|r_i - \chi_c(j)\| = 0 \\ 0 \; De \; otra \; manera. \end{cases} \tag{26}$$

La constante $\alpha \in [0,1]$ regula la intensidad de la penalización a el valor de energí libre.

La función de aptitud se presenta en partes para poder dividir en dos los rendimientos de la población. Aquellos elementos con valor positivo serán

aquellos cuya conformación pertenece al espacio de soluciones factibles. Aquellas cuyo valor de aptitud sea negativo indica que la conformación es no factible por tener solapamientos en su representación espacial. Esto permite adicionar las restricciones inherentes al problema.

Se puede resaltar que la función de aptitud $f(c)$, es una función a ser maximizada. Cuando f se maximiza permite a su vez minimizar energía libre en el plegamiento de una conformación c, A sí mismo, la función f permite penalizar las conformaciones no factibles, permitiendo así, mediante el empleo de un AG, encontrar una población de conformaciones c que busque maximizar f.

3.4.3. Operador de selección

Una vez evaluados los μ individuos de la población en la función de aptitud f se debe seleccionar parte de esta población para conformar las nuevas generaciones del siclo evolutivo. En este trabajo, se seleccionó por el método de la ruleta un número de individuos igual a la población inicial, dando una mayor probabilidad de selección a aquellos individuos cuyo valor de aptitud es más alto, individuos con los que se crearán las nuevas poblaciones en cada iteración. También se selecciona aquel individuo cuyo valor de aptitud destaque entre los demás de la población, Este individuo conservará sus caracteristicas y pasará intacto a la siguiente generación, asegurando de esta manera que se conserve siempre una tendencia a hacer crecer el valor de aptitud y sin que este disminuya.

3.4.4.Operador de cruce

Una vez seleccionada la población que participará en la creación de la siguiente generación de individuos, se procede a compartir su información genética entre sí (compartir su información estructural). La metodología de cruce se realizará de acuerdo con la probabilidad de que se crucen dos individuos. Si esta probabilidad no es superada los dos individuos pasarán

intactos a la siguiente generación, de lo contrarios compartirán información genética con su correspondiente pareja. La metodología de cruce se puede observar en la figura 13.

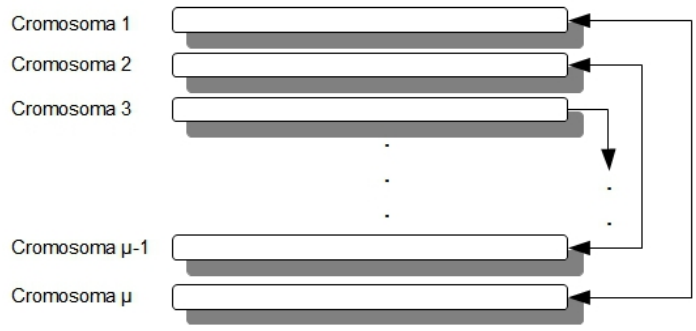

Figura 13: Cromosomas ordenados de forma descendente de acuerdo a su valor de aptitud, se cruzaran cada uno de ellos con aquellos cromosomas con menor valor de aptitud.

3.4.5.Operador de mutación

En el proceso de mutación, en el cual, lo que se busca es cambiar de manera aleatoria parte de la información para un individuo en la población escogido al azar buscando asimilarse a los procesos naturales de la evolución. En este trabajo, este operador trae consigo un proceso adicional, que es la recuperación de la factibilidad debida a posibles problemas introducidos por el gen mutado. En el cuadro I, se muestran los posibles movimientos que puede tener un aminoácido a partir de su antecesor. Sin embargo, al cambiar al azar cualquiera de los posibles movimientos (coordenadas absolutas) en c se puede presentar que el gen mutado genere una infactibilidad en la nueva conformación. Por tal motivo, se hace necesario incluir un operador de recuperación para retornar la factibilidad de la conformación en caso de precentarse alguna. Este proceso se ve reprecentado en la figura 14.

