UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERIA

DOCTORADO EN CIENCIAS APLICADAS

MERIDA – VENEZUELA

"RECONOCIMIENTO DE PATRONES ADAPTATIVOS EN PROTEINAS AMILOIDEAS USANDO EXPRESIONES REGULARES"

Autor: Junior A Altamiranda Pérez

Tutores: Dr. Jose L. Aguilar C.

Dr. Christian Delamarche

Proyecto de Grado presentado ante la ilustre Universidad de los Andes como requisito parcial para optar al Título de Doctor en Ciencias Aplicadas

MÉRIDA, MAYO 2012

A mis padres Miriam y Amilcar

A mis hermanas Leidy y Mirianghely

A mi sobrina L. Valentina

A Damarys

Cree y lo lograras…

Napoleón Hill

AGRADECIMIENTOS

El logro de este trabajo es la suma de esfuerzo, fe y dedicación, pero por sobre todo es el resultado de las palabras de aliento y confianza de mis padres Miriam y Amilcar, además de mis hermanas Leidy y Mirianghely a quienes nunca podré pagarles el amor y la comprensión que me brindaron para alcanzar esta meta.

A mi esposa Damarys Pineda por su amor, por brindarme su apoyo, su comprensión y darme animo durante la realización de esta tesis.

Al Prof. José Aguilar por todo el apoyo y los conocimientos que me brindó para la realización de este proyecto.

Al Prof. Christian Delamarche por su apoyo, su orientación pedagógica, a pesar de la distancia. Además, por poner este trabajo en mis manos y confiar en mi capacidad para lograr finalizarlo.

A la Universidad de Los Andes por haberme permitido formarme como Doctor en Ciencias Aplicadas.

Al Programa de Formación de Generación de Relevo (Plan II) de la Universidad de Los Andes por la confianza depositada en mi.

Al Centro « Structure et Dynamique des Macromolécules » de la Universidad de Rennes I y todos sus miembros por permitirme formar parte de su equipo.

Al Ing. Rafael Torres por los aportes realizados a este proyecto.

Al Prof. Marcos Bastidas por sus consejos, que contribuyeron para la finalización de este proyecto.

Al Prof. Addison Rios y al proyecto FONACIT 2005000170 por su gran apoyo.

A los proyectos I-1237-10-02-AA y I-1238-10-02-ED del CDCHT de la Universidad de Los Andes.

A la Lic. Luisa Diaz por su amistad y siempre darme ánimo en los momentos difíciles.

A mis compañeros en el doctorado Dra. Niriaska Perozo, MSc. Taniana Rodriguez, MSc. Nelson Fernández con quieres compartí todos estos años.

A todos muchas gracias….

RESUMEN

El objetivo principal de este trabajo consiste en la comparación y fusión de motivos de las proteínas amiloideas, extraídas de la base de datos AMYPdb, denotadas como expresiones regulares usando las reglas PROSITE. Así, nuestra tarea radica en analizar un conjunto de posibles motivos relacionados, y detectar si existe semejanza entre ellos. Nosotros creamos un algoritmo para comparar dos motivos de proteínas basado en la técnica evolutiva de Programación Genética para generar la población de secuencias derivada de cada motivo bajo comparación. Además, asignamos un valor de similitud de motivos usando una Red Neuronal de Retropropagación como función de aptitud. El método de fusión de motivos utiliza un algoritmo de optimización combinatoria basado en Colonias de Hormigas. Nosotros usamos los aminoácidos del primer motivo para construir el grafo donde las hormigas caminaran. Entonces, el grafo es recorrido por las hormigas según un mapa de recorrido basado en los aminoácidos que componen el segundo motivo, usado por una función de transición que promueve seguir el camino entre aminoácidos semejantes. Las hormigas al caminar van dejando feromona en los nodos, tal que al final algunos tengan mucho feromona y otros poco, Finalmente, el grafo es recorrido nuevamente para construir la expresión regular resultante conformada por los nodos con la mayor concentración de feromona.

Palabras Claves: Bioinformática, Proteínas Amiloideas, Programación Genética, Redes Neuronales Artificiales, Colonias Artificiales de Hormigas.

ABSTRACT

The main objective of this research is the comparison and fusion of the amyloid protein motifs, extracted from the database AMYPdb, denoted as regular expressions using the rules PROSITE. So our task is to analyze a set of possible motifs, and to detect if exists similarity between them. We created an algorithm to compare two protein motifs based on the evolutive technique of Genetic Programming to generate the population of sequences derived from every regular expression under comparison. In addition, we assigned a motif similarity score using a Neural Network Backpropagation as fitness function.The motifs fusion method use an algorithm of combinatorial optimization based on Ant Colonies. We use the amino acids of the first motif to construct the graph where the ants will walk. Then, the graph is crossed by the ants according to the path of the second motif, used by a transition function that promove to flow the path between similars amino acids. The ants when walking leave pheromone in the nodes, in a way that at the end svevral have a lot of or little pheromone. Finally the graph is crossed again to construct the resultant regular expression composed by the nodes with much pheromone.

Keywords: Bioinformatics, Amyloid Protein, Genetic Programming, Artificial Neural Network, Artificial Ants Colony.

TABLA DE CONTENIDO

AGRADECIMIENTOS ..... **III**

RESUMEN ..... **IV**

ABSTRACT ..... **V**

ÍNDICE DE FIGURAS ..... **IX**

INDICE DE TABLAS ..... **XII**

INDICE DE ECUACIONES ..... **XIV**

CAPÍTULO I: GENERALIDADES **1**

1.1. INTRODUCCION 1
1.2. ANTECEDENTES 4
1.3. PLANTEAMIENTO DEL PROBLEMA 15
1.4. OBJETIVOS 19
1.5. ALCANCE 20
1.6. ORGANIZACIÓN DE LA TESIS 20

CAPÍTULO II: MARCO TEORICO **22**

2.1. BIOINFORMATICA 22
2.2. ASPECTOS DE BIOLOGIA MOLECULAR DE INTERES PARA NUESTRO TRABAJO 24
2.2.1. AMINOACIDOS 25
2.2.2. PROTEINAS 27
2.2.3. PROTEINA AMILOIDEA 30
2.3. EXPRESIONES REGULARES Y SU USO EN BIOLOGÍA 32
2.3.1. MOTIVOS 35
2.3.2. PROSITE 36
2.4. COMPUTACION INTELIGENTE 37
2.4.1. PROGRAMACIÓN GENETICA 38
2.4.2. REDES NEURONALES ARTIFICIALES 40
2.4.2.1. RED NEURONAL DE RETROPROPAGACIÓN 45
2.4.3. ALGORITMO DE OPTIMIZACIÓN DE COLONIAS DE HORMIGAS 49

CAPÍTULO III: DISEÑO DEL SISTEMA **53**

3.1. PRESENTACION DEL DISEÑO 53
3.2. SUB-SISTEMA DE COMPARACIÓN DE MOTIVOS 55
3.2.1. DEFINICIÓN DEL MACRO ALGORITMO GENERAL 55

3.2.2. ESPECIFICACIÓN DE LOS COMPONENTES DEL MACRO ALGORITMO GENERAL 57
3.2.2.1. ESTRUCTURA GRAMATICAL GENERAL 57
3.2.2.2. PROGRAMACION GENETICA 58
3.2.2.3. RED NEURONAL DE RETROPROPAGACION 60
3.3. SUB-SISTEMA DE FUSION DE MOTIVOS 66
3.3.1. DEFINICIÓN DEL MACRO ALGORITMO GENERAL 66
3.3.2. ESPECIFICACION DE LAS ETAPAS DEL MACRO ALGORITMO GENERAL 68
3.3.2.1. CREACION DEL GRAFO DE RECORRIDO 68
3.3.2.2. INICIO DEL RECORRIDO DE LA COLONIA DE HORMIGAS 73
3.3.2.3. SELECCIÓN DE LOS MEJORES NODOS 82
3.3.2.4. CONSTRUCCION DE LA EXPRESION REGULAR DE LA FUSION 84
3.4. DEFINICION FORMAL 87
3.4.1. SIMILITUD DE MOTIVOS 87
3.4.1.1. LENGUAJE PARA DESCRIBIR LOS MOTIVOS 87
3.4.1.2. FUNCION DE PUNTUACIÓN 88
3.4.1.3. COMPARACION DE MOTIVOS 89
3.4.2. FUSION DE MOTIVOS 90
3.4.2.1. MAPA DE RECORRIDO 91
3.4.2.2. RUTA DE RECORRIDO 91
3.4.2.3. CONSTRUCCION DEL MOTIVO RESULTANTE 92

CAPÍTULO IV: PRUEBAS DE ENTONACION, EXPERIMENTOS Y ANALISIS DE RESULTADOS.. 94

4.1. CASO DE ESTUDIO 94
4.1.1. PROTEINA β – AMILOIDEA (APP) 94
4.2. PRUEBAS DE ENTONACIÓN 97
4.2.1. COMPARACIÓN DE MOTIVOS 97
4.2.1.1. PRUEBA ENTONACION: COMPARACION DE DOS MOTIVOS DE PROTEINAS .. 100
4.2.2. FUSION DE MOTIVOS 104
4.2.2.1. PRUEBA DE ENTONACION No. 1: FUSION DE DOS MOTIVOS DE PROTEINAS 106
4.2.2.2. PRUEBA DE ENTONACION No. 2: FUSION DE DOS SECUENCIAS DE MOTIVOS DE PROTEINAS 110
4.3. ANALISIS BIOLOGICOS USANDO NUESTRO SISTEMA 111
4.3.1. COMPARAR UN MOTIVO CON UN CONJUTO DE MOTIVOS 111
4.3.2. COMPARAR UN MOTIVO CON MOTIVOS MODIFICADOS DE ÉL 115
4.3.3. FUSIONAR UN MOTIVO CON OTRO MOTIVO DE PROTEINAS 116
4.3.4. COMPARAR Y FUSIONAR UN MOTIVO CON UN CONJUNTO DE MOTIVOS 119
4.3.5. COMPARAR Y FUSIONAR UN MOTIVO CON UN MOTIVO DESCONOCIDO 122
4.4. COMPARACION CON OTROS TRABAJOS 125

4.4.1. SUB – SISTEMA DE COMPARACION DE MOTIVOS 126
4.4.2. SUB – SISTEMA DE FUSION DE MOTIVOS 129

CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO 134

5.1. CONCLUSIONES 134
5.2. TRABAJO FUTURO 135

BIBLIOGRAFIA 137

APENDICE A: MANUAL DEL USUARIO 147

A.1. INSTALACION DEL SISTEMA 147
A.2. MANUAL DE USUARIO DEL SISTEMA 148

APENDICE B: IMPLANTACION DEL SISTEMA 161

B.1. DESCRIPCION DEL LENGUAJE UTILIZADO 161
B.2. CASOS DE USO DEL SISTEMA 161
B.3. DIAGRAMA DE CLASES 166
B.4. DIAGRAMA DE COMPONENTES SEGÚN SU ARQUITECTURA 169

ÍNDICE DE FIGURAS

FIGURA 1.1 HERRAMIENTAS UTILIZADAS ACTUALMENTE PARA COMPARAR SECUENCIAS PROTEICAS. 7
FIGURA 1.2 EJEMPLO DE ALINEAMIENTO LOCAL Y GLOBAL 8
FIGURA 1.3 MATRIZ PAM250 9
FIGURA 1.4 MATRIZ BLOSUM62 10
FIGURA 1.5 EJEMPLO DE UN ALINEAMIENTO MÚLTIPLE DE 9 SECUENCIAS 11
FIGURA 1.6 EJEMPLO DE UN PERFIL. 12
FIGURA 1.7 EJEMPLO DE HMMS PARA ALINEAMIENTO DE SECUENCIAS. 13
FIGURA 1.8 DIAGRAMA DEL PROBLEMA PROPUESTO. 18
FIGURA 1.9 COMPARAR UN PATRÓN CON LOS PATRONES ALMACENADOS EN UNA BASE DE DATOS. 18
FIGURA 2.1 ESTRUCTURA BÁSICA DE UN AMINOÁCIDO. 25
FIGURA 2.2 ESTRUCTURA DE LOS DIFERENTES AMINOÁCIDOS 25
FIGURA 2.3 FORMACIÓN DE UN ENLACE PEPTÍDICO 26
FIGURA 2.4 ESTRUCTURA PRIMARIA DE UNA PROTEÍNA. 27
FIGURA 2.5 DIFERENTES MODELOS DE LA HÉLICE- A 28
FIGURA 2.6 FORMA TRIDIMENSIONAL DE LA LÁMINA-B 28
FIGURA 2.7 HÉLICE DE COLÁGENO 29
FIGURA 2.8 TIPOS DE ESTRUCTURAS TERCIARIAS 29
FIGURA 2.9 TIPOS DE ESTRUCTURAS CUATERNARIAS 30
FIGURA 2.10 MECANISMOS DE CONVERSIÓN DE PROTEÍNAS NORMALES SOLUBLES EN FIBRA AMILOIDEA 31
FIGURA 2.11 ESQUEMA DE UNA NEURONA NATURAL. 41
FIGURA 2.12 ESQUEMA DE UNA NEURONA ARTIFICIAL. 42
FIGURA 2.13 DIAGRAMA DE UNA RED NEURONAL DE RETROPROPAGACIÓN. 45
FIGURA 2.14 RED NEURONAL DE RETROPROPAGACIÓN 46
FIGURA 2.15 COMPORTAMIENTO DE LA COLONIA DE HORMIGAS PARA OBTENER EL CAMINO MÁS CORTO ENTRE DOS PUNTOS. 50
FIGURA 3.1 DIAGRAMA DE COMPONENTES DEL SISTEMA PROPUESTO. 54
FIGURA 3.2 ESTRUCTURA GENERAL DEL SUB-SISTEMA DE COMPARACIÓN DE MOTIVOS. 55
FIGURA 3.3 ESTRUCTURA DE UN INDIVIDUO 59
FIGURA 3.4 EJEMPLO DE ENTRENAMIENTO DE LA RED NEURONAL AUTO-ASOCIATIVA 62
FIGURA 3.5 FUSIÓN DE DOS EXPRESIONES REGULARES. 66
FIGURA 3.6 FUSIÓN DE N EXPRESIONES REGULARES. 66
FIGURA 3.7 DIAGRAMA GENERAL DEL SUB-SISTEMA DE FUSIÓN DE MOTIVOS. 67
FIGURA 3.8 ESTRUCTURA DE DATOS DE LOS NODOS. 69
FIGURA 3.9TRANSFORMACIÓN DE LA EXPRESION REGULAR ER1 A UN TDA PILA. 70
FIGURA 3.10 CREACIÓN DE LOS NODOS QUE IRÁN A LOS EXTREMOS DEL GRAFO 70

FIGURA 3.11CONSTRUCCIÓN DEL GRAFO UTILIZANDO EL PRIMER ELEMENTO DEL TOPE DE LA PILA. 71
FIGURA 3.12 TRATAMIENTO AL MOMENTO DE HALLARSE UN GAP EN EL TOPE DE LA PILA. 71
FIGURA 3.13 INSERCIÓN DE UN ELEMENTO QUE CONTIENE 2 VALORES DENTRO DE UN PARÉNTESIS EN EL GRAFO. 72
FIGURA 3.14 CULMINACIÓN DE LA CONSTRUCCIÓN DEL GRAFO DE RECORRIDO. 72
FIGURA 3.15 CONSTITUCIÓN DE UNA COLONIA DE HORMIGAS EN LA NATURALEZA. 74
FIGURA 3.16 CONSTRUCCIÓN DE LA PILA UTILIZANDO A ER2 COMO MAPA DE RECORRIDO. 76
FIGURA 3.17 EL AGENTE HORMIGA OBSERVA LOS NODOS DE LA SIGUIENTE POSICIÓN DEL GRAFO. 77
FIGURA 3.18 RULETA CONSTRUIDA PARTIENDO DE LAS PROBABILIDADES DE VISITAR A LOS NODOS DE LA POSICIÓN N°1, OBTENIDAS SEGÚN LA ECUACIÓN 3.15. 78
FIGURA 3.19 DESPLAZAMIENTO DEL AGENTE HORMIGA A LA SIGUIENTE POSICIÓN. 79
FIGURA 3.20 RECORRIDO DEL AGENTE HORMIGA A TRAVÉS DEL GRAFO. 81
FIGURA 3.21 GRAFO DE RECORRIDO CON LOS NIVELES DE FEROMONA DE CADA NODO. 83
FIGURA 3.22 ÚNICO NODO QUE SUPERA EL UMBRAL DE FEROMONA. 84
FIGURA 3.23 UN NODO GAP QUE SUPERA EL VALOR DEL UMBRAL. 85
FIGURA 3.24 UN NODO GAP QUE SUPERA EL UMBRAL PERO NO SUPERA EL VALOR DE FEROMONA DE OTROS NODOS EN ESA POSICIÓN. 85
FIGURA 3.25 OBTENCIÓN DEL PATRÓN DE LA FUSIÓN 86
FIGURA 3.26 ESTRUCTURA DE LA RED NEURONAL DE RETROPROPAGACIÓN UTILIZADA 89
FIGURA 4.1 MECANISMOS MOLECULARES Y CELULARES DE LA ENFERMEDAD DE ALZHEIMER. 96
FIGURA 4.2 VENTANA DE PARAMETROS PARA EL TAMAÑO DE LA MUESTRA DEL MOTIVO A APRENDER EN EL SISTEMA 99
FIGURA 4.3 VENTANA DE PARAMETROS DE LA RED NEURONAL EN EL SISTEMA 99
FIGURA 4.4 VENTANA DE PARAMETROS PARA LA SIMILITUD EN EL SISTEMA 99
FIGURA 4.5 VENTANA DE PARAMETROS PARA LA FUSIÓN EN EL SISTEMA 105
FIGURA 4.6 REPRESENTACIÓN GRAFICA DE LA EXPRESIÓN REGULAR RESULTANTE DE LA FUSIÓN. 110
FIGURA 4.7 RESULTADOS DE LA BÚSQUEDA DEL PATRÓN DE FUSIÓN EN AMYPDB. 117
FIGURA 4.8 RESULTADOS DE LA BÚSQUEDA DEL PATRÓN DE FUSIÓN GENERAL EN AMYPDB. 119
FIGURA 4.9 GRAFO DE RECORRIDO DE H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ] 121
FIGURA A.1 VENTANA PRINCIPAL DEL SISTEMA 148
FIGURA A.2 OPCIONES DEL MENÚ PROYECTO 149
FIGURA A.3 OPCIONES DEL MENÚ TRANSF. FORMATO 150
FIGURA A.4 TRANSFORMAR CADENA DE FASTA A UNA EXPRESIÓN EN PROSITE 150
FIGURA A.5 TRANSFORMAR CADENA DE TRES LETRAS A UNA EXPRESION PROSITE 150
FIGURA A.6 OPCIONES DEL MENÚ COMPARAR MOTIVOS 151
FIGURA A.7 OPCIONES DEL SUB-MENÚ MOTIVO PROTEÍNA Nº1 151
FIGURA A.8 VENTANA DONDE SE MUESTRAN LOS MOTIVOS A COMPARAR 152

FIGURA A.9 VENTANA DONDE SE MUESTRA EL MOTIVO SELECCIONADO COMO OBJETO DE ESTUDIO Y EL CONJUNTO DE SECUENCIAS QUE SE PUEDEN CONSTRUIR A PARTIR DE ÉL. 152
FIGURA A.10 VENTANA PARA LOS PARÁMETROS PARA EL TAMAÑO DE LA MUESTRA DEL MOTIVO A APRENDER. 153
FIGURA A.11VENTANA DEL TAMAÑO DE LA MUESTRA DEL MOTIVO. 153
FIGURA A.12 VENTANA PARA LOS PARÁMETROS DE LA RED NEURONAL PARA EL APRENDIZAJE DEL MOTIVO. 154
FIGURA A.13 VENTANA DE APRENDIZAJE DEL MOTIVO POR LA RED NEURONAL. 155
FIGURA A.14 VENTANA PARA LOS PARÁMETROS DE SIMILITUD. 156
FIGURA A.15 VENTANA DE COMPARACIÓN DE LOS MOTIVOS. 156
FIGURA A.16 VENTANA DE RESULTADOS DE LA COMPARACIÓN DE LOS MOTIVOS. 157
FIGURA A.17 VENTANA DE PARÁMETROS PARA LA FUSIÓN. 158
FIGURA A.18 VENTANA PARA REALIZAR LA FUSIÓN DE LOS MOTIVOS. 158
FIGURA A.19 VENTANA DE RESULTADOS DE LA FUSIÓN DE MOTIVOS. 159
FIGURA A.20 VENTANA DEL VISOR DE IMÁGENES. 160
FIGURA A.21 OPCIONES DEL MENÚ FUSIONAR MOTIVOS 160
FIGURA B.1 ACTORES DEL SISTEMA. 161
FIGURA B.2 INICIO DEL SISTEMA 162
FIGURA B.3 AJUSTE DE LOS PARÁMETROS PARA EL TAMAÑO DE LA MUESTRA 162
FIGURA B.4 AJUSTE DE LOS PARÁMETROS PARA LA RED NEURONAL. 163
FIGURA B.5 AJUSTE DE LOS PARÁMETROS PARA LA SIMILITUD. 163
FIGURA B.6 AJUSTE DE LOS PARÁMETROS PARA LA FUSIÓN. 164
FIGURA B.7 ASIGNAR UN MOTIVO. 164
FIGURA B.8 REALIZAR COMPARACIÓN. 165
FIGURA B.9 REALIZAR FUSIÓN. 165
FIGURA B.10 DIAGRAMA DE CLASES SUB-SISTEMA DEL COMPARACIÓN DE MOTIVOS. 166
FIGURA B.11 DIAGRAMA DE CLASES PARA EL SUB-SISTEMA DE FUSIÓN DE MOTIVOS 168
FIGURA B.12 DIAGRAMA DE COMPONENTES. 169
FIGURA B.13 ESTRUCTURA DE LOS DIRECTORIOS. 171

INDICE DE TABLAS

TABLA 1.1 RELACIÓN ENTRE LAS MATRICES PAM Y BLOSUM 11
TABLA 2.1 LISTA DE AMINOÁCIDOS 26
TABLA 2.2 DEFINICIÓN DE REGLAS SOBRE EL LENGUAJE {A, B, C} 33
TABLA 2.3 FUNCIONES DE TRANSFERENCIA 43
TABLA 3.1 CONVERSOR DE ENTRADA A LA RED DE LOS VALORES DE LOS AMINOÁCIDOS 63
TABLA 3.2 IDENTIFICADORES Y FAMILIAS PARA LOS NODOS ESPECIALES 69
TABLA 3.3 CAMPOS DEL TDA AGENTE HORMIGA. 75
TABLA 3.4 EJEMPLO DE VALORES PARA LA INICIALIZACIÓN DE LOS PARÁMETROS DE UN AGENTE HORMIGA. ... 76
TABLA 4.1 PARÁMETROS TAMAÑO DE LA MUESTRA DEL MOTIVO A APRENDER. 97
TABLA 4.2 PARÁMETROS DE LA RED NEURONAL 98
TABLA 4.3 PARÁMETROS PARA LA SIMILITUD 98
TABLA 4.4 TAMAÑO DE LA MUESTRA PARA EL MOTIVO OBJETO DE ESTUDIO PARA DISTINTOS VALORES DEL ERROR ESTÁNDAR Y FIABILIDAD IGUAL A 0,70 100
TABLA 4.5 TAMAÑO DE LA MUESTRA PARA EL MOTIVO OBJETO DE ESTUDIO PARA DISTINTOS VALORES DE FIABILIDAD Y ERROR ESTÁNDAR IGUAL A 0,02 101
TABLA 4.6 TAMAÑO DE LA MUESTRA PARA EL MOTIVO OBJETO DE ESTUDIO PARA DISTINTOS VALORES DEL ERROR ESTÁNDAR Y FIABILIDAD IGUAL A 0,9 101
TABLA 4.7 DISTINTOS VALORES DE LA TASA DE APRENDIZAJE 102
TABLA 4.8 DISTINTOS VALORES DE MOMENTO 102
TABLA 4.9 DISTINTOS VALORES DE MOMENTO 103
TABLA 4.10 PARÁMETROS DE SIMILITUD EN LOS AMINOÁCIDOS 103
TABLA 4.11 PARÁMETROS DE SIMILITUD PARA LA PROGRAMACIÓN GENÉTICA Y RESULTADOS DE LA SIMILITUD 103
TABLA 4.12 PARÁMETROS PARA EL SUB-SISTEMA DE FUSIÓN DE MOTIVOS. 104
TABLA 4.13 LISTA DE PARÁMETROS QUE SERÁN TOMADOS COMO CONSTANTES, PARA EFECTOS DE LAS PRUEBAS REALIZADAS. 106
TABLA 4.14 BÚSQUEDA DEL NÚMERO DE HORMIGAS E ITERACIONES DE LA COLONIA PARA LA CONVERGENCIA DEL ALGORITMO HACIA LA SOLUCIÓN ESPERADA. 107
TABLA 4.15 RESULTADO DE LA FUSIÓN PARA EL CONJUNTO (4,4). 108
TABLA 4.16 RESULTADO DE LA FUSIÓN PARA EL CONJUNTO (4,8). 108
TABLA 4.17 RESULTADO DE LA FUSIÓN PARA EL CONJUNTO (8,4). 109
TABLA 4.18 RESULTADO DE LA FUSIÓN PARA EL CONJUNTO DE PATRONES (8,8). 109
TABLA 4.19 RESULTADO DE LA FUSIÓN PARA EL CONJUNTO (20,20). 110

TABLA 4.20 MOTIVOS EXTRAÍDOS DE LA BASE DE DATOS AMYPDB DE LA PROTEÍNA B - AMILOIDEA..... 112
TABLA 4.21 PARÁMETROS PARA OBTENER EL TAMAÑO DE LA MUESTRA DEL MOTIVO ESTUDIO. 112
TABLA 4.22 POBLACIÓN DE DE SECUENCIAS DEL MOTIVO DE ESTUDIO. 113
TABLA 4.23 PARÁMETROS PARA EL ENTRENAMIENTO DE LA RED NEURONAL. 113
TABLA 4.24 PARÁMETROS PARA LA SIMILITUD DE LOS MOTIVOS. 113
TABLA 4.25 RESULTADOS DE LA COMPARACIÓN. 114
TABLA 4.26 MOTIVOS MODIFICADOS DEL MOTIVO DE ESTUDIO. 115
TABLA 4.27 MOTIVOS MODIFICADOS DEL MOTIVO DE ESTUDIO. 115
TABLA 4.28 MOTIVOS DE LA PROTEÍNA TAU. 116
TABLA 4.29 RESULTADO DE LA FUSIÓN DE LOS MOTIVOS. 117
TABLA 4.30 MOTIVOS PROTEÍNA B – AMILOIDEA UTILIZADOS PARA COMPARAR. 119
TABLA 4.31 POBLACIÓN DE DE SECUENCIAS DEL MOTIVO DE ESTUDIO. 120
TABLA 4.32 RESULTADOS DE LA COMPARACIÓN. 120
TABLA 4.33 ALGUNAS DE LAS PROTEÍNAS QUE CONTIENEN EL PATRÓN RESULTANTE. 122
TABLA 4.34 MOTIVOS UTILIZADOS EN LA PRUEBA. 123
TABLA 4.35POBLACIÓN DE DE SECUENCIAS DEL MOTIVO OBJETO DE ESTUDIO. 123
TABLA 4.36 RESULTADOS DE LA SIMILITUD DE LOS MOTIVOS. 123
TABLA 4.37 PROTEÍNAS QUE CONTIENEN EL PATRÓN RESULTANTE. 125
TABLA 4.38 COMPARACIÓN CUALITATIVA DE LOS MÉTODOS USADOS EN DIFERENTES TRABAJOS. 127
TABLA 4.39 MOTIVOS USADOS PARA REALIZAR LA COMPARACIÓN CUALITATIVA. 128
TABLA 4.40 RESULTADOS DE LA COMPARACIÓN DE MOTIVOS ENTRE LOS DIFERENTES ENFOQUES. 129
TABLA 4.41 RESULTADOS DE LA COMPARACIÓN DE MOTIVOS ENTRE LOS DIFERENTES ENFOQUES. 131
TABLA 4.42 FUSIÓN DE MOTIVOS. 132
TABLA 4.43 RESULTADOS DE LA FUSIÓN DE LAS SECUENCIAS DE ESCHERICHIA COLI. 133
TABLA B.1 FORMATO Y CONTENIDO DE LOS ARCHIVOS. 173

INDICE DE ECUACIONES

ECUACIÓN 2.1 35
ECUACIÓN 2.2 42
ECUACIÓN 2.3 42
ECUACIÓN 2.4 47
ECUACIÓN 2.5 47
ECUACIÓN 2.6 47
ECUACIÓN 2.7 47
ECUACIÓN 2.8 47
ECUACIÓN 2.9 47
ECUACIÓN 2.10 48
ECUACIÓN 2.11 48
ECUACIÓN 2.12 48
ECUACIÓN 2.13 48
ECUACIÓN 2.14 48
ECUACIÓN 2.15 48
ECUACIÓN 2.16 48
ECUACIÓN 2.17 48
ECUACIÓN 2.18 49
ECUACIÓN 2.19 50
ECUACIÓN 2.20 51
ECUACIÓN 2.21 51
ECUACIÓN 2.22 52
ECUACIÓN 3.1.... 60
ECUACIÓN 3.2 61
ECUACIÓN 3.3 61
ECUACIÓN 3.4 61
ECUACIÓN 3.5 62
ECUACIÓN 3.6 62
ECUACIÓN 3.7 62
ECUACIÓN 3.8 64
ECUACIÓN 3.9 64
ECUACIÓN 3.10 65
ECUACIÓN 3.11 65
ECUACIÓN 3.12…… 65
ECUACIÓN 3.13 65

ECUACIÓN 3.14 65
ECUACIÓN 3.15..... 77
ECUACIÓN 3.16..... 78
ECUACIÓN 3.17..... 81
ECUACIÓN 4.1 126
ECUACIÓN 4.2 126
ECUACIÓN 4.3 129
ECUACIÓN 4.4 129
ECUACIÓN 4.5 130
ECUACIÓN 4.6 130

CAPÍTULO I: GENERALIDADES

1.1. 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. Las bases de datos contienen una enorme cantidad de información útil, difícil de descubrir.

Las herramientas Bioinformáticas más importantes 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 del Centro Nacional para la Información Biotecnológica "National Center for Biotechnology Information" (NCBI) para comparar una secuencia de proteína, o ADN, con otras secuencias almacenadas en éstas bases de datos. La búsqueda en BLAST es una de las vías fundamentales para el aprendizaje acerca de una proteína o gen: la búsqueda revela qué secuencias relacionadas están presentes en el mismo o en otros organismos [1]. FASTA es un programa que puede rápidamente identificar regiones compartidas en dos secuencias de proteínas y le asigna una puntuación por homología. La salida consiste en una lista ordenada de alineamientos entre secuencias. Las regiones con alta similaridad entre secuencias son identificadas por segmentos con aminoácidos comunes a éstas [2].

Para realizar alineamiento múltiple de secuencias de proteínas se utiliza CLUSTAL que 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 [3]. Pero aún quedan problemas de resolver a nivel de descubrimiento de la información, clasificación de datos, entre otros.

La disponibilidad de genomas completos, el volumen de información contenido actualmente en las bases de datos públicas y los ambiciosos proyectos de estudios masivos de interacción entre proteínas, traen consigo dos consecuencias principales: por una parte permiten plantear estudios que hasta ahora eran, simplemente, inabordables; pero, por otra parte, exigen un cambio en la mentalidad y en las herramientas informáticas de tratamiento de datos. Se está pasando de una situación en la que se disponía de un número relativamente bajo de datos, en la que el mayor interés estaba centrado en extraer mucha información de ellos, a un nuevo escenario en el que se dispone de grandes cantidades de datos de los que se sabe mucho menos en detalle, pero de los que se pueden comprender sus propiedades globales como sistema. Este nuevo escenario constituye un verdadero cambio de paradigma en biología, y también en el campo de la Bioinformática, debido a que el tipo de herramientas informáticas y las estrategias para procesar datos biológicos han de ser reorientados hacia la biología molecular, y en concreto hacia la posibilidad que ofrece disponer de secuencias de ADN y proteínas almacenadas en bases de datos. Este conocimiento almacenado se puede utilizar directa o indirectamente. La utilización directa implica encontrar secuencias similares a la secuencia problema, o secuencias que tengan alguna propiedad común. La utilización indirecta usa la información para obtener reglas que permitan posteriormente predecir, con mayor o menor éxito, propiedades en secuencias nuevas.

En este trabajo vamos a estudiar el problema de definir y desarrollar un método computacional de comparación y fusión de motivos de proteínas amiloideas denotados como expresiones regulares. Actualmente existen varios métodos de descubrimiento de motivos (usando Expresiones Regulares, el Modelo Oculto de Markov (HMM), los Autómatas y las Matrices PSSM, entre otros). Las expresiones regulares son las más utilizadas por los biólogos, así como también el método grafico de LOGOS, ya que visualmente son más simples de comprender e interpretar para ellos [4], [5]. Para el descubrimiento de expresiones regulares, por razones históricas, se ha utilizado el método Pratt [6], que está basado en el algoritmo Knuth-Morris-Pratt [7], pero existen otras herramientas, entre las más conocidas tenemos [8], [9], [10], [11], [12], [13]: TEIRESIAS, MEME. Ahora bien, todos esos trabajos previos de descubrimiento de motivos comunes es

para el caso de secuencias homologas [4]. El descubrimiento de motivos comunes entre secuencias que están alejadas en el plano evolutivo (secuencias no-homologas o no-relacionadas) es un problema muy complejo. Además, trabajar con proteínas es más complejo que con cadenas de ADN, a causa del número de letras que las componen (20 en lugar de 4), y la posibilidad de múltiples reagrupamientos entre los aminoácidos: (se pueden hacer distintos tipos de grupos de aminoácidos, lo que aumenta el número de similitudes que pueden existir entre ellos).

En el caso concreto de las proteínas amiloideas, son proteínas normales que, en ciertas condiciones cambian su estructura y se agregan en segmentos internos en forma de fibras (fibras amiloideas) tóxicas para las células. Trabajando sobre la hipótesis de que existen motivos comunes para cada una de las familias de proteínas amiloideas, las dificultades biológicas son las siguientes:

1. Existen más de treinta familias de proteínas amiloideas, por lo que es necesario compararlas entre ellas.

2. En cada familia no se sabe si las proteínas de todas las especies vivientes tienen la capacidad de agregación, o solo ocurre en las proteínas de los vertebrados o de los mamíferos. Existe poco conocimiento biológico a excepción del caso hombre y algunos animales.

3. Existen probablemente varias clases de motivos amiloideas comunes en ciertas familias, pero no en todas ellas. No se tiene conocimiento de cuales familias tienen motivos comunes.

4. Los motivos realmente degenerados (mal conservados) contienen aminoácidos como la valina, la leucina y la isoleucina, muy frecuentes en todas las proteínas.

5. El cambio de la estructura en las proteínas amiloideas puede ser consecuencia de mecanismos biológicos diferentes y complejos: mutaciones, modificaciones post-tradicional, divisiones, etc.

Por otro lado, nuevos descubrimientos biológicos han determinado que los motivos pequeños son muy importantes, existiendo varios programas para buscarlos en proteínas no-homólogas [14], [15], [16], [17], [18]. Además, existen herramientas que permiten comparar expresiones regulares de motivos de ADN y motivos pequeños (SLM por sus siglas en ingles, Short Linear Motifs), tales como Dilimont [19], CompariMotif [20], FunClust [21], y Bio.motif [22]. Ahora bien, dichas herramientas no permiten comparar expresiones regulares largas ni fusionarlas en una expresión común.

En ese sentido, en este trabajo proponemos un método computacional de comparación y fusión de motivos no-homólogos, estudiando el caso concreto de las proteínas amiloideas, las cuales tiene cierta complejidad al analizarlas (sus motivos son largos, degenerados, poseen numerosas variantes (derivadas del número de familias que tienen, de los mecanismos biológicos de cambios que poseen, etc.), etc.). Además, nuestra herramienta presenta la versatilidad que puede trabajar con motivos pequeños (SLM).

En específico, nuestra tarea radica en analizar un conjunto de motivos de proteínas denotados como expresiones regulares almacenados en la base de datos AMYPdb [23], [24] y detectar si existen similitudes entre ellos utilizando métodos computacionales de comparación, para luego construir un patrón general que permita conocer en cuales familias de proteínas se encuentra. Los patrones encontrados pueden ser explicados por la existencia de segmentos que se han preservado por la evolución natural de las proteínas, y sugieren que las regiones obtenidas juegan un rol funcional y estructural en los mecanismos de las proteínas.

1.2. ANTECEDENTES

La Biología fue una ciencia tradicionalmente de observación y descripción a lo largo del siglo XIX, la cual sufrió una serie de explosiones diversificadoras, convirtiéndose en una ciencia caracterizada por la generación de grandes cantidades de información. Para ello utilizó el desarrollo tecnológico, que le permitió extender el análisis hacia niveles de organización biológica cada vez más ricos en información. Así, apareció la Biología Molecular como camino de sistematización de los conocimientos cada vez más caudalosos

acerca del funcionamiento de las moléculas responsables del mantenimiento y la transformación de la vida. De esta forma, comenzó a dibujarse un paradigma que unía la necesidad de relacionar las estructuras biológicas con su funcionamiento. Históricamente, el uso de las computadoras para resolver problemas biológicos comenzó con el desarrollo de algoritmos y su aplicación en el entendimiento de las interacciones de los procesos biológicos y las relaciones filogenéticas entre diversos organismos. También, la computación ha sido necesaria para el almacenamiento de la información en bases de datos adecuadas, la integración de sistemas de información distribuida, y el desarrollo de sistemas para la selección de información relevante a partir de todos los datos acumulados [25], [26], [27].

Todo ello ha venido conformado un área que se ha llamado Bioinformática, la cual es una disciplina científica emergente que utiliza la tecnología de la información y el modelado matemático para aplicarlo en bioquímica, biofísica y biología molecular [28]. Su objeto es la colección, mantenimiento, distribución, análisis y uso de grandes cantidades de datos generados en tales disciplinas, para facilitar el descubrimiento de nuevas ideas biológicas, así como crear modelos globales a partir de los cuales se puedan discernir principios unificadores en biología. La Bioinformática en un comienzo surgió para solucionar problemas de cálculo con información biológica, más tarde, para problemas de almacenamiento y análisis de grandes cantidades de datos extraídos de organismos, y en la actualidad se ha transformado en una disciplina con líneas propias de investigación que dejo de ser solo una herramientas para los biólogos [29], [30].

Durante los últimos 20 años se ha determinado que muchas proteínas de diversos orígenes con una función similar, también tienen secuencias similares de aminoácidos. Así, existen secuencias correspondientes del ADN que son similares en diversas especies, por ejemplo entre los ratones y seres humanos, aunque la proteína bajo análisis no lo sea [26]. Desde principios de los años 90, muchos laboratorios han estado analizando el genoma completo de varias especies tales como bacterias, levaduras, ratones y seres humanos. Durante estos estudios se han generado cantidades enormes de datos, los cuales se recogen y se almacenan en grandes bases de datos. Además de recopilar todos estos datos, es

necesario descubrir secuencias de nucleótidos o de aminoácidos para encontrar semejanzas y diferencias de estos, y poder originar conocimiento de estos hallazgos. Puesto que es muy difícil analizar los datos de varios (cientos) nucleótidos o aminoácidos de manera manual, varias técnicas de computación han sido desarrolladas para solucionar estos problemas.

De esta manera, uno de los campos más importantes en la Bioinformática está relacionado con la identificación de motivos o regiones conservadas en las secuencias en las proteínas. Este cubre un conjunto de objetivos, tales como la búsqueda de genes específicos en los genomas, identificar regiones en las proteínas, predecir dominios proteicos, detectar los lugares de conexión [29]. El método consiste en buscar un conjunto de similitudes en un grupo de aminoácidos, o de proteínas que se piensan pueden estar emparentados. Los problemas que emergen en este sector son interesantes porque requieren el uso de métodos matemáticos e informáticos para comparar los motivos, o secuencias, con tiempos de ejecución de gran complejidad algorítmica.

Las herramientas Bioinformáticas más utilizadas en el análisis de secuencias se esquematizan en la figura 1.1, incluyen métodos para comparar dos secuencias, realizar comparaciones múltiples (CLUSTALW) [31], y la búsqueda de secuencias similares en bases de datos (los programas BLAST [32] y FASTA [33] son los más conocidos). Estas herramientas explotan el principio de la evolución Darwiniana, es decir, las proteínas que tienen un ancestro común (proteínas homologas) divergen entre los organismos debido a la acumulación de mutaciones durante la evolución. Estas proteínas homologas tienen generalmente una estructura y una función conservada. Las herramientas computacionales que comparan las secuencias proteicas utilizan Matrices PSSM deducidas de los conocimientos sobre la evolución de éstas, por lo tanto, solo deberían utilizarse para la comparación de secuencias homologas.

