Similarity of Amyloid Protein Motif using an Hybrid Intelligent System

J. Altamiranda, J. Aguilar, *Member, IEEE*, and C. Delamarche

***Abstract*— The main objective of this research is to define and develop a comparison method of regular expressions, and apply it to amyloid proteins. In general, the biological problem that we study is concerning the search for similarities between non-homologous protein families, using regular expressions, with the goal of discover and identify specific regions conserved in the protein sequence, and in this way determine that proteins have a common origin. From the computer point of view, the problem consists of comparison of protein motifs expressed using regular expressions. A motif is a small region in a previously characterized protein, with a functional or structural significance in the protein sequence. In this work we proposed a hybrid method of motifs comparison based on the Genetic Programming, to generate the populations derived from every regular expression under comparison, and the Backpropagation Artificial Neural Network, for the comparison between them. The method of motifs comparison is tested using the database AMYPdb, and it allows discover possible similarities between amyloid families.**

***Keywords*— Genetic Programming, Neural Network, PROSITE, Motif, Amyloid Protein.**

I. INTRODUCCION

LAS INVESTIGACIONES en ciencias biomédicas están generando un enorme volumen de información biológica, cada vez más compleja, por lo que ellas han empezado a requerir la utilización de técnicas computacionales para su procesamiento. Particularmente, el excesivo aumento de las bases de datos sobre secuencias de proteínas, tanto en el número como en el tamaño de las mismas, provenientes de los experimentos biológicos, ha provocado que la infinita cantidad de información de la que disponemos exceda lo que puede ser procesado y entendido por el ser humano [1]. Las bases de datos contienen una enorme cantidad de información útil, difícil de descubrir, entre estas tenemos: PROSITE [2], Protein Data Bank (PDB) [3], UniProt [4].

Las herramientas informáticas se han desarrollado como una respuesta a las necesidades de obtener nuevos conocimientos sobre las secuencias y motivos de proteínas, aprovechando la información almacenada en esas bases de datos. BLAST (Basic Local Alignment Search Tool) es la principal herramienta para comparar una secuencia de proteína o ADN con otras secuencias en varias bases de datos [5]. La búsqueda en BLAST es una de las vías fundamentales para el aprendizaje acerca de una proteína o gen: la búsqueda revela que secuencias relacionadas están presenten en el mismo o en otros organismos. FASTA es un programa que puede rápidamente identificar regiones compartidas en dos secuencias de proteínas y le asigna una puntuación por homología [6]. La salida consiste en una lista ordenada de secuencias y alineamientos entre secuencias. Las regiones con alta similaridad entre secuencias son identificadas por segmentos con aminoácidos comunes a estas [6]. CLUSTAL es un software que proporciona alineamiento múltiple global usando estrategias progresivas para alinear secuencias de proteínas y ADN de múltiples especies y ayuda a buscar dominios conservados comunes [7]. Pero aún quedan problemas por resolver a nivel de descubrimiento de la información, clasificación de datos, entre otros.

Como aporte a la solución de esos problemas, en este trabajo vamos a estudiar el problema de definir y desarrollar un método computacional de comparación de expresiones regulares de proteínas amiloideas. Este problema es relevante, porque el objetivo de comparar expresiones regulares es encontrar el mayor número de coincidencias entre sus componentes. Las expresiones regulares son usadas para modelar proteínas amiloideas de longitud variable (la próxima sección detalla el interés de ese modelado). Esto va a permitir descubrir relaciones entre proteínas, comparando sus patrones de expresiones regulares. Así, nuestra tarea radica en analizar un conjunto de posibles expresiones regulares, y detectar si existe semejanza entre ellas. El método de comparación de expresiones regulares está basado en la programación genética para construir un conjunto de secuencias validas para las expresiones regulares de la proteína bajo estudio, y la red neuronal de retropropagación para la comparación de las secuencias. Los Algoritmos evolutivos han sido utilizados con éxito en diferentes áreas, como se puede ver en [8], [9], [10], [11], [12], [13], [14], [15]. Igualmente, las redes neuronales de retropropagación se han usado en distintas áreas [16], [17], [18].

Este trabajo está organizado como sigue. En el marco teórico presentamos las dos técnicas de base de nuestra propuesta de comparación de expresiones regulares (la programación genética y las redes neuronales de retropropagación), así como el modelado de proteínas usando expresiones regulares. Seguidamente presentamos la propuesta de comparación de expresiones regulares. La siguiente sección plantea el problema biológico de comparación de motivos de proteínas amiloideas. Finalmente, culminamos con la parte de experimentación.

J. Altamiranda, Universidad de Los Andes, Facultad de Ingeniería, Mérida, Venezuela, altamira@ula.ve

J. Aguilar, Universidad de Los Andes, Facultad de Ingeniería, Mérida , Venezuela, aguilar@ula.ve

C. Delamarche, Université de Rennes I, Rennes, France, christian.delamarche@univ-rennes1.fr

II. MARCO TEÓRICO

A. Programación Genética

La programación genética fue creada por John Koza a finales de los años 80 [19]. La Programación Genética usa los cuatro pasos de la programación evolutiva para la solución de un problema [19], [20], [21]:

1. Genera una población inicial de individuos que representan soluciones potenciales del problema a ser optimizado.
2. Evalúa cada individuo de la población, y asigna un valor de aptitud de acuerdo a que cercano esté dé la solución del problema.
3. Crea una nueva población de programas (fase de reproducción), conformada por los mejores programas existentes y por nuevos programas creados usando los operadores genéticos copia, cruce y mutación, entre otros. Los individuos nuevos reemplazan a los miembros menos aptos de la población.
4. El mejor programa de computación que aparezca en cualquier generación (la mejor solución), es asignado como resultado del proceso evolutivo.

Los elementos de la Programación Genética son [19], [20], [21]:

1. El conjunto de terminales y funciones: cada individuo es una composición de funciones y terminales. El conjunto de terminales está compuesto por átomos, que son las constantes o acciones específicas que son ejecutadas en el programa. El conjunto de funciones pueden ser operaciones aritméticas, lógicas, operadores condicionales, instrucciones de repetición.
2. Los individuos: son estructuras arborescentes.
3. La población inicial: es un conjunto inicial de individuos generados aleatoriamente.
4. El tamaño de la población: el proceso evolutivo se basa en las sucesivas modificaciones realizadas a través de un cierto número de generaciones sobre una población de individuos. Así, cuanto mayor sea la población, más variedad obtendremos durante la evolución. El tamaño de la población se ve principalmente afectado por el tiempo que se tarda en calcular la aptitud de un individuo.
5. El número de generaciones: la evolución se lleva a cabo modificando los individuos que componen la población, a través de un cierto número de generaciones
6. La función aptitud: es una expresión matemática que debe ser capaz de evaluar la calidad de cualquier individuo de la población
7. Los operadores genéticos: los operadores toman individuos de la población actual y producen nuevos individuos para la generación siguiente, aplicando las transformaciones que impongan los operadores. Los operadores clásicos son: copia, cruce, mutación.
8. Los métodos de selección: aquellos utilizados para escoger a un individuo, de entre todos los de la población, para ser utilizado por los operadores genéticos. Los individuos son escogidos dependiendo de su valor de aptitud.
9. El criterio para terminar la ejecución, y el método para designar el resultado final.