Algoritmo 6 Algoritmo genético para el plegamiento de una proteína

Entrada: Secuencia $S \in \{H,P\}$

- Crear la población inicial a partir de S con μ elementos.
- Reparación de la población inicial.
- Evaluación de la población sobre la función de aptitud f.

mientras No se cumpla el criterio de parada **hacer**

- Seleccionar λ individuos de la población.
- Cruzar los λ individuos de la población.
- Repar los nuevos hijos
- Mutar los hijos producto del cruce.
- Reparar los hijos mutados.
- Reemplazar y crear nueva población de μ elementos.
- Evaluación de la población sobre la función de aptitud f.

fin mientras

Salida: Población de μ elelentos con una aptitud mejor que la población inicial.

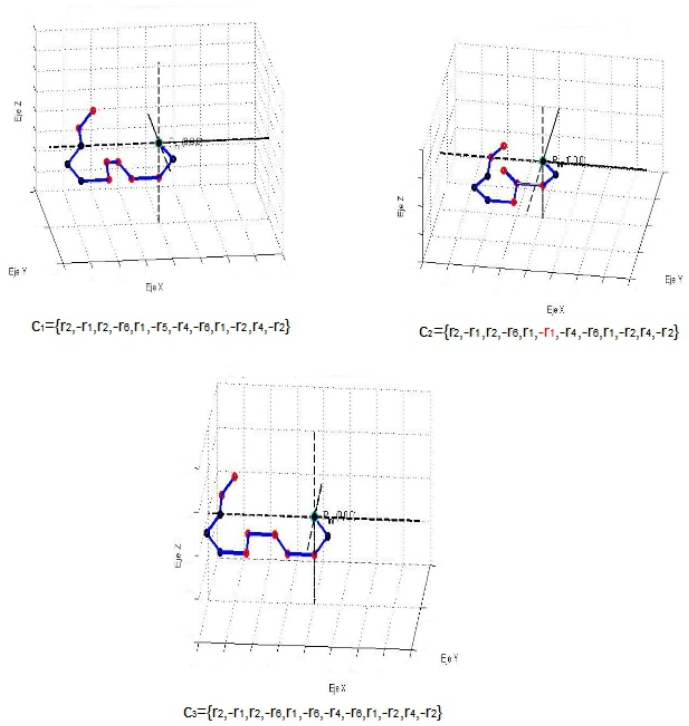

Figura 14: Para la conformación c_1 se realiza una mutación en el gen número 6, en donde dicha mutación genera una infactivilidad observable en la conformación c_2, la cual se recupera introduciendo un elemento que retorne la factibilidad del individuo, lo cual se puede ver en la conformación c_3.

Identificadro	Secuencia	Energia	Fuente
S127	HPPHHPHHPPPHPPPHHHPHHPHHPHH	-12	[66]
S227	PPPHHHHHPHHHHHPPPHHHHHHHHHHP	-6	[66]
S327	HHHHHPPHPHPHPHPHHPPHPPHPHHH	-11	[65]
S427	HPPHHHHHHHHHHPPHHPPHHPHHPHP	-10	[65]
S527	HHHHHHHHHHHHHPHHHPHPPHPHPHPP	-11	[65]
S627	PPPHHPPPPHHHPHPHHPPHHPHHHPP	-15	[65]
S727	HPPPPPPPPPPPHPHPPPPPPPPHPHH	-4	[65]
S827	PPPPPPPHHHPPPHPHHPPPHPPHPPP	-7	[65]

Cuadro 15: Secuencias seleccionadas para verificar el rendimiento del algoritmo implementado.

De forma general, la estructura del algoritmo genético desarrollado se puede observar en el algoritmo 6

3.5. Resultados y discusión

El algoritmo desarrollado se probo con una serie de secuencias empleadas para verficiar la capacidad de otros algoritmos desarrollados. Dichas secuencias permiten comparar los resultados obtenidos en otros trabajos y este trabajo.