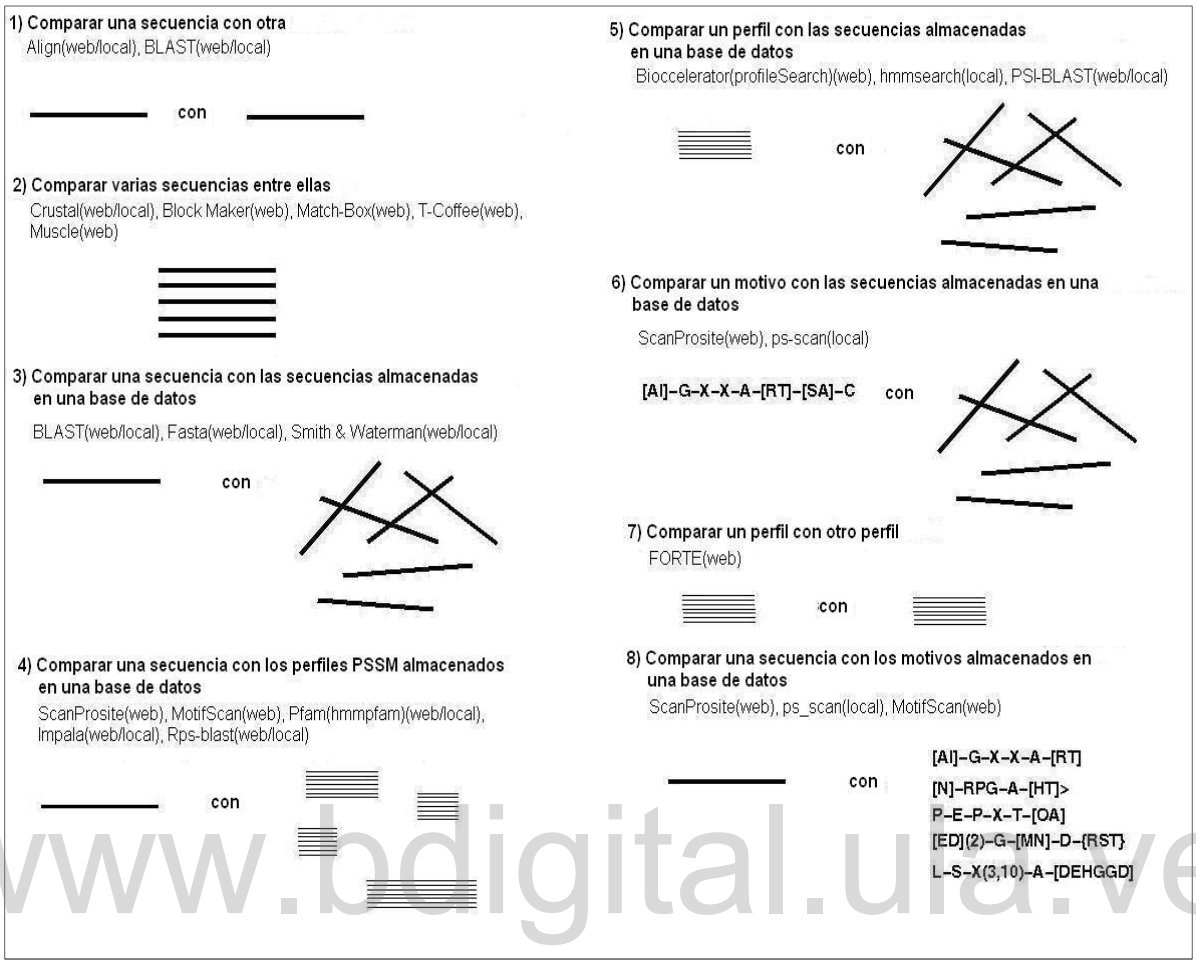


Figura 1.1 Herramientas utilizadas actualmente para comparar secuencias proteicas.

Las herramientas utilizan varios algoritmos de alineamiento y/o bases teóricas para ese proceso, entre estos tenemos:

1. *Alineamiento Local:* un alineamiento local se hace en pequeñas fracciones de la cadena original en donde existen regiones idénticas o de alta similitud. La prioridad dentro de este tipo de alineamiento es encontrar regiones locales, antes que encontrar coincidencias entre cadenas vecinas o pares de aminoácidos. El algoritmo fue desarrollado por Smith y Waterman [34], y está fundamentado en la Programación Dinámica (su complejidad es $O(N^2)$, ver figura 1.2).

2. *Alineamiento Global:* son las posibles coincidencias existentes a lo largo de toda la secuencia de aminoácidos o nucleótidos, tratando siempre de encontrar el mayor

número de aciertos posibles. El algoritmo fue desarrollado por Needleman y Wunsch [35], y está fundamentado en Programación Dinámica (su complejidad es $O(N^2)$, ver figura 1.2).

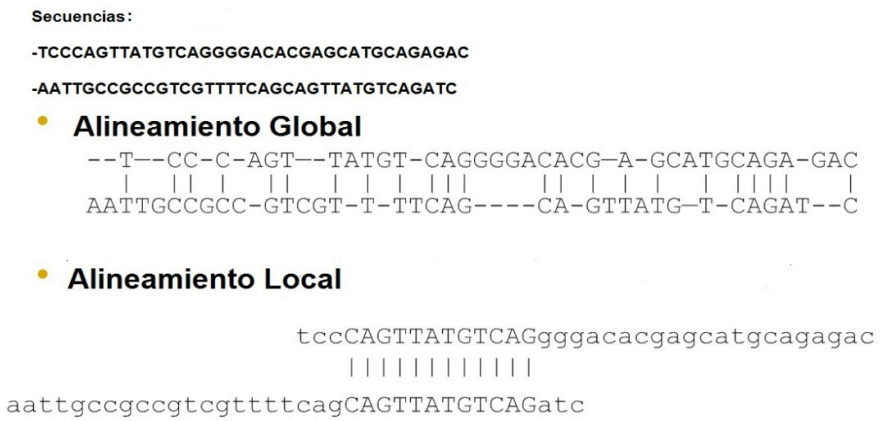


Figura 1.2 Ejemplo de Alineamiento Local y Global

En este contexto, el alineamiento óptimo se obtiene utilizando una valoración a partir de una matriz de puntuación que considera los distintos posibles pares de aminoácidos (o nucleótidos) que quedan enfrentados en el alineamiento, y las penalizaciones por aquellas regiones no coincidentes en el alineamiento [36]. Las matrices de puntuación más utilizadas son:

1. *Matrices PAM (Percent Accepted Mutation):* desarrolladas por Dayhoff [37], se basan en el análisis de mutaciones permitidas en las secuencias de aminoácidos durante la evolución. Se utilizó una base de datos con 1572 cambios en 71 grupos de proteínas muy parecidas, con un porcentaje de identidad igual o mayor al 85%. Una matriz con una distancia evolutiva de 0 PAM tendría solamente unos en la diagonal principal de la matriz y ceros en el resto de la misma. Una matriz con una distancia evolutiva de 1 PAM (PAM1) tendría números cercanos a 1 en la diagonal principal y números pequeños en el resto de la misma. Una PAM1 correspondería a una divergencia de 1% en la secuencia de una proteína (un aminoácido reemplazado por cada 100). Para obtener una matriz PAM que indique un proceso evolutivo más largo (un porcentaje de N mutaciones aceptadas en 100 aminoácidos), se multiplica la matriz PAM1 por sí

misma N veces. Esto se debe a que la construcción de estas matrices se basa en un modelo probabilístico (Markoviano) de evolución, en donde la probabilidad de que ocurra un cambio en un aminoácido es la misma para todos ellos e independiente de los otros aminoácidos. Mientras más alejadas sean las secuencias, su porcentaje de identidad será más pequeño y el valor de la matriz PAM más grande. Con una matriz PAM250 (250 sustituciones por 100 aminoácidos) se tienen proteínas con un 20% de identidad, por lo que se ha visto que esta matriz trabaja bien para proteínas alejadas (ver figura 1.3). En general, mientras más parecidas sean las proteínas entre sí, el valor de la matriz PAM debe ser menor. En [38] se puede calcular una matriz PAM de cualquier valor [39].

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**4**																			
S	0	**3**																		
T	-2	1	**3**																	
P	-3	1	0	**6**																
A	-2	1	1	1	**2**															
G	-3	1	0	-1	1	**5**														
N	-4	1	0	-1	0	0	**2**													
D	-5	0	0	-1	0	1	2	**4**												
E	-5	0	0	-1	0	0	1	3	**4**											
Q	-5	-1	-1	0	0	-1	1	2	2	**4**										
H	-3	-1	-1	0	-1	-2	2	1	1	3	**6**									
R	-4	0	-1	0	-2	-3	0	-1	0	1	2	**6**								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	**5**							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	**6**						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	**5**					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	**6**				
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	**4**			
F	-4	-3	-2	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	**9**		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	**10**	
W	-2	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	**17**

Figura 1.3 Matriz PAM250

2. *Matrices BLOSUM:* En vez de realizar una extrapolación con base en un modelo Markoviano de la evolución y utilizando secuencias muy parecidas, Henikoff [40] en 1992 construyo matrices de sustitución a partir de alineamientos de bloques de secuencias. Un bloque es una matriz cuyas filas representan segmentos de secuencias proteicas alineadas sin interrupciones, estas matrices se denominan BLOSUMnn (BLOck SUbstitution Matrices). En función del grado de similitud entre las secuencias, se obtienen las diferentes matrices. Por ejemplo, la matriz BLOSUM62 se calculó a partir de bloques de proteínas en los que si dos secuencias tenían más de 62 % de identidad, la contribución de esas secuencias se ponderaba para que no tuvieran demasiado peso en los cálculos de frecuencias. En la actualidad, la que más se usa es la

BLOSUM62 (ver figura 1.4). Al contrario de las matrices PAM, mientras más grande sean los número en la matriz más parecidas son las secuencias que se analizan.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**																			
S	-1	**4**																		
T	-1	1	**4**																	
P	-3	-1	1	**7**																
A	0	1	-1	-1	**4**															
G	-3	0	1	-2	0	**6**														
N	-3	1	0	-2	-2	0	**6**													
D	-3	0	1	-1	-2	-1	1	**6**												
E	-4	0	0	-1	-1	-2	0	2	**5**											
Q	-3	0	0	-1	-1	-2	0	0	2	**5**										
H	-3	-1	0	-2	-2	-2	1	1	0	0	**8**									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**								
K	-3	0	0	-1	-1	-2	0	-1	1	1	-1	2	**5**							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**						
I	-1	-2	-2	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**					
L	-1	-2	-2	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**				
V	-1	-2	-2	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	
W	-2	-3	-3	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**

Figura 1.4 Matriz BLOSUM62

La tabla 1.1 muestra la correspondencia entre los dos tipos de matrices y el porcentaje de identidad que se espera en las proteínas. Existen diferencias importantes entre las matrices PAM y las BLOSUM. Las primeras se basan en un modelo explícito de la evolución, en donde las sustituciones o reemplazos de los aminoácidos se dan en las ramas de un árbol filogenético, y se obtuvieron a partir de mutaciones observadas en alineamientos globales utilizando regiones muy conservadas y regiones con un alto grado de sustituciones (tienen la característica de que mientras más grande sea el valor de la matriz, la distancia evolutiva es mayor). Por otro lado, las matrices BLOSUM se basan en un modelo implícito de la evolución, se obtuvieron a partir de alineamientos sin huecos de regiones muy conservadas y, al contrario de las matrices PAM, mientras más alto es el valor de la matriz, menor es la distancia evolutiva [39]. Así, un número bajo de BLOSUM (umbral de identidad bajo) se corresponde con un número alto de PAM (distancia evolutiva alta).

Matriz		% identidad
PAM	BLOSUM	
100	90	43
120	80	38
160	60	30
200	52	24
250	45	20

Tabla 1.1 Relación entre las matrices PAM y BLOSUM

3. *Alineamientos múltiples*: son utilizados para determinar las secuencias consensos. Un consenso proteico representa el resultado de los aminoácidos mayormente encontrados en cada posición de un alineamiento múltiple de secuencias homologas. A nivel biológico, un consenso puede corresponder a un motivo estructural, o funcional común, a un conjunto de proteínas homologas. De esta manera, el consenso de la alineación presentada en la figura 1.5 es K-P-[D, E]-[Q, N]-[Y, F]-K-V-H. Sin embargo, dos inconvenientes principales del consenso son su gran rigidez, así como el hecho de que puede no tener ningún sentido biológico.


```
Secuencia 1          K     P     D     Q     Y     K     V     H
Secuencia 2          K     P     D     Q     Y     K     V     H
Secuencia 3          K     P     D     Q     Y     K     V     H
Secuencia 4          K     P     D     Q     Y     K     V     H
Secuencia 5          K     P     D     N     Y     K     M     H
Secuencia 6          K     -     E     N     Y     K     V     H
Secuencia 7          K     -     E     N     F     K     V     H
Secuencia 8          K     P     E     N     F     K     V     H
Secuencia 9          K     P     E     N     F     K     V     H

Secuencia Consenso   K     P   [D,E][Q,N][Y,F]   K     V     H
```

Figura 1.5 Ejemplo de un alineamiento múltiple de 9 secuencias

4. *Perfil:* es una matriz AxP de pesos asociados, donde "A" corresponde a los 20 aminoácidos y "P" a las distintas posiciones del alineamiento de secuencias. El método utilizado para construir perfiles requiere un alineamiento múltiple de secuencias como entrada, además de una matriz de sustitución de aminoácidos para convertir las frecuencias de los aminoácidos en una posición en pesos. Una vez que disponemos del perfil de un alineamiento es posible iniciar búsquedas de secuencias que encajen en

dicho perfil. Cuando el alineamiento de una secuencia con un perfil da una puntuación por encima de un cierto umbral se considera que en dicha secuencia existe el motivo correspondiente al perfil. Por definición los perfiles, suelen tener una mayor sensibilidad de búsqueda y ser más robustos que los patrones. Esto se debe a que los perfiles integran mejor la información sobre las características conservadas y variables del dominio o región de la secuencia que definen [41]. La figura 1. 6 muestra un perfil a partir de un alineamiento múltiple de secuencias. Para cada posición de la secuencia se asigna un valor a cada aminoácido a partir de su frecuencia, sus propiedades físico-químicas y la matriz BLOSUM62.

	F	K	L	L	S	H	C	L	L	V
	F	K	A	F	G	Q	T	M	F	Q
	Y	P	I	V	G	Q	E	L	L	G
	F	P	V	V	K	E	A	I	L	K
	F	K	V	L	A	A	V	I	A	D
	L	E	F	I	S	E	C	I	I	Q
	F	K	L	L	G	N	V	L	V	C
A	-18	-10	-1	-8	8	-3	3	-10	-2	-8
C	-22	-33	-18	-18	-22	-26	22	-24	-19	-7
D	-35	0	-32	-33	-7	6	-17	-34	-31	0
E	-27	15	-25	-26	-9	23	-9	-24	-23	-1
F	60	-30	12	14	-26	-29	-15	4	12	-29
G	-30	-20	-28	-32	28	-14	-23	-33	-27	-5
H	-13	-12	-25	-25	-16	14	-22	-22	-23	-10
I	3	-27	21	25	-29	-23	-8	33	19	-23
K	-26	25	-25	-27	-6	4	-15	-27	-26	0
L	14	-28	19	27	-27	-20	-9	33	26	-21
M	3	-15	10	14	-17	-10	-9	25	12	-11
N	-22	-6	-24	-27	1	8	-15	-24	-24	-4
P	-30	24	-26	-28	-14	-10	-22	-24	-26	-18
Q	-32	5	-25	-26	-9	24	-16	-17	-23	7
R	-18	9	-22	-22	-10	0	-18	-23	-22	-4
S	-22	-8	-16	-21	11	2	-1	-24	-19	-4
T	-10	-10	-6	-7	-5	-8	2	-10	-7	-11
V	0	-25	22	25	-19	-26	6	19	16	-16
W	9	-25	-18	-19	-25	-27	-34	-20	-17	-28
Y	34	-18	-1	1	-23	-12	-19	0	0	-18

Figura 1.6 Ejemplo de un Perfil.

Por ejemplo: Se desea alinear la secuencia FKTLGCCLLV con el perfil de la figura 1. 6. Se busca cada aminoácido de la secuencia en cada posición del perfil y se obtiene la siguiente puntuación: 60 25 -6 27 28 -26 22 33 26 -16. Si se suman los valores el resultado es 173. Si este valor supera un umbral establecido podremos decir que la secuencia es similar a la información almacenada en el perfil.

5. *Modelos Ocultos de Markov (HMM)*: pueden ser usados como una forma más sensible de búsqueda de similitudes entre secuencias de proteínas basadas en una descripción estadística de la estructura primaria consenso de una familia de secuencias. Un modelo linear de cadenas ocultas de Markov se corresponde con una secuencia de nodos para cada posición en un alineamiento múltiple. En la figura 1.7 se muestra un HMM para un alineamiento de 4 secuencias con tres posiciones o estados (m_1, m_2, m_3) en la terminología de HMMs, cada posición tiene 20 valores de probabilidad de ser uno de los 20 posibles aminoácidos (barras), cuatro estados de inserción (i_0, i_1, i_2, i_3) y tres estados de deleción[1] (d_1, d_2, d_3). Las flechas representan las probabilidades de transición entre estados. Todos o algunos de los parámetros se estiman del alineamiento. Para este caso el HMM funciona de la siguiente manera: Existen tres posibles estados para cada posición de un aminoácido en un alineamiento de secuencias particular: un estado principal donde un aminoácido puede ser emparejado o desemparejado, un estado de inserción donde un nuevo aminoácido puede ser adicionado a una de las secuencias para generar un alineamiento, o un estado de deleción donde un aminoácido puede ser borrado a una de las secuencias para generar un alineamiento. Probabilidades son asignadas a cada uno de estos estados basado en el número de eventos encontrados en la secuencia de alineamiento. Así, cuando se introduce una secuencia el resultado será un valor de probabilidad que representa la similitud entre ésta y la almacena en el HMM. La ventaja de usar HMMs es que tienen unas bases probabilísticas formales, y por tanto, se puede usar teoría probabilística Bayesiana para definir los parámetros del sistema [41].

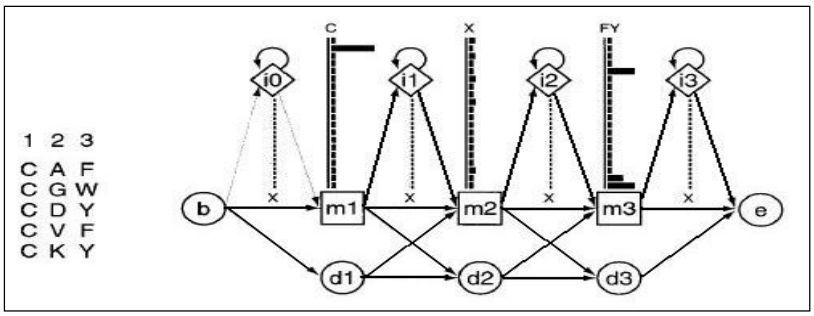


Figura 1.7 Ejemplo de HMMs para alineamiento de secuencias.

[1] Es la pérdida de un fragmento en la secuencia de la proteína.

6. *Motivos:* son otro método de representación de los elementos comunes a un conjunto de datos, utilizando el lenguaje de expresiones regulares. Las expresiones regulares son reglas de escritura utilizadas en computación para manipular cadenas de caracteres. Así, la cadena K-P(0,1)-[DE]-[NQ]-[YF]-K-[VM]-H constituye un motivo posible. De esta manera, se pueden descubrir relaciones entre proteínas comparando sus motivos representados según las reglas en PROSITE (ver sección 2.3.2) utilizando técnicas computacionales. Para ello, es necesario el desarrollo de métodos y algoritmos para el análisis de esta información.

Como ya hemos comentado, la comparación de motivos ha sido objeto de estudio desde hace varios años, para lo cual se han propuesto algunas técnicas y métodos [4]. En particular, algunos trabajos usando la Computación Evolutiva son: en [42] se presenta la Programación Genética utilizando las Funciones Definidas Automáticamente para el descubrimiento automático de motivos de proteínas. Para la evaluación de los motivos se empleó la función de correlación. En [43] se implementa un algoritmo de Programación Genética Lineal para la evolución de un pequeño subconjunto de motivos en PROSITE, utilizan un hardware especializado, llamado "Pattern Matching Chip" (PMC). En [44] se presenta el GPKernel, que genera motivos de proteínas a partir de los datos disponibles en una base de datos, éstos permiten clasificar funciones o familias proteicas. Los motivos son creados desde una gramática de una expresión regular simple y el resultado del emparejamiento contra el conjunto de datos es utilizado para construir vectores de características para una maquina de soporte vectorial. La Programación Genética es utilizada para crear los motivos.

Algunos trabajos basados en Redes Neuronales Artificiales son: en [45] se describe un sistema de red neuronal, basado en un diseño de identificación de motivos (MOTIFIND) que incorpora información de las secuencias de proteínas. Ha sido implementado en gran escala para la identificación de familias de proteínas. Se entrenan Redes Neuronales de Retropropagación por cada familia de proteína, luego son usadas como filtro para detectar nuevos miembros usando las secuencias de proteínas almacenadas en las base de datos SWiss-Prot y PIR. La propuesta de [46] es un sistema para clasificar proteínas basado en

redes neuronales. Se propone un método que asigna una secuencia de la proteína en un espacio de características numéricas mediante la puntuación de la secuencia correspondiente a los grupos de los patrones almacenados (llamado motivos) de las familias de proteínas. En [47] se ha desarrollado una Red de Auto-organización Neuronal con dos capas para resolver el problema de la identificación de motivos de ADN y proteínas. El número de neuronas de la capa de salida es dos que indica si el patrón es un motivo o no igual al número de categorías a clasificar por la red y el número de neuronas de entradas es igual a la longitud del patrón.

Por otro lado, encontrar el motivo común a un conjunto de motivos es un problema de reconocimiento de patrones. La mayoría de los algoritmos de búsqueda de motivos comunes usan técnicas heurísticas para obtener soluciones con un costo computacional relativamente bajo [4]. Por ejemplo, algunos trabajos basado en algoritmos bio-inspirados son: en [48] se presenta un enfoque basado en la Optimización de Colonias de Hormigas que combina una estrategia Max-Min[2] para secuencias con cuatro nucleótidos (A, C, T, G). En [49] se implementa un método basado en la Optimización de Colonias de Hormigas y el algoritmo de maximización de expectativas (EM) para descubrir motivos de ADN (en concreto, para las colecciones de TFBSs) en un conjunto de bio-secuencias.

1.3. PLANTEAMIENTO DEL PROBLEMA

Uno de los campos más importantes en Bioinformática está relacionado con la identificación de motivos en una proteína. Este cubre un conjunto de objetivos, tales como, la búsqueda de genes específicos en los genomas, identificar regiones en las proteínas, predecir dominios proteicos, entre otros. El método consiste en buscar un conjunto de semejanzas en un grupo de genes o de proteínas que se piensan pueden estar emparentados. Los problemas que emergen son interesantes porque enfrentan el uso de métodos matemáticos e informáticos para comparar los motivos. Además, normalmente los tiempos

[2] Solo se agrega el feromona a las mejores soluciones cuando se actualiza el camino recorrido por las hormigas. Además, usa un mecanismo simple para limitar la cantidad de feromona a adicionar y así evitar la convergencia temprana en la búsqueda [105].

de ejecución de los algoritmos para obtener un resultado son de gran complejidad algorítmica.

Para comprender la utilidad de la comparación de motivos es necesario comprender como evolucionan las proteínas, y así conocer porque existen similitudes entre éstas. 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 en 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. Por lo tanto, al comparar motivos de proteínas podemos:

1. Descubrir e identificar regiones concretas conservadas en la secuencia de la proteína, sin que necesariamente dicha similitud sea extensible a la totalidad o resto de la secuencia proteica. Los motivos conservados son el resultado de la presión selectiva sobre regiones restringidas y específicas de las proteínas, impuesta por requerimientos estructurales y/o funcionales importantes.

2. Descubrir que proteínas tienen un origen común.

3. Predecir la estructura de la proteína (proteínas homologas tienen la misma estructura tridimensional).

4. Predecir la función de la proteína (aunque es posible que proteínas con un origen común pueden terminar desarrollando distintas funciones).

Por otra parte, estos motivos existen debido a que ciertas regiones de las proteínas son imprescindibles para mantener sus propiedades biológicas. Estas pequeñas regiones permiten tener una idea de cuál puede ser el papel funcional que una proteína está jugando en la célula. De esta manera podemos construir un patrón común a estos motivos, que representen estas pequeñas regiones conservadas, y así tener una relación de cuales motivos se conservan entre familias de proteínas.

La figura 1.8 muestra esquemáticamente los pasos para comparar motivos de proteínas. Se dispone de un motivo bajo estudio (paso 1), generalmente se puede extraer de las bases de datos la información relacionada con este motivo; buscando otros motivos similares a él, o buscando motivos cuya información asociada tenga algo en común con él (paso 3). Además se pueden buscar patrones funcionales, estos son pequeñas regiones caracterizadas previamente, con un significado funcional o estructural en el motivo bajo estudio (paso 2). Luego se procede a realizar la comparación del motivo bajo de estudio con los motivos almacenados en la base de datos (paso 4), para así encontrar motivos similares al motivo bajo estudio, o que tengan alguna propiedad en común (paso 5). También se puede definir un motivo común a los motivos utilizados en la comparación, y usarlo para recuperar motivos que a pesar de haber perdido la similitud a nivel global, contengan esa región (paso 6), para así posteriormente encontrar nuevos motivos en distintas proteínas, ya que si un motivo está bien definido, detectará una característica biológica común importante en la base de datos (paso 7).

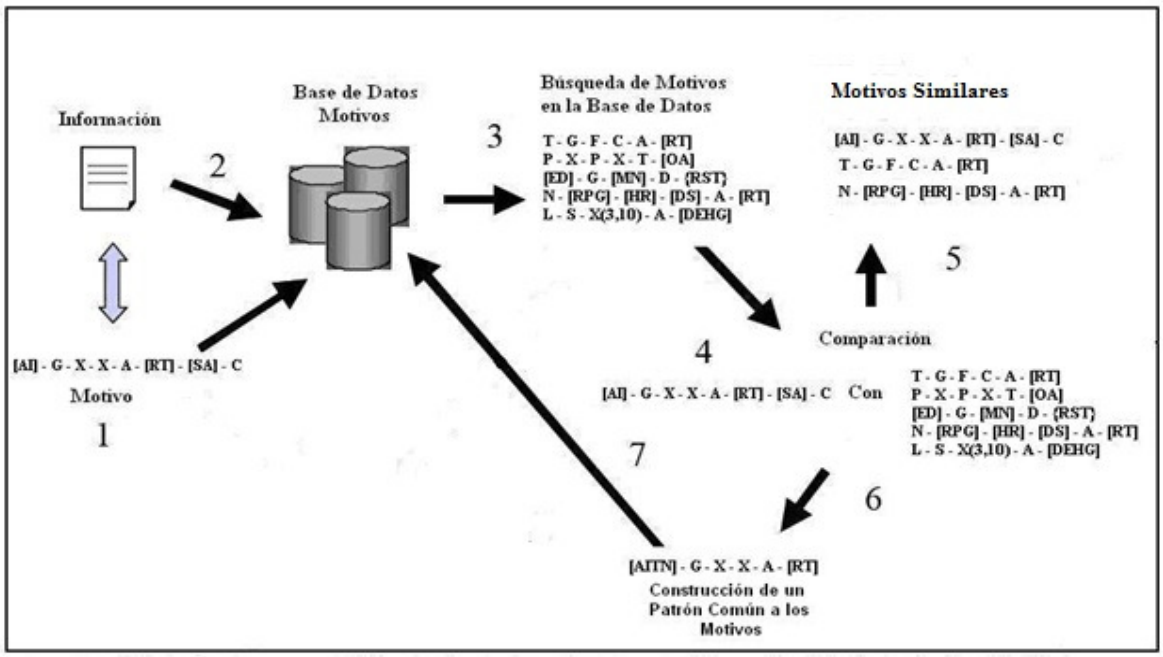


Figura 1.8 Diagrama del problema propuesto.

De esta manera, es necesario definir y desarrollar un método matemático/informático de comparación de expresiones regulares (motivos de proteínas), representadas según las reglas PROSITE (ver sección 2.5), (ver figura 1.9), que permita descubrir relaciones entre motivos de proteínas con más certeza que comparando sus secuencias. Además, dicho método debe hacer posible la construcción de patrones comunes para ese grupo de motivos de proteínas, representados como expresiones regulares. Esto constituye un enfoque innovador que podría tener aplicaciones generales en otros campos.

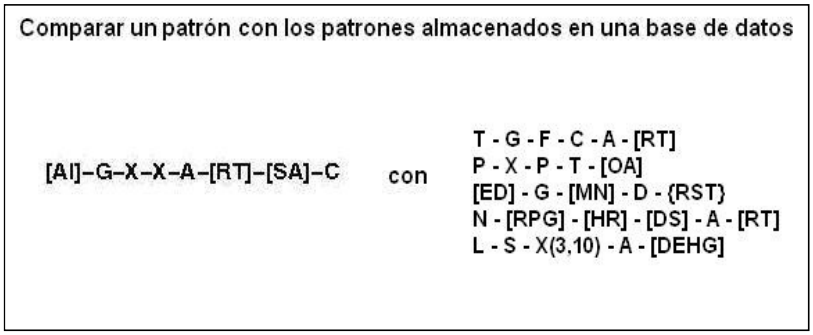


Figura 1.9 Comparar un patrón con los patrones almacenados en una base de datos.

Así, es posible descubrir relaciones entre las proteínas construyendo un motivo común entre varios motivos. Las relaciones que se buscan pueden ilustrarse a través del siguiente ejemplo: Sean 3 motivos S1 (C-x-H-x-[LIVMFY]-C-x(2)-C-[LIVMYA]), S2 (H-A-M-C-x(2)-C) y S3 (H-x-L-C-{R}-C). Se observa que S2 y S3 son sub-motivos de S1. Se puede escribir un motivo común que sería (H-x-[ML]-C-x(2)). Si suponemos que S1, S2, S3 son motivos específicos de 3 familias distintas y que ese motivo común no corresponde a ninguna secuencia de las familias 1, 2 y 3, representa un motivo consenso. Así como ese ejemplo, otros análisis biológicos para grupos de motivos se podrían hacer. Para ello será necesario definir un método de comparación de motivos y de generación de un patrón común para ellos, y luego probar la calidad de éste último.

1.4. OBJETIVOS

OBJETIVO GENERAL

Definir y desarrollar modelos de reconocimiento de patrones de proteínas usando expresiones regulares, e implementarlos en algoritmos eficientes y óptimos para descubrir motivos de proteínas amiloideas, utilizando para ello la base de datos AMYPdb.

OBJETIVO ESPECIFICOS

1. Proponer un algoritmo para comparar dos expresiones regulares escritas según las reglas PROSITE. El método de comparación debe asignar un valor a la comparación según la semejanza de las expresiones regulares.

2. Construir un patrón común según las reglas PROSITE, para las expresiones regulares que tienen un alto grado de semejanza.

3. Utilizar los métodos propuestos en los motivos almacenados en la base de datos AmyPdb (Amyloid Protein Data Bank) para hacer un análisis biológico del contenido de dicha base de datos.

1.5. ALCANCE

Se desea culminar éste proyecto con la implantación de un sistema que pueda realizar satisfactoriamente dos tareas:

1. La comparación de dos motivos de proteínas denotados en expresiones regulares según las reglas PROSITE. El método de comparación de los motivos está basado en la Programación Genética para construir un conjunto de secuencias validas para los motivos bajo estudio, y la Red Neuronal de Retropropagación para la comparación de las secuencias.

2. La fusión de dos motivos denotados en expresiones regulares según las reglas PROSITE, utilizando para ello un algoritmo de Optimización por Colonias de Hormigas.

Además, se desea cumplir con los requerimientos de eficiencia necesarios para que el sistema pueda ejecutarse con los recursos disponibles en una computadora personal moderna bajo los estándares del desarrollo de software libre GNU.

Nuestro sistema podrá comparar y fusionar motivos de proteínas largos, degenerados y flexibles, como los que se encuentran en las bases de datos ScanPROSITE [50] o AMYPdb [24] (no solo motivos de ADN), lo que lo hace más amplio versátil y genérico que las herramientas que existen actualmente. Además, podrá conocer si algunas familias de proteínas tienen puntos comunes.

1.6. ORGANIZACIÓN DE LA TESIS

El Capítulo II comprende los aspectos teóricos necesarios para la comprensión del trabajo desarrollado en la tesis. Los temas principales que aborda ese capítulo son: Bioinformática, Biología Molecular, Expresiones Regulares y las Técnicas Inteligentes utilizadas.

En el Capítulo III se define el Diseño del Sistema. Se realiza la definición formal del esquema de comparación y fusión, además se presenta su estructura global como cada uno

de sus componentes: el Sub-sistema de Comparación de Motivos y el Sub-Sistema de Fusión de Motivos.

El Capítulo IV contiene las pruebas realizadas al sistema implementado y el análisis de los resultados obtenidos. Los casos de prueba permiten determinar la eficacia y eficiencia del Sistema Propuesto.

El Capitulo V contiene las conclusiones generales de la Tesis y los Trabajos Futuros que podrían ser realizados para continuar investigando en este campo del conocimiento.

CAPÍTULO II: MARCO TEORICO

Esta sección presenta los aspectos teóricos más relevantes que serán usados durante el desarrollo de este trabajo en las áreas de: Bioinformática, Biología Molecular, Aminoácidos, Proteínas (presentando particularmente las amiloideas, que serán las estudiadas en nuestro trabajo), Expresiones Regulares, Motivo, PROSITE, y las Técnicas Inteligentes: Programación Genética, Redes Neuronales Artificiales (haciendo hincapié en la usada en esta Tesis, la Red Neuronal de Retropropagación), y Sistemas Artificiales de Hormigas.

2.1. BIOINFORMATICA

La Bioinformática es una disciplina científica emergente, que utiliza la tecnología de la información y el modelado matemático para aplicarlo en Bioquímica, Biofísica y Biología Molecular. Su objeto es la colección, mantenimiento, distribución, análisis y uso de grandes cantidades de datos generados en tales disciplinas, para facilitar el descubrimiento de nuevas ideas biológicas, así como crear modelos globales a partir de los cuales se puedan discernir principios unificadores en la Biología.

La Bioinformática en un comienzo surgió para solucionar problemas de cálculo con información biológica, más tarde problemas de almacenamiento y análisis de grandes cantidades de datos extraídos de organismos, y en la actualidad se ha transformado en una disciplina con líneas propias de investigación que dejo de ser solo una herramientas para los biólogos, [30], [51], [52]. Uno de los retos de la Bioinformática es el desarrollo de métodos que permitan integrar los datos genómicos[3] – de secuencia, de expresión, de estructura, de interacciones, etc. – para explicar el comportamiento global de la célula viva y del organismo, minimizando la intervención humana.

De manera general, las herramientas Bioinformáticas se han desarrollado como respuesta a las necesidades de análisis de la información biológica, entendida esta como la adquisición y consulta de datos, el análisis de las correlaciones entre ellos, la generación de conocimiento desde los datos, entre otras cosas. Este conocimiento almacenado se puede utilizar directa o indirectamente. Una forma de utilización directa implica encontrar secuencias similares a la

[3] La genómica es una nueva sub-disciplina de la Genética cuyo objetivo es el estudio sistemático de genomas completos. [104].

secuencia problema, o secuencias que tengan alguna propiedad en común. Una forma de utilización indirecta consiste en usar la información para obtener reglas que permitan posteriormente predecir, con mayor o menor éxito, propiedades en secuencias nuevas. Algunos de los objetivos fundamentales de la Bioinformática son la predicción de la estructura de las proteínas a partir de su secuencia, la predicción de las funciones biológicas y biofísicas a partir de la secuencia o la estructura, así como simular el metabolismo y otros procesos biológicos basados en esas funciones [30], [51], [52]. Los estímulos principales para el desarrollo de la Bioinformática son [53]:

a) El enorme volumen de datos generados por los distintos proyectos del genoma[4] (humano y de otros organismos) y de la proteómica[5].

b) Los nuevos enfoques experimentales que permiten obtener datos genéticos a gran velocidad, bien de genomas individuales (mutaciones, polimorfismos), de células (expresión génica) y de sus productos (las proteínas).

c) El desarrollo de Internet que permite el acceso universal a las bases de datos de información biológica.

Se debe distinguir tres acepciones en las que se unen la biología y la computación, las cuales tienen objetivos y metodologías bien diferenciadas [54]:

1. *Bioinformática o Biología Molecular Computacional:* Consiste en la aplicación de la informática en la Biología Molecular y la Genética, proporcionando las herramientas y recursos necesarios para favorecer la investigación biomédica. Comprende la investigación y el desarrollo de sistemas útiles para entender el flujo de información desde los genes a las estructuras moleculares.

2. *Biología Computacional:* Consiste en la aplicación de la Informática y Matemáticas en la Biología, para la cual se utiliza la computación para el entendimiento de problemas

[4] Es el contenido total de material genético característico en una especie [106].

[5] Es el estudio de las propiedades de las proteínas (nivel de expresión, modificaciones post-traduccionales, interacciones proteicas, etc.) a gran escala [106].

biológicos básicos, no necesariamente en el nivel molecular, mediante el modelado y la simulación (ecosistemas, modelos fisiológicos).

3. *Biocomputación:* Consiste en la Biología aplicada a la computación mediante el desarrollo y utilización de sistemas computacionales basados en modelos y materiales biológicos (biochips, biosensores, computación basada en ADN, redes neuronales artificiales, computación evolutiva, etc.).

Específicamente, la Bioinformática estudia cómo correlacionar la información disponible para extraer los patrones que se encuentran en ésta, y avanzar así en el conocimiento del funcionamiento de los organismos vivos. De esta manera, la tarea principal de la Bioinformática consiste en proporcionar sentido biológico a los datos, ya que la mera acumulación de los mismos no conlleva un aumento en el conocimiento. La fuente de la que se nutre principalmente la Bioinformática la constituyen una serie de bases de datos de acceso público donde se acumula, continuamente, toda la información disponible. Éstas se han especializado según la naturaleza de los datos almacenados, siendo las principales las que recogen secuencias de genes, genomas y proteínas (Gen-Bank [55], EMBL [56] DDBJ [57], Uniprot [58]), estructuras tridimensionales de macromoléculas (ProteinData Bank [59]), datos de expresión génica (Array Express [60]; Stanford Microarray Database [61]), ontologías (GeneOntology [62]), entre otras (ver PubMed [55] para más detalles).

2.2. ASPECTOS DE BIOLOGIA MOLECULAR DE INTERES PARA NUESTRO TRABAJO

La Biología Molecular es una disciplina que se ocupa del estudio de las bases moleculares de la vida, incluyendo la Bioquímica, de moléculas como el ADN, el ARN o las proteínas, y la estructura molecular y funciones de las diferentes partes de las células [63]. Para comprender el problema de comparación y fusión de motivos de proteínas amiloideas, necesitamos tener claro algunos conceptos sobre las proteínas (aminoácidos, proteínas amiloideas) que se describen a continuación.

2.2.1. AMINOACIDOS

Los aminoácidos son moléculas orgánicas formadas 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); c) un átomo de hidrogeno (–H); y d) una cadena lateral característica (–R) [64] (ver la figura 2.1).

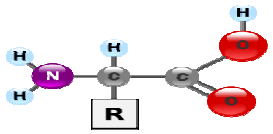


Figura 2.1 Estructura básica de un aminoácido.

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, así como de la carga eléctrica de las moléculas. El conocimiento de las propiedades de estas cadenas laterales es importante para analizar e identificar las proteínas. Así, los aminoácidos con cadenas laterales cargadas, polares o hidrófilas están localizados en la superficie de las proteínas. En cambio, los aminoácidos hidrófobos no polares suelen estar en el interior o en la parte central de la proteína. En la tabla 2.1 se muestran los 20 aminoácidos presentes en las proteínas [64], [65], [66]. La estructura de los diferentes aminoácidos se observa en la figura 2.2.

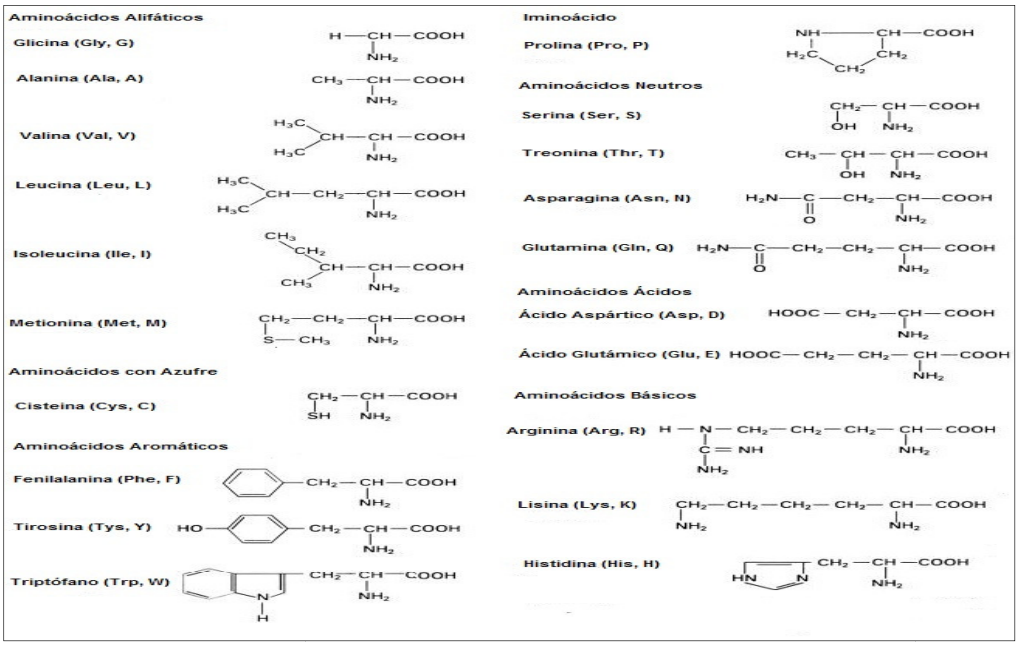


Figura 2.2 Estructura de los diferentes aminoácidos

Abrev.		Nombre	Grupo Funcional
A	Ala	Alanina	Aminoácido Alifático
C	Cys	Cisteína	Aminoácido con Azufre
D	Asp	Ácido aspártico	Aminoácido Acido
E	Glu	Ácido glutámico	Aminoácido Acido
F	Phe	Fenilalanina	Aminoácido Aromático
G	Gly	Glicina	Aminoácido Alifático
H	His	Histidina	Aminoácido Básico
I	Ile	Isoleucina	Aminoácido Alifático
K	Lys	Lisina	Aminoácido Básico
L	Leu	Leucina	Aminoácido Alifático
M	Met	Metionina	Aminoácido Alifático
N	Asn	Asparagina	Aminoácido Neutro
P	Pro	Prolina	Iminoácido
Q	Gln	Glutamina	Aminoácido Neutro
R	Arg	Arginina	Aminoácido Básico
S	Ser	Serina	Aminoácido Neutro
T	Thr	Treonina	Aminoácido Neutro
V	Val	Valina	Aminoácido Alifático
W	Trp	Triptófano	Aminoácido Aromático
Y	Tyr	Tirosina	Aminoácido Aromático

Tabla 2.1 Lista de Aminoácidos

Los aminoácidos se encuentran unidos linealmente por medio de uniones peptídicas. El enlace pepitico es un enlace amida que se forma entre el grupo carboxilo de un aminoácido con el grupo amino de otro, con la eliminación de una molécula de agua (ver figura 2.3) [67].