B. Red Neuronal Retropropagación

Las redes neuronales artificiales son modelos matemáticos simplificados de las redes de neuronas que constituyen el cerebro humano [22], [23]. Estos modelos están compuestos por un conjunto de "neuronas artificiales", o conjunto de unidades, que procesan e intercambian información. En el modelo neuronal usado las neuronas de una red están estructuradas en distintas capas, de forma que cada neurona de una capa está conectada con todas las de la capa siguiente, a través de unos enlaces caracterizados por un peso que da idea de la importancia de la conexión. Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar a partir de casos anteriormente aprendidos, de abstraer características esenciales a partir de entradas que presentan información irrelevante, etc. Aprovechando estas características, se han construido numerosas aplicaciones de redes neuronales que han demostrado ser muy útiles en los campos de reconocimiento de patrones, generalización, entre otros [23], [24], [25].

En este trabajo se usa la red neuronal de retropropagación. De forma simplificada, el algoritmo consiste en el aprendizaje de un número pre-definido de patrones de entrada-salida, empleando un ciclo "propagación-adaptación" con dos fases diferenciadas (ver Fig. 1).

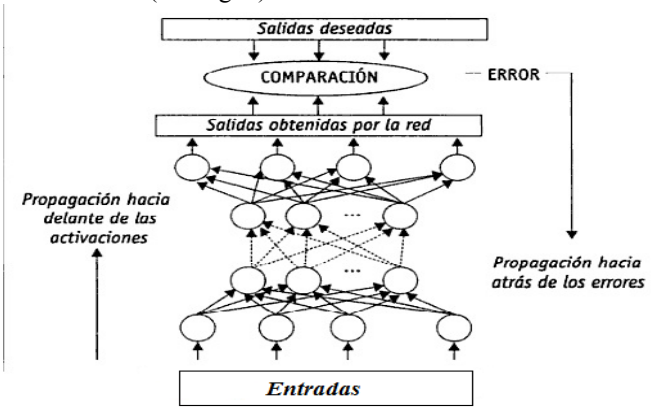

Figura 1. Red Neuronal de retropropagación.

1. Fase de aprendizaje "hacia adelante": los patrones de entrada son presentados a la primera capa de la red, que propaga dicho estimulo a través de todas las capas posteriores hasta generar una salida del sistema.
2. Fase de aprendizaje "hacia atrás": a partir de la comparación generada por la red y la salida deseada, se calcula un valor de error. Este error se transmite por todas las neuronas que conforman la red neuronal. Luego se procede al reajuste de los pesos de las neuronas. Este proceso se repite por un número de ciclos, o hasta que el error sea el deseado por el usuario.

Una vez concluida la fase de aprendizaje se inicia el modo de operación. En dicho modo la red debe generar una salida

próxima a alguna de las aprendidas ante la presencia de un nuevo vector de entrada desconocido, la que más se asemeje a la entrada [23], [24], [25].

C. *Aminoácidos, Proteínas, Motivos y PROSITE*

Los *aminoácidos* son moléculas orgánicas formado por un carbono central (α), al que están unidos cuatro grupos diferentes: a) un grupo amino básico ($-NH_2$); b) un grupo carboxilo acídico (–COOH); un átomo de hidrogeno (–H); d) una cadena lateral característica (–R), como puede verse en la Fig. 2 [26].

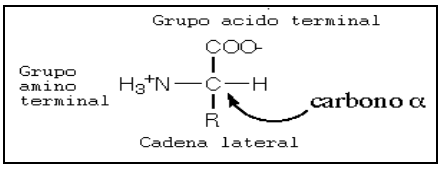

Figura 2. Estructura de los aminoácidos[15].

Las propiedades de los aminoácidos dependen de su cadena lateral característica (–R). Las cadenas laterales son los grupos funcionales que actúan como principales determinantes de la estructura y la función de las proteínas. El conocimiento de las propiedades de estas cadenas laterales es importante a la hora de analizar e identificar proteínas [27], [28]. A continuación se listan los 20 aminoácidos presentes en las proteínas, clasificados según sus propiedades físico-químicas [29]:

Aminoácidos Alifáticos: Glicina (Gly, G), Alanina (Ala, A), Valina (Val, V), Leucina (Leu, L), Isoleucina (Ile, I), Metionina (Met, M).

Aminoácidos con Azufre: Cisteína (Cys, C).

Aminoácidos Aromáticos: Fenilalanina (Phe, F), Tirosina (Tys, Y), Triptófano (Trp, W).

Iminoácido: Prolina (Pro, P).

Aminoácidos Neutros: Serina (Ser, S), Treonina (The, T), Asparagina (Asn, N), Glutamina (Gln, Q).

Aminoácidos Ácidos: Ácido Aspártico (Asp, D), Ácido Glutámico (Glu, E).

Aminoácidos Básicos: Histidina (His, H), Lisina (Lys, K), Arginina (Arg, R).

Las *proteínas* son polímeros de aminoácidos (más de 100 aminoácidos, si son menos de 100 se denomina péptido) que ejecutan la mayor parte de las funciones vitales de las células: el reconocimiento molecular, el transporte de moléculas, la función estructural, la catálisis de las reacciones químicas, inclusive la regulación de la expresión de los genes, está determinada por proteínas que interactúan con el ADN (ácido desoxirribonucleico). Entender estos procesos a nivel molecular es importante por sus consecuencias en el funcionamiento celular, ya que mutaciones en las proteínas, es decir, modificaciones en los residuos originales de la proteína, podrían ocasionar la pérdida o el mal funcionamiento de la misma [27], [30].

Por otro lado, un *motivo* es una región o porción de una secuencia de proteína que posee una estructura específica y describe una función específica de ésta [31]. Las familias de proteínas a menudo son caracterizadas mediante uno o más de tales motivos. Las proteínas tienden a conservar motivos a lo largo de la evolución, ya que estos cumplen requerimientos estructurales y/o funcionales importantes, por lo tanto, no pueden ser suprimidos o modificados. La detección de motivos en proteínas es un problema importante, puesto que los motivos portan y regulan varias funciones, y la presencia de motivos específicos puede ayudar a clasificar una proteína.

Diferentes tipos de representación de motivos han sido propuestos, y se pueden distinguir dos clases principales: probabilístico y determinístico. Un motivo probabilístico consta de un modelo que simula las secuencias, o parte de las secuencias, bajo consideración. Cuando una secuencia de entrada es proporcionada, una probabilidad de comenzar los emparejamientos de los motivos es producida. Las Matrices de Peso por Posición (PWM) y los Modelos Ocultos de Markov (HMM), son ejemplos de motivos probabilísticos [32], [33]. Los motivos determinísticos son descritos en una expresión regular basada en un lenguaje [33]. Estos motivos pueden ser divididos en dos tipos: Longitud fija y longitud variable. Las expresiones regulares son un poderosa notación para caracterizar motivos, indicando cuales posiciones son importantes, cuales pueden variar y que variaciones pueden suceder.