Las secuencias provienen de diferentes trabajos realizados, los valores de energía con los que que se constrasta el algoritmo implementado son los valores máximos encontrados en la literatura. El conjunto de secuencias para la validación se puede ver en el cuadro 15.

Al evaluar el algoritmo genético sobre las secuencias del cuadro 15, se tuvieron en cuanta diversos criterios para la evaluación de éste. El primero de ellos fue los resultados obtenidos por el AG al tratar de maximizar las uniones de aminoácidos Hidrofóbicos . Esto se puede observar en el cuadro 16. En este cuadro se muestran los niveles energéticos que se pueden obtener en este trabajo y el número de generaciones necesarias para alcanzar dichos valores, en contraste con los primero trabajos realizados en el área. Se puede observar que l os cambios en la forma de plantear los algoritmos, tanto en la definición de la información como en el uso de los diferentes operadores genéticos empleados, permiten mejorar el rendimiento de estos en forma notoria.

Id de secuencia	Iteraciones	Energia	Iteraciones Literatura	Energia Literatura
S127	6,000	-13	1,239,519 [63]- 85,426 [65]	-12
S227	5,000	-6	1,225,964 [63]	-6
S327	5,000	-11	1,174,297 [63]- 16,282 [65]	-11
S427	5,000	-10	1,225,281 [63]- 81,900 [65]	-10
S527	5,000	-11	1,226,090 [63]- 16,282 [65]	-11
S627	15,000	-15	1,207,686 [63]- 85,447 [65]	-15
S727	30,000	-3	1,248,118 [63]- 3,603 [65]	-4
S827	20,000	-7	1,198,945 [63]- 10,610 [65]	-7

Cuadro 16: Cuadro comparativo de resultados para el desempeño de el AG desarrollado en comparación con los resultados obtenidos por otros autores.

En la fig 15, se muestra como una secuencia S con una familia de conformaciones iniciales c con valores energéticos no apropiados, después de haber sido procesada por el AG desarrollado permite encontrar otra familia de conformaciones Gor(c) las cuales tienen un valor energético mejor que con el que se inició el proceso.

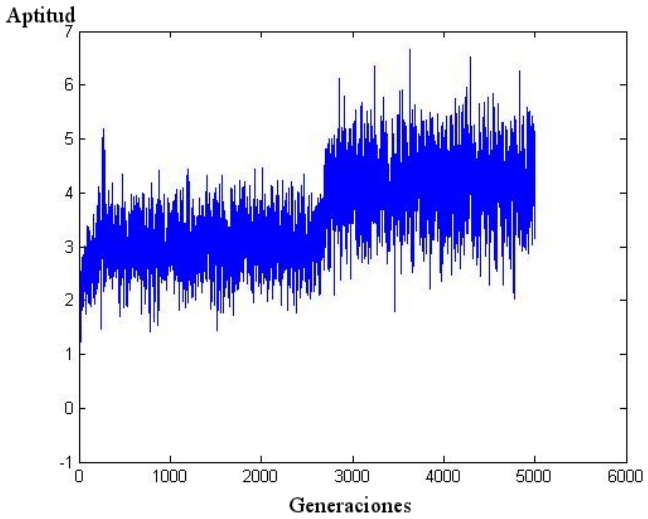

Figura 15: Valor promedio de aptitud en una población de 50 individuos en 5000 generaciones.

Se puede también observar en la Fig 16, como el mejor individuo de cada generación conserva una tendencia de crecimiento en su valor

energético, con lo cual se puede afirmar que el algoritmo está realizando una búsqueda apropiada en el espacio de conformaciones C tratando de encontrar un valor energético apropiado.