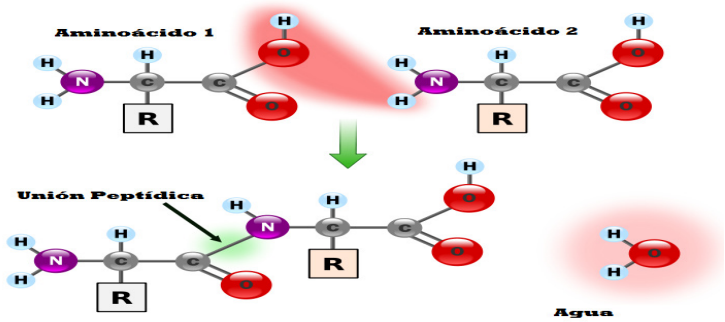


Figura 2.3 Formación de un enlace peptídico

2.2.2. PROTEINAS

Las proteínas son polímeros de aminoácidos (más de 100 aminoácidos, si son menos de 100 se denomina polipéptido) que ejecutan la mayor parte de las funciones vitales de las células [68]: el reconocimiento molecular, el transporte de moléculas, la función estructural, la catálisis de las reacciones químicas, entre otras. 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 las proteínas, podrían ocasionar la pérdida o el mal funcionamiento de las mismas [65]. La estructura de las proteínas se clasifica en varios niveles:

1. *Estructura Primaria:* representa la secuencia de aminoácidos que componen la proteína, ordenados desde el primer aminoácido hasta el último. El primer aminoácido tiene libre el grupo amina, se denomina aminoácido n-terminal. El último aminoácido tiene libre el grupo carboxilo, se denomina aminoácido c-terminal (ver figura 2.4) [65].

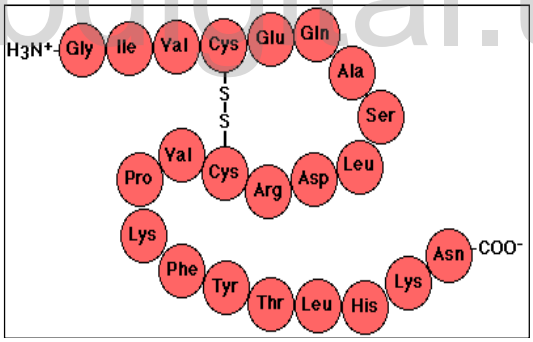


Figura 2.4 Estructura primaria de una proteína

2. *Estructura Secundaria*: describe la orientación de los segmentos de la cadena de la proteína en un patrón regular [69]. Las conformaciones resultantes son:

a. *Hélice-α:* Es una estructura cilíndrica, formada por una cadena polipeptídica que constituye la estructura central, y las cadenas laterales se extienden por fuera de la hélice. Ésta se

estabiliza[6] debido a la cantidad de puentes de hidrógeno que se establecen entre los aminoácidos de la espiral (ver figura 2.5) [70].

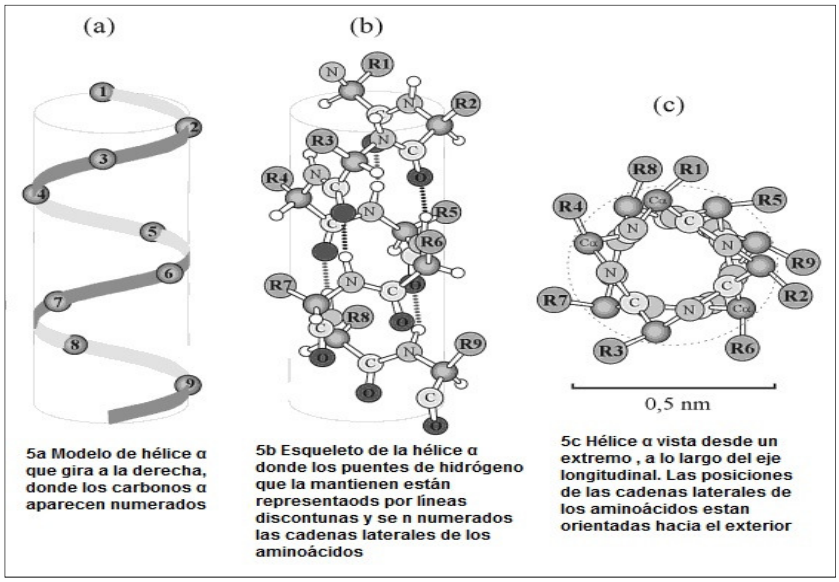


Figura 2.5 Diferentes modelos de la hélice- α

b. *Lamina-β:* Es una estructura en forma de zig–zag, permite la asociación de dos o más cadenas dispuestas una al lado de la otra. De esta forma logran su estabilidad mediante puentes de hidrogeno entre las cadenas subyacentes. Hay dos tipos de láminas-β que se observan en las proteínas: láminas-β paralelas y láminas-β anti–paralelas (ver figura 2.6) [69].

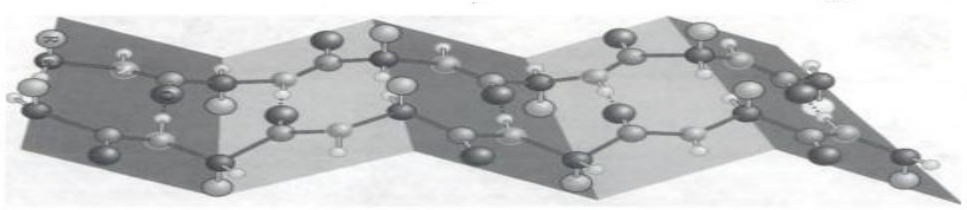


Figura 2.6 Forma tridimensional de la lámina-β

c. *Hélice de Colágeno:* Es una estructura helicoidal, formada por hélices más abiertas y rígidas que en la estructura hélice-α. Esto es debido a la existencia de gran número de aminoácidos

[6] La estructura de las proteínas está estabilizada por diferentes tipos de enlaces, como enlaces covalentes (enlace peptídico, enlace por puentes disulfuro), enlaces por puentes de hidrógeno (interacciones dipolo-dipolo), interacciones hidrofóbicas, enlaces salinos (interacciones electrostáticas) o las fuerzas de los contactos de Van der Waals (atracciones eléctricas débiles entre diferentes átomos) [107].

prolina y glicina, estos aminoácidos tienen una distribución en forma de anillo, formando una estructura rígida en el carbono asimétrico que le imposibilita girar (ver figura 2.7) [67].

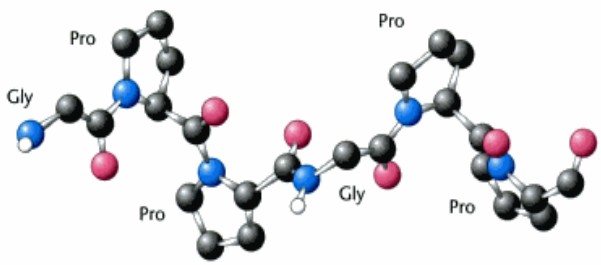


Figura 2.7 Hélice de Colágeno

El resto de los segmentos de la proteína se referencian como una línea, esta región no tiene una estructura secundaria definida.

3. *Estructura Terciaria:* Es la disposición tridimensional de todos los átomos que componen la proteína, la disposición espacial de los distintos tipos de estructuras secundarias determina su interacciones y su función biológica. Existen 2 tipos de estructuras terciarias: de tipo fibroso y de tipo globular (ver figura 2.8) [70].

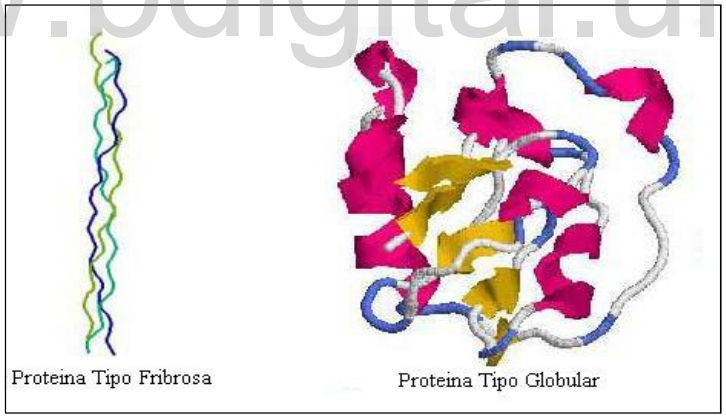


Figura 2.8 Tipos de estructuras terciarias

4. *Estructura Cuaternaria:* Cuando varias proteínas se unen entre sí, forman una organización superior, denominada estructura cuaternaria (ver figura 2.9). Cada proteína componente de la asociación conserva su estructura terciaria. La estructura cuaternaria modula la actividad biológica de la proteína, y la separación de las subunidades conducen a la pérdida de la funcionalidad [70].

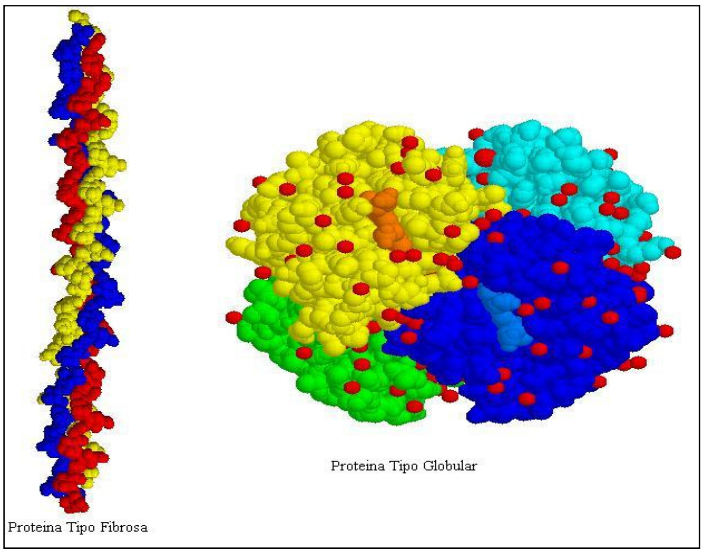


Figura 2.9 Tipos de estructuras cuaternarias

2.2.3. PROTEINA AMILOIDEA

La expresión amiloidea fue utilizada por primera vez por el médico alemán Rudolph Virchow en 1854 [71]. 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 celulosa, denominándola "*amiloidea*" (del latín amylum y del griego amylon, que significa almidón) [71]. 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 [72], [73], [23].

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. Los mecanismos de conversión de proteínas normales solubles en polímeros filamentosos insolubles son aún desconocidos (ver figura 2.10). Se sabe hoy que estos depósitos pueden ser intra 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 se encuentran las enfermedades neurodegenerativas [74]: Alzheimer, Parkinson, Diabetes, entre otras. Distintos

autores sugieren que la capacidad para formar las fibras es una propiedad intrínseca de todas las cadenas polipeptídicas, pero que las proteínas amiloideas son más aptas que otras proteínas para cambiar de estructura *in vivo*, en función de condiciones medioambientales y/o de la presencia de algunas mutaciones [72], [73], [75].

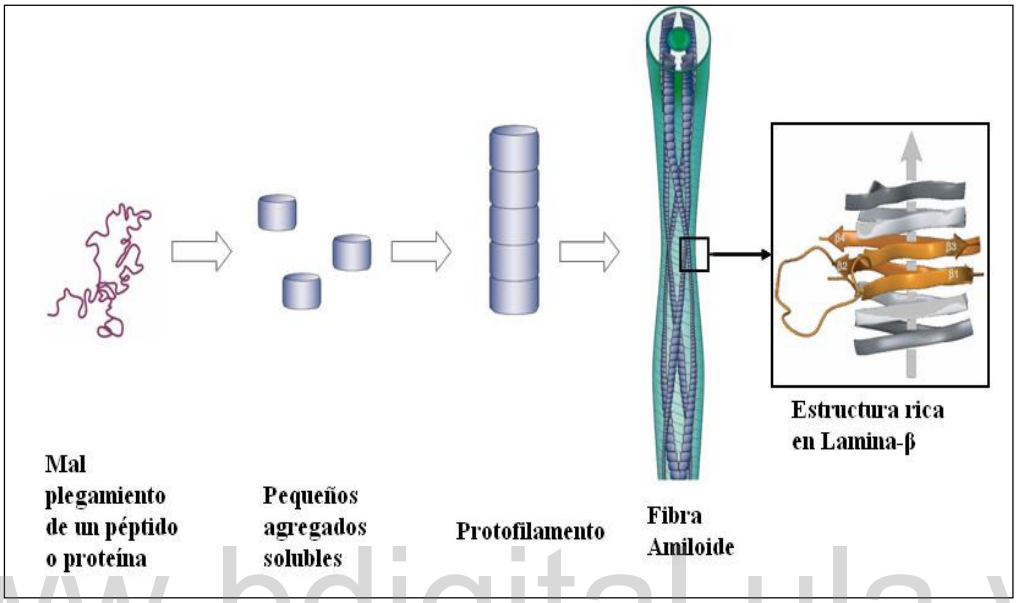


Figura 2.10 Mecanismos de conversión de proteínas normales solubles en fibra amiloidea

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) [24]. 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 amiloideas. 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. Estos patrones están representados en las reglas en PROSITE [23].

Existen dos bases de datos "competidoras" con AMYPdb, pero cuyos objetivos son muy limitados (no contienen patrones PROSITE): Fiblil_one es una recopilación de mutaciones de proteínas amiloideas clasificadas, pero solo reconoce 50 secuencias en total; y PrionDB está limitada a priones, que es un caso particular de las proteínas amiloideas, constituyen pequeñas

partículas infecciosas de naturaleza proteica, carentes de ADN, que se propagan catalizando el cambio de conformación de su forma natural a su forma fibrilar [76].

2.3. EXPRESIONES REGULARES Y SU USO EN BIOLOGÍA

Las expresiones regulares permiten denotar lenguajes regulares, y su estudio resulta de gran interés por su capacidad de especificación mediante un número reducido de operadores. Se denominan expresiones regulares sobre un alfabeto A, a las expresiones que se pueden construir a partir de las siguientes reglas [77]:

1. El símbolo Ø es una expresión regular e indica el lenguaje vacío.

2. El símbolo λ es una expresión regular e indica el lenguaje {λ}.

3. ε es una expresión regular que describe el lenguaje {ε}, esto es el lenguaje que contiene únicamente la cadena vacía.

4. Si a $\in \Sigma$ entonces a es una expresión regular que indica el lenguaje {a}.

5. Si α y β son expresiones regulares entonces:

a. α | β es una expresión regular que indica la unión de los lenguajes mostrados por α y por β, o $L(\alpha \mid \beta) = L(\alpha) \cup L(\beta)$. Por ejemplo a | b corresponde tanto al carácter a como al carácter b, entonces, $L(a|b) = L(a) \cup L(b) = \{a\} \cup \{b\} = \{a, b\}$

b. αβ es una expresión regular que indica la concatenación de los lenguajes mostrados por α y β. Solo es necesario colocar dos expresiones regulares, una a continuación de la otra. Por ejemplo, la concatenación de “a” con “b” sería sencillamente “ab”. En el caso de que las expresiones regulares contengan operadores (como por ejemplo el de disyunción), es posible que se necesiten paréntesis para poder establecer las prioridades de evaluación. Como ejemplo de uso de paréntesis y concatenación podemos ver la expresión (a|b)c que denota las cadenas “ac”, “bc”.

c. α* es una expresión regular que indica la clausura del lenguaje mostrado por α. Por ejemplo a* tienen una "a", ninguna o muchas aes.

d. α+ es una expresión regular que indica que la expresión α debe repetirse al menos una vez. Por ejemplo si tenemos "ba+" entonces "ba", "baa", "baaaaa" son cadenas válidas, pero no es válida "b".

e. α(n) o α(n,m), indica el número de repeticiones de un elemento en una expresión mediante el operador "(n)" o "(n,m)" donde n indica el número mínimo de repeticiones y m el máximo. Por ejemplo:

   a(3) denota las cadena que tiene exactamente 3 "a". aaa

   a(3,) denota la cadena que tiene mínimo 3 "a". aaa, aaaa, aaaaaaa, aaaaaaaaa.

   a(3,5) denota las cadena que tiene mínimo 3 "a" y máximo 5 "a". aaa, aaaa, aaaaaa

f. Si se tiene [α, β], indica que solo una de las expresiones regulares puede ser utilizada. Por ejemplo: [D,F] son cadenas validas "D" o "F" pero no es válido "DF"

g. Si se tiene {α, β}, indica que estas expresiones regulares no pueden utilizarse. Por ejemplo: para {A,C,B} son cadenas validas todas aquellas que no contengan a A, C o B

Ejemplo: para el Lenguaje = {a, b, c}

Reglas	Se corresponde con:
abc	Abc
a | b	a, b
a*	Ø, a aa, aaa,...
b+	b, bb, bbb, bbbb, bbbb, bbbbb, …
b | a*	b, Ø, a aa, aaa,...
(a | b)*	Ø, a, aa, aaa,..., b, bb, bbb, ...
(a | b)(a | b)	aa, ab, ba, bb
ab | cd*	ab, c, cd, cdd, cddd, cdddd …
a(bc)*	a, abc, abcbc, ...
a | b*c	a, c, bc, bbc,bbbc, bbbbc, …
ab(2)c*	abb, abbc, abbcc, abbccc, …
a(2,4)bc	aabc, aaabc, aaaabc
abc+	abc, abcc, abccc, abcccc, …
[ab]c+a	aca, acca, accca, bca, bcca, bccca, …
[ab]c{b}	aca, acc, bca, bcc

Tabla 2.2 Definición de reglas sobre el lenguaje {a, b, c}

La gramática que contiene las expresiones regulares está conformada por un conjunto de reglas. Ejemplos de estas reglas pueden ser [77]:

1. Identificador (id):

Letra = a | A | b | B | ... | z | Z

dígito = 0 | 1 | 2 | 3 | ... | 9

Letra_o_Dígito = letra | dígito

id = Letra(Letra_o_Dígito)*

2. Número:

dígito = 0 | 1 | 2 | 3 | ... | 9

digit = 1 | 2 | 3 | ... | 9

dígitos = $digito^+$

digits = $digits^+$

Opc – fracción = $\left(\frac{\text{digitos}}{\text{digits}}\right)^+$

Opc – exp = $\left(e^{(+|-)digitos^+}\right)^+$

num = $(dígitos|\ Opc-\ fracción|\ digitos\ Opc$ – exp) $^+$

Generalmente, la gramática está compuesta por cuatro conjuntos [77]:

a) Conjunto de Terminales (T). Ejemplo: identificador (id), número (num).

b) Variables Sintácticas o No Terminales (NT). (En el ejemplo que sigue: expresión (expr), operación (op).

c) S, Símbolo del Comienzo.

d) Conjunto de Producciones (P), especifican la manera en que los Terminales y las Variables Sintácticas pueden ser combinados para formar patrones, por ejemplo:

S: expr ⟶ expr op expr | (expr) | -expr | num | id

op ⟶ + | - | * | /

Los motivos de proteínas son descritos como expresiones regulares que permite ser entendidos fácilmente por los humanos. Las expresiones regulares pueden ser usadas para caracterizar los motivos, indicando cuales posiciones son más importantes, cuales pueden variar y qué tipo de variaciones se permiten. A continuación se define que es un motivo y el conjunto de reglas PROSITE que permite definirlos como expresiones regulares.

2.3.1. MOTIVOS

Un motivo es una región o porción de una secuencia de proteína que posee una estructura específica, que describe una función específica de ella [78]. 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. Así, 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 PSSM y los Modelos Ocultos de Markov (HMM) son ejemplos de motivos probabilísticos. Los motivos *determinísticos* son descritos en una expresión regular basada en un lenguaje. Estos motivos pueden ser divididos en dos tipos: longitud fija y longitud extensible. Motivos de longitud fija consisten en una cadena con un tamaño fijo de aminoácidos. Motivos de longitud variable tienen una longitud arbitraria, con un número arbitrario de aminoácidos y gaps[7]. De esta manera, las expresiones regulares son un poderosa notación para caracterizar motivos, indicando cuales posiciones son importantes, cuales pueden variar, y que variaciones pueden suceder [79], [80]. Considere la ecuación 2.1 que representa un patrón abstracto:

$$\boldsymbol{A_i - x(p_1, q_1) - A_2 - x(p_2, q_2) - \cdots - A_n} \qquad \textbf{Ecuación 2.1}$$

[7] Símbolo que sirve para denotar las posiciones en las expresiones regulares, donde podría encontrarse cualquiera de los 20 aminoácidos.

Donde A_i es una secuencia de aminoácidos consecutivos, y $x(p_i, q_i)$ representa un gap mayor o igual que p_i y menor o igual que q_i. El símbolo A representa uno de los veinte aminoácidos (ver tabla 2.1). Tres tipos de motivos de acuerdo a su longitud pueden ser distinguidos:

1. Motivos contiguos o de longitud fija: sin gaps, ejemplo: $p_i = q_i = 0, \forall i$, ejemplo: IPCCPV

2. Motivos con gap rígido: solo contienen gaps con un tamaño fijo, ejemplo: $p_i = q_i, \forall i$. El símbolo x es usado para denotar un gap de tamaño uno y representa cualquier aminoácido. Ejemplo: MNxxAxCA

3. Motivos con gap flexible: permite un número variable de gaps entre eventos de una secuencia, ej: $p_i \leq q_i, \forall i$ ejemplo: ANx(1,3)Cx(4,6)D

2.3.2. PROSITE

PROSITE fue creada en 1988 por Amos Bairoch en el Instituto Suizo de Bioinformática (Swiss Institute of Bioinformatics). Es una base de datos de motivos de relevancia biológica. Su objetivo principal es determinar la función de nuevas proteínas, no caracterizadas en las 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 no aparezcan demasiado frecuentemente por azar, es decir, para que sean específicas de una familia dada [81], [82]. Los motivos en PROSITE son descritos usando las siguientes reglas para representar las expresiones regulares [83]:

1. Para definir cada aminoácido se usa el código estándar de una letra (ver tabla 2.1).

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 se indican por los aminoácidos que no son aceptados en una determinada posición, mediante llaves "{ }". Ejemplo: {AM} 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 indican 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:

ATHD
ATHE

De dicho alineamiento podemos extraer el siguiente motivo común de aminoácidos conservados: A-T-H-[DE]. Por otro lado, uno puede traducir motivos. Por ejemplo, para el motivo: W-x(2)-[LVI]-x(1,3)-{QTS}, la traducción sería: Trp-cualquier aminoácido-cualquier aminoácido-[Leu o Val o Ile]-(cualquier aminoácido una, dos o tres veces)-{cualquier aminoácido excepto Gln, Thr o Ser}.

2.4. COMPUTACION INTELIGENTE

La necesidad de desarrollar sistemas inteligentes se ha incrementado en los últimos años debido a que el conocimiento se ha convertido en un recurso estratégico para ayudarnos en tareas complejas. Esto ha impulsado estudios teóricos dirigidos, fundamentalmente, a lograr una mejor comprensión de los mecanismos de procesamiento de información en los humanos. La comunidad científica computacional ha respondido a estas necesidades a través del área de la Computación Inteligente y, en particular, a través de los mecanismos de integración de las diferentes técnicas inteligentes para conformar los sistemas híbridos inteligentes.

La Computación Inteligente la definiremos como una metodología de cálculo que tiene habilidades para adaptarse a nuevas situaciones, la cual posee atributos tales como: generalización, descubrimiento, asociación, abstracción, entre otros; además, sus respuestas son predicciones o toma de decisiones. De esta manera, la Computación Inteligente envuelve conceptos, paradigmas y algoritmos adaptativos, los cuales permiten generar acciones apropiadas en ambientes complejos y cambiantes (comportamiento inteligente). Los modelos computacionales usados tienen analogía biológica, basados en lo que se ha logrado comprender y concebir como inteligencia.

La Computación Inteligente se ha desarrollado desde dos ángulos:

1. A partir de la implementación de aplicaciones basadas en sus paradigmas para resolver problemas particulares.

2. A partir de la definición de modelos de los diferentes mecanismos de procesamiento de información de los humanos, en este caso se consideran solamente aspectos cognitivos y estructurales.

Por otro lado, las investigaciones sobre como hibridizar las técnicas inteligentes existentes han tenido un papel fundamental en el desarrollo de la Computación Inteligente, con el fin de subsanar las debilidades de cada una de ellas y aumentar el uso potencial de las mismas [84].

A continuación presentamos las técnicas de Computación Inteligente que serán usadas durante esta Tesis: Programación Genética, Redes Neuronales y Colonia de Hormigas.

2.4.1. PROGRAMACIÓN GENETICA

La Computación Evolutiva aplica teorías de la evolución natural y la genética en la adaptación evolutiva de estructuras computacionales, proporcionando un medio alternativo para resolver problemas complejos en diversas aéreas. Una población de posibles soluciones de un problema dado es análoga a una población de individuos vivos que evoluciona en cada generación, la computación evolutiva combina los mejores individuos de la población y transmite las características de dichos individuos a sus descendientes [84].

La Programación Genética es un subdominio de la Computación Evolutiva, creada por John Koza a finales de los años 80. La Programación Genética usa los cuatro pasos de la Computación Evolutiva para la solución de un problema [85], [86]:

1. Generar una población inicial de individuos que representan soluciones potenciales del problema a ser resuelto.

2. Evaluar cada individuo de la población, y asignarle un valor de aptitud de acuerdo a que tan cercano esté de la solución del problema.

3. Crear una nueva población de individuos (fase de reproducción), conformada por: los mejores individuos existentes y por nuevos individuos creados usando los operadores genéticos: copia, cruce, mutación, reemplazo entre otros. Los individuos nuevos reemplazan a los miembros menos aptos de la población.

4. Asignar como resultado del proceso evolutivo, al mejor individuo que aparezca en cualquier generación (la mejor solución).

La Programación Genética hace evolucionar programas, procedimientos o modelos matemáticos, son ellos quienes constituyen su población, más que las soluciones a un problema dado. Los elementos de la Programación Genética son [85], [86]:

1. *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 serán ejecutadas en el programa. El conjunto de funciones pueden ser operaciones aritméticas, lógicas, operadores condicionales, instrucciones de repetición o decisión, etc.

2. *Individuos*: son estructuras arborescentes. Estas estructuras están formadas por nodos funciones y nodos terminales, los cuales son específicos para cada problema (árbol sintáctico).

3. *Población inicial*: es un conjunto inicial de individuos generados aleatoriamente.

4. *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. *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. *Función aptitud:* es una expresión matemática que debe ser capaz de evaluar la calidad de cualquier individuo de la población.

7. *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 en la Programación Genética son: copia, cruce, mutación, intercambio, sustitución.

8. *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. *Criterio para terminar la ejecución*: es el método para designar el resultado final.

Además, la Programación Genética permite la definición automática de funciones (en ingles ADFs), que consiste en una extensión de la Programación Genética mediante la cual se descubren nuevas unidades funcionales de manera automática; esto permite resolver problemas más complejos. El fundamento de la definición automática de funciones es la programación modular, en la cual los programadores escriben funciones o subrutinas que son invocadas una o varias veces durante la ejecución de un programa [86].

2.4.2. REDES NEURONALES ARTIFICIALES

El cerebro consta de un gran número de elementos (aproximadamente 10^{11}) altamente interconectados (aproximadamente 10^4 conexiones por elemento), llamados neuronas. Estas

neuronas tienen tres componentes principales: las dendritas, el cuerpo de la célula o soma, y el axón. Las dendritas son el árbol receptor de la red, el cuerpo de la célula realiza la suma de esas señales de entrada, y el axón lleva la señal desde el cuerpo de la célula hacia otras neuronas (este proceso se llama sinapsis, ver figura 2.11) [84], [87].

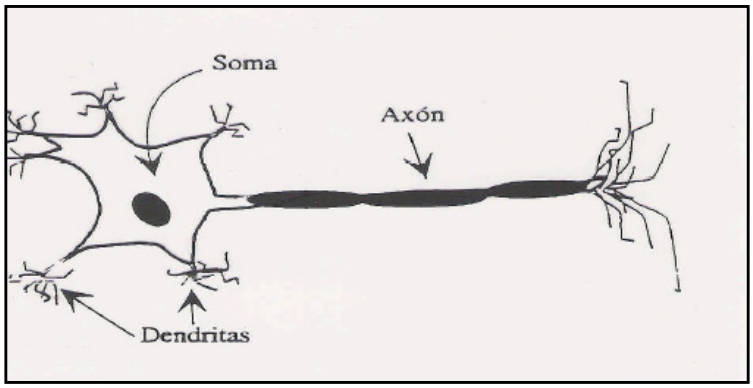


Figura 2.11 Esquema de una neurona natural.

Las Redes Neuronales Artificiales (RNA) emulan las redes neuronales biológicas. Pueden ser consideradas como un sistema de procesamiento de información que tiene las siguientes características:

1. *Adaptabilidad:* es la capacidad para aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.

2. *Auto-organización*: una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.

3. *Tolerancia a Fallas*: es la propiedad que permite a un sistema continuar operando adecuadamente en caso de una falla en alguno de sus componentes.

4. *Robustez*: un sistema es robusto si puede ejecutar diversos procesos de manera simultánea sin generar fallos o bloquearse.

En la figura 2.12 se observa el esquema de una neurona artificial.

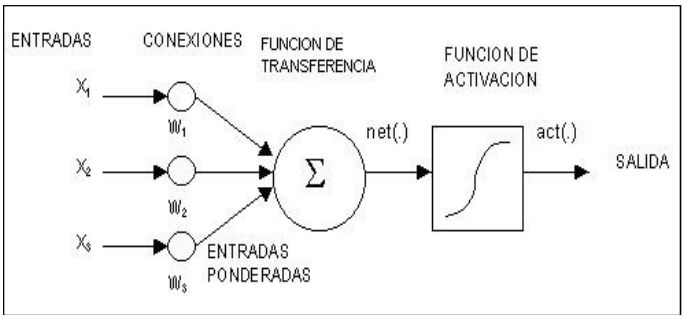


Figura 2.12 Esquema de una neurona artificial.

a) Las entradas X_i representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.

b) Los pesos W_i son la intensidad de la sinapsis que conecta dos neuronas; tanto X_i como W_i son valores reales.

Las señales de entrada a una neuronal artificial X_1, X_2,.., X_n son variables continuas o discretas. Cada señal de entrada pasa a través de una ganancia o peso W_1, W_2,.., W_n, estos pueden ser positivos o negativos, el nodo sumatorio acumula todas las señales de entradas multiplicadas por los pesos, o ponderadas, y las pasa a la salida a través de una función de transferencia, que si es mayor que un umbral activara esa neurona. La entrada neta a la red neuronal puede escribirse de la siguiente manera (ecuación 2.2):

$$\boldsymbol{neta_i = \sum_{i=1}^{n} W_i X_i = \vec{W}\,\vec{X}} \qquad \textbf{Ecuación 2.2}$$

La función de salida J de la red neuronal equivale a (ecuación 2.3):

$$\boldsymbol{J = F_i(neta_i)} \qquad \textbf{Ecuación 2.3}$$

Donde F_i representa la función de activación para esa unidad, que corresponde a la función escogida para transformar la entrada $neta_i$ en el valor de salida *J.* La tabla 2.3 muestra las principales funciones de transferencia empleadas en las redes neuronales.

Nombre	Relación Entrada /Salida	Icono
Limitador Fuerte	$a = 0 \; n < 0$ $a = 1 \; n \geq 0$	
Limitador Fuerte Simétrico	$a = -1 \; n < 0$ $a = +1 \; n \geq 0$	
Lineal Positiva	$a = 0 \; n < 0$ $a = n \; 0 \leq n$	
Lineal	$a = n$	
Lineal Saturado	$a = 0 \; n < 0$ $a = n \; 0 \leq n \leq 1$ $a = 1 \; n > 1$	
Lineal Saturado Simétrico	$a = -1 \; n < -1$ $a = n \; -1 \leq n \leq 1$ $a = +1 \; n > 1$	
Sigmoidal Logarítmico	$a = \frac{1}{1+e^{-n}}$	
Tangente Sigmoidal Hiperbólica	$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$	
Competitiva	$a = 1$ *Neurona con n max* $a = 0$ *El resto de neuronas*	

Tabla 2.3 Funciones de Transferencia

En general, las redes neuronales se pueden clasificar en [84], [87]:

1. *Según su arquitectura*:

a. *Red Unidireccional:* consiste en capas de neuronas donde la salida de una neurona alimenta todas las neuronas de la capa siguiente, hasta llegar a la salida.

b. *Redes Recurrentes*: son aquellas que poseen conexiones de realimentación entre las capas de neuronas.

c. *Monocapa*: todas las neuronas se conectan entre sí.

2. *Valores de entrada***:** en las redes binarias las entradas tienen un valor de 0 o 1. Las redes continuas tienen como entrada valores numéricos enteros o continuos.

3. *Forma de aprendizaje*:

a. *Aprendizaje Supervisado*: se utiliza un agente externo que indica a la red la respuesta deseada para el patrón de entrada.

b. *Refuerzo*: se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. La función del supervisor indica mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1), y en función de ello se ajustan los pesos.

c. *Aprendizaje No Supervisado (auto-organización)*: no existe un agente externo indicando la respuesta deseada para los patrones de entrada.

4. *Algoritmo de Aprendizaje*: según la forma de aprendizaje, estos algoritmos realizan un proceso interactivo de ajustes a los pesos de conexión entre dos neuronas. Algunos de ellos son los siguientes:

a. *Aprendizaje por Corrección de Error*: algoritmo basado en la Regla Delta que busca minimizar la función de error usando un gradiente descendente. Este es el principio usado por el algoritmo retropropagación.

b. *Aprendizaje Competitivo*: dos neuronas de una capa compiten por el privilegio de permanecer activos, por lo tanto, una neurona activa será la única que participará del proceso de aprendizaje. Es usado en mapas de Kohonen y en redes ART.

c. Aprendizaje Hebbiano: dos neuronas al estar simultáneamente activas o desactivas las conexiones entre ellas se deben fortalecer, caso contrario serán debilitadas. Este aprendizaje es utilizado en el Modelo de Hopfield.

A continuación presentamos el modelo neuronal usado durante el desarrollo de este trabajo.

2.4.2.1. RED NEURONAL DE RETROPROPAGACIÓN

En la Red Neuronal de Retropropagación las salidas de las neuronas en una capa pueden estar interconectadas a las entradas de las neuronas de la misma capa o a entrada de neuronas en capas precedentes (ver figura 2.13). Este hecho le proporciona al arreglo neuronal características de procesamiento dinámico en el sentido de que las salidas de la red dependen no solo de sus entradas en un instante dado, sino también de sus entradas y salidas en instantes anteriores.

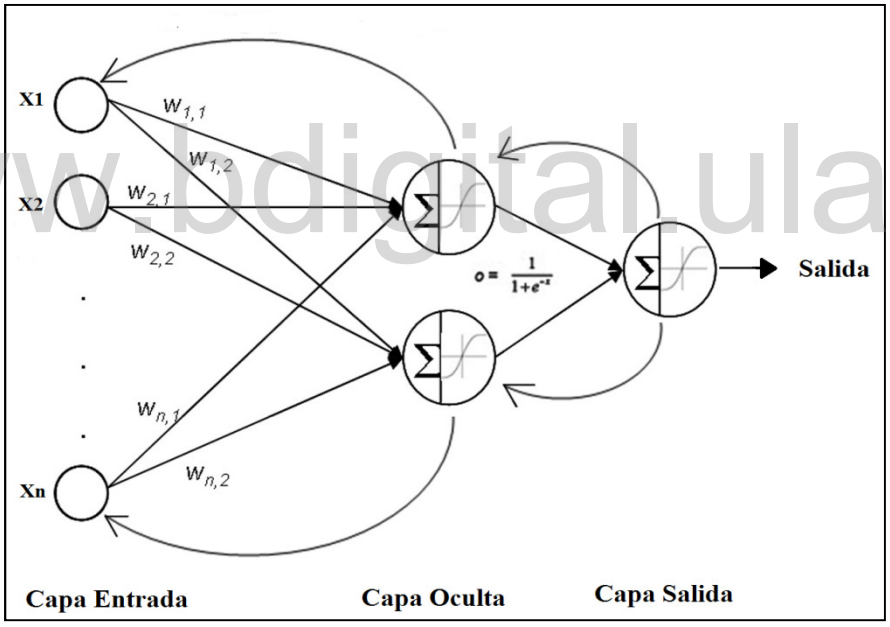


Figura 2.13 Diagrama de una Red Neuronal de Retropropagación.

Este tipo de red neuronal utiliza el aprendizaje supervisado. 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 figura 2.14). Para realizar este proceso se debe inicialmente tener definida la topología de la red: número de neuronas de la capa de entrada (depende del número de componentes del vector de entrada), cantidad de capas ocultas y

número de neuronas de cada una de ellas, número de neuronas en la capa de salida [87]. Las dos fases son descritas a continuación.

1. Fase de aprendizaje "hacia adelante": se aplica un patrón de entrada como estimulo para la primera capa de neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida.

2. Fase de aprendizaje "hacia atrás": los errores obtenidos en la fase anterior se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error. Luego se procede al reajuste de los pesos de las neuronas. Este proceso se repite por un número de iteraciones, o hasta que el error sea el deseado por el usuario.

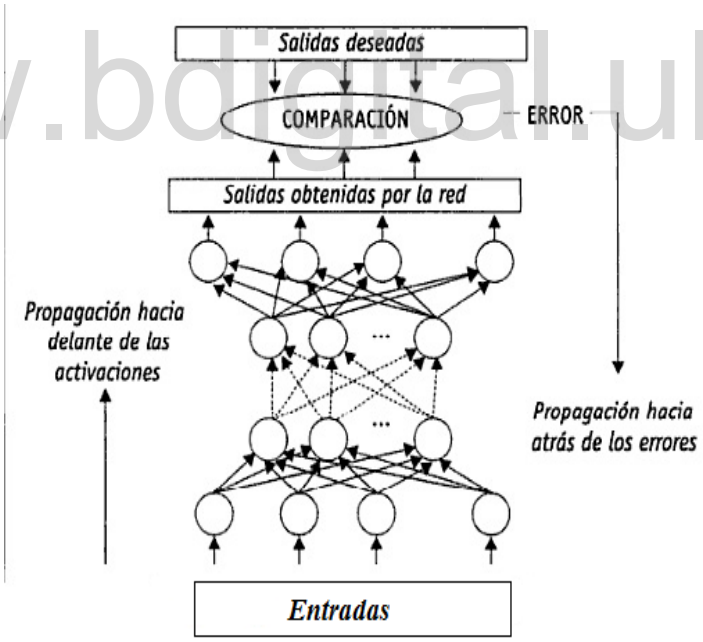


Figura 2.14 Red neuronal de retropropagación

El algoritmo que se emplea para entrenar una red neuronal de retropropagación es la *Regla Delta Generalizada*. Este algoritmo utiliza una función de error asociada a la red, buscando el mínimo error a través del gradiente descendiente. Los pasos del algoritmo son:

Paso 1: Inicializar los pesos de la red con valores aleatorios.

Paso 2: Presentar un patrón de entrada $x_p = (x_{p1}, x_{p2,\ldots,} x_{pn})$ con n componentes, a la capa de entrada de la red y especificar la salida deseada d_p que debe generar ésta.

Paso 3: Calcular el valor de entrada a cada una de las neuronas en las capas ocultas (ecuación 2.4)

$$\boldsymbol{Net_{pj}^{h} = \sum_{i=1}^{n} w_{ji}^{h} * x_{pi} + \theta_{j}^{h}} \qquad \textbf{Ecuación 2.4}$$

Donde w_{ji}^{h} es el peso en la conexión de la unidad de entrada i a la unidad de la capa oculta j, θ_j^h es el valor del bias, y el índice h representa a la capa oculta (hidden).