Por otro lado, *PROSITE* fue creada en 1988 por Amos Bairoch [34], [35], [36]. Es una base de datos de sitios y motivos de relevancia biológica. Su objetivo principal es determinar la función de nuevas proteínas, no caracterizadas en otras bases de datos, por medio de motivos. Así, un motivo de una proteína se incluye en PROSITE si detecta las secuencias que tengan una característica biológica particular. Las expresiones regulares que representan los motivos deben ser lo más cortas posibles, para evitar ambigüedades, pero han de ser suficientemente largas para que sean específicas de una familia dada [34], [36], [37]. Los motivos en PROSITE son descritos usando las siguientes reglas para representar las expresiones regulares [38]:

1. Para definir cada aminoácido se usa el código estándar de una letra.
2. Cuando en una posición puede existir cualquier aminoácido se usa la letra "x".
3. Cuando una posición puede variar entre distintos tipos de aminoácidos, la lista de aminoácidos se indican entre paréntesis cuadrados "[ ]". Ejemplo: [LIV] indica que en dicha posición podemos encontrar tanto una L, como una I o una V.
4. Las posiciones con ambigüedades también pueden indicarse por los aminoácidos que no son aceptados en una determinada posición, mediante llaves "{ }". Ejemplo: {AM} indica que se acepta cualquier aminoácido excepto A y M.
5. Las distintas posiciones en el motivo se separan mediante guiones "-".
6. Las veces que se repite un elemento dentro del motivo se indica con paréntesis "( )", que encierra un número o un rango numérico: ejemplo: x(3) correspondería a x-x-x, y x(2,4) correspondería a x-x o x-x-x o x-x-x-x.
7. Un punto indica el final del motivo

Los motivos se construyen a partir de un alineamiento múltiple de secuencias, donde podemos localizar una región específica relacionada con una determinada función. Por ejemplo:

ALRDFATHDDF
SMTAEATHDSI
ECDQAATHEAS

De dicho alineamiento podemos extraer el siguiente motivo común de aminoácidos conservados: A-T-H-[DE]**.**

III. Modelo de Comparaciones de Expresiones Regulares

La Fig. 3 muestra la estructura general del sistema, éste consta de tres componentes:

1. Una estructura gramatical general, basada en una de las expresiones regulares de los motivos a comparar (estará basada en la que llamaremos expresión regular 1)
2. Una red neuronal retropropagación que actúa como función de aptitud, para evaluar los individuos generados por la programación genética (son individuos derivados de la expresión regular 1) con el aprendido ((la red neuronal aprende a la expresión regular 2, o motivo objeto de estudio).
3. Un algoritmo para realizar la comparación de los motivos basado en la programación genética

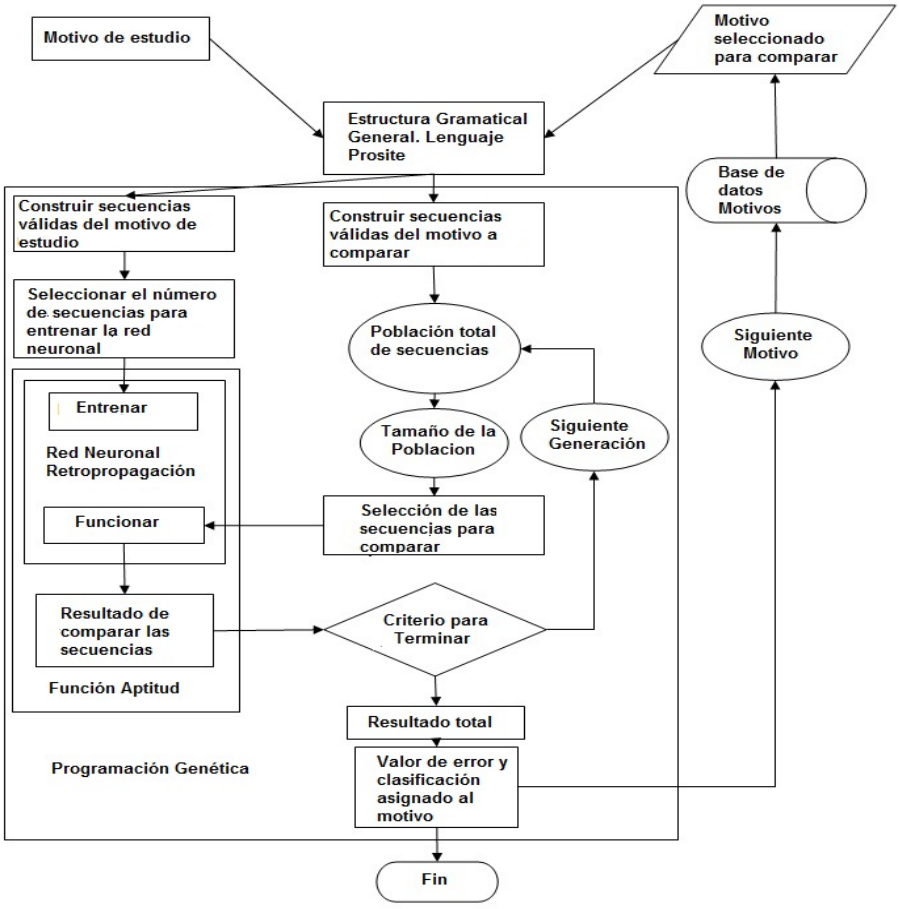

Figura 3. Estructura General del Sistema.

A continuación describimos el macro-algoritmo de la programación genética usado en nuestro enfoque:

1. *Estructura gramatical general para construir las secuencias:* para construir las secuencias de la expresión regular 1 es necesario definir la gramática general que usará la programación genética para esa tarea; esta es una notación formal que describe la sintaxis de un lenguaje dado. El lenguaje usado es PROSITE. La sintaxis consiste en las operaciones que se permiten sobre las secuencias, especificando que atributos pueden aparecer en la construcción de éstas. Este es un punto crítico del sistema, porque si no es definida apropiadamente la gramática, las secuencias generadas por la programación genética no corresponderían a los motivos que se desean comparar.
2. *La población de secuencias del motivo:* es creada por la programación genética usando la gramática general establecida utilizando el lenguaje PROSITE. La gramática es una plantilla bajo la cual se podrán establecer las posibles secuencias del motivo que pueden aparecer. Por ejemplo,

Para el motivo: k[km][ad]a, se pueden formar las secuencias:

- kkaa
- kkda
- kmaa

- kmda

Estas secuencias forman los individuos de la población de secuencias del motivo k[km][ad]a

3. *Tamaño de la población:* representa el número de secuencias del motivo a comparar que se desea en cada generación, para éstas se toman un número de secuencias de la población total. Ésas se utilizan como entrada a la red neuronal en la fase de funcionamiento, para establecer su valor de aptitud.
4. *Número de generaciones:* Es el máximo número de iteraciones que se alcanzan durante la ejecución. Mientras mayor es el número de generaciones el sistema evolucionará por mucho más tiempo, es decir, podemos comparar un mayor número de secuencias del motivo a comparar, pero puede ocurrir que después de un cierto número de generaciones el valor de aptitud de las secuencias del motivo a comparar no mejoren.
5. *Número de secuencias para entrenar la red neuronal:* dada el gran número de secuencias que tiene el motivo objeto de estudio, para la etapa de entrenamiento de la red neuronal es imposible utilizarlos todos. Por lo tanto, es necesario calcular el tamaño de la población que represente todas las posibles secuencias que pueden ser generadas de un motivo. Esa será la población de secuencias del motivo objeto de estudio con la se entrenara la red neuronal (expresión regular 2) , para ello se utiliza:

$$n' = \frac{S^2}{\sigma^2} \quad (1)$$

Donde:

n' = tamaño de la muestra

σ^2 = varianza de la población

Como la varianza de la población es desconocida, se utiliza el error estándar cuadrado $(se)^2$, de esta manera $\sigma^2=(se)^2$

S^2 = varianza de la muestra de la población, la cual puede ser determinada como $S^2 = P(1-P)$

P = Fiabilidad deseada

El tamaño final de la población de secuencias es:

$$n = \frac{n'}{1+\frac{n'}{N}} \quad (2)$$

Donde:

n = tamaño final de la población de secuencias

N = número total de secuencias que puede ser generado usando el motivo de estudio.