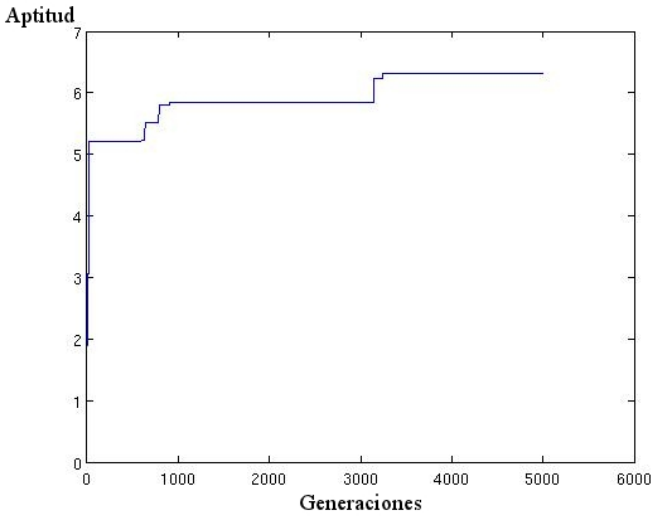

Figura 16: Individuo con mejor aptitud en todas las generaciones.

El comportamiento energético de la población en el AG también se ve reflejado en las conformaciones que se van generando a medida que el algoritmo itera. Ver fig 17.

Se tuvieron en cuenta para la evaluación del algoritmo dos operadores más, uno para evaluar la diversidad de posibles soluciones en cada población y otro para controlar la intensidad de selección del mejor individuo por generación. Estos operadores se denominan: Intensidad de selección τ, ver ecuación 27, y la proporción de selección P_s, ver ecuación 28.

$$\tau = \frac{\bar{g} - \bar{f}}{\sigma_f} \tag{27}$$

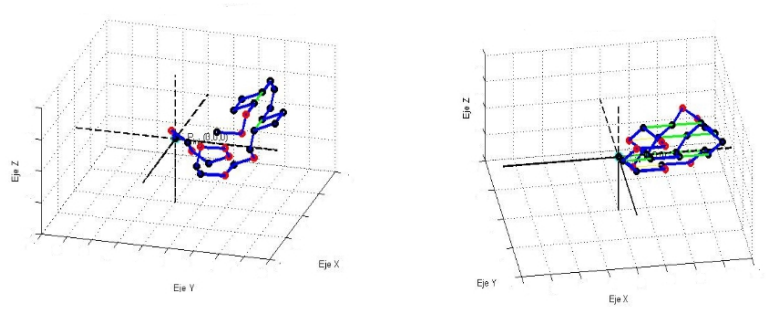

Figura 17: Conformaciones con el mejor nivel de aptitud en el inicio y fin de la ejecución del AG.

$$P_s = \frac{\hat{f}}{\bar{f}} \tag{28}$$

Donde $\bar{g}$ representa el promedio de los λ rendimientos de las conformaciones que se seleccionan para realizar el cruce, $\bar{f}$ es el promedio de los μ rendimientos de las poblaciones antes de la selección y σ_f la varianza de los μ rendimientos antes de realizar la selección. τ da una idea de que tan efectiva es la solución del mejor individuo de cada generación para la conformación de la siguiente. Si los valores de τ son altos, muy seguramente se presentará una convergencia prematura del algoritmo, de lo contrario el algoritmo muestra un comportamiento normal.

Respecto a P_s, $\hat{f}$ representa el individuo con mejor valor de aptitud entre los μ elementos de la población, y $\bar{f}$ el promedio de los rendimientos de los individuos. Sí $P_s = 1$ indica que todos los individuos tienen la misma probabilidad de ser seleccionados, indicando ausencia de presión de selección τ.

En la Fig 18, se puede observar que existe una presión de selección τ. Sin embargo, vista desde la curva generada por P_s no es un τ alto, lo que asegura que no se están presentando problemas de convergencia prematura a un mínimo local. Con esto, se puede inferir que el algoritmo tratará de buscar soluciones sin quedarse estancado por llenar la población de copias

del mejor individuo.