Paso 4: Calcular la salida de la capa oculta (ecuación 2.5)

$$\boldsymbol{y_{pj} = f_j^h(Net_{pj}^h)} \qquad \textbf{Ecuación 2.5}$$

Paso 5: En la capa de salida, calcular el valor de entrada a cada neurona (ecuación 2.6)

$$\boldsymbol{Net_{pk}^{o} = \sum_{j=1}^{L} w_{kj}^{o} * I_{pj} + \theta_{k}^{o}} \qquad \textbf{Ecuación 2.6}$$

Donde L es el número de neuronas de la capa oculta, y el índice o representa a la capa de salida (output)

Paso 6: Calcular la salida (ecuación 2.7)

$$\boldsymbol{y_{pk} = f_k^o(Net_{pk}^o)} \qquad \textbf{Ecuación 2.7}$$

Paso 7: Calcular el error para las neuronas de la capa de salida (ecuación 2.8).

$$\boldsymbol{\delta_{pk}^{o} = (y_{pk} - d_{pk}) * f_k^{o\prime}(Net_{pk}^{o})} \qquad \textbf{Ecuación 2.8}$$

La función f debe ser derivable. Las funciones de salida más utilizadas son (ver ecuación 2.9 y 2.11):

$$\textbf{La función lineal: } \boldsymbol{f_k^o(Net_{jk}^o) = Net_{jk}^o} \quad \textbf{donde} \quad \boldsymbol{f_k^{o\prime}(Net_{pk}^o) = 1} \quad \textbf{Ecuación 2.9}$$

El error para las neuronas de salida para la función lineal es (ecuación 2.10):

$$\delta_{pk}^{o} = (y_{pk} - d_{pk}) * f_{k}^{o'}(Net_{pk}^{o}) = (d_{pk} - y_{pk}) \qquad \textbf{Ecuación 2.10}$$

La función sigmoidea: $\frac{1}{(1+e^{-Net_{jk}^{o}})}$ **donde** $f_{k}^{o'}(Net_{pk}^{o}) = f_{k}^{o}(1 - Net_{pk}^{o}) = y_{pk}(1 - y_{pk})$

Ecuación 2.11

El error para las neuronas de salida para la función sigmoidea es (ecuación 2.12):

$$\delta_{pk}^{o} = (y_{pk} - d_{pk}) * f_{k}^{o'}(Net_{pk}^{o}) = (d_{pk} - y_{pk}) y_{pk}(1 - y_{pk}) \qquad \textbf{Ecuación 2.12}$$

Paso 8: Calcular el error en las neuronas de la capa oculta (ecuación 2.13):

$$\delta_{pj}^{h} = f_{j}^{h'}(Net_{pj}^{h}) \sum \delta_{pk}^{o} * w_{kj}^{o} \qquad \textbf{Ecuación 2.13}$$

El error en la capa oculta depende de todos los términos del error de la capa de salida. De aquí surge el término retropropagación o propagación hacia atrás.

Paso 9: Actualizar los pesos en capa de salida (ecuación 2.14):

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \alpha \delta_{pk}^{o} * y_{pj} \qquad \textbf{Ecuación 2.14}$$

Donde α representa la tasa de aprendizaje, el cual es un valor en el intervalo [0, 1] que permite aumentar la velocidad de convergencia del error. A mayor tasa de aprendizaje, mayor es la modificación de los pesos en cada iteración, pero puede dar lugar a oscilaciones en este valor. Por lo tanto, se puede agrega un término que tiende a mantener los cambios de los pesos en una misma dirección, llamado momento β (ecuación 2.15).

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \alpha \delta_{pk}^{o} * y_{pj} + \beta(w_{kj}^{o}(t) - w_{kj}^{o}(t-1)) \qquad \textbf{Ecuación 2.15}$$

Donde β es un valor en el intervalo [0, 1].

Paso 10: Actualizar los pesos en la capa oculta (ecuación 2.16)

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \alpha \delta_{pj}^{h} * x_{p} \qquad \textbf{Ecuación 2.16}$$

Utilizando el término momento (ecuación 2.17)

$$w_{ji}^{h}(t+1) = w_{ij}^{h}(t) + \alpha \delta_{pj}^{h} * x_{p} + \beta(w_{ij}^{h}(t) - w_{ji}^{h}(t-1)) \qquad \textbf{Ecuación 2.17}$$

Paso 11: El proceso se repite hasta que el error para el patrón de entrada sea mínimo o se cumpla un número máximo de iteraciones (ecuación 2.18).

$$E_p = \frac{1}{2}\sum_{k=1}^{M} \delta_{pk}^2 \qquad \textbf{Ecuación 2.18}$$

Donde M es el número de neuronas de la capa de salida.

Una vez concluida la fase de aprendizaje se inicia el modo de operación. La red debe generar una salida próxima a una de las aprendidas, ante la presencia de un nuevo patrón de entrada desconocido [84], [87].

2.4.3. ALGORITMO DE OPTIMIZACIÓN DE COLONIAS DE HORMIGAS

Los Algoritmos basados en Colonias de Hormigas (ACO) son un tipo de metaheurística basada en población, cuya filosofía está inspirada en el comportamiento de las hormigas reales cuando buscan comida (ver figura 2.15). La idea principal consiste en usar hormigas artificiales que simulan dicho comportamiento en un escenario también artificial: un grafo. En los últimos años ACO ha demostrado su efectividad en la resolución de diferentes problemas de optimización combinatoria considerados difíciles [84], [88], [89], [90].

El modo de operación de un algoritmo de optimización de colonias de hormigas es como sigue: las *m* hormigas (artificiales) de la colonia se mueven, concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema (que son los nodos del grafo). Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes (nodos). Esta información local incluye la información heurística y memorística (rastros de feromona) para guiar la búsqueda. Al moverse por el grafo, las hormigas construyen incrementalmente soluciones. Además, las hormigas depositan feromona en los arcos (conexiones) que componen la solución que descubren (*actualización de los rastros de feromona*). Esa actualización puede ser en línea (en función de la calidad de la solución parcial construida) o una vez que cada hormiga ha generado una solución (depositan una cantidad de feromona que es función de la calidad de la solución obtenida). Además, se debe incluir la evaporación de los rastros de feromona en el entorno; se usa como un mecanismo que evita el estancamiento en la búsqueda, y permite que las hormigas busquen y exploren nuevas regiones

del espacio [84], [88], [89], [90]. Esta información guiará la búsqueda de las otras hormigas de la colonia en el futuro.

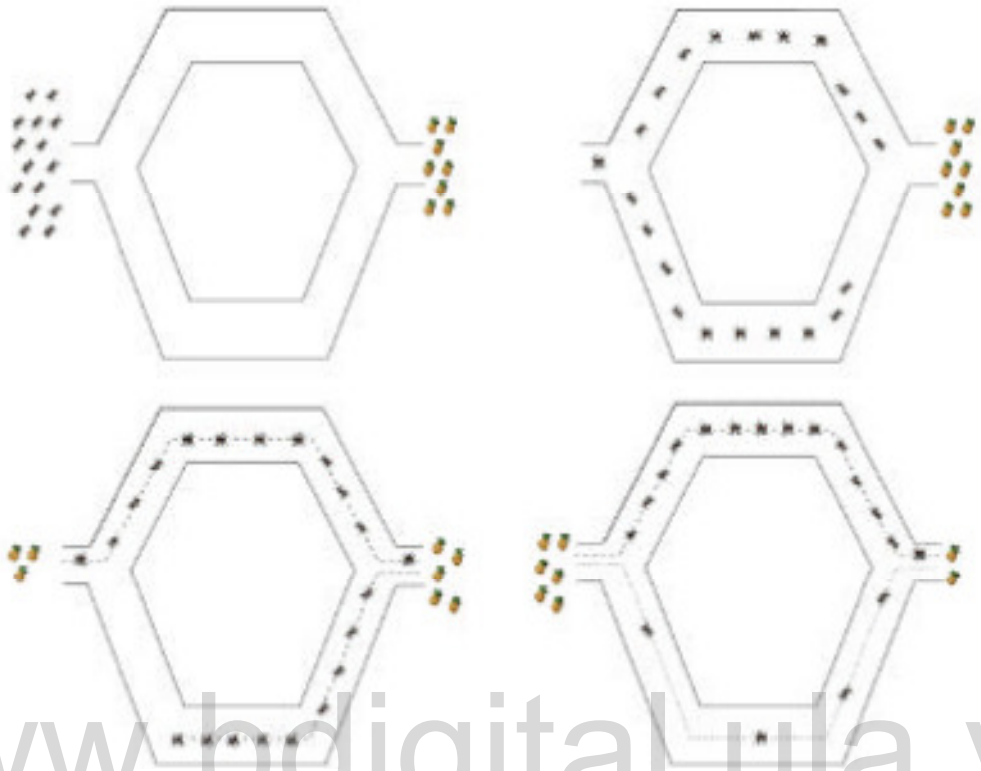


Figura 2.15 Comportamiento de la colonia de hormigas para obtener el camino más corto entre dos puntos.

Los pasos a seguir para resolver un problema utilizando el algoritmo de Optimización de Colonias de Hormigas es [84], [88], [89], [90]:

1. Representar el problema como un conjunto de componentes y transiciones, a través de un grafo ponderado que será recorrido por las hormigas para construir soluciones.

2. En cada paso, una hormiga *k* escoge ir al siguiente nodo r con una probabilidad que se calcula como (ecuación 2.19):

$$p_r^k = \begin{cases} \frac{[\tau_r]^\theta [\eta_r]^\beta}{\sum_{u \in N_i^k} [\tau_r]^\theta [\eta_r]^\beta} & \text{Si s} \in N_k(i) \\ 0 & \text{en otro caso} \end{cases} \qquad \textbf{Ecuación 2.19}$$

Donde τ_r es la cantidad de rastro de feromona en el nodo r del grafo, y η_r es la visibilidad del nodo r, que frecuentemente es $\frac{1}{d_{ir}}$, donde d_{ir} es la distancia que existe entre el

nodo actual i y el nodo r donde se quiere ir; $N_k(i)$ es el vecindario alcanzable por la hormiga *k* cuando se encuentra en el nodo *i*, θ y $\beta \in R$ son dos parámetros que ponderan la importancia relativa de los rastros de feromona y la visibilidad. Cada hormiga *k* almacena la secuencia que ha seguido hasta el momento.

Volviendo a los parámetros *θ y β*, si θ=*0* aquellos nodos con una preferencia heurística (visibilidad) mejor tienen una mayor probabilidad de ser escogidos, haciendo el algoritmo muy similar a un algoritmo probabilístico clásico. Si *β=0*, sólo se tienen en cuenta los rastros de feromona para guiar el proceso constructivo, lo que puede causar un rápido estancamiento, esto es, una situación en la que los rastros de feromona asociados a una solución son ligeramente superiores que el resto, provocando por tanto que las hormigas siempre construyan las mismas soluciones, normalmente óptimos locales. Por tanto, es preciso establecer una adecuada proporción entre ambos valores.

3. Actualización de feromona. Ese proceso tiende a asignar una gran cantidad de feromona a los caminos más cortos. Este esquema es similar al aprendizaje reforzado, en el cual, mientras mejor es una solución más se refuerzan los elementos que caracterizan dicha solución. El feromona depositado en los nodos tiene el rol de una memoria distribuida de largo plazo. Esta memoria no es guardada localmente en cada individuo, sino distribuida a través de los nodos del grafo. Esto permite una forma directa de comunicación. Clásicamente, la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, según la ecuación 2.20:

$$\boldsymbol{\tau_r(t+1) = \alpha\tau_r(t) + \Delta\tau_r} \qquad \textbf{Ecuación 2.20}$$

Donde: $\tau_r(t)$ = intensidad de la traza depositada en el nodo r en el momento t, α = coeficiente tal que (1 – α) representa la tasa de evaporación de la feromona entre el intervalo de tiempo t y t+1 y $\Delta\tau_r$ viene dado por la ecuación 2.21.

$$\boldsymbol{\Delta\tau_r = \sum_{k=1}^{m} \Delta\tau_r^k} \qquad \textbf{Ecuación 2.21}$$

Donde: $\Delta\tau_r^k$= cantidad de feromona dejada en un nodo r por la k^{esima} hormiga entre el intervalo t y t+1, la cual es calculada utilizando la ecuación 2.22 (en la literatura existen otras formas matemáticas para la ecuación 2.22, ver [84], [88], [89], [90]):

$$\Delta\tau_r^k = \begin{cases} \frac{Q}{L_k} \; \textbf{\textit{Si la hormiga k pasa por el nodo r en su recorrido (entre t y t + 1}} \\ \mathbf{0} \;\; \textbf{\textit{En otro caso}} \end{cases}$$

Ecuación 2.22

Donde: Q es una constante, L_k = la longitud del recorrido hecho por la hormiga k

El coeficiente α debe ser menor que uno (1) para evitar acumular cantidades de feromona ilimitadamente. Por otro lado, la intensidad de la feromona en el momento t = 0, $\tau_r(0)$, puede ser escogida aleatoriamente.

CAPÍTULO III: DISEÑO DEL SISTEMA

En este capítulo se define el Diseño del Sistema, se presenta su estructura global así como cada uno de sus componentes: el Sub-Sistema de Comparación de Motivos y el Sub-Sistema de Fusión de Motivos. Además, se realiza la definición formal del esquema de comparación y fusión. Para más detalles sobre la implantación del sistema ver Apéndice B.

3.1. PRESENTACION DEL DISEÑO

El problema de comparar y fusionar motivos de proteínas representados como expresiones regulares denotadas según las reglas PROSITE, implica la resolución de dos sub-problemas principales:

1. Clasificar y asignar un valor de similitud entre motivos de proteínas representados como expresiones regulares denotadas según las reglas PROSITE.

2. Construir un patrón común que caracterice a los motivos con alto grado de similitud.

Para ello es necesario:

1. Diseñar un algoritmo para comparar dos expresiones regulares denotadas por las reglas PROSITE. Además, diseñar un método para asignar un valor de similitud entre éstas.

2. Construir una expresión regular que represente un patrón común denotado según las reglas PROSITE para las expresiones regulares que tengan un algo grado de similaridad.

Por lo tanto, nuestro Sistema consta de dos sub-sistemas: un Sub-Sistema de Comparación y un Sub-Sistema de Fusión de Motivos de proteínas representados como expresiones regulares denotadas según las reglas PROSITE (ver figura 3.1).

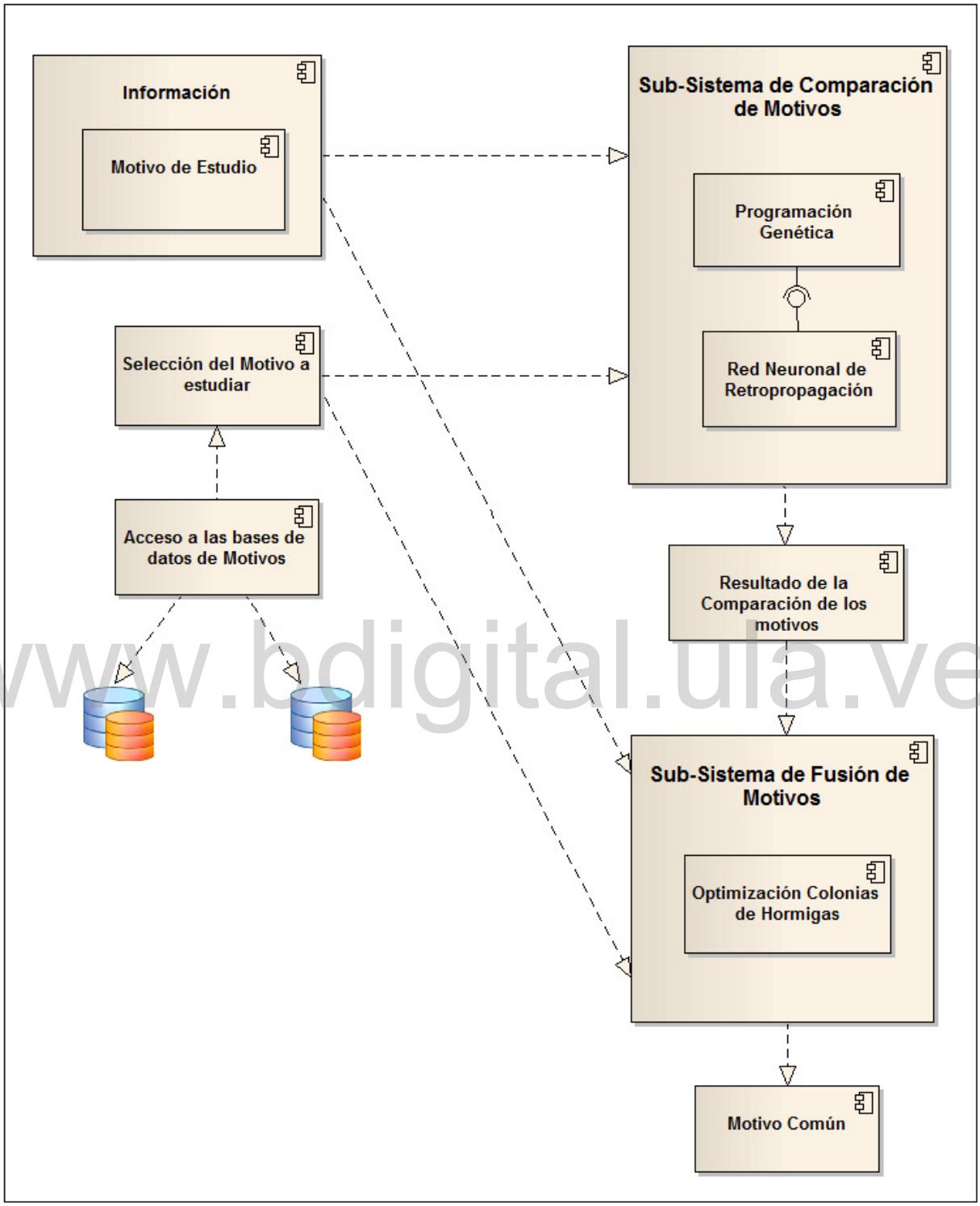


Figura 3.1 Diagrama de Componentes del Sistema Propuesto.

3.2. SUB-SISTEMA DE COMPARACIÓN DE MOTIVOS

3.2.1. DEFINICIÓN DEL MACRO ALGORITMO GENERAL

Se diseño un algoritmo que permite comparar un motivo bajo estudio con motivos almacenados en una base de datos de proteínas (AMYPdb [23], [24]) denotados como expresiones regulares según las reglas PROSITE. La figura 3.2 muestra la estructura general del Sub-Sistema de Comparación de motivos.

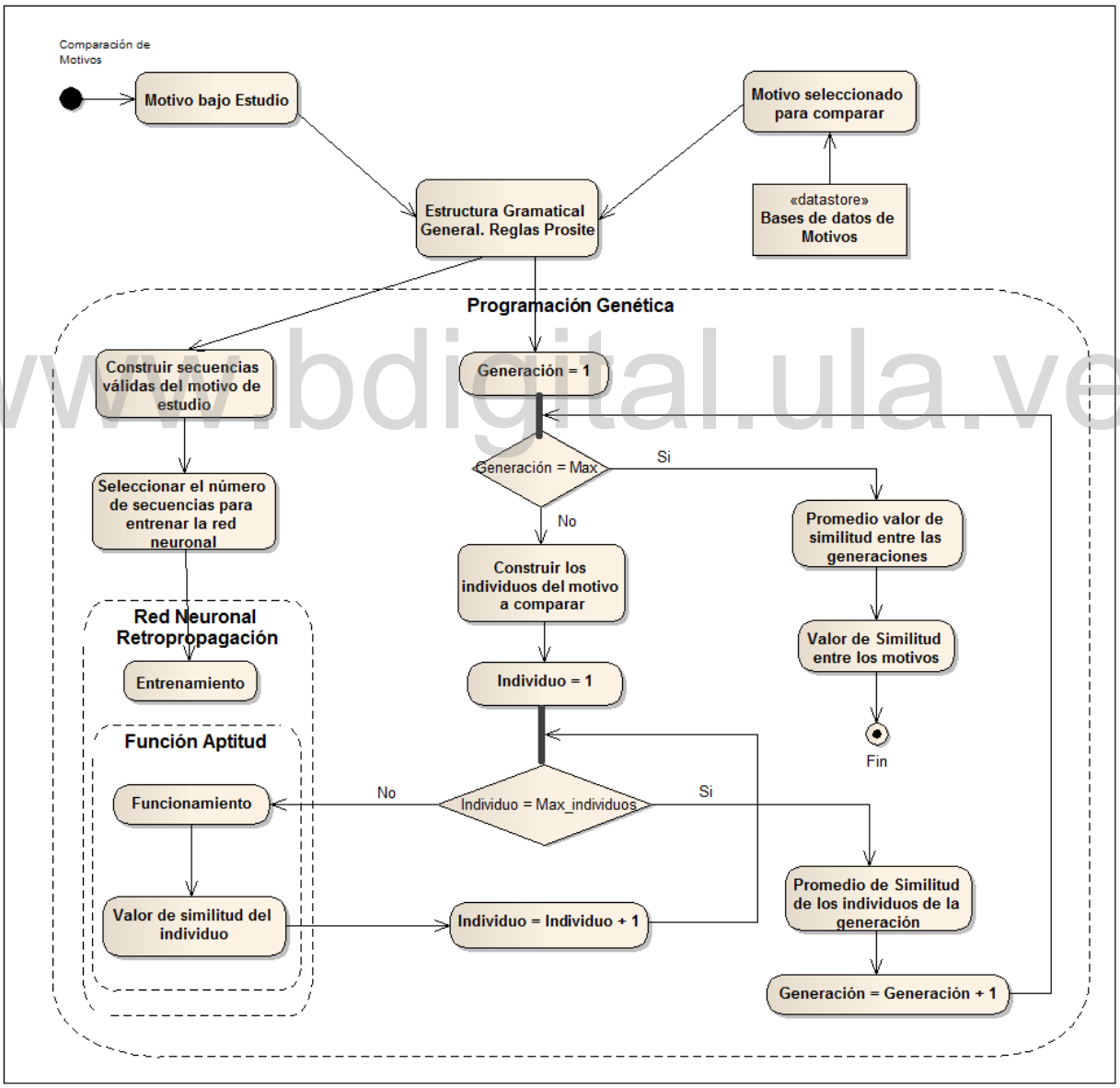


Figura 3.2 Estructura general del sub-sistema de comparación de motivos.

De manera general el algoritmo trabaja según los siguientes pasos:

1. Se tiene un motivo de proteínas (expresión regular 1) bajo estudio.

2. Se construye un conjunto de secuencias validas para el motivo de estudio.

3. Del total de secuencias del paso anterior se selecciona una muestra que es introducida en una Red Neuronal de Retropropagación para su aprendizaje (entrenamiento de la Red Neuronal de Retropropagación).

4. Se extrae un motivo de proteínas de una base de datos (expresión regular 2).

5. Se construye aleatoriamente un conjunto de secuencias validas para el motivo extraído de la base de datos que representan los individuos de la población (población total de secuencias).

6. Cada uno de los individuos de la población total de secuencias (secuencias del motivo extraído de la base de datos) es presentado a la Red Neuronal de Retropropagación, para compararlo con los aprendidos anteriormente. La Red Neuronal de Retropropagación obtiene un valor que representa el error entre ésta y las aprendidas en el paso 3 (esta fase representa el cálculo de la función aptitud).

7. Con el valor del error se calcula la similitud para cada uno de los individuos del motivo extraído de la base de datos.

8. Los pasos 6 y 7 son repetidos para todos los individuos que conforman la población. Luego, se calcula un valor promedio de similitud para esta población que representa el valor de similitud de esa generación entre los motivos.

9. Los pasos 5, 6, 7 y 8 son repetidos para cada una de las generaciones, hasta completar un número máximo de generaciones dadas por el usuario.

10. Al final, con los valores de similitud de cada una de las generaciones se calcula un valor promedio de similitud general que representa la similitud entre los motivos.

El sub-sistema de comparación de motivos consta de tres componentes:

1. Una Estructura Gramatical General que permite construir secuencias de los motivos denotados como expresiones regulares según las reglas PROSITE.

2. La Programación Genética, que usa la estructura gramatical para construir un conjunto de secuencias validas a partir de las expresiones regulares a estudiar, y compara dichas secuencias para asignarles un valor de similitud.

3. Una Red Neuronal de Retropropagación, que actúa como la función de aptitud para evaluar las secuencias de las expresiones regulares generadas por la Programación Genética.

En la siguiente sección detallamos cada una de estos componentes.

3.2.2. ESPECIFICACIÓN DE LOS COMPONENTES DEL MACRO ALGORITMO GENERAL

3.2.2.1. ESTRUCTURA GRAMATICAL GENERAL

Es una plantilla que utiliza la Programación Genética para establecer las normas o reglas que permiten construir las posibles secuencias validas de los motivos a comparar denotados como expresiones regulares. En nuestro caso vamos a utilizar las reglas PROSITE (ver sección 1.5). Estas reglas describen las operaciones que se admiten sobre las secuencias, especificando que atributos pueden aparecer en la construcción de éstas. Este es un punto crítico del sistema, porque si la gramática no es definida apropiadamente, las secuencias generadas por la Programación Genética no corresponderían a los motivos utilizados. La plantilla se presenta a continuación:

amino = A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y
digito = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
aminos = $amino^{+}$
gap = x

paréntesis = (amino(digito) | amino(digito, digito) | gap(digito) | gap(digito, digito))$^+$

corchete = [amino,(amino)*]$^+$

llaves = {amino, (amino)*}$^+$

secuencia = (aminos | gap | paréntesis | corchete | llaves)$^+$

Para la definición de las reglas utilizadas ver sección 2.3

3.2.2.2. PROGRAMACION GENETICA

A continuación se especifican los componentes de la Programación Genética utilizados en el sub-sistema de comparación de motivos [91], [92]:

1. *Los individuos:* representan cada una de las secuencias validas que se pueden formar desde los motivos de proteínas denotados como expresiones regulares, usando la gramática general establecida utilizando las reglas PROSITE. Por ejemplo, para el motivo de proteínas: K-[KM]-[AD]-A se pueden formar las siguientes secuencias:
   - KKAA
   - KKDA
   - KMAA
   - KMDA

Cada secuencia es un individuo del motivo K-[KM]-[AD]-A. Los individuos son estructuras arborescentes (ver figura 3.3) formados por nodos terminales y funciones. En nuestro caso en específico serán:

a. Conjunto de terminales: son átomos que representan las constantes. En nuestro caso, existen 3 tipos: amino, digito y gap (ver plantilla de la sección 3.2.2.1)

b. Conjunto de funciones: todas aquellas establecidas en la plantilla de la sección anterior para concebir secuencias de motivos (paréntesis, corchetes, etc.).

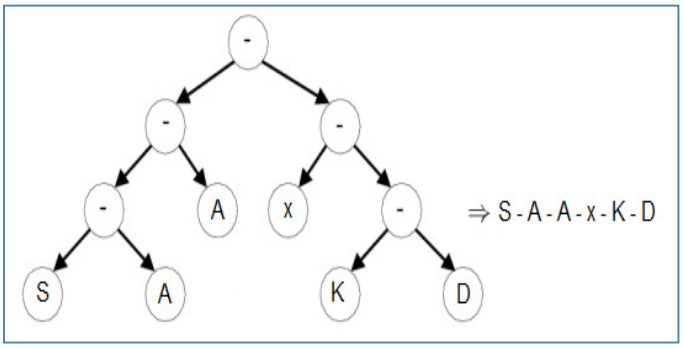


Figura 3.3 Estructura de un individuo

2. *Tamaño de la población:* representa el número de secuencias que se desean en cada generación del motivo de proteínas extraído de la base de datos (expresión regular 2). Su rango va desde 1 hasta el número máximo de secuencias que se pueden generar para ese motivo.

3. *Número de generaciones:* Es el máximo número de iteraciones que se alcanzan durante la ejecución su rango puede estar entre 1 y 10000000. 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 bajo estudio con el motivo extraído de la base de datos, pero puede ocurrir que después de un cierto número de generaciones el valor de similitud entre las secuencias de los motivos comparados no mejore.

4. *Función de Aptitud:* se debe asignar un valor cuando comparamos las secuencias del motivo bajo estudio (expresión regular 1) con las secuencias del motivo extraído de la base de datos (expresión regular 2). Este valor es determinado por la Red Neuronal de Retropropagación. Para esto, la red neuronal es entrenada previamente con las secuencias del motivo bajo estudio (expresión regular 1). Luego es usada para evaluar, tal que se le introducen las secuencias del motivo extraído de la base de datos y ella determina un valor de reconocimiento entre ambos motivos.

5. *Operadores Genéticos:* en nuestro Sistema se utiliza el reemplazo como método para sustituir los individuos de la población de una generación a otra. El reemplazo consiste en

construir nuevos individuos de la población y reemplazar los individuos actuales por éstos en cada generación.

6. *Criterio para terminar:* un número máximo de generaciones es establecido para comparar las secuencias del motivo bajo estudio con las secuencias del motivo de proteínas extraído de la base de datos; al cumplirse esta, el valor de reconocimiento promedio obtenido representa la similitud entre ambos motivos.

7. *Resultado total:* Al final, la Programación Genética muestra un valor de similitud entre el motivo bajo estudio (expresión regular 1) y el motivo extraído de la base de datos (expresión regular 2). A nivel biológico, este valor representa que tan parecidos son los motivos que se están comparando. De esta manera, si el valor de similitud es alto o cercano a 100 % significa que existe coincidencia entre los aminoácidos y las posición que ocupan en ambos motivos comparados/estudiados, esto puede representar que son motivos homólogos o tienen un ancestro en común, motivos no relacionados que tienen puntos en común, o motivos no homólogos con estructuras análogas. Si el valor de similitud se acerca a cero la similitud es baja, es decir, no existe relación entre ambos motivos.

3.2.2.3. RED NEURONAL DE RETROPROPAGACION

La Red Neuronal de Retropropagación será utilizada por la Programación Genética como función de aptitud [91], [92]. A continuación se detalla su funcionamiento:

1. *Número de secuencias para entrenar la Red Neuronal:* dado el gran número de secuencias que tiene el motivo bajo estudio, para la etapa de entrenamiento de la Red Neuronal de Retropropagación es imposible utilizarlos todos. Por lo tanto, es necesario calcular una muestra que represente una porción del total de las posibles secuencias que pueden ser generadas a partir del motivo bajo estudio. Dicha muestra será la población de secuencias del motivo bajo estudio con la se entrenará la red neuronal ((expresión regular 1), ver ecuación 3.1).

$$n' = \frac{s^2}{\sigma^2} \qquad \textbf{Ecuación 3.1}$$

Donde: n' = tamaño de la muestra, σ^2 = varianza de la población, S^2 = varianza de la muestra de la población, la cual puede ser determinada utilizando la ecuación 3.2:

$$S^2 = P(1 - P) \qquad \textbf{Ecuación 3.2}$$

Donde: P = Fiabilidad deseada[8], (valor en el intervalo [0, 1]).

Como la varianza de la población es desconocida, se utiliza el error estándar cuadrado[9] $(se)^2$, de esta manera $\sigma^2=(se)^2$. El valor del error estándar se encuentra en el intervalo [0, 1] y puede calcular utilizando la ecuación 3.3:

$$se = \frac{\mu}{\sqrt{N}} \qquad \textbf{Ecuación 3.3}$$

Donde: μ = desviación estándar, N = número total de secuencias que pueden ser generadas usando el motivo bajo estudio.

El tamaño final de la población de secuencias que se utilizará en el entrenamiento de la Red Neuronal de Retropropagación está dado por la ecuación 3.4:

$$n = \frac{n'}{1+\frac{n'}{N}} \qquad \textbf{Ecuación 3.4}$$

Donde: n = tamaño final de la población de secuencias usadas en la fase de entrenamiento.

2. *Diseño y Entrenamiento de la Red Neuronal:* Los individuos generados a partir del motivo bajo estudio son utilizados para entrenar la Red Neuronal Retropropagación (el número de individuos a generar es determinado según el paso anterior), que consta de 3 capas (entrada – oculta – salida). El número de neuronas de la capa oculta se ha escogido mediante un proceso de ensayo y error (demasiadas neuronas ocasionaría un sobre-ajuste y la red perdería la capacidad de generalizar. Por otra parte si el número de neuronas en la capa oculta es reducido produciría un sub-ajuste, en la cual la red no sería capaz de aprender correctamente las secuencias del motivo objeto de estudio). En todo caso, nuestro objetivo será dotar a la red de un número adecuado de neuronas en la capa oculta capaz de aprender los aminoácidos que conforman las secuencias del motivo objeto de

[8] Representa la calidad de la muestra (entre más cercana a uno significaría que se desea una muestra más fiable)

[9] Proporciona una medida de la precisión de la estimación de la media poblacional a partir de una muestra.

estudio (ese valor es determinado por la ecuación 3.5, 3.6 y 3.7). Así, la arquitectura de la Red Neuronal de Retropropagación utilizada es (ver figura 3.4):

$$\boldsymbol{Numero\ de\ neuronas\ de\ entrada = tamaño\ de\ la\ secuencia} \qquad \textbf{Ecuación 3.5}$$

$$\boldsymbol{Numero\ de\ neuronas\ de\ la\ capa\ oculta = \frac{Nro\ neuronas\ de\ entrada}{2} + 1} \qquad \textbf{Ecuación 3.6}$$

$$\boldsymbol{Número\ de\ neuronas\ de\ salida = Número\ de\ neuronas\ de\ entrada} \qquad \textbf{Ecuación 3.7}$$

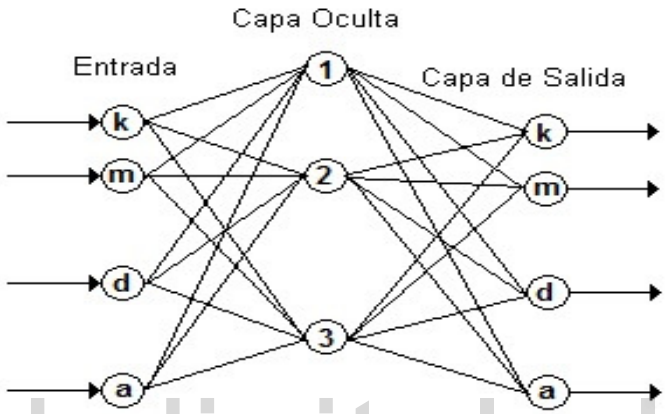


Figura 3.4 Ejemplo de entrenamiento de la red neuronal auto-asociativa

Ahora bien, como las secuencias de los motivos están formadas por caracteres que representan los aminoácidos, 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[10]. De esta forma, tenemos la tabla 3.1 que determina los valores de entrada a la red neuronal.

[10] 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.

Aminoácido	Símbolo	Nro.	Escalado
Familia: Aminoácidos Alifáticos			
Glicina	Gly, G	0	0/80
Alanina	Ala, A	1	1/80
Valina	Val, V	2	2/80
Leucina	Leu, L	3	3/80
Isoleucina	Ile, I	4	4/80
Metionina	Met, M	5	5/80
Familia: Aminoácidos Neutros			
Serina	Ser, S	20	20/80
Treonina	The, T	21	21/80
Asparagina	Asn, N	22	22/80
Glutamina	Gln, Q	23	23/80
Familia: Aminoácidos Básicos			
Histidina	His, H	30	30/80
Lisina	Lys, K	31	31/80
Arginina	Arg, R	32	32/80
Familia: Aminoácidos Ácidos:			
Ácido Aspártico	Asp, D	40	40/80
Ácido Glutámico	Glu, E	41	41/80
Familia: Aminoácidos Aromáticos			
Fenilalanina	Phe, F	50	50/80
Tirosina	Tys, Y	51	51/80
Triptófano	Trp, W	52	52/80
Familia: Aminoácidos con Azufre			
Cisteína	Cys, C	60	60/80
Familia: Iminoácido			
Prolina	Pro, P	70	70/80
Gaps			
Gap	X	80	80/80

Tabla 3.1 Conversor de entrada a la red de los valores de los aminoácidos

Como se dijo antes, para entrenar la Red Neuronal de Retropropagación se calcula el tamaño final de la población de secuencias a entrenar del motivo bajo estudio utilizando las ecuaciones 3.1 y 3.4; luego se itera hasta que el error de la red sea menor que el error dado por el usuario, o cuando se alcanza el máximo número de épocas[11] definido para ésta.

3. *Funcionamiento de la Red Neuronal:* después que la Red Neuronal de Retropropagación fue entrenada se procede a presentar a ésta las secuencias del motivo extraído de la base de datos (expresión regular 2), creadas por la Programación Genética, para conocer el error de reconocimiento, en cada una de las generaciones. De esta manera, la red neuronal realiza la comparación de cada aminoácido de las secuencias del motivo extraído de la base de datos con el motivo aprendido en la fase de entrenamiento. Para esto calcula para cada aminoácido (neurona) un error (ver ecuación 3.8), que es el valor absoluto de la diferencia entre el valor obtenido de la red neuronal y el valor que se presenta en la secuencia que se desea reconocer (tasa de reconocimiento).

$$\boldsymbol{error_i = |valorneurona_i - valordeseado_i|} \qquad \textbf{Ecuación 3.8}$$

Donde: $error_i$ es igual al error del aminoácido en la posición i de la secuencia; $valorneurona_i$es igual al valor de la salida i de la red neuronal, después de entrenada la red neuronal; $valordeseado_i$ es igual al valor del aminoácido de entrada en la posición i.

De la ecuación 3.8 se pueden obtener varios valores del error. Así:

a. Si $error_i \leq \frac{1}{80}$, significa que el aminoácido en la posición i en las secuencias es el mismo (secuencia presentada y secuencias aprendidas por la Red Neuronal de Retropropagación). Entonces:

$$\boldsymbol{simil_i = indice\ de\ similitud\ de\ amonoácidos\ iguales} \qquad \textbf{Ecuación 3.9}$$

[11] Cada una de las iteraciones que realiza la red neuronal de retropropagación para disminuir el error en la fase de entrenamiento.

Donde: $simil_i$ es igual al valor de similitud para el aminoácido i de la secuencia. Cuyo valor esta en el intervalo [0, 10]

b. Si $\frac{1}{80} < error_i \leq \frac{5}{80}$, significa que el aminoácido en la posición i es diferente en las secuencias comparadas (secuencia presentada y secuencias aprendidas por la Red Neuronal de Retropropagación), pero pertenecen a la misma familia. Entonces:

$$\boldsymbol{simil_i = índice\ de\ similitud\ de\ aminoácidos\ de\ la\ misma\ familia}$$

Ecuación 3.10

c. Si $error_i > \frac{5}{80}$ significa que el aminoácido en la posición i es diferente en las secuencias comparadas (secuencia presentada y secuencias aprendidas por la Red Neuronal de Retropropagación. Entonces:

$$\boldsymbol{simil_i = índice\ de\ similitud\ de\ aminoacidos\ diferentes} \qquad \textbf{Ecuación 3.11}$$

Para cada secuencia que representa un individuo de la población del motivo extraído de la base de datos, se calcula la similitud, de la siguiente manera.

$$\boldsymbol{similind_j = \sum_{i=1}^{b} \frac{simil_i}{b}} \qquad \textbf{Ecuación 3.12}$$

Donde: $similind_j$ es igual a la similitud para cada individuo j de la población del motivo; *b* es el número de aminoácidos que componen el individuo.

Después, para cada generación del motivo extraído de la base de datos, se calcula la similitud.

$$\boldsymbol{similgen_p = \sum_{j=1}^{k} \frac{similindj}{k}} \qquad \textbf{Ecuación 3.13}$$

Donde: $similgen_p$ es igual a la similitud para cada generación p del motivo; *k* es igual al número de individuos que componen cada generación.

Al final, se calcula la similitud total, que representa el resultado dado por la Programación Genética.

$$\boldsymbol{similtotal = \sum_{p=1}^{m} \frac{similgen_p}{m}} \qquad \textbf{Ecuación 3.14}$$

Donde: *similtotal* es igual a la similitud total; *m* es igual al número de generaciones

3.3. SUB-SISTEMA DE FUSION DE MOTIVOS

3.3.1. DEFINICIÓN DEL MACRO ALGORITMO GENERAL

Para realizar la fusión de motivos de proteínas denotados como expresiones regulares según las reglas PROSITE se ha tomado el algoritmo clásico de Sistemas de Hormigas como base [84], [88], [89], [90], al cual se la han realizado modificaciones para adaptarlo a nuestro caso. De esta manera, para cada ejecución del algoritmo se toman dos expresiones regulares y dará como resultado una nueva expresión regular que representa un patrón común a las anteriores (ver figura 3.5).

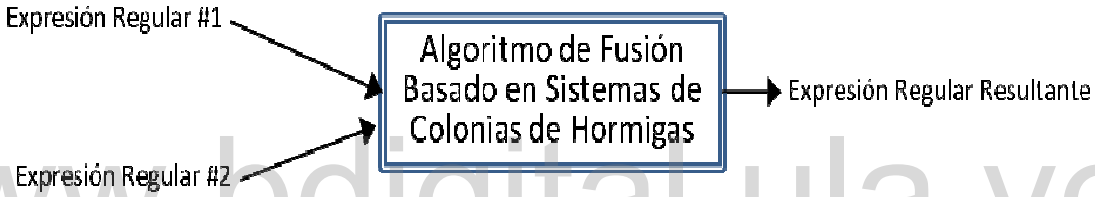


Figura 3.5 Fusión de dos expresiones regulares.