5. *Función Aptitud:* se debe asignar un valor cuando comparamos las secuencias del motivo de estudio con las secuencias del motivo extraído de la base de datos. Este valor es determinado por la red neuronal retropropagación. Para esto, la red neuronal es entrenada para reconocer el motivo objeto de estudio (expresión regular 2). Luego, es usada como función aptitud, tal que se introduce una secuencia de otro motivo y determina un error de reconocimiento, que en nuestro caso será interpretado como un valor de similitud entre ambos motivos.

Entrenamiento Red Neuronal: Los individuos generados a partir del motivo objeto de estudio son utilizados para entrenar la red neuronal retropropagación. Esta es una red auto–asociativa, es decir, la información de entrada a la red debe ser la misma de salida (ver Fig. 4). La red retropropagación trabaja bajo aprendizaje supervisado, y por tanto, necesita un conjunto de entrenamiento que describa cada valor de entrada y de salida (en nuestro caso la salida es la misma que la entrada, por ser auto-asociativa, es decir cada secuencia del motivo extraído de la bases de datos).

La arquitectura de la red neuronal es:

Número de neuronas de entrada = tamaño de la secuencia

Número de neuronas de salida = número de neuronas de entrada

$$\text{Neuronas de la capa oculta} = \frac{\text{Nroneuronasentrada}}{2} + 1 \quad (3)$$

El proceso de aprendizaje es el clásico de las redes neuronales por retropropagación.

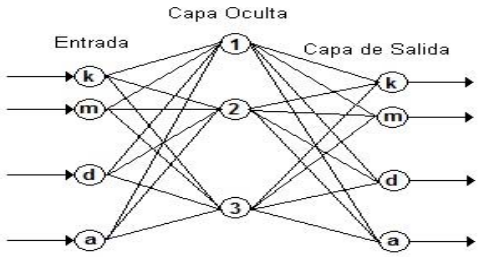

Figura 4. Ejemplo del entrenamiento de la red neuronal auto–asociativa.

Ahora bien, como los motivos están formados por caracteres, es necesario hacer una transformación para convertirlos a números. Así, cada aminoácido está representado por un número. Además como los aminoácidos están agrupados en familias, los aminoácidos de una misma familia tienen números próximos. Por otra parte, los valores de las neuronas de entrada en la red neuronal deben ser normalizados (ya que las entradas deben estar en el intervalo [0 1]. Es necesario escalar los valores dados a los aminoácidos. Para esto se utiliza el método de escalado por el valor máximo. Este método divide el valor de un aminoácido entre el valor máximo dado a un aminoácido, de esta manera se tiene un valor entre 0 y 1 para cada uno de ellos; y así se puede presentar como entradas a la red neuronal retropropagación. De esta forma, tenemos la Tabla I.

TABLA I

CONVERSOR DE AMINOÁCIDOS

AMINOÁCIDO	SÍMBOLO	NUMERO	ESCALADO
FAMILIA: AMINOÁCIDOS ALIFÁTICOS			
GLICINA	GLY, G	0	0/70
ALANINA	ALA, A	1	1/70
VALINA	VAL, V	2	2/70
LEUCINA	LEU, L	3	3/70
ISOLEUCINA	ILE, L	4	4/70
METIONINA	MET, M	5	5/70
FAMILIA: AMINOÁCIDOS NEUTROS			
SERINA	SER, S	20	20/70
TREONINA	THR, T	21	21/70
ASPARGINA	ASN, N	22	22/70
GLUTAMINA	GLN, Q	23	23/70
FAMILIA: AMINOÁCIDOS BÁSICOS			
HISTIDINA	HIS, H	30	30/70
LISINA	LYS, K	31	31/70
ARGININA	ARG, R	32	32/70
FAMILIA: AMINOÁCIDOS ÁCIDOS			
ÁCIDO ASPARTICO	ASP, D	40	40/70
ÁCIDO GLUTAMICO	GLU, E	41	41/70
FAMILIA: AMINOÁCIDOS AROMATICOS			
FENILALANINA	PHE, F	50	50/70
TOROSINA	TYS, Y	51	51/70
TRIPTOFANO	TRP, W	52	52/70
FAMILIA: AMINOÁCIDOS CON AZUFRE			
CISTEINA	CYS, C	60	60/70
FAMILIA: IMINOACIDOS			
PROLINA	PRO, P	70	70/70

Para entrenar la red neuronal de retropropagación, se toman n secuencias del motivo de estudio utilizando las ecuaciones 1 y 2; luego se itera hasta que el error de la red neuronal es menor que el error dado por el usuario, o cuando se alcanzan el máximo número de épocas para la red neuronal.

Funcionamiento de la Red Neuronal: después que la red neuronal fue entrenada se procede a introducir las secuencias del motivo extraído de la base de datos (expresión regular 1), generadas por la programación genética, como entradas para saber si éstas son reconocidas, en cada una de las generaciones. De esta manera, la red neuronal realiza la comparación de cada aminoácido de las secuencias del motivo de entrada con el motivo aprendido en la fase de entrenamiento. Para esto calcula para cada aminoácido (neurona) un error, que es el valor absoluto de la diferencia entre el valor obtenido de la red neuronal y el valor que se introdujo que se desea reconocer (tasa de reconocimiento).

$$error_i = \left| valorneurona_i - valordeseado_i \right| \quad (4)$$

Donde:

$error_i$ = error del aminoácido en la posición i de la secuencia.

$valorneurona_i$ = valor de la salida i de la red neuronal, después de entrenada la red neuronal.

$valordeseado_i$ = valor del aminoácido de entrada en la posición i.

De la ecuación 4 se pueden obtener varios valores del error. Así:

1. Si $error_i \leq \frac{1}{70}$. Representa que el aminoácido en la posición i es igual para la secuencia en entrada y las secuencias aprendidas en la red neuronal. En este caso se definen los siguientes valores::

$$error_i = error_i \quad (5)$$

$$claf_i = 1 \quad (6)$$

Donde:

$claf_i$ = valor por la clasificación en la correcta familia del aminoácido.

2. Si $\frac{1}{70} < error_i \leq \frac{5}{70}$. Representa que el aminoácido en la posición i es diferente entre la secuencia de entrada y las secuencias aprendidas por la red neuronal, pero pertenecen a la misma familia. En este caso se definen los siguientes valores:

$$error_i = \frac{1}{70} \quad (7)$$

$$claf_i = 0{,}8 \quad (8)$$

3. Si $error_i \geq \frac{5}{70}$ Representa que el aminoácido en la posición i es diferente en la secuencia de entrada con respecto a las secuencias aprendidas por la red neuronal. En este caso se definen los siguientes valores:

$$error_i = \frac{5}{70} \quad (9)$$

$$claf_i = 0 \quad (10)$$

Para cada individuo de la población del motivo a comparar que representa una secuencia, se calcula el error por cada posición y la clasificación hecha a cada posición, de la siguiente manera.

$$errorind = \sum_{i=i}^{n} \frac{error_i}{n} \quad (11)$$

$$clafind = \sum_{i=i}^{n} \frac{claf_i}{n} \quad (12)$$

Donde:

errorind = error para cada individuo de la población.

clafind = clasificación para cada individuo de la población

n = número de aminoácidos que componen el individuo.

Para cada generación del motivo a comparar, se calcula el error y la clasificación.

$$errorgen = \sum_{i=i}^{k} \frac{errorind_i}{k} \quad (13)$$

$$clafgen = \sum_{i=i}^{k} \frac{clafind_i}{k} \quad (14)$$

Donde:
errorgen = error para cada generación del motivo.
clafind = clasificación para cada generación del motivo
k = número de individuos que componen cada generación.