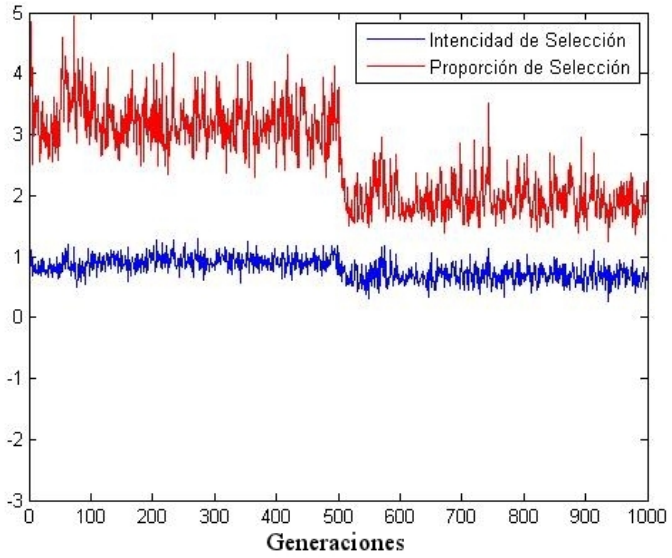

Figura 18: Posibilidad de selección del mejor individuo (Rojo) y diversidad de la población (Azul)

3.5.1. Discusión de los resultados

La fig 15 y 16, se puede observar que el algoritmo mantiene una tendencia de mejoramiento de la población y soluciones, en la fig 15. Se puede observar que en promedio el valor de aptitud de la población al finalizar el siclo genético es mejor que la población con la que se inicio el algoritmo. De esta manera, se asegura que las soluciones que se encuentran serán mejores que aquellas con las que se inicia. En la fig 16, se muestra el comportamiento de la mejor solución en cada generación, manteniendo esta una tendencia a la alza, pues en cada generación ésta siempre pasará intacta a la siguiente. Esto se produce siempre y cuando el producto del cruce de dos individuos genere un mejor individuo que el de la anterior generación.

Este comportamiento elitista permite que se conserven las mejores soluciones y el algoritmo continúe creciendo en aptitud. Sin embargo, la generación de una super solución, una solución con un valor de aptitud muy grande en comparación con las demás soluciones de la misma población, puede provocar una sobre selección del mejor individuo, llenando de copias la población y provocando una convergencia prematura.

La permanencia de dicha diversidad se puede constatar en la fig 18, donde las dos curvas permiten dilucidar el hecho de que la población no está constituida solo por copias del mejor individuo y que los procesos de selección, a pesar de que el mejor individuo tiende a ser muchas más veces que los demás, éstos otros también están siendo seleccionados para la creación de las nuevas generaciones.

3.6.Conclusiones

Se puede concluir por tanto, que el AG desarrollado en este trabajo, asegura la existencia de una diversidad permanente en la población de individuos evitando de esta forma la convergencia prematura de éste.

El algoritmo asegura la búsqueda permanente de mejores soluciones a medida que avanzan las generaciones, pues su carácter elitista le impide desmejorar su rendimiento.

4. Conclusiones y recomendaciones generales

En este trabajo se muestran los fundamentos básicos para abordar diferentes problemas relacionados con la información estructural de las proteínas empleando técnicas de reconocimiento de patrones y optimización heurística.

Los temas que se abordaron en este trabajo fueron, la predicción del contenido estructural, la predicción de la estructura secundaria y el plegamiento de proteínas empleando técnicas de inteligencia artificial, sin embargo se abordaron los tres problemas de forma separada.