En aquellos casos donde se desee fusionar N expresiones regulares, se debe ejecutar el algoritmo N-1 veces, teniendo siempre en cuenta que una vez finalizada cada iteración (ejecución), la expresión regular resultante será una de dos expresiones regulares de entrada del algoritmo en su próxima ejecución (ver figura 3.6).

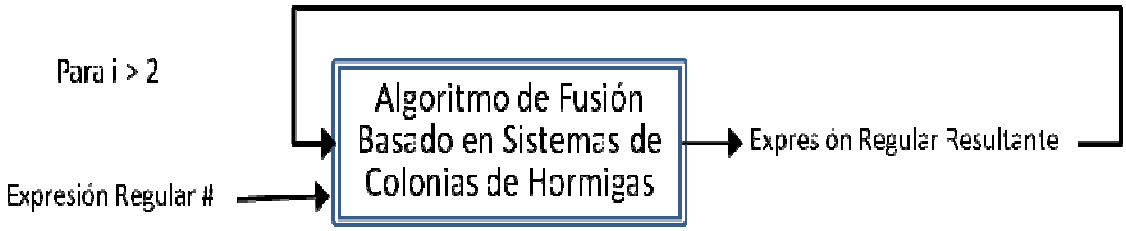


Figura 3.6 Fusión de N expresiones regulares.

De manera general, el macro algoritmo para el proceso de la fusión de patrones proteicos

propuesto en este trabajo, es el siguiente (ver figura 3.7) [92], [93], [94], [95]:

1. Creación del grafo de recorrido.

2. Recorrido de la colonia de hormigas.

3. Selección de los mejores nodos.

4. Construcción de la Expresión Regular Resultante.

En la siguiente sección detallamos cada una de esas fases.

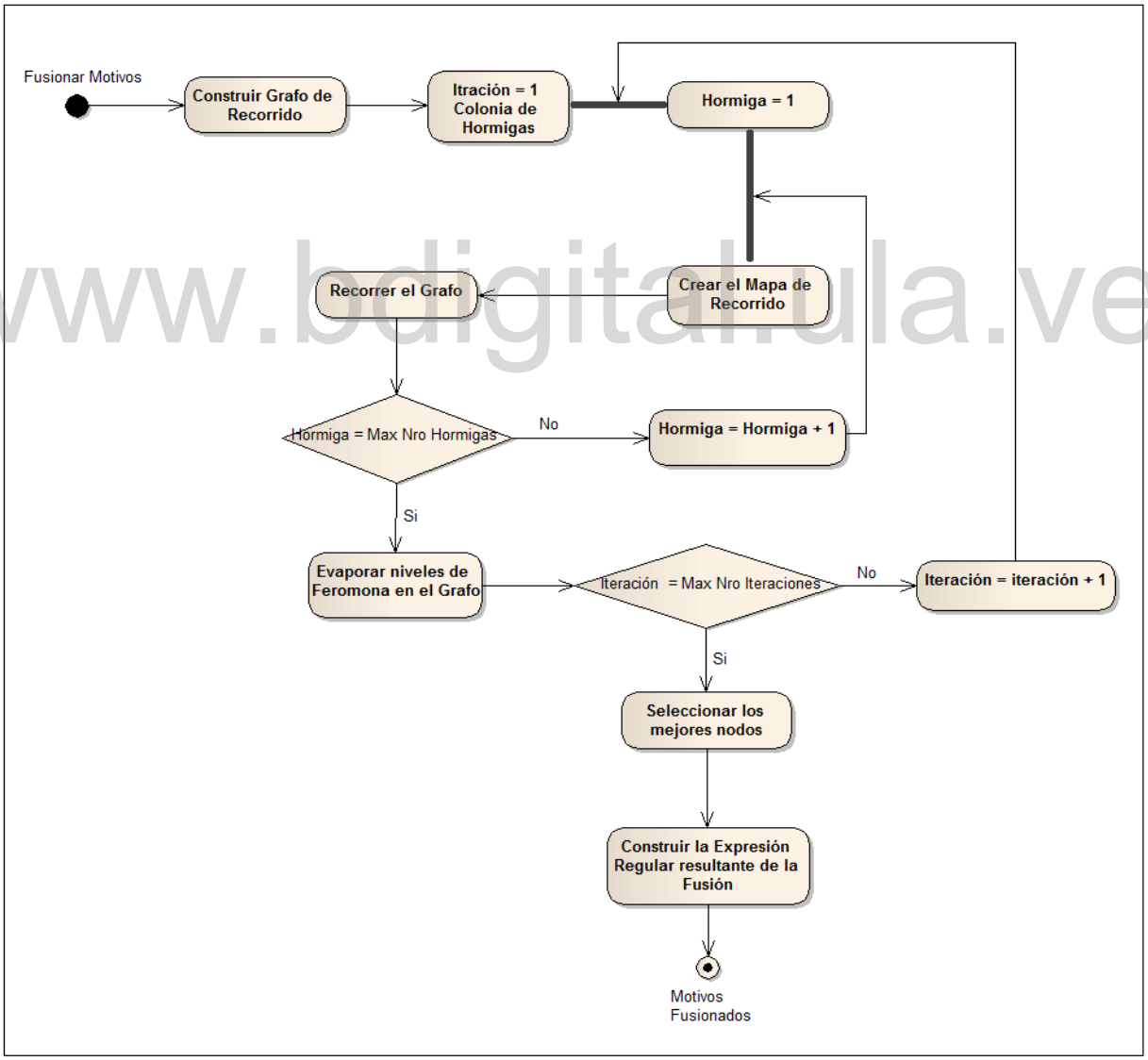


Figura 3.7 Diagrama general del Sub-sistema de Fusión de Motivos.

3.3.2. ESPECIFICACION DE LAS ETAPAS DEL MACRO ALGORITMO GENERAL

3.3.2.1. CREACION DEL GRAFO DE RECORRIDO

Existen diferentes formas para representar los motivos, entre las cuales se encuentran matrices, listas, arboles y grafos, en nuestro caso vamos a utilizar estos últimos. La razón de esta decisión es que este tipo de estructura de datos utiliza poca memoria, es escalable para las diferentes expresiones regulares a fusionar, y adicionalmente, es fácil realizar el recorrido entre sus nodos[12]. Particularmente, nosotros usaremos un tipo de grafo llamado dígrafo, el cual tienen los nodos conectados en dos sentidos.

Así, debido a que el problema de la fusión de motivos nace a partir del estudio de la estructura primaria de las proteínas, la cual es una estructura lineal conformada por aminoácidos, se establecen dos condiciones básicas para el diseño del grafo:

1. En la construcción de motivos, lo esencial es la posición de los distintos aminoácidos a lo largo de las cadenas proteicas, las cuales pueden ser vistas como un arreglo unidimensional.

2. El producto de la fusión de motivos deberá generar un nuevo patrón que caracterice a las expresiones regulares fusionadas. La estructura de datos debe ser capaz de construir estas cadenas lineales con el orden deseado, por lo tanto, en el grafo solo se permitirá que cada nodo tenga arcos conectados a sus vecinos de la derecha e izquierda, permitiendo que el recorrido solo sea en dirección horizontal.

Teniendo clara las necesidades para la construcción del grafo se procede a definir el diseño de los arcos y nodos:

1. Los arcos utilizados en el grafo de recorrido cumplirán la función de interconectar dos nodos, por lo que ellos solo tendrán la información referente a las direcciones de los

[12] También conocido como vértices

nodos que se encuentran interconectados por él (los arcos que lo interconectan tanto por la derecha como por la izquierda).

2. Los nodos adicionalmente almacenan los niveles de feromona depositados por las hormigas que transitaran a través de ellos, y la información biológica del aminoácido al que representan. Esta información estará constituida por el tipo de aminoácido en notación de 1 letra (ver tabla 2.1) y un clasificador para determinar la familia a la que este pertenece (ver tabla 3.1); ver la figura 3.8 para todos los detalles de información en un nodo.

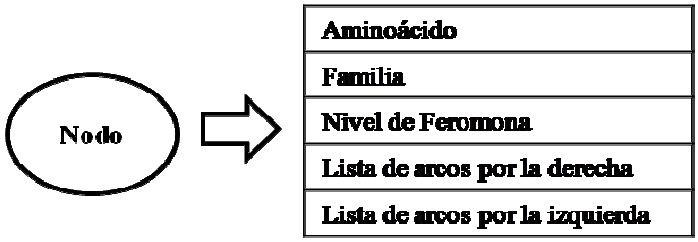


Figura 3.8 Estructura de datos de los nodos.

La estructura utilizada para los nodos permite reconocer los gaps y vacios[13] usados en los motivos denotados por las reglas PROSITE, además del inicio y fin del recorrido, para lo cual se utilizaran identificadores especiales para todos estos casos (ver tabla 3.2), diferente a los que se usaron como clasificadores para sus familias.

Información a representar	Identificador especial	Clasificador para las nuevas Familias
Gap	X	0
Vacio	_	-1
Inicio	Ini	-2
Fin	Fin	-2

Tabla 3.2 Identificadores y familias para los nodos especiales

Para construir el grafo de recorrido se escoge la primera expresión regular que se desea fusionar (es el grafo que será transitado por las hormigas). Para ello, se transforma la

[13] Nodo que no contiene ni aminoácidos, ni gap.

expresión regular (ER1) a un TDA[14] LIFO[15], debido a que esta estructura de datos facilita la construcción del grafo. Por ejemplo, sea la expresión regular "S-A(1,3)-x-[KV]", la figura 3.9 muestra la transformación de esa expresión regular a una TDA LIFO o TDA Pila.

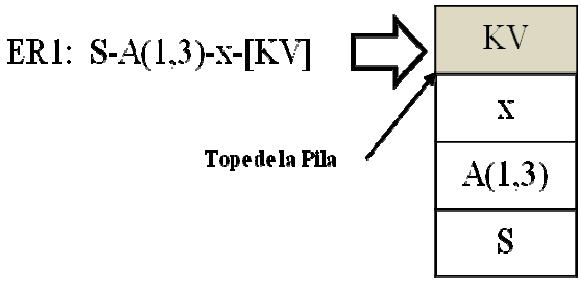


Figura 3.9 Transformación de la expresion regular ER1 a un TDA Pila.

Una vez ingresado el contenido de ER1 a un TDA Pila, se construyen dos nodos (ver figura3.10), que sirven de guía para la construcción del grafo y para indicar a las hormigas donde deben iniciar y finalizar su recorrido.

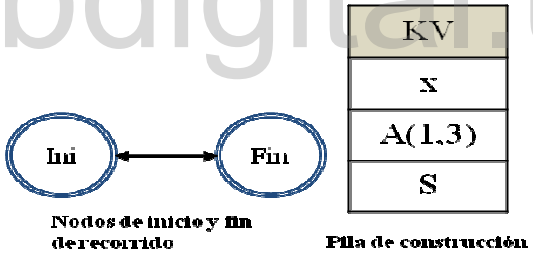


Figura 3.10 Creación de los nodos que irán a los extremos del grafo

Luego, se procede a extraer los elementos que se encuentran al tope de la pila iterativamente: cada elemento constituye una posición de la expresión regular y definen los nodos del grafo (ver figura3.11). Además, se agrega un nodo gap, que sirve como una ruta auxiliar para los casos en los cuales una hormiga no se dirija a ninguno de los nodos con aminoácidos disponibles, ayudando de esta manera a evitar la detención del recorrido de ésta.

[14] TDA (Tipo de Dato Abstracto) es una estructura de datos con una colección de operaciones definidas sobre ella [108].

[15] "Last In, First Out", también conocido como un TDA Pila.

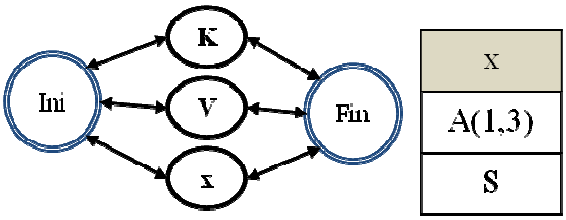


Figura 3.11 Construcción del grafo utilizando el primer elemento del tope de la pila.

Cuando el tope de la pila indique que se ingresará un nodo gap al grafo, no se necesita la construcción de ningún nodo adicional (ver figura 3.12).

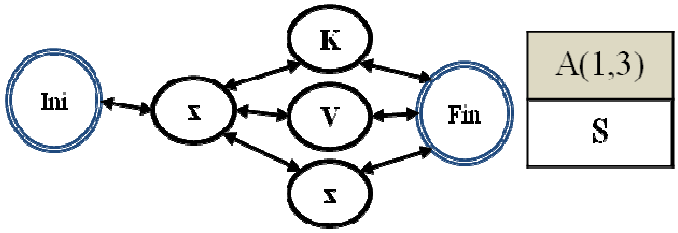


Figura 3.12 Tratamiento al momento de hallarse un gap en el tope de la pila.

Veamos el caso cuando se debe ingresar al grafo un elemento que contiene 2 valores dentro de un paréntesis (por ejemplo A(1,3) indica que en esa posición pueden haber desde una hasta tres Alaninas, ver figura 3.13). En este caso se observa que no se permite la existencia de otros aminoácidos que no sea el indicado en esas posiciones, haciendo inservible el uso de un nodo gap para evitar la detención de los agentes hormigas. Por ello se utiliza un nodo especial (nodo vacio ("_")), que ayuda a evitar el estancamiento del recorrido de la hormiga por el grafo. Además, los arcos que conducen a estos nodos deben cumplir ciertas condiciones, ya que cuando una hormiga decida dirigirse a un nodo vacio continuara su recorrido por nodos de este tipo hasta que no encuentre otro nodo vacio, porque haría el recorrido solo por una de las opciones que representa este elemento. Por lo tanto, se deben ingresar tantos elementos vacios como posiciones existan entre el menor y mayor numero del paréntesis (en nuestro caso, en A(1,3) existen 2 posiciones entre 1 y 3).

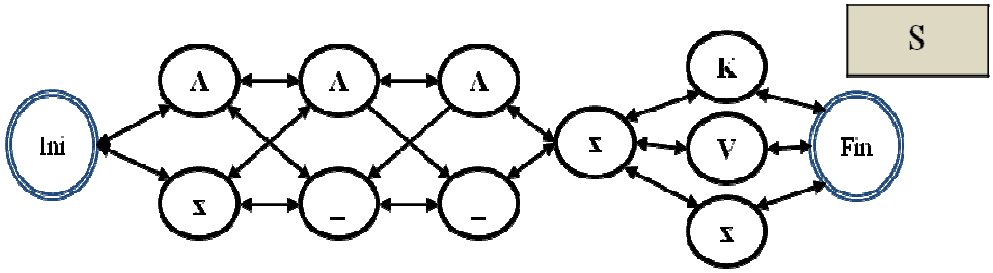


Figura 3.13 Inserción de un elemento que contiene 2 valores dentro de un paréntesis en el grafo.

En la figura3.14, se observan otros dos casos, en primer lugar, cuando en el tope de la pila se encuentra un solo aminoácido este siempre será acompañado por un nodo gap en la misma posición, esto para efectos de evitar detenciones en las hormigas; en segundo lugar, el punto de parada para la construcción del grafo será cuando se halla vaciado la pila.

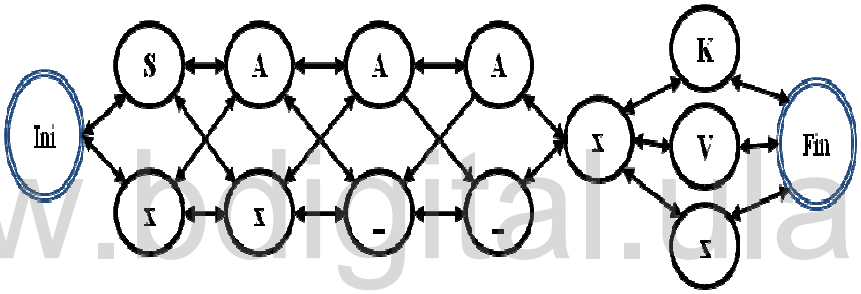


Figura 3.14 Culminación de la construcción del grafo de Recorrido.

A continuación se muestra el macro-algoritmo de generación del grafo de recorrido.

```
Función ConstruyeGrafo(nodoI, nodoF, eR) devuelve el grafo de recorrido
    Entradas:    nodoI, una variable de tipo Nodo.
                 nodoF, una variable de tipo Nodo.
                 eR cadena que contiene la expresión regular de un patrón proteico.
    Variables:   pilaConst, variable del tipo Pila
                 amin, variable del tipo carácter, almacenara aminoácidos de la PilaConst

    pilaConst = ConviertePila(eR)
    Inicializa(nodoI, 'Ini', 1.0)
    Inicializa(nodoF, 'Fin', 1.0)
    CreaArcos(nodoI, nodoF)
    Repita Mientras pilaConst no este vacia:
        amin = Tope(pilaConst)
        En Caso de:
            1° amin = ']' Entonces: CrearSegmentoCorchete(amin, pilaConst, nodoI, nodoF)
            2° amin =')' Entonces: CrearSegmentoParentesis(amin, pilaConst, nodoI, nodoF)
            3° amin ¡= Gap Entonces: CreaSegmentoAmino((amin, pilaConst, nodoI, nodoF)
            Cualquier otro: CreaSegmentoGap( pilaConst, nodoI, nodoF)
```


Algoritmo 3.1. Creación del Grafo de Recorrido.

3.3.2.2. RECORRIDO DE LA COLONIA DE HORMIGAS

La colonia de hormigas artificiales mantiene el mismo comportamiento de las colonias de hormigas naturales, donde la evolución de su actuación sobre el entorno en el que se encuentra no es sino el producto de las acciones que realizan cada una de sus integrantes (ver figura3.15). En esta fase, algunas consideraciones se deben hacer:

1. La simulación de la colonia se realiza como resultado de N-iteraciones de la clase Agente Hormigas, por lo tanto, se debe definir el número de individuos que integran la colonia, antes de que se inicie el recorrido sobre el grafo.

2. Como el número de habitantes de la colonia es invariable se impide el nacimiento de

nuevas hormigas dentro de ésta, por lo tanto, cada generación se simula repitiendo el recorrido de las hormigas tantas veces como generaciones se deseen, donde cada nueva generación toma como punto de partida el trabajo realizado por la anterior.

Figura 3.15 Constitución de una Colonia de Hormigas en la naturaleza.

El TDA agente Hormiga está compuesto de 9 elementos: nodo inicial, mapa de recorrido, coeficiente de incremento de feromona, los diferentes índices de similitud entre el aminoácido buscado por la hormiga y el representado por cada (para casos igualdades, para familia, para aminoácidos diferentes, para los gaps, etc.), además, la similitud aprobatoria y el número máximo de fracasos. Todos son descritos en la tabla3.3; y representa toda la información necesaria para que las hormigas puedan hacer su recorrido en el grafo.

Campo	Utilidad
Nodo Inicial	Tendrá la dirección del nodo de inicio en el grafo de recorrido para cada agente Hormiga.
Mapa de Recorrido	Es una pila que contiene la segunda expresión regular a fusionar, y le sirve a los agentes Hormiga para conocer que nodos deben visitar.
Coeficiente de Incremento de Feromona	Número Real perteneciente al conjunto [0,1] que representa la concentración de feromona que depositan los agentes hormiga en cada nodo visitado del grafo.
Índice de similitud para igualdades	Número entero perteneciente al conjunto [0,10], que influirá en el nivel de feromona depositado por el agente Hormiga. Este es el caso cuando el aminoácido del nodo visitado en su recorrido es idéntico al esperado según el mapa de recorrido.
Índice de similitud para las familias	Número entero perteneciente al conjunto [0,10] que influirá en el nivel de feromona depositado por el agente Hormiga. Este es el caso cuando si el aminoácido del nodo visitado en su recorrido pertenece a la misma familia del aminoácido indicado en el mapa de recorrido.
Índice de similitud para las diferencias	Número entero perteneciente al conjunto [0,10], que influirá en el nivel de feromona depositado por el agente Hormiga. Este es el caso cuando el aminoácido del nodo visitado en su recorrido no es igual, ni corresponde a la familia del aminoácido indicado en el mapa de recorrido.
Índice de similitud para los gaps	Número entero perteneciente al conjunto [0,10], que influirá en el nivel de feromona depositado por el agente Hormiga, si el nodo visitado en su recorrido contiene un gap y en el mapa de recorrido espera un gap.
Similitud aprobatoria	Es un número entero perteneciente al conjunto [0,10] que le indica al agente Hormiga el nivel de similitud mínimo para considerar que el nodo visitado es un hallazgo exitoso, y continuará la búsqueda de otro aminoácido distinto en el mapa de recorrido, en caso contrario continuara buscando en el grafo de recorrido el mismo aminoácido.
Número máximo de fracasos	Es un número entero mayor a -1, representa el mayor número de hallazgos no exitosos que realiza el agente Hormiga antes realizar la búsqueda de un nuevo aminoácido del mapa de recorrido en el grafo. Permite que el agente Hormiga no permanezca indefinidamente buscando nodos con un índice de similitud menor a la similitud aprobatoria, ya que al agente Hormiga igualar el número de hallazgos fallidos con el máximo número de fracasos, dudara sobre la factibilidad de encontrar un nodo que supere la similitud aprobatoria, y con una probabilidad de 0,5 decidirá si accede al mapa de recorrido para buscar un nuevo aminoácido o continua buscando el mismo aminoácido.

Tabla 3.3 Campos del TDA agente Hormiga.

La expresión regular (ER1) será fusionada con la expresión regular (ER2) "L(2)-A(2)-Q", con ésta se construye un nuevo TDA Pila que representa el *mapa de recorrido* que utilizará el agente Hormiga para transitar a través el grafo (ver figura 3.16).

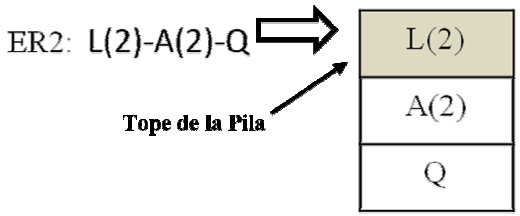


Figura 3.16 Construcción de la Pila utilizando a ER2 como Mapa de Recorrido.

Antes de que el agente Hormiga pueda recorrer el grafo se necesitan ajustar unos parámetros que están contenidos en su estructura de datos (ver tabla 3.4), los cuales le permitirán identificar si un nodo visitado pertenece a una solución exitosa (*índices de similitud esperado en un nodo en el caso de que los aminoácidos sean iguales, de la misma familia, etc.*), establecer la cantidad de feromona que depositarán en cada nodo (*coeficiente de incremento de feromona*), y si un aminoácido no es encontrado en un nodo seguir la búsqueda en los nodos vecinos (*similitud aprobatoria, máximo número de fracasos*).

Parámetro	Valor
Nodo Inicial	Nodo Ini
Coeficiente de incremento de feromona	0,10
Índice de similitud para igualdades	10
Índice de similitud para las familias	8
Índice de similitud para las diferencias	1
Índice de similitud para los gaps	3
Similitud aprobatoria	2
Máximo número de fracasos	1

Tabla 3.4 Ejemplo de valores para la inicialización de los parámetros de un agente Hormiga.

Cuando ya se hayan asignado todos los valores para los campos del TDA agente Hormiga, éste se posiciona en el nodo inicial del grafo e inicia el recorrido usando su mapa de recorrido y observando los nodos contiguos a su derecha (ver figura 3.17).

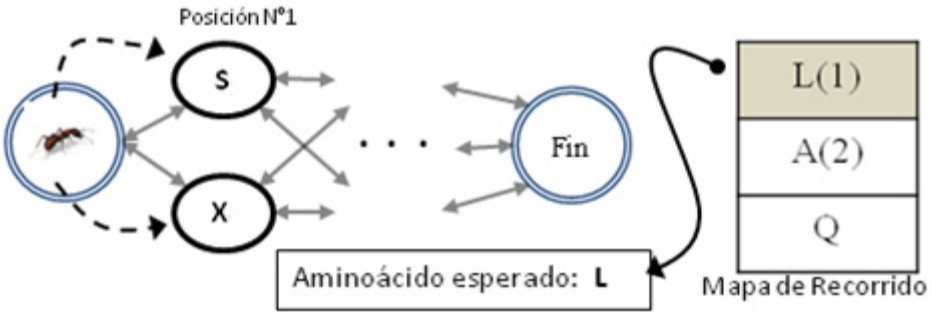


Figura 3.17 El agente Hormiga observa los nodos de la siguiente posición del grafo.

Luego, el agente Hormiga ejecuta la función de transición desde el nodo actual para cada uno de los nodos que puede visitar en la próxima posición. Esta función consta de dos etapas:

1. Se calcula la probabilidad de visitar a cada uno de los nodos contiguos ($P_n^k(r)$) en función de su nivel de feromona "τ_r" e índice de similitud "φ_r" de cada nodo ("r" indica el nodo vecino de la posición "k", y "n" es el número de nodos vecinos a la derecha para esa posición 'k' (ver ecuación 3.15).

$$P_n^k(r) = \begin{cases} \frac{\tau_r * \varphi_r}{\sum_{i=1}^{n} \tau_i * \varphi_i} & si\ n > 1 \\ 1 & si\ n = 1 \end{cases} \qquad \textbf{Ecuación 3.15}$$

2. Se toma la decisión del nodo a visitar utilizando una simulación de Monte Carlo. Para ello se sitúan en una ruleta todas las probabilidades calculadas en el paso anterior, y se escogerá un número aleatorio entre 0 y 1 (simulando el movimiento de la bola en la ruleta). El valor obtenido será escalado por el perímetro de la ruleta (ver figura3.18) y se dirige al agente Hormiga al nodo que pertenezca ese espacio.

Continuando con la ejecución del recorrido de nuestro ejemplo del agente Hormiga, supongamos que se obtienen los siguientes valores de probabilidad para su desplazamiento cuando se está en la posición 1 (ver figura 3.18):

- Probabilidad de Visitar el nodo S: 25%
- Probabilidad de visitar al nodo Gap: 75%

Figura 3.18 Ruleta construida partiendo de las probabilidades de visitar a los nodos de la posición N°1, obtenidas según la ecuación 3.15.

Trasladando los valores de las probabilidades a la ruleta, el espacio (0;0,25) está asociado a desplazarse al nodo S, mientras que [0,25; 1) representa la opción de desplazarse al nodo gap. Se genera un número aleatorio, en nuestro caso suponemos que es "0,16", y según este valor se hace que el agente Hormiga se dirija al nodo S y deposite allí su rastro de feromona.

En específico, la actualización del feromona será equivalente a la suma de la concentración actual de feromona en el nodo más el producto del coeficiente de incremento por su índice de similitud con respecto al aminoácido esperado según el mapa (ver ecuación 3.16).

$$\tau_r = \tau_r + \sigma * \varphi_r \qquad \textbf{Ecuación 3.16}$$

El índice de similitud se establece de la siguiente manera: el agente Hormiga extrae un aminoácido del tope de la pila del mapa de recorrido y observa el aminoácido del nodo a visitar en el grafo. Si los aminoácidos son iguales se utiliza el índice de similitud para las igualdades, si los aminoácidos pertenecen a la misma familia se utiliza el índice de similitud para las familias, si el nodo visitado contiene un gap, entonces se usa el índice para gaps, de lo contrario, se usa el índice de similitud para las diferencias (ver tabla 3.4).

En el ejemplo para el agente Hormiga (ver figura3.17), se observa que el aminoácido S no es igual al aminoácido esperado (L), de hecho, no pertenece a la misma familia, por lo tanto, el índice de similitud retornara el valor indicado para las diferencias. El valor obtenido es "1", el índice similitud aprobatoria es "2" (ver tabla 3.4). De esta manera, se considera que el movimiento no ha sido exitoso. Sin embargo, como es apenas la primera equivocación que

comete la hormiga, y según la variable que almacena el máximo número de fracasos para el ejemplo en curso una hormiga puede equivocarse una vez antes de dudar si continua buscando el aminoácido extraído del tope del mapa (ver campo máximo número de fracasos tabla 3.4), la hormiga no extrae otro elemento del mapa de recorrido y continua buscando el mismo aminoácido en la siguiente posición (ver figura 3.19). Esta vez el agente excede el máximo de equivocaciones consecutivas permitidas, por lo que se decide con probabilidad 0,5[16] si se continúa buscando el mismo aminoácido o se procede a extraer otro elemento del mapa de recorrido. Esta acción le da al agente la posibilidad de evitar búsquedas frustradas, o por otro lado, explorar nuevas soluciones. Cabe destacar que como el número de errores cometidos por el agente no se actualiza hasta hallar el aminoácido o extraer uno nuevo del mapa de recorrido, será cada vez menos probable que continúe buscando el mismo aminoácido sin tener éxito.

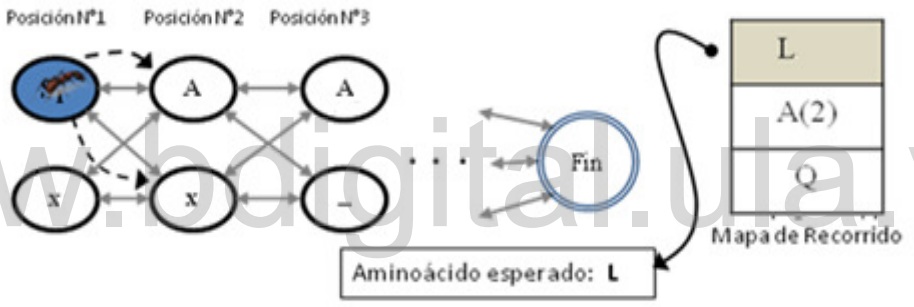
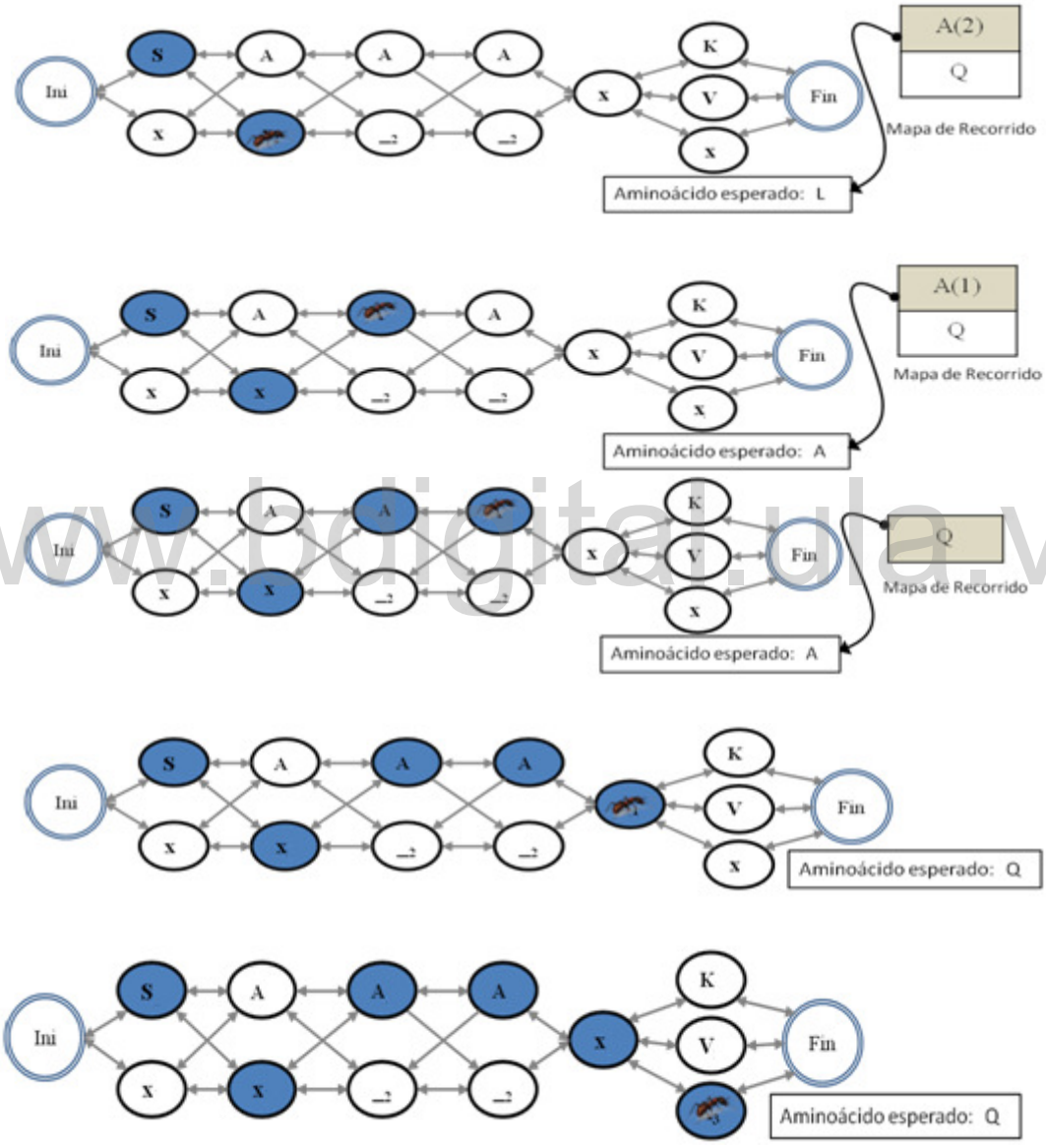


Figura 3.19 Desplazamiento del agente Hormiga a la siguiente posición.

A continuación se muestra el desplazamiento del agente Hormiga a través de las posiciones del grafo de recorrido (ver figura 3.20). La hormiga se encontraba en el nodo S (primera posición del grafo), por el procedimiento anterior se desplaza al nodo X en la siguiente posición. Luego se extrae un elemento del mapa de recorrido (en este caso A), la hormiga realiza la función de transición, y como son aminoácidos iguales el que se encuentra en el mapa y en el grafo de recorrido se dirige a él. En la siguiente posición ocurre lo mismo que lo anterior. A continuación se extrae el último elemento del mapa "Q", como solo existe un nodo X en esa posición se dirige la hormiga a ese nodo. Como los aminoácidos no son iguales y no pertenecen a la misma familia, se sigue buscando "Q" en el mapa, ya que el

[16] Probabilidad uniforme para el número de nodos en esa posición del grafo. En este caso como existen dos nodos S y x la probabilidad para cada uno es 0,5.

número máximo de fracasos es "2". Con la ecuación 3.15 se calcula la probabilidad de dirigirse a los nodos de la siguiente posición. Utilizando la simulación de Monte Carlo se escoge un número aleatorio (supongamos 0,91) y se selecciona el nodo que corresponde a ese valor en la ruleta (en este caso contiene un gap). Finalmente la hormiga se dirige a esa posición, terminando su recorrido en el nodo fin.



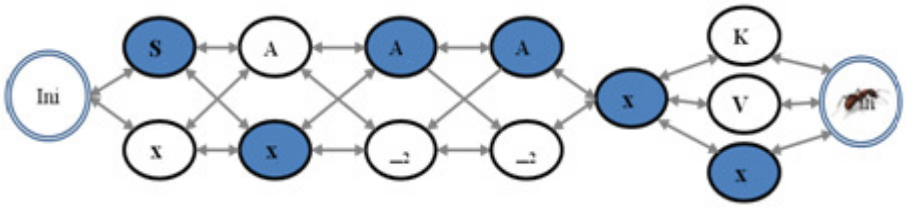


Figura 3.20 Recorrido del agente Hormiga a través del grafo.

El proceso descrito anteriormente se repetirá tantas veces como miembros tenga la colonia de hormigas. Luego se realiza la evaporación de las trazas de feromona, la cual consiste en disminuir los niveles de feromona en todos los nodos del grafo utilizando la ecuación 3.17, donde "ρ" es el coeficiente de evaporación de feromona. En general, esto se realiza recursivamente hasta completar el número de generaciones deseadas para la colonia.

$$\boldsymbol{\tau_r^k = (1 - \rho) * \tau_r^k} \qquad \textbf{Ecuación 3.17}$$

A continuación el macro-algoritmo que simula el recorrido de la colonia de hormigas a través de diferentes generaciones.

Función *RecorreGrafoACO(grafo, mapa, similMin, numMaxF)* ***devuelve*** *grafo de recorrido con las variaciones en los niveles de feromonas de sus nodos*

Entradas: *grafo, variable de tipo Nodo, contiene la información del nodo inicial del Agente.*

mapa, cadena que contiene la expresión regular de un patrón proteico.

similMin, índice mínimo de similitud para considerar acertado el hallazgo del agente hormiga en una posición especifica.

numMaxF, máximo número de hallazgos fracasados consecutivos, necesarios para dudar sobre el tipo de búsqueda.

Variables:*hormiga,es una instancia de la clase AgenteHormiga*

pilaRecorrido, pila que contiene el recorrido esperado de cada hormiga.

aminE, carácter que contiene el aminoácido que desea encontrar la hormiga.

pos, posición del nodo que se visitara.

numFallas, entero, cuenta los desplazamientos consecutivos que no superaron el índice de similitud mínimo.


```
Repita Desde i=1 hasta i = número de Generaciones de la Colonia:
    Repita Desde j = 1 hasta j = Población de la Colonia:
        hormiga.Inicializar(mapa, grafo, similMin, numMaxF)
        Repita Mientras  hormiga .recorrido.arcosDerecha[0].aminoacido  ¡= 'Fin' :
            Repita Desde k=1 Hasta k=Longitud de (hormiga.recorrido.arcosDerecha):
                listaProbAcum.Agregar( Probabilidad de ( aminE,
hormiga.recorrido.arcosDerecho[k].valor.aminoacido)
```

Algoritmo 3.2. Recorrido de la Colonia de Agentes Hormigas a través del Grafo de Recorrido.

3.3.2.3. SELECCIÓN DE LOS MEJORES NODOS

Una vez que la colonia haya concluido su trabajo, se recorrerán todos los nodos del grafo, eliminando los arcos que conduzcan hacia aquellos nodos que contienen un nivel de feromona por debajo del umbral que haya seleccionado el usuario (por ejemplo, cuando el umbral de feromona sea igual a 1,0), lo que ayudara a pre-seleccionar a los aminoácidos que contribuirán a las mejores soluciones.

En la figura 3.21 se observa el grafo de recorrido con los nodos seleccionados (marcados en azul) que superaron el umbral. En la figura se observa que se eliminaron los arcos que conducían al nodo V (valina) en la posición 6 junto a los que conectaban a los nodos vacios ubicados en las posiciones 3 y 4 dado que sus valores son menores al umbral fijado.

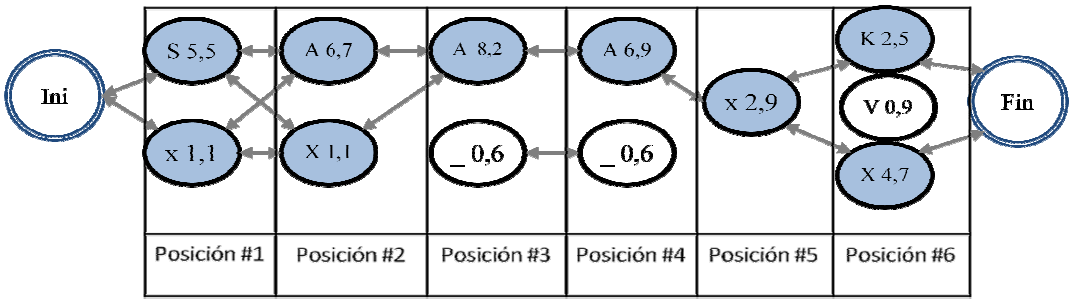


Figura 3.21 Grafo de recorrido con los niveles de feromona de cada nodo.

A continuación el macro-algoritmo para realizar esta tarea.

```
Función IdentificaNodos(grafo, umbral) devuelve grafo que solo contiene nodos cuyo nivel de
feromona supera el umbral escogido.
        Entradas:       grafo, variable tipo Nodo, contiene el grafo de recorrido
                        Umbral, variable tipo Flotante, indicador del nivel de feromona mínimo
para considerar un nodo como parte de una fusión viable

        Repita Mientras grafo.arcosDerecha[0]. aminoacido  != 'Fin' :
                Repita Desde i = 1  Hasta i = longitud(grafo.arcosDerecha):
                        Si grafo.arcosDerecha[i].feromona > umbral Entonces:
                                listAux.Agregar(grafo.arcosDerecha[i])
                grafo.arcosDerecha = listAux
                listAux = Lista Vacia
```

orrido

3.3.2.4. CONSTRUCCION DE LA EXPRESION REGULAR DE LA FUSION

Finalmente, luego de los cambios realizados al grafo en la sección 3.3.2.3, es recorrido dicho grafo para filtrar la información irrelevante y ajustar el patrón resultante. Para realizar esta tarea se analizan los nodos marcados del grafo, posición por posición, y se insertan en una lista, que contendrá el valor de los aminoácidos correspondientes a cada posición del patrón. Para lograr tal objetivo se aplican los siguientes criterios:

1. Si en una posición del grafo existe solamente un nodo que haya superado el umbral de feromona (ver figura 3.22), se ingresa a la lista el aminoácido correspondiente a dicho nodo y se continua con la siguiente posición del grafo.

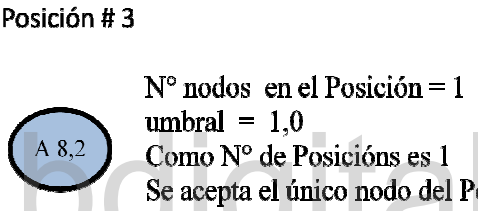


Figura 3.22 Único nodo que supera el umbral de feromona.