Al final, se calcular el error total y la clasificación total

$$errortotal = \sum_{i=i}^{m} \frac{errorgen_i}{m} \quad (15)$$

$$claftotal = \sum_{i=i}^{m} \frac{clafgen_i}{m} \quad (16)$$

Donde:
claftotal = clasificación total
m = número de generaciones.

6. *Criterio para terminar:* un número máximo de generaciones es establecido para evolucionar las secuencias.
7. *Resultado total:* Al final, la programación genética muestra los valores obtenidos en las ecuaciones 15 y 16. Estos valores representan el valor de error y clasificación para el motivo de entrada.

Estas ecuaciones representan a nivel biológico que tan similares son las expresiones regulares que se están comparando. Mientras el valor del error sea más pequeño o se aproxime a cero, las expresiones tienen gran similitud, en caso contrario no, este error mide que tan cercano son los valores de salida de la red neuronal entre el aminoácido que se presenta como entrada y el que la red da como salida. Igualmente, si el valor de la clasificación es alto o cercano a 100 % significa que son muy similares las familias a las que pertenece los aminoácidos aprendidos y el que estamos valorando. En general, que sean similares en ambos casos significa que existe coincidencia en los aminoácidos y en las posiciones que ocupan en ambas expresiones regulares o motivos, que pueden representar que son motivos homólogos o tienen un ancestro en común, mientras este valor se acerca a cero la similitud es baja, es decir, no existe relación entre ambas expresiones regulares o motivos.

IV. Planteamiento del Problema de Comparación de Motivos de Proteínas

Para comprender la utilidad de la comparación de motivos, es necesario comprender como evolucionan las proteínas y así conocer porque existen similitudes entre éstas. De esta forma, cuando una proteína diverge de otra del mismo tipo, se establece, a partir de ese momento, dos versiones de una misma proteína; cuando comienzan a evolucionar los cambios ocurren al azar en cada versión, la estructura empieza a cambiar lentamente en forma independiente en ambas, pero conservando ciertas regiones idénticas. Otra forma de evolución ocurre cuando dos proteínas presentan una estructura homologa, sin haber tenido un ancestro común, esto se conoce como evolución convergente. Estas regiones de las proteínas existen porque son imprescindibles para mantener sus propiedades biológicas. Estas pequeñas regiones sufren fuertes restricciones estructurales a lo largo de la evolución, de forma que pueden ser reconocibles mediante análisis de motivos. El análisis de estos cambios permite inferir el origen de determinados motivos, o descubrir nuevos motivos en las proteínas. Por lo tanto, el estudio de la similitud de motivos consiste en la identificación de pequeñas regiones conservadas en una proteína que pueden ser identificadas en otras, y poder medir el grado de semejanza existente entre éstas, que no es posible detectar por métodos clásicos de computación. De esta manera, los motivos que observamos ahora reflejan toda una historia evolutiva en la que las proteínas han adquirido nuevas funciones adaptándose a nuevos entornos.

De esta manera, es posible descubrir relaciones entre proteínas comparando sus patrones PROSITE de expresiones regulares, con más certeza que comparando sus secuencias.

Así, el problema que se desea resolver consiste en definir y desarrollar un método para comparar una expresión regular contra un conjunto de expresiones regulares, y determinar la similitud entre ellas. El problema biológico está dirigido a la búsqueda de similitudes entre familias de proteínas usando expresiones regulares, y aplicarlo en proteínas amiloideas (ver sección II C). Particularmente, nosotros trabajaremos con la base de datos AMYPdb [28], la cual contiene patrones de proteínas amiloideas. Los resultados de la comparación de los motivos contenidos en la base de datos AMYPdb, deberían permitir descubrir posibles vínculos entre algunas familias amiloideas. Esto constituye un enfoque innovador que podría tener aplicaciones generales, más allá del estudio de proteínas

A. Proceso general de análisis de motivos descritos como expresiones regulares

La figura 5 muestra esquemáticamente los pasos que proponemos para el análisis de motivos descritos usando expresiones regulares. Se tiene un motivo conocido (paso 1), generalmente es posible extraer información relacionada a él de las bases de datos (proteína a la que pertenece, especie animal o vegetal que posee el motivo, etc.). Los motivos que se utilizaran para comparar con el motivo anterior son buscados en la misma u otras bases de datos (pasos 2 y 3). Después se procede a realizar la comparación del motivo conocido con cada uno de los motivos seleccionados de la base de datos, utilizando las técnicas de Programación Genética y Redes Neuronales (paso 4), de esta manera, se pueden encontrar motivos similares al motivo conocido. Luego, los motivos que contienen gran similitud podrían ser agrupados (paso 5). Este último paso no es estudiado en este trabajo.

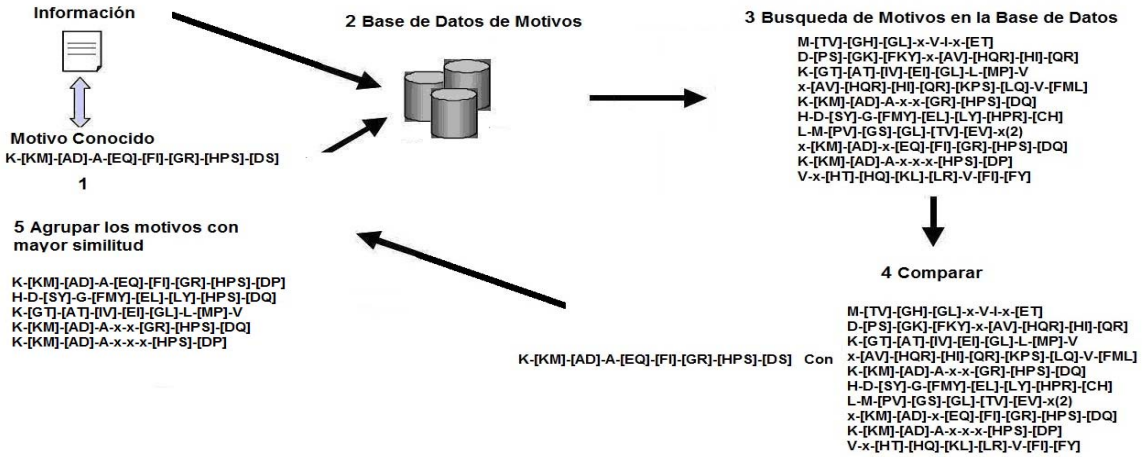

Figura 5. Pasos para analizar de motivos descritos como expresiones regulares.

Particularmente, en este trabajo se estudia el paso 4, cómo realizar la comparación de motivos. Como se muestra en la Fig. 6, se desea conocer si el motivo de la izquierda tiene semejanza con cada uno de los motivos de la derecha, para agruparlos y asignarle un valor representativo de la comparación a cada uno de ellos. Las letras idénticas y en posiciones especificas constituyen puntos angulares para la existencia de semejanzas, mientras que las ambigüedades de tipo x, [], {} constituyen elementos flexibles que disminuyen el valor de la comparación. La comparación de motivos es exactamente lo que permite hacer el sistema propuesto en la sección 3.

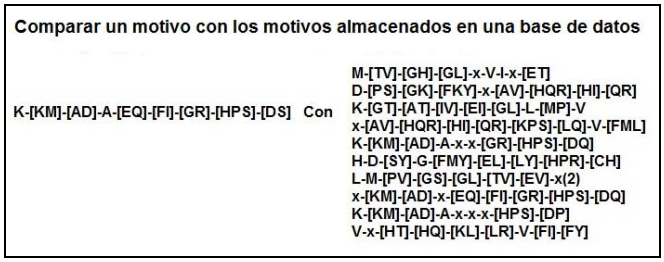

Figura 6. Comparar un motivo con los motivos almacenados en una base de datos.