Al abordar los diferente problemas mostrados en este trabajo de forma individual, se puede tener una idea mas clara de las dimensiones reales del problema de la predicción de la estructura 3D, así como también, se pueden ver estrategias acerca de como abordar dicho problema en futuros trabajos. A continuación, se mostrarán algunas de las conclusiones y recomendaciones generales obtenidas del trabajo de investigación realizado.

La información del contenido estructural, así como la información de la estructura secundaria puede ser empleada de forma conjunta para refinar los modelos de plegamiento basados en mallas. Es decir, emplear esta información como punto de partida para el algoritmo de plegamiento, permitiendo con esto reducir aún mas el espacio de búsqueda.

Los modelos de plegamiento basados en mallas como lo es el modelo HP, generan aproximaciones que no muestran la posible ubicación de todos

los átomos que componen la secuencia , solo se genera un conjunto de posibles ubicaciones en las que los aminoácidos que componen algún péptido toman respecto a los otros. Por tanto, si se quiere generar una aproximación más adecuada a el plegamiento de una proteína se debe tener en cuenta no solo la inclusión de mas elementos estructurales (átomos que componen los aminoácidos) si no que también seria adecuado generar algoritmos numéricos que permitieran realizar una aproximación de las coordenadas de todos los átomos que componen la secuencia a partir de aquellos que se les aproxime su ubicación.

Tanto en los problemas de reconocimiento de patrones (Contenido estructural y estructura secundaria) como en los problemas de búsqueda y optimización (plegamiento) asociados a la búsqueda de información estructural de las proteínas, se presentan inconvenientes con el costo computacional de los algoritmos que se implementan, por tanto es necesario plantear dichas soluciones en términos de computación de alto rendimiento, creando enfoques que traten de paralelizar o distribuir los algoritmos y los procesos que conllevan dichas problemáticas.

Bibliografía

[1] Peña Díaz Antonio. *Bioquímica*. 2 edition, 1988.

[2] D Voet and G Voet J. *Biochemestry*. Second edition edition, 1995.

[3] John E. Coligan, Ben M. Dunn, David W. Speicher, Paul T. Wingfield, and Hidde L. Ploegh. *Current protocols in protein science*. 2000.

[4] Juli G Pertó. *Fundamentos de Bioqímica*. Universitat de Valéncia, 2007.

[5] Tsingenly Igor F. *Protein structure prediction bioinformatic approach*. September 2002.

[6] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, December 1983.

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

[8] C. Chothia M. Levitt. Structural patterns in globular proteins. *Nature,*, pages 552–557, 1996.

[9] Jishou Ruan, Kui Wang, Jie Yang, Lukasz A. Kurgan, and Krzysztof J. Cios. 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, 2005.

[10] A.D. Oregon C.A. &Thorton J.M. Michie. Analysis of domain structural class using an automated class assignment protocol. *J. Mol.Biol*, (206):168–185, 1994.

[11] Vladimir N. Vapnik. The nature of statistical learning theory. 1995.

[12] Vladimir Vapnik Corinna Cortes. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[13] Chao Chen, Yuanxin Tian, Xiaoyong Zou, Peixiang Cai, and Jinyuan Mo. Prediction of protein secondary structure content using support vector machine. *Talanta*, 71(5):2069–2073, 2007.

[14] Sujun Hua and Zhirong Sun. 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, 2001.

[15] Jieyue He Yi Pan Wei Zhong. Multiclass fuzzy clustering support vector machines for protein local structure prediction. *IEEE.*, 2007.

[16] Wei Zhong Robert Harrison Phang C. Tai Jieyue He and Yi Pan. Clustering support vector machines and its application to local protein tertiary structure prediction. *Springer-Verlag Berlin Heidelberg*, pages 710–717,, 2006.

[17] ZIDING ZHANG, ZHI-RONG SUN, and CHUN-TING ZHANG. A new approach to predict the helix/strand content of globular proteins. *Journal of Theoretical Biology*, 208(1):65–78, 2001.