2. Si existen más de un nodo en la misma posición del grafo que superen el umbral de feromona, se debe verificar si alguno de estos es de tipo gap o vacio. En caso de no existir de estos tipos, se ingresa a la lista una cadena con cada uno de los aminoácidos correspondientes a dicha posición y se continúa el recorrido. Si por el contrario se encuentra un nodo de tipo gap o vacio, se aplican las siguientes condiciones:

a. Si el nivel de feromona del nodo gap es superior al resultado de la multiplicación del umbral por el número de nodos de esa posición, entonces se crea un nuevo umbral para la posición estudiada, que tendrá el mismo valor del nodo gap, y se evalúa si los demás nodos de la misma posición superan el nuevo umbral (esto se realiza para garantizar que esto no se deba al azar). Si al menos uno de ellos supera el nuevo nivel de feromona se descarta la inserción del gap en la lista, y en su lugar se agrega una cadena compuesta por los aminoácidos que tengan mayor nivel de feromona que éste, en caso contrario se ingresa a la lista solamente el valor del gap (ver figura 3.23).

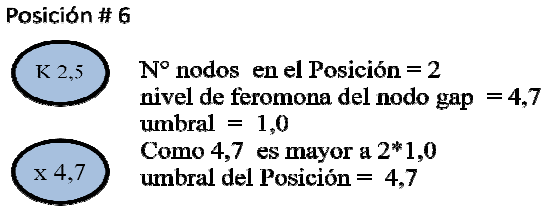


Figura 3.23 Un nodo gap que supera el valor del umbral.

b. Si el nivel de feromona del nodo gap es inferior al umbral multiplicado por el número de nodos de la posición en estudio, se descarta dicho nodo y se inserta a la lista una cadena compuesta por los aminoácidos que tengan mayor nivel de feromona que éste (ver figura 3.24)

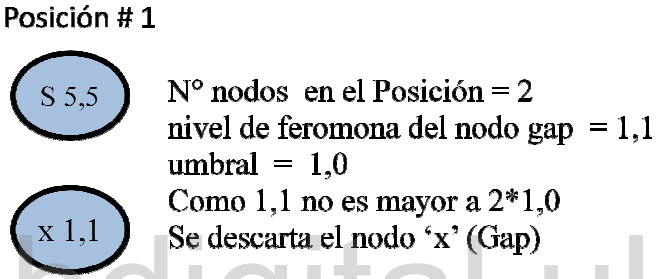


Figura 3.24 Un nodo gap que supera el umbral pero no supera el valor de feromona de otros nodos en esa posición.

c. En caso de que en una posición del grafo se encuentre un nodo del tipo vacio que haya superado el umbral, se aplicaran las mismas condiciones que con el nodo gap (puntos a y b), para verificar que esta no haya superado el umbral a raíz de un fenómeno fortuito. Si éste supera los criterios anteriormente establecidos, se omite la inserción de esa posición en la lista (esa posición desaparece en el patrón final).

Al concluir el recorrido sobre el grafo, se obtiene la lista que contiene los aminoácidos correspondientes a cada posición del patrón de fusión. Después, se agrupan los segmentos contiguos, con un solo carácter igual. Al final se construye la expresión regular resultante denotada por las reglas PROSITE que representa el patrón que caracteriza ambas expresiones regulares (constituye un motivo de proteínas, ver figura 3.25).

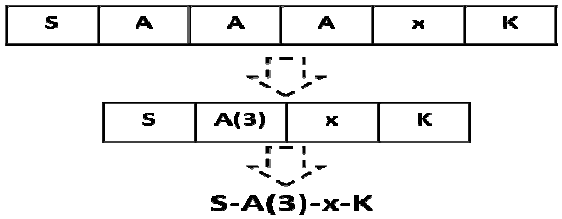


Figura 3.25 Obtención del Patrón de la Fusión

A continuación el macro-algoritmo para realizar esta tarea

```
Función ConstruyePatronFinal(grafo) devuelve  Cadena con el patrón resultante
        Entradas: grafo, variable tipo Nodo, contiene el grafo de recorrido

        Repita Mientras grafo.arcosDerecha[0]. aminoacido  != 'Fin' :
            cadenaAux = CADENA VACIA
            Si grafo.arcosDerecha[longitud(grafo.arcosDerecha)].aminoácido == 'x' Entonces:
              umbralEdo = CompruebaFero(grafo.arcosDerecha[longitud(grafo.arcosDerecha), longitud(grafo.arcosDerecha)].feromona ,
umbral)
            Repita Desde i = 1  Hasta i = longitud(grafo.arcosDerecha)-1:
              Si grafo.arcosDerecha[i].feromona > umbralEdo Entonces:
                cadenaAux.Concatena(grafo.arcosDerecha[i].aminoacido)
            Si cadenaAux = CADENA VACIA:
                cadenaAux.Concatena('x')
            listAux.Agregar(cadenaAux)
          Sino:
            Si grafo.arcosDerecha[longitud(grafo.arcosDerecha)]. Aminoácido == '_' Entonces:
                umbralEdo = CompruebaFero(grafo.arcosDerecha[longitud(grafo.arcosDerecha),
longitud(grafo.arcosDerecha)].valor.feromona , umbral)
              Repita Desde i = 1  Hasta i = longitud(grafo.arcosDerecha)-1:
                  Si grafo.arcosDerecha[i].feromona > umbralEdo Entonces:
                    cadenaAux.Concatena(grafo.arcosDerecha[i].aminoacido)
              Si cadenaAux != CADENA VACIA:
                  listAux.Agregar(cadenaAux)
            Sino:
                Repita Desde i = 1  Hasta i = longitud(grafo.arcosDerecha):
                  cadenaAux.Concatena(grafo.arcosDerecha[i].aminoacido)
            listAux.Agregar(cadenaAux)
        cadenaPatron = ConstruyePatron(listAux)
 Devuelve cadenaPatron
```

n grafo.

3.4. DEFINICION FORMAL

3.4.1. SIMILITUD DE MOTIVOS

Definición 1: *Similitud de motivos.* Puede ser considerado a nivel computacional como la similitud de cadenas de caracteres.

Los motivos pueden ser representados mediante diferentes formalismos, por ejemplo, expresiones regulares, HMM o matrices PSSM. Cuando se realiza la comparación de motivos es necesario asignar un valor de semejanza entre ellos. Por lo tanto, es necesario establecer un mecanismo formal de valoración de semejanza. Para ello, desde el punto de vista computacional, la similitud de motivos puede ser estudiada como un problema de determinar la similitud de cadenas de caracteres. De esta manera, el problema de la similitud de motivos puede ser dividido en tres sub-problemas:

1. Escoger un lenguaje apropiado para describir los motivos.
2. Escoger una función de puntuación para asignar un valor a la comparación de los motivos.
3. Realizar la comparación entre los motivos.

Pasemos a ver como se resolvería cada sub-problema

3.4.1.1. LENGUAJE PARA DESCRIBIR LOS MOTIVOS

Definición 2: *Lenguaje.* Sea L un lenguaje, este debe ser definido por un alfabeto y una gramática. Además, un lenguaje tiene individuos que no son más que instancias de él.

Definición 3*: Alfabeto*. Sea Σ el alfabeto que denota un conjunto finito de caracteres.

En nuestro caso es el alfabeto de aminoácidos Σ = {A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V}. En general, el tamaño de un alfabeto Σ es |Σ|, el tamaño del alfabeto de aminoácidos es 20 y el de ADN es 4.

Definición 4: *Gramática*. Consiste en conjunto de reglas para construir las secuencias de caracteres.

En nuestro caso, para construir las secuencias de los motivos de proteínas se usa una plantilla, bajo la cual se establecen las posibles secuencias validas de los motivos. En particular, la gramática esta descrita según las reglas PROSITE (ver sección 2.5) y la plantilla definida en la sección 3.2.2.1

Definición 5. *Individuos del lenguaje*. Los individuos son estructuras arborescentes (ver figura 3.3) que están formados por terminales y funciones, que representan cadenas de caracteres. Esas cadenas de caracteres conforman las posibles secuencias de los motivos

En nuestro caso, el conjunto de terminales más el conjunto de funciones deben ser capaces de expresar una secuencia de un motivo. Basado en ello, un individuo es definido según se indico en la sección 3.2.2.2.

3.4.1.2. FUNCION DE PUNTUACIÓN

Definición 6: *Función de Puntuación*. Determina el grado de similaridad entre los motivos en comparación.

La función de puntuación en nuestro caso es definida por una Red Neuronal de Retropropagación entrenada para reconocer una secuencia de motivo (ver figura 3.26). De esta manera, el problema de similaridad se define como un problema de reconocimiento de patrones, tal que al introducirle una secuencia del otro motivo se produce un error de reconocimiento. La red neuronal es auto–asociativa, es decir, la información de entrada a la red debe ser la misma de la salida (tanto para el entrenamiento como para cuando se usa en tareas de reconocimiento), y el error de reconocimiento es el grado de similaridad entre los motivos comparados (con el que se entreno a la red y el que se introduce después para calcular el error de reconocimiento).

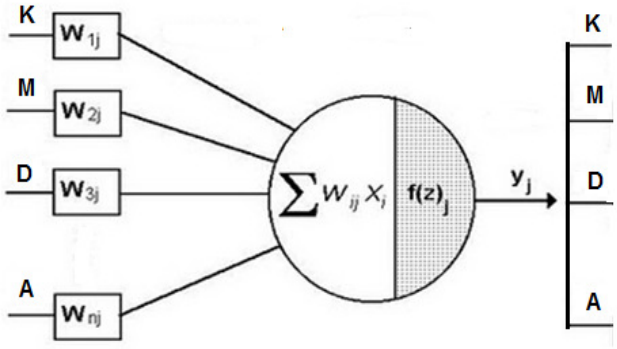


Figura 3.26 Estructura de la Red Neuronal de Retropropagación utilizada

Definición 7: *Aprendizaje.* Es el proceso por el cual la red neuronal aprende patrones.

Las redes neuronales tienen muchos mecanismos de aprendizaje, en nuestro caso se uso el aprendizaje supervisado (ver [84], [87] donde se demuestra por qué este proceso converge al culminar el proceso de aprendizaje y la sección 2.4.2.1 donde se explica el proceso de aprendizaje usado en este trabajo). En nuestro caso, la red neuronal aprenderá las secuencias de uno de los motivos de proteínas a comparar.

3.4.1.3. COMPARACION DE MOTIVOS

Definición 8: *Comparación de motivos*. Consiste en tomar un motivo, introducirlo a la red neuronal entrenada con otro motivo y obtener un error de reconocimiento.

En nuestro caso se toman las secuencias (individuos) de uno de los motivos a comparar de la población generada por la Programación Genética, se introducen a la red neuronal entrenada con el otro motivo a comparar (función de aptitud) y se calcula una tasa de semejanza (error de reconocimiento) para cada aminoácido que componen la secuencia. En la sección 3.2.2.3 (funcionamiento de la red neuronal) se indica cómo se calcula a partir de ese error de reconocimiento la similitud entre los dos motivos.

Definición 9: *Valor de Similitud*. La media del error de reconocimiento contiene toda la información de la similitud de los aminoácidos de las secuencias a comparar, lo que le da un carácter de medida representativa.

El valor de similitud en nuestro caso viene dado por la ecuación 3.14 Como se dijo antes, el proceso de calcular ese valor se explica en la sección 3.2.2.3. Básicamente, el valor de similitud es el error de reconocimiento entre los dos motivos porque determina que tan parecidos son ambos.

3.4.2. FUSION DE MOTIVOS

Definición 10: *Fusión de motivos*. Consiste en construir un patrón común para los motivos de proteínas que tienen un alto grado de semejanza.

La fusión de motivos de proteínas es parte de los objetivos de la biología moderna. Para ello, es necesario plantear nuevos conceptos que permitan encontrar patrones comunes a motivos con alto grado de similitud, que estén representados por medio de expresiones regulares. Nosotros hemos propuesto usar la técnica de Optimización de Colonias de Hormigas para resolver ese problema, la cual se basa en el comportamiento de las hormigas para hallar caminos más cortos entre los hormigueros y las fuentes de alimento.

Se ha observado que muchas especies de hormigas que caminan hacia o desde un depósito de comida, van dejando en el suelo una sustancia conocida como feromona, que tiende a evaporarse al transcurrir cierto tiempo. Esta sustancia posee una influencia en otras hormigas al momento de elegir su ruta, es decir, las hormigas tienden a elegir caminos con una alta concentración de feromona. Esto les permite seguir un camino de feromona que las lleva a encontrar buenas fuentes de alimentos que han sido previamente identificadas por otros individuos de la colonia. Inspirado en lo anterior, nosotros hemos propuesto un algoritmo de fusión de motivos. Para poder ser usado ese modelo de optimización en nuestro problema, es necesario determinar:

1. Cómo se realiza la construcción del mapa de recorrido.

2. Cómo se forma en que se establece la ruta de recorrido

3. Cómo es la construcción del motivo resultante.

3.4.2.1. MAPA DE RECORRIDO

Definición 11: *Mapa de Recorrido*. Grafo donde los nodos caracterizan cada uno de los aminoácidos que compone un motivo de proteínas dado y los arcos el orden en que aparecen en dicho motivo.

Los nodos del grafo de recorrido constituyen una estructura que contiene información sobre los aminoácidos que representan (en algunos casos, también representan identificadores especiales requerido por el proceso diseñado). Los arcos representan las relaciones entre los aminoácidos, conectando nodos según como está establecido en el motivo representado por el grafo. Así, para la construcción del grafo es importante la posición de los distintos aminoácidos a lo largo de la cadena proteica, como también a quienes están conectados a su derecha e izquierda, para que de esta manera las hormigas puedan desplazarse por el grafo en dirección horizontal como si se estuvieran desplazando sobre el motivo (ver sección 3.3.2.1).

3.4.2.2. RUTA DE RECORRIDO

Definición 12: *Ruta de recorrido*. Las hormigas realizarán el recorrido del grafo utilizando la información del mapa de recorrido y una función de transición que establece que nodo vecino del mapa de recorrido visitar.

Antes de iniciar el recorrido de las hormigas por el grafo (mapa de recorrido) se debe fijar el número de individuos (hormigas) de la colonia. Además, cada generación de la colonia se simula repitiendo el recorrido del grafo de las hormigas tantas veces se desee, lo que permite el comportamiento evolutivo y emergente porque cada nueva generación toma como partida para su recorrido el trabajo realizado por la generación anterior. Ahora bien, para que cada hormiga haga el recorrido usa la información que contiene el mapa de recorrido (valores de similitud para los aminoácidos, cantidad de intentos para encontrar un aminoácido en el nodo antes de continuar con otro, etc.), así como una función de transición que determina el desplazamiento de las hormigas (ver definición 12 y sección 3.3.2.2).

Definición 13: *Desplazamiento de las hormigas*. Para desplazarse entre el nodo inicial y el nodo final del grafo, la hormiga utiliza una función de transición que establece a que nodo dirigirse en un momento dado según la similitud entre los posibles nodos a visitar en el mapa de recorrido y los aminoácidos que componen al otro motivo a fusionar.

La función de transición es el mecanismo de decisión que usan las hormigas para determinar hacia que nodos moverse en el mapa de recorrido. Ella utiliza el mapa de recorrido e información del otro motivo a fusionar (ver sección 3.3.2.2. para detalles de este proceso) y permite:

1. Calcular la probabilidad de que la hormiga visite cada uno de los nodos vecinos al nodo en que se encuentra en un momento dado (ver ecuación 3.15)

2. Decidir que nodo va a visitar la hormiga utilizando los valores obtenidos en el punto anterior.

La función de transición, en general, hace que se visiten con una gran probabilidad los nodos cuya similitud con el aminoácido que se está tratando de buscar es alta.

3.4.2.3. CONSTRUCCION DEL MOTIVO RESULTANTE

Definición 14: *Motivo Resultante*. El motivo resultante está representado por los nodos (aminoácidos) en cada posición del grafo que contienen una cantidad de feromona superior a un umbral dado.

Cuando una hormiga se desplaza a un nodo del grafo de recorrido deposita un rastro de feromona según la ecuación 3.16. Después que todas las hormigas de la colonia han hecho el recorrido del grafo en cada generación, se realiza la evaporación de la traza de feromona utilizando la ecuación 3.17. Una vez que la colonia haya concluido su trabajo, nuestro sistema recorrerá todos los nodos del grafo, eliminando los arcos que conduzcan hacia aquellos nodos que contienen un nivel de feromona por debajo del umbral (ver sección 3.3.2.3 para más detalles). Los nodos resultantes constituyen el motivo que caracterizan la fusión (detalles de ese proceso son dados en la sección 3.3.2.4).

Ese proceso de determinación del motivo fusión se basa en el hecho de que las hormigas han recorrido los nodos (que representan los aminoácidos de uno de los motivos a fusionar) mas similares a los aminoácidos del otro motivo a fusionar (fusión de transición, ver definición 13), dejando trazas sobre esos recorridos. Eso, aunado al proceso de evaporación de las trazas, hace que hacia el final de la evolución de nuestro sistema vayan quedando trazas intensas en los nodos que corresponden a las secuencias más comunes a los dos motivos a fusionar.

CAPÍTULO IV: PRUEBAS DE ENTONACION, EXPERIMENTOS Y ANALISIS DE RESULTADOS.

Este capítulo contiene las pruebas realizadas al sistema implementado y el análisis de los resultados obtenidos. Los casos de prueba permiten determinar la eficacia y eficiencia del Sistema propuesto. Se comienza con el proceso de entonación del sistema, en donde se estudian los diferentes parámetros del mismo. Después, se lleva a cabo la comparación y fusión de patrones con la finalidad de hacer un análisis biológico de motivos, para conocer si los resultados obtenidos son útiles a los biólogos. Finalmente, se comparar nuestro sistema con trabajos previos. Para ver los detalles del funcionamiento del sistema ver Apéndice A.

4.1. CASO DE ESTUDIO

4.1.1. PROTEINA β – AMILOIDEA (APP)

La principal fuente de constitución de la proteína β – amiloidea (APP) son las placas amiloideas, la cual ha sido objeto de numerosos estudios (su expresión y metabolismo) (ver sección 2.2.3). La acumulación de péptidos amiloideas β (Aß) en estas placas fue la primera evidencia de que la APP podría ser producida de manera anormal en las enfermedades neurodegenerativas, presentando dos manifestaciones patológicas características:

a) El depósito extracelular de péptido β – amiloidea (Aß) en el parénquima cerebral[17] (placas seniles) y en los vasos sanguíneos cerebrales (amiloidea vascular).

b) La formación de ovillos neurofibrilares intraneuronales, formados por agregados de una proteína neuronal asociada a los mocrotúbulos Tau (ver sección 4.1.2).

El péptido Aß deriva de la APP. La mayor parte de la de la APP se libera de forma soluble mediante la acción de una α-secretasa[18]. Sin embargo, la formación de Aß procede del procesado de APP en sitios de corte diferentes, mediante secretasas β y γ. El péptido (Aß) forma agregados que, al parecer, son los responsables de iniciar una cascada patogénica que

[17] Es el tejido que compone el cerebro.

[18] Son enzimas que realizan cortes a la proteína y se incorporan a la membrana celular.

conduce, en último término, a la perdida de neuronas y a la demencia. Mientras que las placas amiloideas son especificas de la enfermedad de Alzheimer, los ovillos neurofibrilares se encuentran en otros trastornos.

La enfermedad de Alzheimer es multifactorial, y en su patogenia están implicados factores tanto genéticos como ambientales. Existen dos formas principales de la enfermedad: familiar y no familiar. La forma familiar es poco frecuente y de aparición temprana, mientras la forma no familiar es la más frecuente y de aparición tardía. El análisis genético de la forma familiar de la enfermedad de Alzheimer ha permitido identificar mutaciones en los genes responsables de la síntesis de Aß correspondiente a la proteína precursora amiloidea (APP) y a los que codifican las enzimas claves para la generación de Aß, las proteínas presenilina 1 (PS1) y 2(PS2); estas mutaciones ocasionan un aumento en la producción de Aß.

Estudios epidemiológicos indican que entre los factores de riesgo se encuentran la edad, el sexo, y la existencia de una lesión cerebral aguda previa o enfermedad cardiovascular. Por otra parte, el síndrome de Down incrementa el riesgo de desarrollar la enfermedad, así como las mutaciones del gen de apoliproteína E4 (apo E4), que predisponen a la enfermedad de Alzheimer, probablemente debido a que la expresión de proteínas E4 anómalas facilita la agregación de Aß.

Para los mecanismos moleculares y celulares de la enfermedad (ver figura 4.1) se han planteado dos hipótesis:

a) La hipótesis de la cascada amiloidea, que afirma que el proceso neurodegenerativo consiste en una serie de de sucesos desencadenados por el péptido Aß, generado por el procesado anormal de la APP.

b) La hipótesis de la degeneración del citoesqueleto neuronal, que propone que son cambios en el citoesqueleto, en los que participa la proteína Tau que son responsables de la enfermedad.

En cualquier caso, la disfunción celular originada en estas circunstancias causa la muerte de neuronas. La mayor vulnerabilidad la presentan las células piramidales[19] que contienen aminoácidos excitadores, y las neuronas colinérgicas[20], noradrenérgicas[21] y serotininérgicas[22] [96], [97], [98].

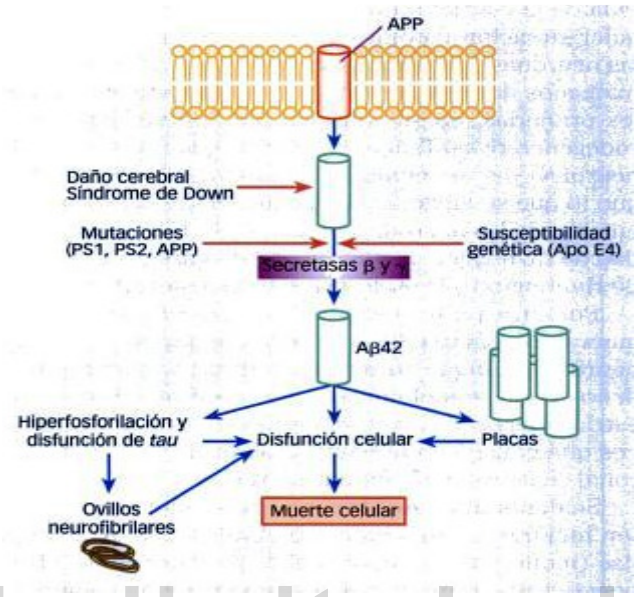


Figura 4.1 Mecanismos moleculares y celulares de la enfermedad de Alzheimer.

Por lo tanto, el análisis de los motivos de la proteína β – amiloidea (APP) permitirá conocer el grado de similitud entre ese conjunto de motivos extraídos de la base de datos AMYPdb, y de esta manera, conocer como están relacionados entre ellos. Estudios estructurales y biofísicos han aportados evidencias que sugieren que, a través de la evolución se han incorporado elementos a las secuencias y/o motivos estructurales cuya función es protegerlas de la agregación. Estos hallazgos pueden interpretarse en términos de cómo las proteínas evolucionaron no sólo para plegarse de forma funcional y estable, sino también para

[19] Llevan ese nombre por su forma, se encuentran en la corteza cerebral.

[20] Neuronas que liberan acetilcolina, que es un neurotransmisor distribuido por el sistema nervioso central y el sistema nervioso periférico. Su función es mediar en la actividad sináptica del sistema nervioso

[21] Neuronas que liberan noradrenalina que funciona como hormona y neurotransmisor. Un incremento en los niveles de noadreanalina en sistema nervioso incrementa el ritmo de las contracciones del corazón.

[22] Neuronas que liberan serotonina, que es un neurotransmisor que permite la inhibición de: la ira, la agresión, la temperatura corporal, el humor, el sueño, el vomito, la sexualidad y el apetito. Estas inhibiciones están relacionadas directamente con los síntomas de depresión.

asegurar que este estado se alcance a través de una vía que no represente riesgo de acumulación de formas no nativas con propiedades potencialmente perjudiciales para la célula. Por otra parte, se han identificado motivos estructurales comunes en las proteínas de tipo β a los que se le atribuye función inhibidora de la agregación, debido a su potencial capacidad para impedir o desfavorecer las interacciones intermoleculares que pueden causarla [99]. Por las razones antes expuestas, en la esta sección nos dedicaremos a estudiar a esta proteína.

4.2. PRUEBAS DE ENTONACIÓN

El objetivo de las pruebas de entonación es establecer valores idóneos de los parámetros del sistema para obtener los mejores resultados. Se desea establecer una regla genérica para determinar los valores ideales de los parámetros de nuestro sistema, para llevar a cabo la comparación y fusión de motivos de proteínas denotados como expresiones regulares usando las reglas PROSITE. El sistema fue desarrollado tratando de dar la mayor cantidad de libertades al usuario para fijar los parámetros de acuerdo a sus necesidades[23].

4.2.1. COMPARACIÓN DE MOTIVOS

Para realizar la comparación de motivos es necesario ajustar un conjunto de valores necesarios para el funcionamiento del sistema:

1. Parámetros para calcular el tamaño de la muestra del motivo a aprender (ver tabla 4.1): los parámetros son error estándar y fiabilidad (ver sección 3.2.2.3, donde está la descripción de esos parámetros) los cuales nos permiten calcular el número de individuos del motivo de estudio utilizados en la fase de aprendizaje en la Red Neuronal de Retropropagación, es decir, la muestra utilizada para entrenar la red (ver ecuación 3.4).

Parámetros tamaño de la muestra del motivo a aprender
Error Estándar
Fiabilidad

Tabla 4.1 Parámetros tamaño de la muestra del motivo a aprender.

[23] Las posibilidades para combinar aminoácidos que puedan generar nuevas cadenas de proteínas aun no estudiadas son prácticamente infinitas.

2. Parámetros de la Red Neuronal (ver tabla 4.2): son la tasa de aprendizaje, el momento, el error de la red y el número de iteraciones (ver sección 2.4.2.1, donde está la descripción de estos parámetros), los cuales regulan la velocidad de convergencia y la calidad del proceso de aprendizaje de la red neuronal.

Parámetros de la Red Neuronal
Tasa de Aprendizaje
Momento
Error
Número de Iteraciones

Tabla 4.2 Parámetros de la Red Neuronal

3. Parámetros para evaluar la similitud entre motivos (ver tabla 4.3): son el número de generaciones y el número de individuos (ver sección 3.2.2.2, donde está la descripción de esos parámetros), sirven para establecer cuantos individuos del motivo a comparar van a ser utilizados en cada generación/iteración de la Programación Genética, y hasta cuando evolucionará para encontrar el valor de similitud entre los motivos. Además, otros parámetros que se requieren definir son (ver sección 3.2.2.2, donde está la descripción de esos parámetros): índice de similitud de aminoácidos iguales; índice de similitud de aminoácidos de la misma familia e índice de similitud de aminoácidos diferentes. Todos estos valores de similitud, en nuestro caso, son asignados por el usuario.

Parámetros para la Similitud
Número de Generaciones
Número de Individuos
Índice de Similitud Aminoácidos Iguales
Índice de Similitud Aminoácidos familia
Índice de Similitud Aminoácidos diferentes

Tabla 4.3 Parámetros para la similitud

En general, todos estos parámetros son introducidos por el usuario en el sistema (ver figura 4.2, 4.3 y 4.4).

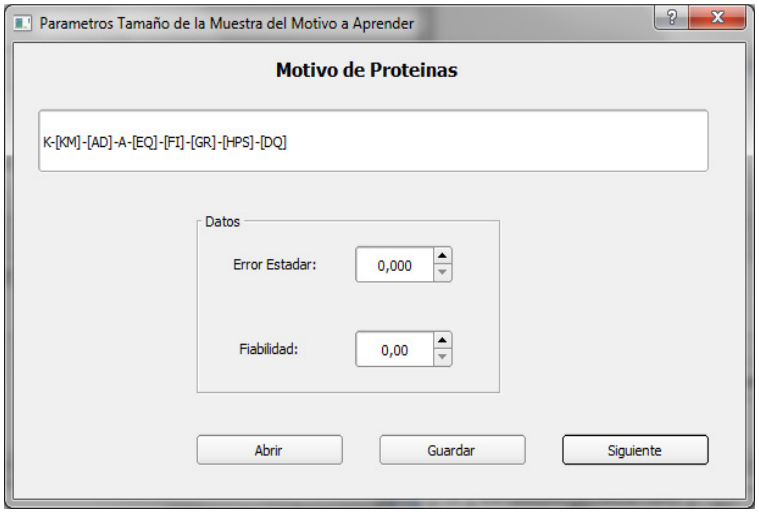


Figura 4.2 Ventana de parametros para el tamaño de la muestra del motivo a aprender en el sistema

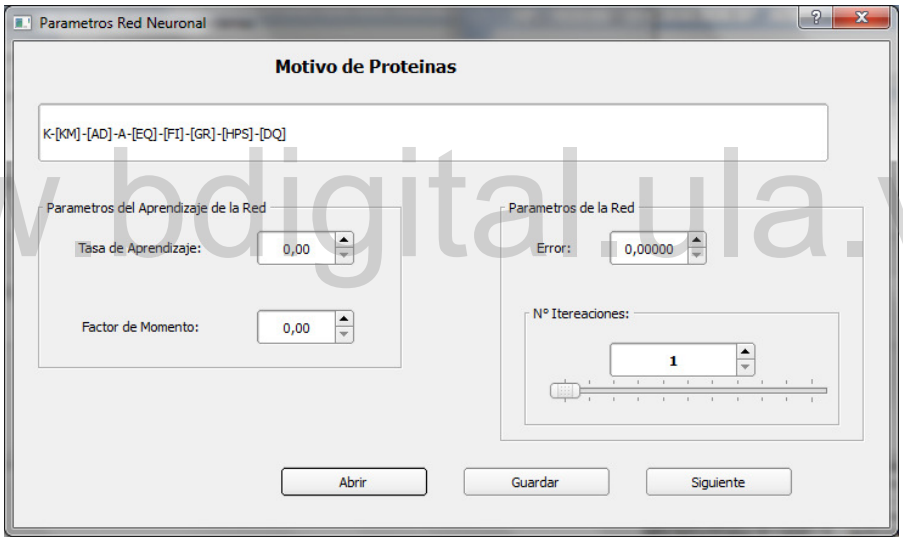


Figura 4.3 Ventana de parametros de la Red Neuronal en el sistema

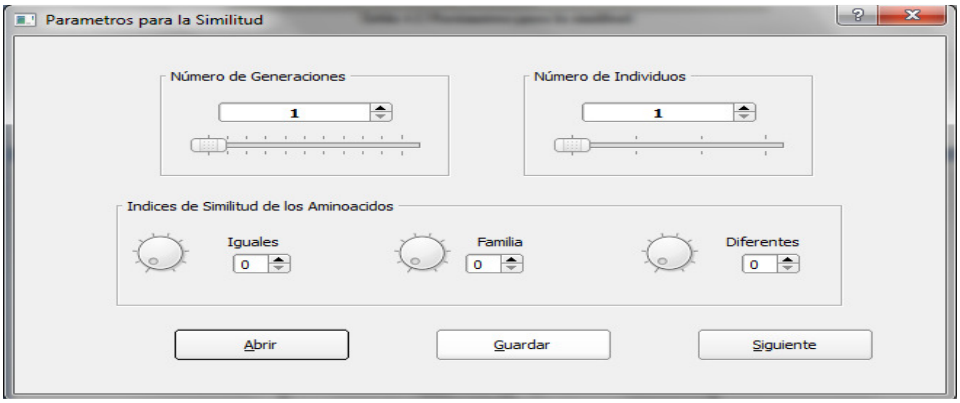


Figura 4.4 Ventana de parametros para la similitud en el sistema

4.2.1.1. PRUEBA ENTONACION: COMPARACION DE DOS MOTIVOS DE PROTEINAS

L-M-[PV]-[GS]-[GL]-[TV]-[EV]-x(2)-T y L-M-[ENPQTV]-[AGS]-[AGILV]-[ACHMTV]-[ENPQTV]-x(2)-T son motivos de la proteína β - amiloidea (APP), denotados como expresiones regulares según las reglas PROSITE, extraídos de la base de datos AMYPdb. Se toma la expresión regular L-M-[PV]-[GS]-[GL]-[TV]-[EV]-x(2)-T como motivo objeto de estudio, y la expresión regular L-M-[ENPQTV]-[AGS]-[AGILV]-[ACHMTV]-[ENPQTV]-x(2)-T como motivo a comparar. El propósito de esta prueba es obtener los valores idóneos de los parámetros del sub-sistema de comparación que permita a éste proporcionar los mejores resultados.

A continuación vamos a calcular el tamaño de la muestra del motivo a aprender, esta representa el número de secuencias del motivo que se va a introducir a la Red Neuronal de Retropropagación para su aprendizaje. La muestra debe ser lo suficientemente grande para cubrir todo el espacio de posibles secuencias que se podrían formar utilizando el motivo objeto de estudio, para ello es necesario variar los valores del error estándar (ver tabla 4.4) y fiabilidad (ver tabla 4.5). Para cada caso se deja uno de ellos fijos, para así calcular una muestra representativa de secuencias del motivo objeto de estudio (en esta prueba hipotéticamente suponemos que el tamaño de la población de motivos es 32).

Error Estándar	Fiabilidad	Tamaño de la muestra	Tamaño de la población
0,5	0,7	1	32
0,2	0,7	5	32
0,1	0,7	13	32
0,05	0,7	23	32
0,04	0,7	26	32
0,03	0,7	28	32
0,02	0,7	30	32
0,015	0,7	31	32
0,010	0,7	32	32
0,005	0,7	32	32

Tabla 4.4 Tamaño de la muestra para el motivo objeto de estudio para distintos valores del error estándar y fiabilidad igual a 0,70

Fiabilidad	Error Estándar	Tamaño de la muestra	Tamaño de la población
0,1	0,02	28	32
0,2	0,02	30	32
0,3	0,02	30	32
0,4	0,02	30	32
0,5	0,02	30	32
0,6	0,02	30	32
0,7	0,02	30	32
0,8	0,02	30	32
0,9	0,02	28	32
0,95	0,02	25	32

Tabla 4.5 Tamaño de la muestra para el motivo objeto de estudio para distintos valores de fiabilidad y error estándar igual a 0,02

Al observar los resultados de la tabla 4.4 y 4.5, vemos que el tamaño de la muestra depende sensiblemente de ambos parámetros. En general, un valor pequeño del error estándar nos habla de un mejor proceso de aprendizaje pero requiere de un número de secuencias grande, un valor de fiabilidad elevado para tener mayor confianza requiere que la muestra sea representativa de todas las secuencias posibles. La tabla 4.6 considera estos aspectos para determinar los valores idóneos del error estándar y de fiabilidad.

Error Estándar	Fiabilidad	Tamaño de la muestra	Tamaño de la población
0,04	0,9	20	32
0,03	0,9	24	32
0,020	0,9	28	32
0,015	0,9	30	32
0,010	0,9	31	32
0,005	0,9	32	32

Tabla 4.6 Tamaño de la muestra para el motivo objeto de estudio para distintos valores del error estándar y fiabilidad igual a 0,9

Como la muestra va a ser utilizada en el proceso de aprendizaje de la red neuronal, y el tiempo de éste depende de la cantidad de secuencias a ser introducidas en la red neuronal, vamos a utilizar los valores de error estándar y fiabilidad iguales a 0,02 y 0,90, respectivamente, para calcular el tamaño de la muestra (representan un % importante de la población, sin penalizar la calidad ni el tiempo de ejecución).

A continuación vamos a calibrar los parámetros de la Red Neuronal de Retropropagación: la tasa de aprendizaje (ver tabla 4.7), el momento (ver tabla 4.8), y el error de aprendizaje (ver tabla 4.9). Para cada caso se deja uno de ellos fijo, mientras que el número de iteraciones representa el número de veces que se repite el proceso de aprendizaje hasta converger el sistema.

Tasa de Aprendizaje	Momento	Error	Número de Iteraciones
0,1	0,2	0,1	64
0,2	0,2	0,1	67
0,3	0,2	0,1	86
0,4	0,2	0,1	205
0,5	0,2	0,1	350
0,6	0,2	0,1	1495

Tabla 4.7 Distintos valores de la tasa de aprendizaje

En la tabla 4.7 se observa que cuando la tasa de aprendizaje es mayor a 0,6 la red neuronal no puede alcanzar el error de aprendizaje de 0,1.

Momento	Tasa de Aprendizaje	Error	Número de Iteraciones
0,1	0,1	0,1	60
0,2	0,1	0,1	61
0,3	0,1	0,1	75
0,1	0,4	0,1	56
0,2	0,4	0,1	76
0,3	0,4	0,1	1507

Tabla 4.8 Distintos valores de momento

En la tabla 4.8 se observa la variación de la tasa de aprendizaje y el número de iteraciones que realizo la red neuronal para alcanzar el error de 0,1. En este caso se va tomar el valor de momento 0,1 y la tasa de aprendizaje 0,4, porque con ellos se realizaron el menor número de iteraciones para llegar al error de 0,1.

Error	Momento	Tasa de Aprendizaje	Número de Iteraciones
0,1	0,1	0,4	56
0,01	0,1	0,4	651
0,001	0,1	0,4	2076
0,0001	0,1	0,4	40365

Tabla 4.9 Distintos valores de momento

Como se observa en la tabla 4.9, a medida que se disminuye el error el número de iteraciones que debe realizar la red neuronal aumenta. En nuestro caso vamos a utilizar el valor de error de 0,0001. Con respecto, al número de iteraciones, la red neuronal puede hacer un número máximo de 1000000, para garantizar que pueda iterar un gran número de veces hasta alcanzar el error.

En cuanto a los parámetros de similitud, sus valores fueron seleccionados por nosotros para darle mayor importancia al caso de los aminoácidos iguales, después a los que pertenecen a la misma familia, y finalmente el caso de los aminoácidos diferentes (ver tabla 4.10). Por otro lado, en cuanto a los parámetros de la Programación Genética número de individuos y número de generaciones, probamos distintos valores solo para el número de generaciones, ya que el número de individuos en este caso no es importante porque es una muestra de la población del motivo a comparar que en casa generación se actualiza (ver tabla 4.11). Nosotros usamos como tamaño de la población 25.

Parámetros de Similitud	Valor Establecido
Índices de Similitud para los aminoácidos que sean iguales	10
Índices de Similitud para los aminoácidos que sean familia	8
Índices de Similitud para los aminoácidos que sean diferentes	0

Tabla 4.10 Parámetros de similitud en los aminoácidos

Número de Generaciones	Número de Individuos	Similitud de los Motivos (%)
10	25	80,78
30	25	85,34
40	25	88,22
50	25	90,56
100	25	90,56

Tabla 4.11 Parámetros de similitud para la Programación Genética y Resultados de la Similitud

Como se observa en la tabla 4.11 el mejor valor para el número de generaciones es 50 ya que se obtiene la mejor similitud entre los motivos.

4.2.2. FUSION DE MOTIVOS

Para realizar la fusión de motivos es necesario ajustar un conjunto de parámetros (ver tabla 4.12 y figura 4.5). Todos estos parámetros están descritos en la sección 3.3.

Parámetros fijados por el usuario
Número de Hormigas en la Colonia
Número de Iteraciones de la Colonia
Índices de Similitud para los aminoácidos que sean iguales
Índices de Similitud para los aminoácidos que sean familia
Índices de Similitud para los aminoácidos que sean diferentes
Índices de Similitud para los Gaps
Índice de Similitud Aprobatoria
Máximo número de Fracasos
Coeficiente del incremento de feromona
Índice de Evaporación de feromona
Nivel inicial feromona en los nodos del grafo
Umbral de Feromona

Tabla 4.12 Parámetros para el Sub-sistema de Fusión de Motivos.

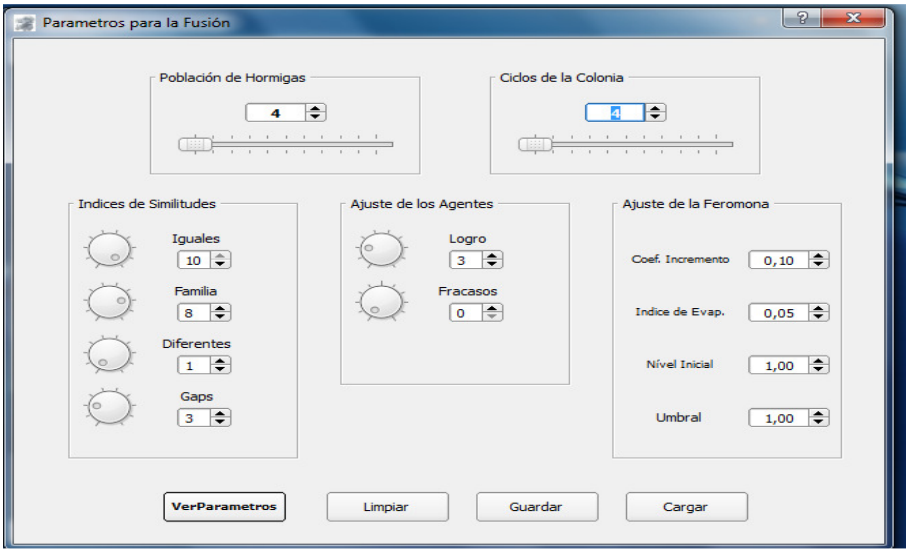


Figura 4.5 Ventana de parametros para la fusión en el sistema

Debido a que el número de parámetros ajustables en el sistema desarrollado es bastante amplio se decidió fijar los valores de algunos de ellos (ver tabla 4.13), quedando solo como parámetros variables el número de hormigas y las iteraciones que tendrá la colonia Esta decisión se tomo, debido a que la convergencia y la calidad de la solución dependen principalmente de ellos, ya que el aporte a la solución final de una hormiga es despreciable con respecto al comportamiento de toda la colonia. Lo anterior es debido a la propiedad emergente para encontrar la solución de la colonia, la cual surge de la acción autónoma de numerosos agentes simples (hormigas) [84], [88]. En ese proceso de búsqueda de la solución de la colonia subyace un mecanismo auto-organizativo que le permite adaptarse a los cambios en el ambiente con una gran robustez. Por lo tanto, asignamos un valor pequeño al índice de incremento de feromona, que representa la feromona dejada por una hormiga al visitar un nodo. También, el índice de evaporación de feromona será la mitad del índice de incremento. Como el valor de la feromona en los nodos del grafo va a aumentar o disminuir dependiendo del número de hormigas que transiten por él, tomamos el valor de uno (1) para el nivel inicial de ésta en los nodos y para el umbral. Finalmente, los valores de similitud de los aminoácidos fueron seleccionados para encontrar soluciones donde el criterio de elegir los nodos en el grafo al cual dirigirse las hormigas sea a los aminoácidos iguales o pertenecientes a la misma familia.