A. Proteínas Amiloideas

El término amiloidea es utilizado en biología para definir un conjunto de enfermedades caracterizadas por la presencia en órganos específicos (cerebro, riñón, ojo, piel, corazón, páncreas) de depósitos insolubles, esencialmente de carácter proteico. Se sabe hoy que estos depósitos pueden ser intras o extracelulares, y que el componente principal es un enredo fibrilar de proteínas enteras o fragmentos polipéptidicos. Las proteínas que componen los depósitos de amilosis son proteínas normales del organismo. Es solamente en condiciones "patológicas" que estas proteínas tienen la capacidad de cambiar su estructura y auto–ensamblarse en fibras. Entre estas patologías que ocasiona esta proteína se encuentran enfermedades neurodegenerativas: Enfermedad de Alzheimer, Parkinson, Diabetes, entre otras [39], [40], [41].

El término amiloidea fue utilizado por primera vez por el médico alemán Rudolph Virchow en 1854. Estudiando un tejido cerebral de *corpora amylacea* de aspecto macroscópico anormal, observó que al teñirlo con yodo adquiría un color azul pálido, el cual se transformaba en violeta tras tratar el tejido con ácido sulfúrico. Este hecho le hizo llegar a la conclusión de que la sustancia que producía la anormalidad macroscópica era una celulosa denominándola "amiloidea" (del latín amylum y del griego amylon). Más tarde, en 1859, Fridreich y Kekule demostraron que el componente mayoritario de la sustancia amiloidea no era carbohidrato, sino que estaba constituido por proteína [39], [40], [41].

Para facilitar la comparación de las proteínas involucradas en la formación de fibrillas amiloideas, se ha creado la base de datos de proteínas amiloideas (AMYPdb) [42], [43]. El principal objetivo de esta base de datos relacional es proporcionar un acceso actualizado a las secuencias y patrones que describen cada una de las proteínas. Existen 3621 patrones de secuencias de aminoácidos almacenados en la base de datos que pueden ser estudiados para facilitar la asignación de nuevas secuencias a una familia particular, y la formulación de hipótesis sobre sus funciones. Los patrones conservados en las familias pueden también ayudar en la extracción de reglas sobre los mecanismos de formación de fibras [39].

V. EXPERIMENTOS

A. Caso de estudio

En este trabajo, para probar nuestra propuesta de comparación de Proteínas Amiloideas modeladas usando expresiones regulares, estudiamos, por su gran importancia, la proteína precursora del β-amiloide (APP). La proteína precursora del β-amiloide (APP) está conectada a la enfermedad de Alzheimer por vía bioquímica y genética. Ya que la principal fuente de su constitución son las placas amiloideas, APP ha sido objeto de numerosos estudios de su expresión y el metabolismo. La acumulación de péptidos amiloideas β-péptido (Aß) en estas placas fue la primera evidencia de que APP podría ser producida de manera anormal en las enfermedades de Alzheimer, lo que supone que

los síntomas clínicos, anatomía patológica, y la fatalidad es el resultado de la acción de Aβ en el cerebro [44], [45]. Por lo tanto, el estudio de la APP permitirá un mayor conocimiento del funcionamiento de esté en el desarrollo de la enfermedad de Alzheimer.

B. Experimentos

Para la construcción del sistema se utilizo Matlab 7.6. Para realizar la parte experimental se utilizaron 34 expresiones regulares de motivos de la proteína precursora del β–amiloide (APP) de la base de datos AMYPdb, como se muestra en la Tabla II

Para el aprendizaje de la red neuronal retropropagación se utilizó la expresión regular de la posición 1 de la Tabla II: km [k] [ad] a [eq] [fi] [gr] [hps] [dq], que representan un motivo de la proteína precursora β-amiloide (APP). La arquitectura de red neuronal es la siguiente:

Neuronas de entrada: 9
Neuronas de salida: 9
Neuronas capa oculta: 5
Número de épocas: 100000
Secuencias del motivo para entrenar la red neuronal: 130
Error: 0.050813

En la etapa de funcionamiento de la red neuronal, los siguientes parámetros, para cada una de las expresiones regulares de la Tabla I, se utilizaron:
Tamaño de la población: 50 secuencias
Número Generaciones: 50

TABLA II
EXPRESIONES REGULARES DE LA PROTEÍNA PRECURSORA B-AMILOIDE

NUMERO	EXPRESIÓN REGULAR
0	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]
1	M-[TV]-[GH]-[GL]-x-V-I-x-[ET]
2	H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]
3	D-[AP]-[EK]-[FK]-x-[AH]-[DQ]-x-[GR]
4	Q-K-[EL]-[QV]-x-[FY]-[AS]-[DE]-D
5	V-x-[ACM]-[DPV]-A-E-[AF]-[EGR]-[HR]
6	M-[DE]-[AT]-E-x-[GR]-[HQ]-[DS]-[ST]
7	Q-K-[EHIKLMQV]-[ENPQTV]-x-[FWY]-[AGS]-[DE]-D
8	G-x-[ES]-V-x-[HW]-[LQ]-[KL]-L
9	L-M-[PV]-[GS]-[GL]-[TV]-[EV]-x(2)
10	N-[KQ]-[GS]-[AL]-x-[IL]-[GL]-[LY]-x
11	V-I-x-[ET]-x-[IM]-[NV]-[IQ]-[ST]
12	K-[GT]-[AT]-[IV]-[EI]-[GL]-L-[MP]-V
13	x-[FWY]-[AGS]-[DE]-D-V-[AGILV]-[AGS]-N
14	V-[FI]-[FY]-x-[DER]-x-[NV]-[GQ]-S
15	K-[AGS]-A-[HIKLMQR]-I-[DEGHKNPQRSTY]-x-[HIKLMQR]-V
16	[KL]-[LR]-V-[FI]-[FY]-x-[DER]-x-[NV]
17	A-[IV]-[AI]-[DEG]-[EL]-[IM]-[QV]-[DG]-[EG]
18	L-x-[FV]-[FI]-x-E-[DR]-[MV]-[GN]
19	A-[DET]-[EV]-I-[QV]-x-[ET]-[LV]-[DV]
20	x-[IM]-[NV]-[IQ]-[ST]-L-x-[LM]-L
21	K-[EL]-[QV]-x-[FY]-[AS]-[DE]-D-V
22	D-[PS]-[GK]-[FKY]-x-[AV]-[HQR]-[HI]-[QR]
23	V-x-[HT]-[HQ]-[KL]-[LR]-V-[FI]-[FY]
24	M-x-[AC]-[EW]-[AF]-[GHR]-[AH]-D-[ST]
25	x-[AV]-[HQR]-[HI]-[QR]-[KPS]-[LQ]-V-[FML]
26	G-[FLY]-[EL]-[AV]-[EHR]-[HP]-Q-[KV]-[AL]
27	K-x(2)-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]
28	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-x(2)
29	K-[KM]-x-A-[EQ]-[FI]-[GR]-x-[DQ]
30	K-x-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-x
31	K-[KM]-[AD]-A-x(2)[GR]-[HPS]-[DQ]
32	x-[KM]-[AD]-x-[EQ]-[FI]-[GR]-[HPS]-[DQ]
33	K-[KM]-[AD]-A-x(3)-[HPS]-[DQ]
34	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]

Se obtuvieron los resultados que se muestra en la Tabla III. Comparando la expresión regular aprendida por la red neuronal con las expresiones regulares que representan los números 28, 29, 30, 31, 32, 34, podemos ver que el errores son: 0,07, 0,07, 0,08, 0,07, 0,05 y la clasificación es 91,45%, 91,16%, 90,27%, 92,7%, 94,40%, 90,83%, respectivamente, así que podemos agruparlos y decir que se incluyen en la expresión original. Esto nos puede indicar que pueden representar que son motivos homólogos, y podrían estar representados en una sola expresión regular general. De esta manera, para los biólogos sería más sencillo buscar estas expresiones significativas en otras proteínas que buscar la gran cantidad de expresiones regulares existentes. Con respecto a la expresión regular 33, se diferencia de la expresión regular en estudio en que los aminoácidos fijos fueron reemplazados por x, el error fue 0,53, y la clasificación 17,94%. No es posible clasificar a esta expresión regular como familia de la expresión regular en estudio, dado a los valores obtenidos, esto nos dice que las posiciones en la expresión regular donde se encuentra solo un aminoácido, juegan un papel importante en las proteínas. Con el resto de las expresiones regulares presentes en la Tabla III, los valores de los errores son altos y la clasificación es baja, lo que demuestra que las expresiones regulares no tienen relación con la expresión regular que aprendió la red neuronal por retropropagación.