[18] Lukasz A. Kurgan and Leila Homaeian. Prediction of structural classes for protein sequences and domains-impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. *Pattern Recogn.*, 39(12):2323–2343, 2006.

[19] Kanaka Durga Kedarisetti, Lukasz Kurgan, and Scott Dick. Classifier ensembles for protein structural class prediction with varying homology. *Biochemical and Biophysical Research Communications*, 348(3):981–988, 2006.

[20] Susan Costantini and Angelo M. Facchiano. Prediction of the protein structural class by specific peptide frequencies. *Biochimie*, 91(2):226–229, 2009.

[21] S. Brenner Steven E. T. Hubbard C. Chothia A. Murzin. Scop: a structural classification of protein database for the investigation of sequence and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[22] Steven M. Muskal and Sung-Hou Kim. Predicting protein secondary structure content : A tandem neural network approach. *Journal of Molecular Biology*, 225(3):713–727, 1992.

[23] Umar Syed and Golan Yona. Using a mixture of probabilistic decision trees for direct prediction of protein function. pages 289–300, 2003.

[24] Bin W X Yang. Weave amino acid sequences for protein secondary structure prediction. pages 80–87, 2003.

[25] Tibshirani Hatie T. Classification by pairwise coupling. *The Annals of Statistics*, 26:451–471, 1997.

[26] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.*, 2(1):263–286, 1994.

[27] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, 2001.

[28] M. Shoyaib, S.M. Baker, T. Jabid, Firoz Anwar, and H. Khan. Protein secondary structure prediction with high accuracy using support vector machine. *Computer and information technology, 2007. iccit 2007. 10th international conference on*, pages 1–4, Dec. 2007.

[29] Yu-Dong Cai. Xiao-Jun Liu. Xue biao Xu and Guo-Ping Zhou. Support vector machines for predicting protein structural class. *Guo-Ping Zhou*, pages 1471–2105–2–3, 2001.

[30] Chih-Wei Hsu, Chih-Chung Chan, and Chih-Jen Lin. Support vector machine (svm) is a popular technique for classification. 2011-04-28 00:51:57 2000.

[31] Yanning Z Jianyu Shi. Using decision templates to predict subcellular localization of protein. *Springer-Verlag Berlin Heidelberg*, pages 71–83,, 2007.

[32] Samad Jahandideh, Parviz Abdolmaleki, Mina Jahandideh, and Sayyed Hamed Sadat Hayatshahi. Novel hybrid method for the evaluation of parameters contributing in determination of protein structural classes. *Journal of Theoretical Biology*, 244(2):275–281, 2007.

[33] L. Yuan Y. Cai Y. Dong. Using bagging classifier to predict protein domain structural class. *Using bagging classifier to predict protein domain structural class*, 24:239–242, 2006.

[34] Lixia Jin, Weiwu Fang, and Huanwen Tang. Prediction of protein structural classes by a new measure of information discrepancy. *Computational Biology and Chemistry*, 27(3):373–380, 2003. Computers and Chemistry.

[35] Yu-Dong Cai, Kai-Yan Feng, Wen-Cong Lu, and Kuo-Chen Chou. Using logitboost classifier to predict protein structural classes. *Journal of Theoretical Biology*, 238(1):172–176, 2006.

[36] Kanaka Durga Kedarisetti, Lukasz Kurgan, and Scott Dick. Prediction of protein structural classes by a new measure of information discrepancy. *Computational Biology and Chemistry*, 30:393–394, 2006.

[37] Lukasz Kurgan and Ke Chen. Prediction of protein structural class for the twilight zone sequences. *Biochemical and Biophysical Research Communications*, 357(2):453–460, 2007.

[38] Cios K Chen K. Kurgan L. Scpred: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. *BMC Bioinformatics.*, 1(9):226., 2008.

[39] Barton GJ Cuff JA. . evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 1999.