Parámetros del Sistema	Valor Establecido
Índices de Similitud para los aminoácidos que sean iguales	10
Índices de Similitud para los aminoácidos que sean familia	8
Índices de Similitud para los aminoácidos que sean diferentes	1
Índices de Similitud para los Gaps	3
Índice de Similitud Aprobatoria	3
Máximo número de Fracasos	0
Índice de incremento de feromona	0,10
Nivel inicial feromona en los nodos del grafo	1,0
Índice de evaporación de feromona	0,05
Umbral de feromona	1,0

Tabla 4.13 Lista de parámetros que serán tomados como constantes, para efectos de las pruebas realizadas.

4.2.2.1. PRUEBA DE ENTONACION No. 1: FUSION DE DOS MOTIVOS DE PROTEINAS

[ST]-x(2)-[ST] y [ST]-x-[RK] son motivos de proteínas denotados como expresiones regulares según las reglas PROSITE, extraídos de la base de datos AMYPdb. Se toma la expresión regular [ST]-x(2)-[ST] como ER1 (para la construcción del grafo de recorrido), y la expresión regular [ST]-x-[RK] como ER2 (mapa de recorrido de las hormigas).

Como se justifico anteriormente, para la fusión se utilizan los valores de los parámetros descritos en la tabla 4.13. Además, los parámetros variables serán el número de hormigas y las iteraciones que tendrá la colonia. Para la prueba nosotros tomamos como valor inicial para los parámetros número de hormigas e iteraciones de la colonia la longitud de ER1. Luego, procederemos a tomar valores para estos parámetros (ver tabla 4.14) hasta que confirmamos que el algoritmo ha llegado a soluciones que no cambian a pesar que cambiemos los valores de los parámetros (convergencia de la solución). Para hacer el análisis se corre 3 veces el algoritmo con el mismo conjunto de parámetros, luego se observa si existe convergencia en los resultados (debido a que se está usando un algoritmo heurístico para fusionar patrones, y las soluciones obtenidas no siempre son iguales para todas sus ejecuciones). La tabla 4.14 muestra

las 3 fusiones obtenidas para las pruebas de entonación utilizando diferentes números de hormigas e iteraciones (en este caso, la longitud de ER1 es 4).

N° Hormigas	N° Iteraciones Colonia	Expresiones Regulares Resultantes			Convergencia de los resultados
		Primero	Segundo	Tercero	
1	1	S-x(2)-T	S-x(3)	S-x(3)	Regular
1	2	T-x(2)-T	T-x(2)	[ST]-x(3)	Mala
2	2	T-x(3)	[ST]-x(3)	[ST]-x(2)-T	Mala
2	4	S-x(3)	[ST]-x(3)	S-x(2)-T	Regular
2	8	T-x(3)	[ST]-x(3)	S-x(3)	Regular
4	2	S-x(3)	[ST]-x(3)	T-x(3)	Regular
4	4	[ST]-x(3)	[ST]-x(3)	S-x(3)	Regular
4	8	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	Buena
8	2	T-x(3)	[ST]-x(3)	S-x(3)	Regular
8	4	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	Buena
8	8	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	Buena

Tabla 4.14 Búsqueda del número de hormigas e iteraciones de la colonia para la convergencia del algoritmo hacia la solución esperada.

En la tabla 4.14 podemos observar que el sub-sistema de fusión basado en colonias de hormigas comienza a converger la solución, es decir, las hormigas caminan siempre por los mismos nodos del grafo, cuando el número de hormigas y de iteraciones son iguales al máximo número de posiciones de ER1, dando como resultado en promedio la expresión regular [ST]-x(3). Con los mejores valores de los parámetros (4,4)[24], (4,8), (8,4) y (8,8), se realizaron treinta[25] corridas del sistema para las mismas expresiones regulares para observar los resultados y medir el tiempo utilizado por el algoritmo para realizar cada una de las fusiones. De este modo, tenemos otro criterio[26] que ayude a establecer cuando un conjunto de parámetros es mejor a otro.

[24] Notación que representa (número de hormigas, número de iteraciones).

[25] Número escogido basado en la ley de los grandes números consiste en que la solución final de un experimento converge a la media de las n soluciones obtenidas a lo largo de él.

[26] El tiempo en realizar la fusión es importante cuando aumenta la cantidad de posiciones en las expresiones regulares (el problema de fusión de motivos es N-P completo).

- Para el conjunto (4,4), el algoritmo converge en promedio 90% a la expresión regular [ST] – x(3) y 10 % a otras expresiones (ver tabla 4.15), con un tiempo promedio para cada fusión de 0,76 segundos[27].

Resultado	Resultado	Resultado	Resultado	Resultado
S-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	T-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	T-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)

Tabla 4.15 Resultado de la fusión para el conjunto (4,4).

- Para el conjunto (4,8), el algoritmo converge en promedio 92,33% a la expresión regular [ST] – x(3) y 7,77 % a otras expresiones (ver tabla 4.16). Además, el tiempo promedio para cada fusión es de 1,10 segundos.

Resultado	Resultado	Resultado	Resultado	Resultado
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	S-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	T-x(3)	[ST]-x(3)

Tabla 4.16 Resultado de la fusión para el conjunto (4,8).

- Para el conjunto (8,4), el algoritmo converge en promedio 96,66% a la expresión regular [ST] – x(3) y 4,44% a otras expresiones (ver tabla 4.17). Además, el tiempo promedio para cada fusión es de 0,89 segundos.

[27] Para la eficiencia del algoritmo, las pruebas realizadas se llevaron a cabo bajo el sistema operativo Microsoft Windows 7 Profesional x64, sobre una laptop HP Pavilion modelo Dv5-1132la, 2 GB de memoria RAM. El algoritmo también fue probado en Linux OpenSUSE 11.2 x64, utilizando las mismas especificaciones de hardware).

Resultado	Resultado	Resultado	Resultado	Resultado
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	S-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)

Tabla 4.17 Resultado de la fusión para el conjunto (8,4).

- Para el conjunto (8,8), el algoritmo converge en promedio 100% a la expresión regular [ST] – x(3) (ver tabla 4.18). Además, el tiempo que emplea para cada fusión es de 1,09 segundos.

Resultado	Resultado	Resultado	Resultado	Resultado
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)
[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)	[ST]-x(3)

Tabla 4.18 Resultado de la fusión para el conjunto de patrones (8,8).

Observando los resultados obtenidos para cada uno de los conjuntos seleccionados, concluimos que la solución de la fusión de las expresiones regulares [ST]-x(2)-[ST] y [ST]-x-[RK] converge a [TS]-x(3), la cual se obtiene cuando el número de hormigas y las iteraciones de la colonia duplican la longitud de la expresión regular ER1. Por lo tanto, se recomienda utilizar estos valores para que de esta manera se garantice la convergencia del resultado de la fusión. Si utilizamos un número mayor de hormigas e iteraciones se obtiene el mismo resultado, pero con un tiempo para realizar la fusión mucho más grande.

4.2.2.2. PRUEBA DE ENTONACION No. 2: FUSION DE DOS SECUENCIAS DE MOTIVOS DE PROTEINAS

El propósito de esta prueba consiste presentar al sistema dos secuencias de motivos de proteínas más grande que las de la prueba anterior, y aplicar la regla esbozada al final de la sección anterior, para conocer si el sistema es capaz de realizar la fusión de manera satisfactoria. La secuencia I(2)-G-L-M-V-G(2)-V(2) se utiliza para la construcción del grafo de recorrido (ER1) y la secuencia K-G-A-I(2)-G-L-M-V-G se usa como el mapa de recorrido de las hormigas (ER2).

Partiendo de la regla definida en la sección anterior (sección 4.2.2.1), en este caso se utiliza el conjunto de parámetros (20,20), ya que la longitud de ER1 es 10. La fusión realizada converge 100% (ver tabla 4.19) a la expresión regular x-I-G-L-M-V-G(2)-V(2) (ver figura 4.6). Además, el tiempo promedio para la obtención de las mismas fue de 3,86 segundos.

Resultado	Resultado	Resultado
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)
x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)	x-I-G-L-M-V-G(2)-V(2)

Tabla 4.19 Resultado de la fusión para el conjunto (20,20).

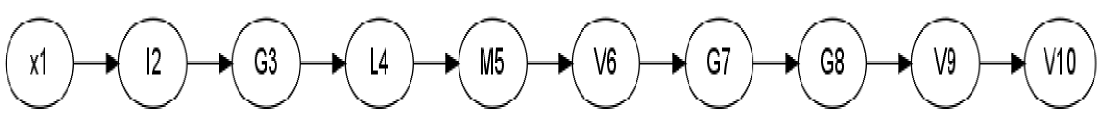


Figura 4.6 Representación grafica de la expresión regular resultante de la fusión.

Luego de realizadas las pruebas de entonación para la fusión de motivos podemos concluir que para garantizar una rápida y buena convergencia del sistema a la solución, se

recomienda que el número de hormigas de la colonia y las iteraciones de la misma sean el doble del número de posiciones que contenga la expresión regular ER1. Los otros valores de la tabla 4.13 funcionaron bien en el proceso de fusión de las expresiones regulares.

4.3. ANALISIS BIOLOGICOS USANDO NUESTRO SISTEMA

Ahora vamos a realizar la comparación y fusión de motivos de proteínas denotados como expresiones regulares según las reglas PROSITE, analizando el sentido biológico de los resultados obtenidos, para comprobar si las expresiones regulares resultantes generadas por el sistema son de utilidad para el estudio de los motivos de proteínas.

4.3.1. COMPARAR UN MOTIVO CON UN CONJUTO DE MOTIVOS

Se desea comparar un motivo de proteína con un conjunto de motivos almacenados en la base de datos AMYPdb. Se tomo el motivo de estudio denotado como expresión regular usando las reglas PROSITE K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ], el cual corresponde a la proteína β – amiloidea (APP), y los motivos a comparar tomados de la base de datos AMYPdb se muestran en la tabla 4.20.

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

Tabla 4.20 Motivos extraídos de la base de datos AMYPdb de la proteína β - amiloidea.

Se toma una muestra aleatoria de las cadenas que se pueden formar con el motivo de estudio para entrenar la Red Neuronal de Retropropagación usando las ecuaciones 3.1 y 3.4, y los valores de los parámetros fueron determinados después de un análisis experimental (ver sección 4.2.1). Se usaron los parámetros mostrados en la tabla 4.21 para obtener el tamaño de la muestra del motivo objeto de estudio (ver tabla 4.22).

Parámetro	Valor
Error Estándar	0.02
Fiabilidad	0,90

Tabla 4.21 Parámetros para obtener el tamaño de la muestra del motivo de estudio.

Población	Individuos
Población total de secuencias	192
Muestra aleatoria de secuencias	104

Tabla 4.22 Población de secuencias del motivo de estudio.

Para entrenar la Red Neuronal de Retropropagación se utilizaron los parámetros de la tabla 4.23, y la muestra aleatoria de 104 secuencias del motivo de estudio determinada en la fase anterior.

Parámetro	Valor
Tasa de Aprendizaje	0,40
Momento	0,10
Número de Iteraciones	1000000
Error	0,0001

Tabla 4.23 Parámetros para el entrenamiento de la Red Neuronal.

Para realizar la comparación del motivo objeto de estudio con cada uno de los motivos presentados en la tabla 4.20, se utilizaron los parámetros mostrados en la tabla 4.24.

Parámetro	Valor
Tamaño de la población	50
Número de Generaciones	25
Índice de Similitud Aminoácidos Iguales	10
Índice de Similitud Aminoácidos familia	8
Índice de Similitud Aminoácidos diferentes	0

Tabla 4.24 Parámetros para la similitud de los motivos.

Usando los parámetros anteriores, se obtuvieron los resultados que se muestran en la tabla 4.25.

N°	Motivo	Similitud %
1	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]	100
2	M-[TV]-[GH]-[GL]-X-V-I-X-[ET]	21,14
3	H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]	71,29
4	D-[AP]-[EK]-[FK]-X-[AH]-[DQ]-X-[GR]	16,9
5	Q-K-[EL]-[QV]-X-[FY]-[AS]-[DE]-D	29,36
6	V-X-[ACM]-[DPV]-A-E-[AF]-[EGR]-[HR]	10,07
7	M-[DE]-[AT]-E-X-[GR]-[HQ]-[DS]-[ST]	8,19
8	Q-K-[EHIKLMQV]-[ENPQTV]-X-[FWY]-[AGS]-[DE]-D	21,09
9	G-X-[ES]-V-X-[HW]-[LQ]-[KL]-L	20,84
10	L-M-[PV]-[GS]-[GL]-[TV]-[EV]	17,20
11	N-[KQ]-[GS]-[AL]-X-[IL]-[GL]-[LY]	30,94
12	V-I-X-[ET]-X-[IM]-[NV]-[IQ]-[ST]	11,87
13	K-[GT]-[AT]-[IV]-[EI]-[GL]-L-[MP]-V	78,41
14	[FWY]-[AGS]-[DE]-D-V-[AGILV]-[AGS]-N	13,93
15	V-[FI]-[FY]-X-[DER]-X-[NV]-[GQ]-S	17,21
16	K-[AGS]-A-[HIKLMQR]-I-[DEGHKNPQRSTY]-X-[HIKLMQR]-V	45,25
17	[KL]-[LR]-V-[FI]-[FY]-X-[DER]-X-[NV]	32,55
18	A-[IV]-[AI]-[DEG]-[EL]-[IM]-[QV]-[DG]-[EG]	16,57
19	L-X-[FV]-[FI]-X-E-[DR]-[MV]-[GN]	14,46
20	A-[DET]-[EV]-I-[QV]-X-[ET]-[LV]-[DV]	20,87
21	[IM]-[NV]-[IQ]-[ST]-L-X-[LM]-L	17,08
22	K-[EL]-[QV]-X-[FY]-[AS]-[DE]-D-V	38,00
23	D-[PS]-[GK]-[FKY]-X-[AV]-[HQR]-[HI]-[QR]	3,37
24	V-X-[HT]-[HQ]-[KL]-[LR]-V-[FI]-[FY]	0,00
25	M-X-[AC]-[EW]-[AF]-[GHR]-[AH]-D-[ST]	8,89
26	[AV]-[HQR]-[HI]-[QR]-[KPS]-[LQ]-V-[FML]	16,20
27	G-[FLY]-[EL]-[AV]-[EHR]-[HP]-Q-[KV]-[AL]	19,60

Tabla 4.25 Resultados de la Comparación.

En general, comparando el motivo de estudio con los motivos que se encuentran en la tabla 4.25 podemos ver que los motivos nro. 3 y 13 tienen similitud de 71,29% y 78,41%, respectivamente. Existe una gran similitud entre estos dos motivos y el motivo objeto de estudio, que no podemos ver solo observando su estructura. Esto permite a los biólogos establecer relaciones entre motivos que no pueden ser observadas a simple vista. Con el resto de los motivos presentes en la tabla 4.25, los valores de la similitud son bajos, lo que muestra que los motivos no tienen relación con el motivo objeto de estudio que aprendió la Red Neuronal de Retropropagación.

4.3.2. COMPARAR UN MOTIVO CON MOTIVOS MODIFICADOS DE ÉL

Al motivo objeto de estudio de la prueba anterior (sección 4.3.1), se le cambiaron algunas posiciones por gap, creando así nuevos motivos (ver tabla 4.26). Esta prueba nos permite conocer si el sistema es capaz de reconocer motivos modificados de él. Para este experimento se utilizaron los mismos parámetros dados anteriormente (tablas 4.21, 4.22, 4.23, 4.24).

N°	Motivo
1	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]
2	K-X-X-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]
3	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]
4	K-[KM]-X-A-[EQ]-[FI]-[GR]-X-[DQ]
5	K-X-[AD]-A-[EQ]-[FI]-[GR]-[HPS]
6	K-[KM]-[AD]-A-X(2)-[GR]-[HPS]-[DQ]
7	[KM]-[AD]-X-[EQ]-[FI]-[GR]-[HPS]-[DQ]
8	K-[KM]-[AD]A-X(3)-[HPS]-[DQ]

Tabla 4.26 Motivos modificados del motivo de estudio.

Los resultados de la comparación se muestran en la tabla 4.27.

N°	Motivo	Similitud %
1	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]	100
2	K-X-X-A-[EQ]-[FI]-[GR]-[HPS]-[DQ]	91,45
3	K-[KM]-[AD]-A-[EQ]-[FI]-[GR]	91,16
4	K-[KM]-X-A-[EQ]-[FI]-[GR]-X-[DQ]	90,27
5	K-X-[AD]-A-[EQ]-[FI]-[GR]-[HPS]	92,70
6	K-[KM]-[AD]-A-X(2)-[GR]-[HPS]-[DQ]	94,40
7	[KM]-[AD]-X-[EQ]-[FI]-[GR]-[HPS]-[DQ]	17,94
8	K-[KM]-[AD]A-X(3)-[HPS]-[DQ]	90,83

Tabla 4.27 Motivos modificados del motivo de estudio.

Comparando el motivo objeto de estudio con los motivos nro. 2, 3, 4, 5, 6, 8 en la tabla 4.11, podemos ver que la similitud es 91,45%, 91,16%, 90,27%, 92,70%, 94,40%, 90,83%, respectivamente, así que podemos afirmar que son bastante semejantes con el motivo objeto de estudio. El motivo n° 7 de la tabla 4.26 se diferencia del motivo objeto de estudio en que en las posiciones del motivo donde se encuentra solo un aminoácido fue reemplazado por un gap (en los otros casos los gaps se colocaron en posiciones donde podían estar varios aminoácidos), bajando su similitud a 17,94%. No es posible clasificar éste motivo como semejante al motivo objeto de estudio, dado que los valores obtenidos nos dicen que estas posiciones juegan un papel importante dentro del motivo.

El número de motivos de proteínas denotados como expresiones regulares según las reglas PROSITE que se encuentran en la base de datos AMYPDB es inmenso. Si utilizamos nuestro sistema es posible agruparlos 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 del sistema que estamos presentando. Esto ayudaría a los biólogos a estudiar tales motivos, o a descubrir motivos con alto grado de similitud que hasta ahora son desconocidos dentro de la base de datos AMYPdb. Esto permitirá hacer estudios sobre la similitud de los motivos de la Proteína β – amiloidea y estudiar los motivos modificados de éstas proteínas que pueden estar implicados en las enfermedades neurodegenerativas.

4.3.3. FUSIONAR UN MOTIVO CON OTRO MOTIVO DE PROTEINAS

Se desea realizar la fusión de 2 motivos de la proteína TAU extraídos de la base de datos AMYPdb (ver tabla 4.28), para encontrar un motivo (patrón) común a ellos.

Nro	Motivo
1	K-x-G-S-L-[DGK]-N-[AIV]-T-H-V-[AP]-G-G-G-[AHN]-[KV]-[KQ]-I-E-[NST]-[HR]-K-L-[DST]-F-[RS]-x-[AN]-[AS]-[KP]-x-[KV]-[GT]-[DS]-[HK]-[GT]-[AN]-[EY]-[IQ]-[PV]-x-K-S-[DP]-[GV]-[HKV]
2	G-S-[KT]-D-N-[IM]-[KNR]-H-x-P-G-G-G-[KNS]-V-Q-I-[FV]-[DHY]-[EK]

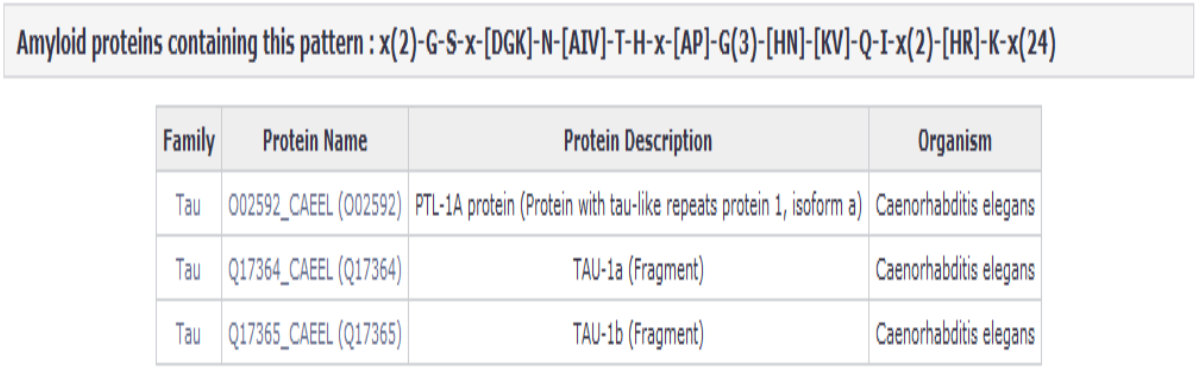
Tabla 4.28 Motivos de la proteína TAU.

Antes de iniciar la fusión, utilizamos los valores de los parámetros mostrados en la tabla 4.13. Por otra parte, tomamos la regla establecida en la sección 4.2.2.1. De esta manera, como el máximo número de posiciones que puede tener el motivo nro. 1 de la tabla 4.28 es 47, utilizamos 94 hormigas e igual número de iteraciones para llevar a cabo la fusión. Una vez definidos los parámetros, se procede a correr tres veces el sistema y se obtienen los motivos de fusión resultante en cada corrida (ver tabla 4.29).

Ejecución	Motivo Resultante
1	x(2)-G-S-x-[DGK]-N-[AIV]-T-H-x-[AP]-G(3)-[HN]-[KV]-Q-I-x(2)-[HR]-K-x(24)
2	x(2)-G-S-x-[D**G**K]-N-[AIV]-T-H-x-P-G(3)-[**A**HN]-V-Q-I-x(2)-[H**R**]-K-x(24)
3	x(2)-G-S-x-[DK]-N-[AIV]-T-H-x-[**A**P]-G(3)-[**A**HN]-V-Q-I-x(2)-H-K-x(24)

Tabla 4.29 Resultado de la fusión de los motivos.

A simple vista observamos que los tres motivos obtenidos solo varían 4 aminoácidos en 47 posibles posiciones. Además, los únicos cambios detectados pertenecen al caso donde se encuentran agrupados por corchetes. A continuación se procede a comprobar la si estos motivos existen en la base de datos AMYPdb. Consultamos en la base de datos AMYPdb, y obtenemos que dentro de la base de datos no se encuentran el 2° y 3er patrón (sin embargo el hecho que un patrón no exista en AMYPdb no significa que el mismo no sea de utilidad para posteriores estudios, ya que la base de datos de patrones proteicos se encuentra en constante modificación y crecimiento). Sin embargo, al buscar el 1er patrón obtenemos los siguientes resultados (ver figura 4.7).


Amyloid proteins containing this pattern : x(2)-G-S-x-[DGK]-N-[AIV]-T-H-x-[AP]-G(3)-[HN]-[KV]-Q-I-x(2)-[HR]-K-x(24)

Family	Protein Name	Protein Description	Organism
Tau	O02592_CAEEL (O02592)	PTL-1A protein (Protein with tau-like repeats protein 1, isoform a)	Caenorhabditis elegans
Tau	Q17364_CAEEL (Q17364)	TAU-1a (Fragment)	Caenorhabditis elegans
Tau	Q17365_CAEEL (Q17365)	TAU-1b (Fragment)	Caenorhabditis elegans

Figura 4.7 Resultados de la búsqueda del patrón de fusión en AMYPdb.

En la figura 4.7 se observa que el motivo resultante pertenece a la proteína TAU. TAU y PTL-1 son proteínas con estructuras asociadas a los microtúbulos[28]. Además todas las secuencias que contienen este motivo son de origen mamífero. En este caso el organismo que contiene estas proteínas es el nematodo[29] "*Caenorhabditis elegans*", que es expresado en dos lugares conocidos: en primer lugar en el epidermis del embrión (donde microtúbulos orientados circunferencialmente ayudan a distribuir las fuerzas generadas durante la elongación), y en segundo lugar, se encuentra en las neuronas mecanosensoriales (las cuales contienen 15 microtúbulos de protofilamento[30] necesarios para la respuesta al tacto [100]).

Por otro lado, si observamos los motivos obtenidos en la tabla 4.29, podemos eliminar los gaps que se encuentran al principio y al final de los mismos, ya que estos no aportan información determinante a la hora de definir un motivo, (este criterio se realiza a disposición del investigador).

Además podemos agregar una alanina (A) en la posición 16 del motivo nro. 1. de la tabla 4.29 ya que los motivos nro. 2 y 3 de dicha tabla lo contienen y así se puede obtener un motivo más general:

G-S-x-[DGK]-N-[AIV]-T-H-x-[AP]-G(3)-[AHN]-[KV]-Q-I-x(2)-[HR]-K

Se consulta la base de datos AMYPdb para conocer si el motivo resultante modificado existe (ver figura 4.8). Se obtiene los mismos resultados que en la figura 4.7. De esta manera, al fusionar los motivos se podría comenzar a formar una nueva categorización de estos según la estructura de los mismos.

[28] Son estructuras tubulares de 25 nm de diámetro exterior y 12 nm de diámetro interior, se extienden a lo largo de todo el citoplasma de la célula. Los microtúbulos intervienen en diversos procesos celulares que involucran desplazamiento de vesículas de secreción, movimiento de orgánulos, transporte intracelular de sustancias, división celular (mitosis y meiosis), junto con los microfilamentos y los filamentos intermedios, forman el citoesqueleto. Además, constituyen la estructura interna de los cilios y los flagelos [111].

[29] Los nematodos, o larvas de suelo, son gusanos cilíndricos no segmentados que viven en grandes cantidades en el suelo y agua. Algunas especies son parasitarias de plantas [109].

[30] Son unidades proteicas y longitudinales.13 protofilamentos forman un microtúbulo.

Amyloid proteins containing this pattern : G-S-x-[DGK]-N-[AIV]-T-H-x-[AP]-G(3)-[AHN]-[KV]-Q-I-x(2)-[HR]-K

Family	Protein Name	Protein Description	Organism
Tau	O02592_CAEEL (O02592)	PTL-1A protein (Protein with tau-like repeats protein 1, isoform a)	Caenorhabditis elegans
Tau	Q17364_CAEEL (Q17364)	TAU-1a (Fragment)	Caenorhabditis elegans
Tau	Q17365_CAEEL (Q17365)	TAU-1b (Fragment)	Caenorhabditis elegans

Figura 4.8 Resultados de la búsqueda del patrón de fusión general en AMYPdb.

4.3.4. COMPARAR Y FUSIONAR UN MOTIVO CON UN CONJUNTO DE MOTIVOS

La prueba está dividida en dos partes: en la primera parte se propone comparar un motivo de proteínas con un conjunto de motivos de proteínas almacenados en una base de datos. En la segunda parte se construye un motivo común entre los dos motivos con mayor grado de similitud. Para realizar la prueba se utilizaron 6 motivos de la proteína β – amiloidea (APP) extraídos de la base de datos AMYPdb, como se muestra en la Tabla 4.30.

Nro.	Motivo
1	H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]
2	H-D-[GPSTWY]-G-[FILMVY]-[EHIKLMQV]-[AGILV]-[DEGHKNPQRSTY]-[ACHMTV]-[ACGHNPQST]
3	L-M-[PV]-[GS]-[GL]-[TV]-[EV]-x(2)-T
4	Q-K-[EL]-[QV]-X-[FY]-[AS]-[DE]-D-V
5	V-I-x-[ET]-x-[IM]-[NV]-[IQ]-[ST]-L
6	V-x-[ACM]-[DPV]-A-E-[AF]-[EGR]-[HR]-D

Tabla 4.30 Motivos Proteína β – Amiloidea utilizados para comparar.

El motivo Nro. 1de la tabla 4.30 representa el motivo objeto de estudio, se toma una muestra aleatoria de las secuencias que se pueden formar con éste para entrenar la Red Neuronal de Retropropagación usando las ecuaciones 3.1 y 3.4. Se usaron los parámetros definidos en la tabla 4.21 para obtener el tamaño de la muestra del motivo de estudio (ver tabla 4.31).

Población	Individuos
Población total de secuencias	288
Muestra aleatoria de secuencias	126

Tabla 4.31 Población de de secuencias del motivo de estudio.

Para entrenar la Red Neuronal de Retropropagación se utilizaron los parámetros de la tabla 4.23 y una muestra aleatoria de 167 secuencias generadas del motivo de estudio. Para la similitud de los motivos, los parámetros mostrados en la tabla 4.24 fueron utilizados. Los resultados obtenidos se muestran en la tabla 4.32.

Nro.	Motivo	Similitud %
1	H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]	100
2	H-D-[GPSTWY]-G-[FILMVY]-[EHIKLMQV]-[AGILV]-[DEGHKNPQRSTY]-[ACHMTV]-[ACGHNPQST]	87,34
3	L-M-[PV]-[GS]-[GL]-[TV]-[EV]-x(2)-T	39,73
4	Q-K-[EL]-[QV]-X-[FY]-[AS]-[DE]-D-V	47,05
5	V-I-x-[ET]-x-[IM]-[NV]-[IQ]-[ST]-L	32,22
6	V-x-[ACM]-[DPV]-A-E-[AF]-[EGR]-[HR]-D	46,67

Tabla 4.32 Resultados de la Comparación.

Según la tabla 4.32 el motivo nro. 2 tiene un alto grado de similitud, por lo que podemos generar un motivo común entre éste y el motivo objeto de estudio. De esta manera, se tendrá un motivo general que contiene las características comunes de ambos motivos que podrían representar una función u otra propiedad importante que los biólogos podrían buscar en la base de datos AMYPdb (si este motivo resultante se encuentra en otras proteínas). Si observamos los motivos nro. 3, 4, 5, 6 el valor de similitud es muy bajo, lo que demuestra que estos motivos tienen poca relación con el motivo objeto de estudio.

Ahora se realiza la fusión del motivo 1 y el motivo 2 de la Tabla 4.32, utilizamos los valores de los parámetros mostrados en la. Por otra parte, tomamos la regla establecida en la sección 4.2.2.1. De esta manera, como el máximo número de posiciones que puede tener el motivo nro. 1 de la tabla 4.32 es 10, utilizamos 20 hormigas e igual número de iteraciones para llevar a cabo la fusión.

El algoritmo toma el motivo nro. 1 de la Tabla 4.32 para la construcción del grafo de recorrido (ver figura 4.9), y con el motivo nro. 2 de la misma tabla las hormigas definen el mapa de ruta.

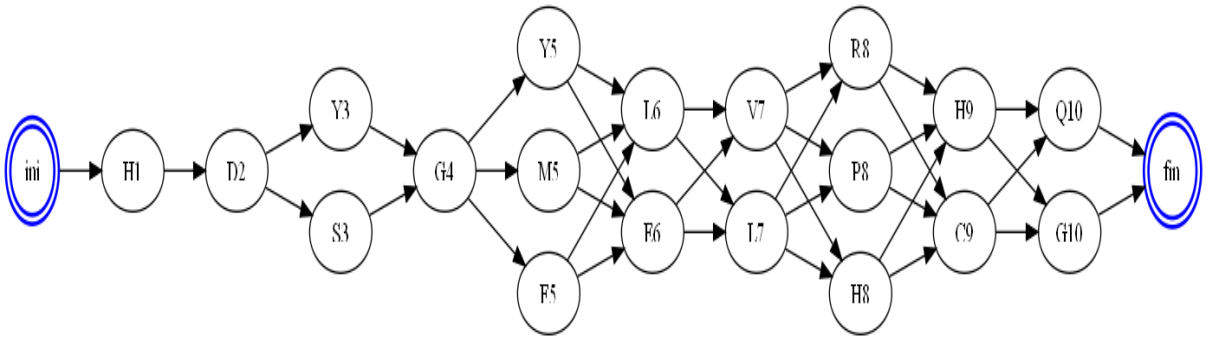


Figura 4.9 Grafo de recorrido de H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]

.

El motivo de fusión resultante es:

H-D-[SY]-G-[FMY]-[EL]-[LV]-[HPR]-[CH]-[GQ]

Podemos ver que el motivo resultante es el motivo objeto de estudio, el cual está contenido en el motivo nro.2 de la Tabla 4.32. Al consultar la base de datos AMYPdb, algunas de las proteínas que lo contienen son mostradas en la tabla 4.33. Vemos que el motivo resultante de la fusión pertenece a la proteína amiloidea beta A4 que es un péptido que se encuentra en los depósitos amiloideos cerebro-vasculares, en el núcleo de las placas neuríticas[31], y en el tejido cerebral de los pacientes con enfermedad de Alzheimer [101].

[31] Las placas seniles (neuriticas) son estructuras esféricas localizadas en el espacio extracelular donde se desplazan las terminaciones nerviosas. Se encuentran generalmente en el cerebro de los ancianos [110].

Familia	Nombre de la Proteína	Organismo
Proteína Precursora Amiloidea (APP)	A4_BOVIN (Q28053)	Bos tauros (Bovino)
Proteína Precursora Amiloidea (APP)	A4_CANFA (Q28280)	Canis familiaris (Dog)
Proteína Precursora Amiloidea (APP)	A4_CAVPO (Q60495)	Cavia porcellus (Cerdo de Guinea)
Proteína Precursora Amiloidea (APP)	A4_FUGRU (O93279)	Fugu rubripes (pez globo japonês)
Proteína Precursora Amiloidea (APP)	A4_HUMAN (P05067)	Homo sapiens (Humano)
Proteína Precursora Amiloidea (APP)	A4_MACFA (P53601)	Macaca fascicularis (Macaco Cangrejero)
Proteína Precursora Amiloidea (APP)	A4_MOUSE (P12023)	Mus musculus (ratón)
Proteína Precursora Amiloidea (APP)	A4_PANTR (Q51S80)	Pan troglodytes (Chimpanzee)
Proteína Precursora Amiloidea (APP)	A4_PIG (P79203)	Sus scrofa (cerdo)
Proteína Precursora Amiloidea (APP)	A4-RABIT (Q28748)	Oryctolagus cuniculus (conejo)
Proteína Precursora Amiloidea (APP)	A4-RAT (P08592)	Rattus norvegicus (rata)

Tabla 4.33Algunas de las Proteínas que contienen el patrón resultante.

4.3.5. COMPARAR Y FUSIONAR UN MOTIVO CON UN MOTIVO DESCONOCIDO

En esta prueba se desea conocer la similitud entre un motivo objeto de estudio y un motivo desconocido, luego realizar la fusión de ambos para obtener un motivo común. Para realizar esta prueba se utilizaron 2 motivos de la proteína β – amiloidea (APP) que se encuentran en la base de datos AMYPdb, y un motivo desconocido como se muestra en la tabla 4.34

Nro	Motivo	Proteína
1	N-x(2,4)-A-x(1,2)-I-[GQ]-x-[EM]-V-[DG]-[EG]-[LV]-[LV]-[IQ]-x-[ET]	β–amiloidea (APP)
2	I-x(3,4) –M-[TV]-[GH]-[GL]-X-V-I-X-[ET]-X-[IM]-[NV]-[IQ]-[ST]-L	β–amiloidea (APP)
3	K-x(0,1)-A-x(0,1)-I-x(3,4)-M-[ACHMTV]-[ACGHMPT]-[AGILV]-[AGILV]-V-I-x-[ENPQTV]-x-[CILMVY]	Desconocida

Tabla 4.34 Motivos utilizados en la prueba.

El motivo nro. 1 de la tabla 4.34 representa el motivo objeto de estudio. Por otro lado, se usa los parámetros definidos en la tabla 4.21 para obtener el tamaño de la muestra del motivo (ver tabla 4.35).

Población	Individuos
Población total de secuencias	1536
Muestra aleatoria de secuencias	196

Tabla 4.35Población de de secuencias del motivo objeto de estudio.

Para entrenar la Red Neuronal de Retropropagación se usaron los parámetros de la tabla 4.23, solo que el error para este caso es 0,012 y una muestra aleatoria de 317 secuencias generadas del motivo de estudio. Para la similitud de los motivos, los parámetros mostrados en la tabla 4.24 fueron utilizados, pero se duplico el valor del tamaño de la población y el número de generaciones debido a que estos motivos tienen una longitud mayor a los utilizados anteriormente. Los resultados obtenidos se muestran en la tabla 4.36

Nro	Motivo	Similitud %
1	N-x(2,4)-A-x(1,2)-I-[GQ]-x-[EM]-V-[DG]-[EG]-[LV]-[LV]-[IQ]-x-[ET]	100
2	I-x(3,4) -M-[TV]-[GH]-[GL]-X-V-I-X-[ET]-X-[IM]-[NV]-[IQ]-[ST]-L	60,26
3	K-x(0,1)-A-x(0,1)-I-x(3,4)-M-[ACHMTV]-[ACGHMPT]-[AGILV]-[AGILV]-V-I-x-[ENPQTV]-x-[CILMVY]	84,92

Tabla 4.36 Resultados de la similitud de los motivos.

Según la tabla anterior, como el motivo nro. 3 tiene un alto grado de similitud con el motivo objeto de estudio podemos generar un motivo común a ambos. Si observamos el

motivo nro. 2 el valor de similitud es 60,26%, lo que muestra que este motivo tiene algo de relación con el motivo objeto de estudio, pero no suficiente como para fusionarlo con él. El motivo resultante no tendría gran significancia por la cantidad de gaps que pudiera tener.

Ahora se pasa a fusionar el motivo nro. 1 y el motivo nro. 3, para comprobar si el motivo de fusión generado por el sistema es de utilidad para el estudio de esta proteína. Se utilizan los parámetros mostrados en la tabla 4.13. Por otra parte, tomamos la regla establecida en la sección 4.2.2.1, utilizamos 40 hormigas e igual número de iteraciones para llevar a cabo la fusión.

El algoritmo toma el motivo nro. 1 de la tabla 4.37 para la construcción del grafo de recorrido, y con el motivo nro. 3 de la misma tabla las hormigas definen el mapa de ruta. La ejecución de la fusión se realizó tres veces, y se obtuvo el siguiente motivo:

x(5)-A-x(5)- M-V-G(2)-[LV]-[LV]-I-x-T

Se procede a eliminar el segmento "x(5)-A-x(5)" (localizado al principio), ya que el mismo no aporta información importante porque posee una gran cantidad de gaps. El patrón de fusión resultante será:

M-V-G(2)-[LV]-[LV]-I-x-T

Luego se busca en la base de datos AMYPdb si existe el patrón hallado, y observamos que se encuentra en la lista de motivos de la proteína β – amiloidea (APP). Esto le servirá al investigador no solo para clasificar el patrón resultante, sino también como punto de partida para iniciar estudios sobre el motivo nro. 3 de la tabla 4.37 del cual aun no se poseía información. Según los resultados obtenidos por nuestro sistema y al consultar la base de datos AMYPdb, algunas de las proteínas que lo contienen se encuentran en la tabla 4.38. Además, el motivo resultante se encuentra en la proteína amiloidea beta A4 descrita anteriormente.

Familia	Nombre de la Proteína	Organismo
Proteína Precursora Amiloidea (APP)	A4_BOVIN (Q28053)	*Bos taurus* (Bovino)
Proteína Precursora Amiloidea (APP)	A4_CANFA (Q28280)	*Canis familiaris* (Perro)
Proteína Precursora Amiloidea (APP)	A4_CAVPO (Q60495)	*Cavia porcellus* (Cerdo de Guinea)
Proteína Precursora Amiloidea (APP)	A4_FUGRU (O93279)	*Fugu rubripes* (Pez Globo Japones)
Proteína Precursora Amiloidea (APP)	A4_MACFA (P53601)	*Macaca fascicularis* (Macaco Cangrejero)
Proteína Precursora Amiloidea (APP)	O35463_CRIGR (O35463)	*Cricetulus griseus* (Hamster Chino)

Tabla 4.37 Proteínas que contienen el patrón resultante.

4.4. COMPARACION CON OTROS TRABAJOS

A continuación se van a realizar dos tipos de comparaciones:

1. *Comparación cualitativa*: realizamos la comparación de las características más resaltantes, los aspectos iguales, diferentes y limitaciones, de nuestro sistema con otros trabajos realizados en esta área.

2. *Comparación cuantitativa*: tomamos experimentos realizado con otros enfoques y los realizamos con nuestro sistema, para comparar los resultados obtenidos en ambos.

A continuación presentamos la comparación de cada uno de nuestros sub-sistemas.