En general, el número de expresiones regulares o motivos que se encuentran en la base de datos de la proteína APP es inmenso, si utilizamos nuestro sistema es posible agruparlos en familias por similitud, o hacer búsquedas de motivos por similitud, lo que haría posible encontrar relaciones entre ellos que hasta ahora no es posible realizar sin la utilización de herramientas computacionales como la que estamos presentado. Nuestra herramienta ayuda a los biólogos a estudiar tales motivos, descubriendo motivos con alto grado de similitud, que hasta ahora son desconocidos dentro de las bases de datos de motivos.

TABLA III
RESULTADOS

NÚMERO	ERROR	SIMILITUD %
1	0,01	100
2	0,51	21,14
3	0,21	71,29
4	0,54	16,90
5	0,46	29,36
6	0,58	10,07
7	0,59	8,19
8	0,51	21,09
9	0,51	20,84
10	0,54	17,20
11	0,45	30,94
12	0,57	11,87
13	0,16	78,41

14	0,56	13,93
15	0,54	17,21
16	0,36	45,25
17	0,44	32,55
18	0,54	16,57
19	0,55	14,46
20	0,51	20,87
21	0,54	17,08
22	0,40	38,00
23	0,62	3,37
24	0,64	0,0
25	0,59	8,89
26	0,54	16,20
27	0,52	19,60
28	0,07	91,45
29	0,07	91,16
30	0,08	90,27
31	0,07	92,70
32	0,05	94,40
33	0,53	17,94
34	0,08	90,83

VI. CONCLUSIONES

Nosotros proponemos en este trabajo un sistema donde es posible descubrir relaciones entre proteínas, comparando sus patrones PROSITE de expresiones regulares con más certeza que comparando sus secuencias, ya que en este caso se realiza la comparación de familias de secuencias agrupadas en expresiones regulares, lo que las hace candidatos ideales para las reconstrucciones de relaciones ancestrales Esto constituye un enfoque innovador que podría tener aplicaciones generales en otros campos.

Nuestro sistema lleva a cabo la comparación de las expresiones regulares pudiendo reconocer aquellas que tienen gran similitud. Nuestro sistema utiliza el error de la comparación obtenido de la red retropropagación como valor para saber que tan parecidos son las expresiones regulares. Además, se agrega la utilización de un valor de clasificación que representa mejor la similitud entre expresiones. Dicho valor toma en cuanta la familia a la que pertenece el aminoácido cuando se está calculando la similitud.

REFERENCIAS

[1] Altamiranda J., Aguilar J., Hernández L., "Sistema de Reconocimiento de Patrones en Bioinformática" Carmen Mueller-Kargen, Sara Wong, Alexandra La Cruz (Eds): CLAIB 2007, IFMBE Proceeding 18,Springer pp 573 – 577.

[2] "*Database PROSITE*". Disponible en [http://www.expasy.ch/prosite/]

[3] "Protein Data Bank". Disponible en: [http://www.pdb.org/pdb/home/home.do]

[4] "Uniprot". Disponible en: [http://www.uniprot.org]

[5] Pevsner J., "*Bioinformatics and Functional Genomics*", Second Edition, Wiley – Backwell, 2009

[6] Mathura V., Kangueane P. "*Bioinformatic A Concept-Based Introduction*", Springer, 2009

[7] Srinivas V., "*Bioinformatics A modern Approach*", Eastern Economy Edition, 2005

[8] D. S. Ramos, G. L. Susteras, "Applying Genetic Algorithms for Predicting Distribution Companies Power Contracting", IEEE LATIN AMERICA TRANSACTIONS, Vol. 4, No. 4, pp. 268-278, June 2006.

[9] Aguilar J. "La Programación Evolutiva en la Identificación de Sistemas Dinámicos a Eventos Discretos", IEEE LATIN AMERICA TRANSACTIONS, Vol. 5, No. 5, pp. 301-310, September 2007.

[10] I. Silva Abreu, J. Viana Fonseca, M. d. P. Melo Wolff, O. F. Silva, P. H. Moraes Rêgo, "A Genetic Algorithm Convergence and Models for Eigenstructure Assignment via Linear Quadratic Regulator (LQR)", IEEE LATIN AMERICA TRANSACTIONS, Vol. 6, No. 1, pp. 1-9, March 2008.

[11] R. Fernandes de Mello, S. Mazzini Bruschi, "Logical process partitioning in distributed simulation using genetic algorithms", IEEE LATIN AMERICA TRANSACTIONS, Vol. 6, No. 1, pp. 97-105, March 2008.

[12] .Martínez Torres, E. Caicedo Bravo, H. Loaiza Correa, "Combination of faces recognition techniques in infrared images by using genetic algorithms", IEEE LATIN AMERICA TRANSACTIONS, Vol. 6, No. 2, pp. 201-206, June 2008.

[13] I. P. Abril, "Genetic Algorithm for the Load Balance on Primary Distribution Circuits", IEEE LATIN AMERICA TRANSACTIONS, Vol. 8, No. 5, pp. 526-532, Sept. 2010.

[14] G. Oliveira, J. Xexéo, R. Torres, R. Linden, W. Souza, "Identification of Keys and Cryptographic Algorithms Using Genetic Algorithm and Graph Theory.", IEEE LATIN AMERICA TRANSACTIONS, Vol. 9, No. 2, pp. 178-183, April 2011.

[15] L. Cunha Brito, M. Lajovic Carneiro, P. C. Miranda Machado, P. H. Portela Carvalho, S. Granato Araújo, "Genetic programming applied to programmable logic controllers programming", IEEE LATIN AMERICA TRANSACTIONS, Vol. 9, No. 3, pp. 270-279, June 2011.

[16] J. Yaljá Montiel Pérez, J. M. d. l. Rosa-Vázquez, "Identification of modes TEM laser by means of an analysis with a back propagation neural network", IEEE LATIN AMERICA TRANSACTIONS, Vol. 4, No. 5, pp. 326-331, Sept. 2006.

[17] E. Gonçalves Pelaes, M. E. de Lima Tostes, R. N. das Merces Machado, R. C. Limão de Oliveira, U. Holanda Bezerra, "Use of Wavelet Transform and Generalized Regression Neural Network (GRNN) to the Characterization of Short-Duration Voltage Variation in Electric Power System.", IEEE LATIN AMERICA TRANSACTIONS, Vol. 7, No. 2, pp. 329-334, June 2009.