[40] BURKHARD ROST and CHRIS SANDER. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Biophysics*, 90,:7558–7562, 1993.

[41] D. Nelson and M. Cox. Lehninger principles of biochemestry amino. *Worth Publishers*, 2000.

[42] K.H. Rosen S. G. Krantz D. Zwillinger. Standar mathematical tables and formulae (30th edition). *CRC Press*, 1996.

[43] J. Garnier, D. J. Osguthorpe, and B. Robson. 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, 1978.

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

[45] Kit Fun Lau and Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[46] Yu Xia, Enoch S. Huang, Michael Levitt, and Ram Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *Journal of Molecular Biology*, 300(1):171–185, 2000.

[47] KAIZHI YUE KLAUS M. FIEBIG DAVID P. YEE PAUL D. THOMAS 2 DILL, SARINA BROMBERG and SUN CHAN. Principles of protein folding -a perspective from simple exact models. *Protein Science*, 4:561–602, 1995.

[48] Ken A. Dill. Dominant forces in protein folding. *Journal of the American Chemical Society*, 1990.

[49] CYRUS LEVINTHA. Are there pathways for protein folding? *Extrait du Journal de Chimie Physique*, 1:44, 1968.

[50] Tom Leightont Bonnie Berger. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. *RECOMB*, 1998.

[51] A.R. Leach. *Molecular Modelling: Principles and Application*. LongMan, Harlow, England,, 1996.

[52] Ulrich H.E. Hansmann and Yuko Okamoto. New monte carlo algorithms for protein folding. *Current Opinion in Structural Biology*, 9(2):177–183, 1999.

[53] Harold A. Scheraga. Recent developments in the theory of protein folding: searching for the global energy minimum. *Biophysical Chemistry*, 59(3):329–339, 1996.

[54] Michael Levitt and Shneior Lifson. Refinement of protein conformations using a macromolecular energy minimization procedure. *Journal of Molecular Biology*, 46(2):269–279, 1969.

[55] Jan T Pedersen and John Moult. Genetic algorithms for protein structure prediction. *Current Opinion in Structural Biology*, 6(2):227–231, 1996.

[56] Cheng-Jian Lin and Ming-Hua Hsieh. An efficient hybrid taguchi-genetic algorithm for protein folding simulation. *Expert Systems with Applications*, 36(10):12446–12453, 2009.

[57] Xinchao Zhao. Advances on protein folding simulations based on the lattice hp models with natural computing. *Applied Soft Computing*, 8(2):1029–1040, 2008.

[58] Alena Shmygelska, Rosalía Aguirre Hernández, and Holger H. Hoos. An ant colony optimization algorithm for the 2d hp protein folding problem. pages 40–53, 2002.

[59] Luchian Andrei Băutu. Protein structure prediction in lattice models with particle swarm optimization. *Lecture Notes in Computer Science*, 6234:512–519, 2010.

[60] Rolf Backofen. *Optimization Techniques for the Protein Structure Prediction Problem*. PhD thesis, Ludwig-Maximilians-Universität München institut für informatik, München Alemanha, 1999.

[61] Jerniga G. Raghunathan. Ideal architecture of residue packing and its observation in protein structures. *Protein Sci.*, 6(10):2072–2083, 1997;.

[62] David C. Becerra Fernando Niño Yoan J. Pinzón Sergio R. Duarte. A novel ab-initio genetic-based approach for protein folding prediction. In *GECCO'07, July 7–11, 2007, London, England, United Kingdom.*, 2007.

[63] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231(1):75–81, 1993.

[64] Carlos Cott. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In *IWANN'03: Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks, Berlin, Heidelberg: Springer-Verlag,*, 2003.

[65] A.L.; Punch III W.F.; Goodman E.D. Patton. A standar ga approach to native protein conformation prediction. In *Proceedings of international conference on Genetic Algorithms*, 1995.

[66] Sorin Istrail William Hart. Hp benchmarks.