[18] A. García, B. Ruiz, I. González, J. L. López, R. Colomo-Palacios, "Methodology for software development estimation optimization based on neural networks", IEEE LATIN AMERICA TRANSACTIONS, Vol. 9, No. 3, pp. 391-405, June 2011.

[19] Koza J., "*Genetic programming: on the programming of computers by means of natural*" MIT Press, 1992

[20] Koza J., Bennett F., Andre D., Keane M. "*Genetic Programming III: Darwinian Invention and Problem Solving"*, Morgan Kaufmann, 1999

[21] Altamiranda J., Aguilar J., "*A data Mining Algorithm based on the Genetic Programming*", Proceeding of the World Multiconference on Systeminc, Cybernetics and Informatics, Vol. IX, pp. 234-239 USA. 1994

[22] Hilera J., Matinez V. *"Redes Neuronales Ratificales: Fundamentos, Modelos y Aplicaciones"*. Editorial Addison – Wesley. 1995

[23] Aguilar J., Rivas F. (Ed.), *"Introducción a la Computación Inteligente"*, MERITEC, Venezuela. 2001.

[24] Mehrotra K., Mohan C., Ranka S. "*Elements of Artificial Neural Networks*" MIT 2000

[25] Bishop. C., "*Neural Network for Patterns Recognition"* Oxford University Press. 2004

[26] Baynes J., Dominiczak M., *"Bioquímica Médica"* Segunda Edición, Elsevier España, 2006

[27] Petsko G., Ringe D., "*Protein Structure and Function*", New Science Press Ltd, USA, 2004

[28] Teijon J., "*Bioquímica Estructural*" Editorial Tébar, España 2001

[29] Baynes J., Dominiczak M., *"Bioquímica Médica"* Segunda Edición, Elsevier España, 2006

[30] González M., Silva D., Hernández I., Vázquez E., Sosa A., "*La Estructura y La Visualización Molecular de Proteínas"* Flores O., Rendón E., Riveros H., Sosa A., Vázquez E., Vázquez I., (eds). Mensaje Bioquímico, Vol XXIX. Depto Bioquímica, Fac Medicina, Universidad Nacional Autónoma de México. México. 2005. pp. 157 - 180.

[31] Herrera M., "*Una Interfaz Web basada en Perl para el análisis de secuencias*". Disponible en: http://ociologia.org/consol/consol/2004/comas/general/material/75/ProSA.pdf

[32] Ferreira P., Azevedo P., *"Evaluating deterministic motif significance measures in protein databases"*Algorithms for Molecular Biology 2:16 2007

[33] Yang J., Deugun J., Sun Z. "*A New Scheme for Protein Sequence Motif Extraction*". Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Vol. 09, pp. 280.1, 2005
[34] Bairoch A. "*Bioinformatics in protein analysis*". Nucleic Acids Research. Vol. 19:Suppl, pp. 2241-2245, April 1991.
[35] Bairoch A. *"Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through exciting times!"*. Bioinformatics Vol. 16, No 1, pp. 48–64, 2000
[36] Database PROSITE Disponible en: http://www.expasy.ch/prosite/
[37] Bairoch A., Bucher P., Hofmann K., "*The PROSITE database, its status in 1997"* Nucleic Acids Research, Vol. 25, No. 1, pp. 217-221, 1997
[38] Seckbach J., Rubin E. "*The new avenues in bioinformatics*" Kluwer Academic Publisher, 2004
[39] Cruz M., "*Diseño, síntesis y evaluación de inhibidores de la proteina β-amiloide. Desarrollo de un modelo de fibrilogénesis*" Universidad de Barcelona, Facultad de Química, España, 2003
[40] Sipe, J., Cohen, A., "*Review*: *History of the Amyloid Fibril*" Journal of Structural Biology, Vol. 130, No. 2-3, pp. 88-98, June 2000.
[41] Westermark P., "*Classification of amyloid fibril proteins and their precursors: An ongoing discussion",* Amyloid: The Journal of Protein Folding Disorders, 1744-2818, Vol. 4, Issue 3, pp. 216-218, 1997.
[42] Pawlicki S., Le Béchec A., Delamarche C., *"AMYPdb: A database dedicated to amyloid precursor proteins"* BMC Bioinformatics*,* Vol. 9, pp**.** 273-28, 2008
[43] "AMYPdb" Disponible en: http://amypdb.univ-rennes1.fr
[44] Hooper N. "*Alzheimer's disease: Methods and Protocols"*. Humana Press. 2000
[45] Xia W., Xu H., *"Amyloid Precursor Protein. A Preactical Approach"* CRC Press 2005.

Junior Altamiranda obtuvo el grado de Ingeniero de Sistemas en 2002 en la Universidad de los Andes en Mérida–Venezuela y Maestría en Computación en 2006 en la Universidad de Los Andes en Mérida–Venezuela. Actualmente realiza estudios de Doctorado en Ciencias Aplicadas en la Universidad de Los Andes en Mérida–Venezuela. Realizo una estancia de investigación en el Máster 2 de Bioinformática en 2007 en la Universidad de Rennes 1 en Francia. Forma parte del Plan II: Programa de Formación de la Generación de Relevo de la Universidad de Los Andes e e investigador del Centro de Microelectrónica y Sistemas Distribuidos (CEMISID) de la misma universidad. Sus áreas de interés son: Bioinformática, Computación Inteligente, Minería de Datos y Sistemas Multiagentes donde ha publicado artículos y capítulos de libros

Jose Aguilar obtuvo una Maestría en Informática en 1991 en la Universidad Paul Sabatier-Toulouse-France, y el Doctorado en Ciencias Computacionales en 1995 en la Universidad Rene Descartes-Paris-France. Además, realizó un Postdoctorado en el Departamento de Ciencias de la Computación de la Universidad de Houston entre 1999 y 2000. Es profesor del Departamento de Computación de la Universidad de los Andes, Mérida-Venezuela, e investigador del Centro de Microcomputación y Sistemas Distribuidos (CEMISID) de la misma universidad. El Dr. Aguilar ha sido profesor/investigador visitante en varias universidades y laboratorios (Université Pierre et Marie Curie Paris-France, Laboratorie d'Automatique et Analyses de Systemes Toulouse-France, Universidad Complutense de Madrid-España, entre otras). Sus áreas de interés son los sistemas paralelos y distribuidos, computación inteligente, (redes neuronales artificiales, lógica difusa, sistemas multiagente, computación evolutiva etc.), optimización combinatoria, reconocimiento de patrones, sistemas de control y automatización industrial. Ha publicado más de 200 artículos y varios libros en las áreas de Sistemas Computacionales y Gestión en Ciencia y Tecnología, y editor de varias actas de conferencias y de libros. Ha formado parte de varios jurados de premios científicos; presidido varios simposios, talleres, etc.; y es revisor de revistas internacionales permanentemente. Además, ha recibido varios premios nacionales como internacionales.

Chistian Delamarche realizó el tercer ciclo en Biología Molecular en 1984 en la Universidad Paris VI en Francia, y Doctorado en Ciencias Biológicas en 1990 en la Universidad Rennes 1 en Francia. Es profesor del Departamento de Biología de la Universidad Rennes 1, e investigador del grupo de Structure et Dynamique des Macromolécules CNRS 6026 de la misma universidad. Sus áreas de interés son: Bioquímica Fundamental, Biología Molecular y Celular, Microbiología Molecular, Bioinformática: Análisis de Secuencias y Búsqueda en Base de Datos, Algoritmos para comparación de Secuencias, descubrimiento de Patrones. Áreas donde ha escrito numerosos artículos. Desarrolló la Base de Datos "AMYPdb" para el estudio de las proteínas amiloideas.