4.4.1. SUB – SISTEMA DE COMPARACION DE MOTIVOS

La tabla 4.38 muestra la comparación de nuestro enfoque con trabajos donde se utiliza la Programación Genética y las Redes Neuronales sobre motivos y/o secuencias de proteínas. En [43] se implementa un algoritmo de Programación Genética Lineal muy simple, la arquitectura fue construida en la misma forma que en [42] con funciones definidas automáticamente (ADFS). Además, el algoritmo se entrenó con secuencias positivas (verdaderas) de familias de proteínas que se desean comparar, y negativas (falsas tomadas de la base de datos ScanProsite), formando dos conjuntos diferenciados. Como función de aptitud se usa el coeficiente de correlación (ver ecuación 4.1), cuyo resultado se encuentra en el intervalo [-1, 1]. -1 indica una correlación perfecta negativa y 1 indica una correlación perfecta positiva.

$$\boldsymbol{C = \frac{TP*TN-FN*FP}{\sqrt{(TN+FN)(TN+FP)(TP+FN)(TP+FP)}}} \qquad \textbf{Ecuación 4.1}$$

Donde TP, FP son el número de verdaderos y falsos positivos, respectivamente; FP, FN son el número de verdaderos y falsos negativos.

En [44] se utilizó la Programación Genética para evolucionar una secuencia de motivo. La Programación Genética es entrenada de la misma forma que en [43], y la función de aptitud (ver ecuación 4.2) de una solución candidata asigna un valor que depende del emparejamiento de los dos conjuntos descritos anteriormente.

$$\boldsymbol{f = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) = \frac{1}{2}(Se + Sp)} \qquad \textbf{Ecuación 4.2}$$

Donde Se y Sp son las tasas de clasificación correcta positivas y negativas respectivamente. Estas tasas son conocidas como sensibilidad y especificidad.

En el enfoque [46] se estudia el problema de clasificar un conjunto de N secuencias de proteínas en K clases. Se propone un esquema de clasificación de proteínas que consiste de una herramienta de búsqueda (aprendizaje supervisado) para descubrir motivos probabilísticos en un conjunto de K familias de proteínas, se convierte las secuencia de proteínas en un vector de características (Matriz PPSM) que es aprendido por una red neuronal, y un módulo de decisión (red neuronal) asigna una familia de proteínas para cada secuencia de entrada.

Nuestro enfoque utiliza la Programación Genética con el fin de crear una población de secuencias válidas de un motivo a comparar, sin necesidad de utilizar las Funciones Definidas

Automáticamente (ADFS). Además, utiliza una Red Neural de Retropropagación como función de aptitud, esta red neuronal previamente ha sido entrenada con las secuencias del motivo objeto de estudio, con el fin de asignar un valor de similitud entre estos motivos, lo que lo hace diferente con respecto a trabajos anteriores. Con respecto a [46], se diferencia porque en este caso la red neuronal aprende una Matriz PSSM que representa las secuencias, y en nuestro caso aprende las secuencias directamente del motivo previamente transformadas según una escala numérica (ver tabla 3.1). Además, puede ser utilizado para cualquier motivo denotado como una expresión regular según las Reglas PROSITE. En [43], [44], [46] se han utilizado solamente para motivos y/o secuencias de proteínas específicos.

Característica	Nuestro enfoque	[43]	[44]	[46]
Función de Aptitud	Red Neuronal de Retropropagación	Función de Correlación	Función de Correlación que utiliza valores de sensibilidad y especificidad	Red Neuronal de Retropropagación
Medida de Similitud	Utiliza el resultado de la Red Neuronal de Retropropagación calcula un valor de similitud entre dos motivos de proteínas	Función de Correlación con casos positivos y negativos	Función de Correlación con casos positivos y negativos	Utiliza el resultado de la Red Neuronal de Retropropagación que asigna una familia de proteínas para cada secuencia de entrada
Funciones Definidas Automáticamente (ADFs)	No	Si	No	No
Elementos Utilizados	Motivos de proteínas	Secuencias de Proteínas	Secuencias de Proteínas	Secuencias de Proteínas

Tabla 4.38 Comparación cualitativa de los métodos usados en diferentes trabajos.

Además, hemos llevado a cabo un análisis de comparación cuantitativo entre nuestro enfoque y los motivos utilizados en [43] (ver tabla 4.39).

Familia Proteína	Motivo	Patrón ScanProsite
EGF_1	C – [LIQHDE] – C – [VPHK] – [GNHE]	C – x – C – x(5) – G – x(2) – C
GAPDH	A – A – C – T – [FT] – [AVTN]	[ASV] – S – C – [NT] – T – x – x – [LIM]
TRYPAIN_HIS	[LIVWH] – [STR] – A – [GA] – H – C	[LIVM] – [ST] – A – [STAG] – H – G
TUBULIN	G – G – T – G – [AS] – G	[SAG] – G – G – T – G – [SA] – G

Tabla 4.39 Motivos usados para realizar la comparación cualitativa.

La tabla 4.40 resume los resultados obtenidos utilizando diferentes enfoques (L(istGP), A(ggressive), K(oza)[42] y C(ommon)) y nuestro sistema. Para medir la calidad se utiliza la función de correlación [43]. Con nuestro sistema se llevaron a cabo dos tipos de pruebas:

1. *Motivo completo*: se compara el motivo con el patrón que se encuentra en ScanProsite, adicionando gaps al motivo para que ambos tenga la misma longitud. Los gaps se agregan al principio o al final del motivo más corto. En este caso, si agregamos gaps el valor de similitud puede ser peor (ver tabla 4.40 columna con titulo "motivo completo), porque no siempre al agregar gaps coinciden los aminoácidos iguales o pertenecen a la misma familia en ambos motivos. Sólo en un caso nosotros obtuvimos mejores resultados (TRYPAIN_HIS), en este caso no se agregaron gaps porque ambos motivos tienen la misma longitud.

2. *Fragmentos del motivo:* se toma un segmento del motivo para comparar con el patrón que se encuentra en ScanProsite. En este caso, los resultados mejoran (los resultados son buenos en todas las pruebas, ver tabla 4.40 columna con titulo "Fragmento de motivo). Sólo para la familia GAPDH tenemos un resultado que no es mejor que los mostrados en [43], esto se debe a los gaps que contiene el patrón que se encuentra en ScanProsite que no aportan ningún valor a la solución.

Familia Proteína	Nuestro Sistema		L	A	K	C
	Motivo Completo	Fragmentos del Motivo				
EGF_1	0,54	0,56	0,34	0,29	0,25	0,30
GAPDH	0,69	0,92	0,94	0,69	0,56	0,93
TRYPAIN_HIS	0,97	0,97	0,93	0,93	0,81	0,93
TUBULIN	0,82	1,00	1,00	1,00	0,61	0,92

Tabla 4.40 Resultados de la Comparación de motivos entre los diferentes enfoques.

Así, la tabla 4.40 muestra que nuestro sistema da muy buenos resultados. Nuestro algoritmo supera en casi todos los casos a ListGP, Agresivo, Koza y los algoritmos comunes. Sólo para la familia GAPDH el rendimiento de nuestro enfoque no es mejor por la cantidad de gaps que contienen los motivos. Nuestro sistema es muy sensible a los motivos con una gran cantidad de vacíos (elementos aleatorios). El enfoque de fragmentos es necesario porque permite comparar sub-secciones donde cierta similitud puede existir, la limitación es como escoger estos pequeños fragmentos para obtener buenos resultados. Esta tarea debería hacerla un experto siguiendo la idea de "hot spots" [14], [17] que son segmentos cortos de alrededor de 6 aminoácidos de motivos de proteínas.

4.4.2. SUB – SISTEMA DE FUSION DE MOTIVOS

En la tabla 4.41 comparamos nuestra propuesta con trabajos basados en Colonia de Hormigas que utilizan motivos de proteínas y/o secuencias de ADN. En el enfoque MFACO [48] la selección de un patrón de motivos se basa en una función de puntuación de consenso (CSc) (ver ecuación 4.3) y la información contenida (IC) (ver ecuación 4.4).

$$\boldsymbol{CSc} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{l}} \left(\boldsymbol{max}_{\boldsymbol{t} \in \{\boldsymbol{A,T,C,G}\}} (\boldsymbol{C(i,j)}) \right) \qquad \textbf{Ecuación 4.3}$$

Donde el patrón del motivo candidato puede ser representado por un perfil C, tal que, C(i,j) es un contador de i nucleótidos en la columna j de la matriz de alineamiento.

$$\boldsymbol{IC} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{l}} \sum_{\boldsymbol{t} \in \{\boldsymbol{A,T,C,G}\}} \boldsymbol{Q(i,j)} * \boldsymbol{log_2} \frac{\boldsymbol{Q(i,j)}}{\boldsymbol{B_0(i)}} \qquad \textbf{Ecuación 4.4}$$

Donde cada elemento Q(i,j) indica la frecuencia del nucleótido i en la posición j del motivo patrón y $B_0(i)$ denota su frecuencia histórica (por ejemplo, la frecuencia observada del nucleótido i en todas las secuencias del conjunto de datos).

En [49] se utiliza un algoritmo que combina la Optimización de Colonias de Hormigas (ACO) y la Expectativa de Maximización (EM). El punto crítico es la función de aptitud que utiliza la puntuación de consenso (ver ecuación 4.5) y la información contenida (ver ecuación 4.4).

$$\sum_{i=1}^{N} \boldsymbol{d_H(S_i, m_i)} \qquad \textbf{Ecuación 4.5}$$

Donde d_H es la distancia de Hamming[32] y se desea descubrir una secuencia S_o de un conjunto de subsecuencias m_i de la secuencia S_i que representa la solución. El objetivo es encontrar un conjunto que pueda minimizar la suma de la distancia de Hamming (d_H).

Además, [48], [49], sólo se han utilizado para la fusión de motivos de ADN que contienen cuatro nucleótidos (A, T, C, G).

En [49] luego de obtener los resultados se realizan una serie de procedimientos para refinar su calidad. El proceso de post-procesamiento incluye tres componentes:

1. Para reducir el costo computacional y ampliar la búsqueda de motivos, en el algoritmo se tiene una variable contador cuyo valor es tres (3) que indica cuantas iteraciones para que un motivo sea etiquetado como convergente.

2. Para evitar una convergencia prematura a algunas posiciones de los nucleótidos se utiliza un valor elevado de IC.

3. Definición de una puntuación de similitud (sim) (ver ecuación 4.6):

   $$\boldsymbol{sim} = \sum_{j=1}^{w} \boldsymbol{f_{jb}} \frac{\boldsymbol{f_{jb}}}{\boldsymbol{P_b}} \qquad \textbf{Ecuación 4.6}$$

   Donde f_{jb} corresponde a la frecuencia normalizada de aparición del nucleótido b en la posición j del motivo. La puntuación sim puede ser usada para calcular la similitud de una subsecuencia del motivo. En [49] se calcula la media, la mediana y la desviación estándar

[32] La distancia de Hamming en este caso representa 0 si los secuencias son iguales y 1 si son diferentes

(std) para la puntuación sim. Usando estas estadísticas, todas las secuencias de entrada son filtradas y las subsecuencias con puntuación sim mayor que la mediana y la desviación estardar son adicionadas a un conjunto; subsecuencias con puntuación sim mas baja son removidos.

Nuestro enfoque propone un motivo para la construcción del grafo de recorrido y otro motivo define el mapa de recorrido que las hormigas utilizan para caminar. Además, las hormigas ejecutan la función de transición para cada uno de los nodos que se pueden visitar en la siguiente posición utilizando el índice de similitud entre los aminoácidos del mapa y los nodos del grafo de recorrido. Algunos de los procesos de la etapa de post-procesamiento en [49] son realizados por el sub-sistema de comparación de motivos (ver sección 3.2)

Característica	Nuestro enfoque	[48]	[49]
Modelado del proceso de fusión usando Grafos	Grafo dirigido ponderado. Las posiciones en el nodo representan a cada uno de los aminoácidos en los motivos. Hay nodos especiales para casos específicos.	Grafo dirigido ponderado. Donde hay 4 posibles letras que representan los nucleótidos por cada posición	Grafo dirigido ponderado. Donde hay 4 posibles letras que representan los nucleótidos por cada posición
Elementos del Motivo	20 letras (A, C, E, F, G, H, I, K, L, M, N, P, Q, R, S; T, V, W, Y) que representan los aminoácidos y "x" representa un gap	4 nucleotídeos (A, C, G, T)	4 nucleotídeos (A, C, G, T)
Medida de Similitud	Índice de similitud esperada en un nodo	Puntuación de consenso y la información contenida (IC)	Puntuación de consenso y la información contenida (IC)
Post-Procesamiento	No	No	Si

Tabla 4.41 Resultados de la Comparación de motivos entre los diferentes enfoques.

Para la comparación cualitativa de nuestro método con trabajos anteriores, llevamos a cabo experimentos con conjuntos de datos reales ya construidos en [102] (ver los resultados

en la tabla 4.42). En este caso se realiza la fusión de S1 con S2, el motivo resultante con S3, y así sucesivamente.

Secuencias	[102]	Nuestro enfoque
S1 : ATCATCCGTGTAGCTCAAAA S2: : ATCATCCGTGTAGCTCAAAA	ATCATCCGTGTAGCTCAAAA	ATCATCCGTGTAGCTCAAAA
S3 : AGATCCGTAACGAAGTTTAC	ATCCGT	AxxATCCGTxxxGxxxxxxA
S4 : CCCCATCCGTAATTACCTAT	ATCCGT	xxxxATCCGTxxxxxxxxxxx

Tabla 4.42 Fusión de motivos.

La sub-secuencia ATCCGT representa el consenso. Este estudio sugiere que los resultados proporcionados por nuestro sistema son similares a los resultados que se encuentran en [102], con la ventaja adicional de que nuestro sistema no requiere el uso de post-procesamiento.

Realizamos una segunda comparación cualitativa con conjuntos de datos reales de las secuencias de Escherichia coli (que tiene dos partes bien conservadas, llamadas regiones -35 y -10) [103]. La fusión de un conjunto de estas secuencias se observa en la tabla 4.43. En [103] no se presenta el motivo consenso de cada fusión. En nuestro caso, nosotros vamos fusionando el motivo resultante de las dos filas previas con el siguiente hasta llegar al final.

Secuencias	Fusión usando nuestro enfoque
Bgl R mut : A A C T G T G A G C A T G G T C A T A T T T	A(2)-C-T-G-T-G-A-G-C-A-T-G(2)-T-C-A-T-A-T(3)
Deo P2 site 1 : A A T T G T G A T G T G T A T C G A A G T G	A(2)-x-T-G-T-G-A-x(6)-T-C-x(2)-A-x-T-x
Lac site 1: T A A T G T G A G T T A G C T C A C T C A T	x-A-x-T-G-T-G-A-x(6)-T-C-x(6)
Lac site 2: A A T T G T G A G C G G A T A A C A A T T T	x-A-x-T-G-T-G-A-x(14)
Mal k: T T C T G T G A A C T A A A C C G A G G T C	x(3)-T-G-T-G-A-x(14)
Mal T: A A T T G T G A C A C A G T G C A A A T T C	x(3)-T-G-T-G-A-x(14)
Tna A: G A T T G T G A T T C G A T T C A C A T T T	x(3)-T-G-T-G-A-x(14)
Uxu AB: T G T T G T G A T G T G G T T A A C C C A A	x(3)-T-G-T-G-A-x(14)
pBR P4: C G G T G T G A A A T A C C G C A C A G A T	x(3)-T-G-T-G-A-x(14)
Cat site 2: A C C T G T G A C G G A A G A T C A C T T C	x(3)-T-G-T-G-A-x(14)
Tdc: A T T T G T G A G T G G T C G C A C A T A T	x(3)-T-G-T-G-A-x(14)

Tabla 4.43 Resultados de la fusión de las secuencias de Escherichia Coli.

De acuerdo con [103], las secuencias consenso son TTGACA y TATAAT. En nuestro caso, la secuencia consenso es TGTGA. Se obtuvo una secuencia consenso para todas las secuencias de la tabla 4.43, en contraste con [103]. Nuestro sistema caracteriza muy bien las posiciones conservadas, en [103] no está claro qué posiciones son absolutamente conservadas ya que las secuencias consenso presentadas no se encuentran dentro de las secuencias mostradas en la tabla 4.43.

CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO

5.1. CONCLUSIONES

El descubrimiento de motivos comunes entre secuencias que están alejadas en el plano evolutivo (secuencias no-homólogas o no-relacionadas) es un problema muy complejo. Actualmente solo existen herramientas que permiten comparar motivos de ADN y motivos cortos (SLM Short Linear Motifs), además no permiten la fusión en un motivo común. Por ejemplo, La herramienta Pratt, que se utiliza para descubrir patrones que son conservados en un conjunto de secuencias de proteínas, no puede encontrar motivos biológicos significativos para datos heterogéneos [6]. Con un conjunto de secuencias proteicas que pertenecen a 2 o 3 familias diferentes, la herramienta Pratt encontrará solamente motivos no significativos muy cortos.

El objetivo de nuestro sistema es encontrar motivos de proteínas amiloideas a través de la comparación y fusión de éstos, para ello se integraron técnicas y métodos de la Computación Inteligente, Computación Evolutiva y Computación Emergente, haciendo que nuestro trabajo de investigación sea nuevo y original en esta área. Además, nuestro sistema permite comparar y fusionar motivos de proteínas como los que se encuentran en las bases de datos ScanProsite [50] o AMYPdb [24] (no solo motivos de ADN), lo que lo hace más amplio, versátil y genérico que las herramientas que existen actualmente. En general, nuestro sistema permite la comparación y fusión de motivos largos, degenerados y flexibles, esto representa un aporte también. Además, utilizando nuestro sistema es posible conocer si algunas familias de proteínas tienen puntos comunes.

Trabajar con proteínas es más complejo que con cadenas de ADN a causa del número de letras que las componen (20 en lugar de 4), y la posibilidad de múltiples reagrupamientos entre los aminoácidos (se pueden hacer distintos tipos de grupos de aminoácidos, lo que aumenta el número de similitudes que pueden existir entre ellos). Esto hace que la herramienta haya sido diseñada para ser eficiente en el uso de la memoria, en el procesamiento de datos, en sus tiempos de respuesta, para lograr resultados satisfactorios. En general, nuestro Sistema cumple con los requerimientos de eficiencia necesarios para que pueda ejecutarse con los recursos

disponibles en una computadora personal moderna, independientemente del sistema operativo que utilice.

El diseño, desarrollo e implantación del Sub-Sistema de Comparación utilizando la Programación Genética y las Redes Neuronales Artificiales, y del Sub-Sistema de Fusión basado en la Optimización de Colonias de Hormigas, mostraron ser una solución factible para el descubrimiento de patrones proteicos debido a su rápida convergencia hacia soluciones con sentido biológico, (ver Capítulo IV para más detalles al respecto). Específicamente, nuestro sistema contribuye a estudiar, clasificar y generar nuevos motivos de proteínas, actividades que resultarían difíciles de realizar sin herramientas como ésta. Nuestra herramienta permite determinar biológicamente por medio de la comparación cuales motivos son similares, si están relacionados o contienen información asociada. Además, por medio de la fusión se puede establecer un motivo común a un grupo de motivos, en el cual puede caracterizase su familia proteica o el motivo funcional[33]. Utilizando estos motivos de la fusión se pueden buscar en bases de datos proteicas para conocer si están vinculados a otros, y establecer relaciones funcionales y estructurales entre los motivos a fusionar y el motivo resultante, o entre familias de proteínas (para más detalles ver Capítulo IV). Además por medio del motivo resultante se pueden definir nuevas zonas conservadas o determinar las variaciones que sufren las proteínas, y conjuntamente con el conocimiento previo que tienen los biólogos de las propiedades de estos motivos se puede especificar que regiones están implicadas en la funcionabilidad de las proteínas.

5.2. TRABAJO FUTURO

Antes que nada, se deben continuar realizando experimentos biológicos con motivos de proteínas amiloideas, ya que existen más de treinta familias de estas proteínas que son necesarias compararlas entre ellas. Particularmente, deben existir probablemente varias clases de motivos amiloideas comunes en ciertas familias, pero no en todas ellas, eso son parte de los estudios a realizar. Además, los motivos realmente degenerados (mal conservados) contienen aminoácidos como la valina, la leucina y la isoleucina, muy frecuentes en todas las proteínas.

[33] Pequeñas regiones de la proteínas caracterizadas previamente, con significado funcional o estructural y que son indispensables para éstas.

Por otro lado, es necesario demostrar que los motivos comunes encontrados a través de la fusión contribuyen a la formación de fibras en las proteínas. Todo lo anterior, hacen muy interesantes y fascinantes, pero complejos y difíciles, estos estudios por realizar.

Otro aspecto importante por realizar con nuestra herramienta es masificar su uso. El número de motivos que se encuentran en la base de datos AMYPdb es de 4000 aproximadamente, y 1300 motivos en ScanProsite. Es imposible compararlos todos, por lo tanto, es necesario desarrollar una versión web de nuestro sistema que permita a los biólogos realizar la comparación y fusión de motivos en cualquier lugar que se encuentren, y así permitirles realizar sus estudios (agrupar motivos en familias por similitud, hacer búsquedas por similitud, descubrir motivos con alto grado de similitud, realizar la fusión de motivos que hasta ahora son desconocidos, etc.).

Existen nuevos conocimientos biológicos que dicen que los motivos pequeños (SLM Short Linear Motifs) son muy importantes. En la actualidad existen varios programas para buscar SLM en las proteínas no-homólogas [9], [17], [18], [19]. Esta es un área de gran competencia entre los bioinformáticos. Algunas herramientas son muy selectivas, otras son muy sensibles, pero en general permiten predecir las regiones donde se encuentran en la proteína. Por lo anterior, es necesario desarrollar un método eficiente y eficaz para buscar éstos motivos en las proteínas. En el caso concreto de las proteínas amiloideas, un aspecto fundamental es desarrollar herramientas que permitan cubrir el objetivo biológico de comprender el "código amiloidea" (conjunto de reglas comunes a todas las proteínas amiloideas que definen el orden de secuencia de los aminoácidos que constituyen patrones cortos localizados en lugares específicos de la estructura de proteínas, que, en condiciones favorables desencadenan eventos o cambios en la conformación de las proteínas y su agrupamiento en fibras). En las proteínas amiloideas se utiliza el término "hot spots", que son segmentos cortos de aproximadamente 6 aminoácidos que se asocian para formar fibras amiloideas. Los "hot spots" son solo una parte del "código amiloidea". Es necesario desarrollar una herramienta que permita buscar los "hot spots" en los motivos de proteínas, para luego realizar todo el proceso de comparación y fusión entre ellos.

BIBLIOGRAFIA

[1] Pevsner J. "*Bioinformatics and Functional Genomics*". Segunda Edición. Wiley – Backwell. 2009.

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

[3] Srinivas V**.** "*Bioinformatics. A modern Approach*". Eastern Economy. 2005.

[4] Sandve G., Drablos F. *"A survey of motif discovery methods in an integrated framework"*. Biology Direct. Vol. 1 (11). 2006.

[5] Habib N., Kaplan T., Margalit H., Friedman N., "*A Novel Bayesian DNA Motif Comparison Method for clustering and retrieval"*. Plos Comput Biol. Vol. 4(2). pp. 1-17. 2008.

[6] *Pratt Pattern Matching*. Disponible en: http://www.ebi.ac.uk/Tools/pratt/.

[7] Gusfield D. "*Trees, and Sequences: Computer Science and Computational Biology"*. Press University of Cambridge. 1999.

[8] *Functional Sites in Proteins*. Disponible en: http://elm.eu.org/links.html.

[9] *Protein Patterns & Motifs*. Disponible en : http://www.geneinfinity.org/sp/sp_proteinmotifs.html.

[10] *Teiresias*. Disponible en: http://cbcsrv.watson.ibm.com/Tspd.html.

[11] *Meme*. Disponible en: http://meme.sdsc.edu/meme/doc/examples/meme_example_output_files/meme.html.

[12] Bailey TL., Boden M., Buske FA., Frith M., Grant CE., Clementi L., Ren J., Li WW., Noble Ws."*MEME Suite: tools for motif discovery and searching"*. Nucleic Acids Research. Vol. 37. pp. W202-W208. 2009.

[13] Dogruel M., Down T., Hubbard T. "*NestedMICA as an ab initio protein motif discovery tool"*. BMC Bioinformatics. Vol. 9(19). pp 1-12. 2008.

[14] Edwards R., Davey N., Shields D**.** "*SLiMFinder: A Probabilistic Method for Identifying Over-Represented, Convergently Evolved, Short Linear Motifs in Proteins"*. PLOS ONE. Vol. 2 (10). pp. 1-11. 2007.

[15] Sayadi A., Briganti L., Tramontano A., Via A. "*Exploiting Publicly Available Biological and Biochemical Information for the Discovery of Novel Short Linear Motifs"*. Plos One. Vol. 6(7). pp. 1-11. 2007.

[16] Lieber D., Elemento O., Tavazoie S. "*Large-Scale Discovery and Characterization of Protein regulatory motifs in eukaryotes"*. Plos One. Vol. 5(12). pp 1-12. 2010.

[17] Davey N., Haslam N., Shields D., Edwards R. "*SLiMFinder: a web server to find novel, significantly over-represebted, short protein motifs"*. Nucleic Acids Research. Vol. 38. pp. W534 – W539. 2010.

[18] Neduva V., Russell R. "*DILIMOT: discovery of linear motifs in proteins"*. Nucleic Acids Res. Vol. 34. pp. W350 – W355. 2006.

[19] *Dilimot*. Disponible en: http://dilimot.russelllab.org/.

[20] Edwards R., Davey N., Shields D. "*CompariMotif: quick and easy comparisons of sequence motifs"*. Bioinformatics. Vol. 24 (19). pp. 1307-1309. 2008.

[21] *FunClust*. Disponible en: http://pdbfun.uniroma2.it/funclust/.

[22] *Bio.Motif*. Disponible en: http://www.bio-cloud.info/Biopython/en/ch13.html#motif-objects.

[23] Pawlicki S., Le Bechec A., Delamarche C. "*AMYPdb: A database dedicated to amyloid precursor proteins"*. BMC Bioinformatics. Vol. 9. pp. 273. 2008.

[24] *A Database Dedicated to Amyloid Proteins*. Disponible en: http://amypdb.univ-rennes1.fr/e107_plugins/amypdb_project/project.php.

[25] Gomis A. "*La Biología en el siglo XIX"*. AKAL. 1991.

[26] Cadeño R. Arencibia R. **"***Bioinformática: en busca de los secretos moleculares de la vida"*. Revista del Centro Nacional de Información de Ciencias Médicas de Cuba (ACIDEM). Vol. 12 pp. 2004. Disponible en: http://bvs.sld.cu/revistas/aci/vol12_6_04/aci02604.htm

[27] Joyanes L. *"La Bioinformatica como convergencia de convergencia de la biotecnología y la informatica"*.1ra Jornada de Biotecnología y Sociedad. 2003

[28] Polanski A., Kimmer M. "*Bioinformatics"*. Springer. 2007.

[29] Buehler L., Rashidi H. "*Bioinformatics basics: applications in biological science and medicine"*. Segunda Edición. CRC Press. 2005.

[30] Tramontano A. "*Introduction to Bioinformatics"*.CRC Press. 2006.

[31] *CLUSTALW*. Disponible en: http://www.ebi.ac.uk/Tools/clustalw2/index.html.

[32] *BLAST*. Disponible en: http://blast.ncbi.nlm.nih.gov/Blast.cgi.

[33] *FASTA*. Disponible en: http://www.ebi.ac.uk/Tools/fasta33/index.html.

[34] SmithT., Watermann M. "*Identification of common molecular subsequences"*. J. Mol. Biol. Vol. 147. pp. 195-197. 1981.

[35] Needleman S., Wunsch C. "*A general method applicable to search for similarities in the amino acid sequences of two protein"*. J. Mol. Biol. Vol. 48. pp. 444 - 453. 1970

[36] Dopazo J., Valencia A. *"Bioinformatica y genómica"*. Disponible en: http://www.fbmc.fcen.uba.ar/materias/ga/teoricas/bibliografia-de-las-teoricas/Bioinformatica%20para%20Genomica.pdf

[37] Dayhoff M., Schwartz R., Orcutt B. *"A model of evolutionary change in protins."* In Atlas of Proteins sequences and structures. Vol. 5. Suppl 3. pp. 345-352. 1972.

[38] *Calculate PAM Matrix.* Disponible en: http://www.bioinformatics.nl/tools/pam.html.

[39] Flores O., Rendon J., Martinez F., Guerra G., Sierra E., Pardo J. **"***Las Herramientas del Modelo Molecular"*. Disponible en: http://bq.unam.mx/wikidep/uploads/MensajeBioquimico/Mensaje_Bioq08v32p95_134_Pardo.pdf, 2008.

[40] Henikoff S., Henikoff J. "*Amino acid sustitution matrices form protein blocks"*. Proc Nactl Acad Sci. Vol. 22. pp. 10915–10919. 1992.

[41] Abascal F. "*Patrones, perfiles y dominios.*" Disponible en: http://darwin.uvigo.es/people/fabascal/Teaching/Patrones_perfiles_dominios/teoria.html.

[42] Koza J., Andre D. *"Automatic discovery of protein motifs using genetic programming"*. Ed. Xin Yao. Evolutionary Computation: Theory and Applications. World Scientific. pp. 1-28. 1996.

[43] Seehuus R., Tveit A., Edsberg O. "*Discovering biological motifs with genetic programming"*. GECCO'05 Procceding of the conference on Genetic and evolutionary computation. pp. 401-408. 2005

[44] Handstad T., Hestnes A., Sætrom P. "*Motif kernel generated by genetic programming imprives remote homology and fold detection"*. BMC Bioinformatics. Vol. 8(23). 2007. Disponible en: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1794419/

[45] Wu C., Sheng Z., Simmon K., Shivakumar S. **"***Motif Neural Network desing for large-scale protein familiy indentification"*. Neural Networks 1997. International Conference. pp. 86-89. 1997.

[46] Blekas, K, Fotiadis, D y Likas, A. *"A Motif-based Protein Sequence Calssification using Neural Networks"*. Journal of Computational Biology. Vol. 12(1). pp. 64-82. 2005.

[47] Liu D., Xiong X., Das Gupta B. "*A self-organizing Neural Network structure for motif identification in DNA sequences"*. Proceedings of the IEEE international conference on networking sensing and control. pp. 129-134. 2005.

[48] Bouamama A., Boukerram A., Al-Bederneh A. "*Motif Finding usin Ant Colony Optimization"*. Ed. Dorigo M. et al. Springer-Velang Berlin Heidelberg. ANTS 2010. pp. 464-471. 2010.

[49] Chen-Hong Yang, Yu-Tang Liu, Li-Yeh Chuand. *"DNA Motif Discovery Based on Ant COlony Optimization and Expectation Maximization"*. IMECS 2011. Proceeding of the international Multiconference of Engineers and Computer Scientists. pp. 169-174, 2011.

[50] *ScanProsite*. Disponble en: http://prosite.expasy.org/scanprosite/.

[51] Polanski A., Kimmel M. "*Bioinformatics"*. Springer. 2007.

[52] Ramsden J. "*Bioinformatics An Introduction"*. Segunda Edición. Springer. 2009.

[53] Febles J. González A. *"Aplicación de la minería de datos en bioinformática"*. Revista del Centro Nacional de Información de Ciencias Médicas de Cuba ACIMED. Vol. 10(2). pp 69-76, 2002.

[54] *"Acceso a fuentes de información genómica y herramientas bioinformáticas básicas"*. Disponible en: http://bvs.isciii.es/bib-gen/Actividades/curso_virtual/Introduccion/bioinformatica.htm.

[55] *The National Center for Biotechnology Information.* Disponible en: http://www.ncbi.nlm.nih.gov/.

[56] *EMBL - EBI European Bioinformatics Institute*. Disponible en: http://www.ebi.ac.uk/.

[57] *DNA Data Bank of Japan*. Disponible en: http://www.ddbj.nig.ac.jp/.

[58] *UniProt.* Disponible en: http://www.uniprot.org/.

[59] *Protein Data Bank.* Disponible en: http://www.rcsb.org/pdb/home/home.do.

[60] *ArrayExpress.* Disponible en: http://www.ebi.ac.uk/arrayexpress/.

[61] *Stanford Microarray Database*. Disponible en: http://smd.stanford.edu/

[62] *The Gene Ontology*. Disponible en: http://www.geneontology.org/.

[63] Sanchez J. *Diccionario de la Ciencia.* Drakontos Bolsillo. 2006.

[64] Baynes J., Dominiczak M. "*Bioquímica Médica"*. Elsevier. 2006.

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

[66] Teijon J. "*Bioquímica Estructural"*. Tébar. 2001.

[67] Berg J., Stryer L., Tymoczko J., "*Biochemestry"*. Quinta Edición. W. H. Freeman. 2002.

[68] Peyrefitte G. "*Principios Básicos de Biología Humana"*. Masson. 1995.

[69] McMurry J. "*Química Organica"*. Sexta Edición. Brook Cole. 2004.

[70] CampBell M. "*Bioquímica"*. Cuarta Edición. Thomson. 2004.

[71] Cruz M. *"Diseño, síntesis y ecaluación de inhibidores de la proteína B - amiloidea. Desarrollo de un modelo de fibrogénesis"*. 2003. Disponible en: http://www.tdx.cat/handle/10803/2788;jsessionid=15D00759219553E8B67874350CD5B092.tdx2

[72] Sipe J., A Cohen. *"Review: History of the Amyloid Fibril"*. Journal of Structural Biology. Vol. 130(2-3), pp. 88-98. 2000.

[73] Westermark, P. *"Classification of amyloid fibril proteins and their precursors: An ongoing discussion"*. Amyloid: The Journal of Protein Folding Disorders, Vol. 4(3). pp. 216-218. 1997.

[74] Dinoto L., Deture M., Purich D. "*Structural insights into Alzheimer filament assemby pathways based on site-directed mutagenesi and S-glutathionylation of three-repeat neural Tau protein"*. Miscros. Re. thech. Vol. 67, pp. 156-163. 2002.

[75] Stefani M., Dobson C. "*Protein aggregation and aggregate toxicity: new insights into protein folding, misfolding diseases and biological evolution"*. J. Mol. Med. Vol. 81. pp. 678-699. 2003.

[76] Prusiner S. *"Prions"*. Proc. Nalt. Acad. Sci, (95) pp. 13363-13383. 2006.

[77] Aguilar J. "*Compiler Course"*. Houston University. 2000.

[78] 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.

[79] Ferreira P., Azevedo. *"Evaluating deterministic motif significance measures in protein databases"*. Algorithm for Molecular Biology, Vol. 2. 2007.

[80] Yang J. Deugun J. Sun Z. "*A new sheme fpr protein sequence motif extraction"*. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Vol. 9, pp 280.1. 2005.

[81] Bairoch A. "*Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through excitinh times!"* Bioinformatics. Vol. 16(1). pp. 48-64. 2000.

[82] Bairoch A., Bucher P., Hofmann K. "*The PROSITE data base its status in 1997"*. Nucleic Acids Research. Vol. 25(1). pp. 217 – 221. 1997.

[83] Seckbach J., Rubin E. "*The new avenues in bioinformatics"*. Kluwer Academic Publisher. 2004.

[84] Aguilar J., Rivas F., "*Introducción a las técnicas de Computación Inteligente"*. Meritec. 2001.

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

[86] Koza J.**,** Forrest B., Andre D., Keane M. "*Genetic Programming III: Darwinian Invention and Probrem Solving"*.Morgan Kaufmann. 1999.

[87] Hilera J., Martinez V. "*Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones"*. Addison – Wesley. 1995.

[88] Dorigo M., Birattari M., Stutzle T. *"Ant Colony Optimization"*. Computational Intelligence Magazine IEEE. Vol. 1(4). pp. 28-39. 2006.

[89] Dorigo M., Di Caro G., Sampels M. "*Ant Algorithm"*. Springer, 2002.

[90] Dorigo M., Stützle T. *"Ant Colony Optimization"*. MIT. 2004.

[91] Altamiranda J., Aguilar J., Delamarche C. "*Similarity of Amyloid Protein Motif using an Hybrid Intelligent System"*. IEEE Latin America transactions, Vol. 9(5). pp. 700 - 710. 2011.

[92] Altamiranda J., Aguilar J., Delamarche C. "*Comparación y fusión de Motivos de Proteína β-amiloidea utilizando Computación Inteligente"*. Aceptado para publicar. Revista de Estudios Transdiciplinarios RET, Vol. 3(1). 2011.

[93] Torres R., Altamiranda J., Aguilar J., Delamarche C. "*Regular Expression Fusion using Emergent Computing"*. Proceeding of the 9th WSEAS International COnference on Advances in E-activities, Information Security and Privacy. pp. 64 - 71. 2010.

[94] Altamiranda J., Aguilar J., Delamarche D. "*Comparison and Fusion Model of Protein Motifs using Intelligent Computing"*. En preparación. 2012.

[95] Altamiranda J., Aguilar J., Delamarche C. "*Fusion Method of Protein Motifs based on Ant Colony Optimization"*. Applied Soft Computing. En revisión. 2012.

[96] Hooper N. "*Alzheimer's disease: Methods and Protocols"*. Humana Press. 2000.

[97] Xia W., Xu H. "*Amyloid Precursor Protein. A Practical Approach"*. CRC Press. 2005.

[98] Lorenzo P., Moreno A., Lizasoain I., Leza J. "*Farmacología Básica y Clinica"*. 18a. Edición. Médica Panamericana. 2008.

[99] Yauner L., Rodriguez S., Becerril B. *"Las Amiloidosis humanas: Cuando las proteínas muestran su lado oscuro"*. 2008. Disponible en:. http://bq.unam.mx/wikidep/uploads/MensajeBioquimico/Mensaje_Bioq08v32p79_94_Becerril.pdf.

[100] Goedert M., Baur C., Ahringer J., Hasegawa M., Spillantini M., Smith M., Hill F. *"PTL-1, a microtubule-associated protein with tau-like repeats from the nematode Caenorhabditis elegans"*. Journal of Cell Science. Vol. 109. pp. 2661-2672. 1996

[101] Roch J., Shapiro P., Sundsmo M., Otero D., Refolo L. "*Bacterial Expression, Purification and Functional Mapping of the Amyloid β/A4 Protein Precursor"*.The Journal of Biological Chemistry, Vol. 267(4), pp. 2214-2221. 1992.

[102] Wei Z., Jensen T. *"GAME: detecting cis-regulatory elements using a genetic algorith"*. Bioinformatics, Vol. 22, pp. 1577-84. 2006

[103] Stormo G., Hartzell G. "*Identifing protein-binding site from unaligned DNA fragments"*. Proc. Natl. Acad. Sci, Vol. 86(4), pp. 1183-1187. 1989.

[104] Piqueras J., Fernández A., Santos J., González J. "*Genética"*. Ariel Ciencia, 2002.

[105] Stützle T., Hoos H. "*Max-Min ant System"*. Future Generation Computer Systems, Vol. 16, pp. 889-914. 2000.

[106] Oliva R., Vidal J. *"Genoma Humano Nuevos avances en investigación, diagnóstico y tratamiento"*. Publicaciones de la Universidad de Barcelona. 2009.

[107] "*Fuerzas de la estructura proteíca*". Disponible en: http://gmein.uib.es/moleculas/fuerzas_proteinas/fuerzaproteinasjmol.html.

[108] Aho A., Hopcroft J., Ullman J. "Data Structures and Algorithms" Addsion-Wesley Series in Computer Science and Information Processing. 1983.

[109] *Nematodos*.Disponible en: http://apps.cimmyt.org/spanish/docs/field_guides/enfplagastrigo/Nematodos.pdf.

[110] Castro A., Martinez A. *"La enfermedad de Alzheimer: Bases moleculares y aproximaciones terapéuticas."* Disponible en: http://www.aecientificos.es/empresas/aecientificos/intereshtml/alzheimer/alzheimer.htm.

[111] Welsch U., *"Histología"* Editorial Médica Panamericana, Segunda edición. 2010.

APENDICE A: MANUAL DEL USUARIO

A.1. INSTALACION DEL SISTEMA

Antes de iniciar la instalación del sistema desarrollado, se debe comprobar que el computador donde se desee ejecutar el mismo cuente previamente con los siguientes programas:

1. Sistema Operativo Windows XP/Vista/7 o Linux a partir del kernel 2.6.28 (en caso de disponer de versiones anteriores no se garantiza la funcionalidad del sistema).

2. Python 2.6 o superior

3. Pyqt4

4. GraphViz-2

5. Setuptools-0.6

En caso de no disponer de los mismos, puede descargarlos a través de los gestores de paquetes de la distribución de Linux que esté utilizando, o bien bajar los archivos desde los siguientes enlaces web:

- http://www.python.org/
- http://www.riverbankcomputing.co.uk/software/pyqt/intro
- http://www.graphviz.org/
- http://pypi.python.org/pypi/setuptools

Luego se necesita descargar la biblioteca Pynu-0.1.1, realizada la descarga se procede a su instalación, se descomprime el archivo y se guarda la carpeta en el lugar de su preferencia, luego, dentro de la carpeta “Pynu-0.1.1” se abre una consola de comandos con privilegios de

usuario administrador o root (dependiendo del sistema operativo que se esté utilizando) y se escribe:

"python setup.py isntall"

Después de haber instalado los paquetes necesarios para la puesta en funcionamiento del sistema, se ingresa a la carpeta ComFusiPro (bien sea a través de una consola de comandos o de un explorador de ventanas) y luego corremos el archivo "FusiProACO.py" (si accedimos a él a través de una consola se deberá escribir "python FusiProACO.py")

A.2. MANUAL DE USUARIO DEL SISTEMA

Comenzamos mostrando la pantalla principal del sistema ComFusiPro (ver figura A.1), desde ésta se pueden acceder a los menús desplegables que permiten trabajar con los Proyectos, Transformar Formato, Comparar Motivos y Fusionar Motivos.

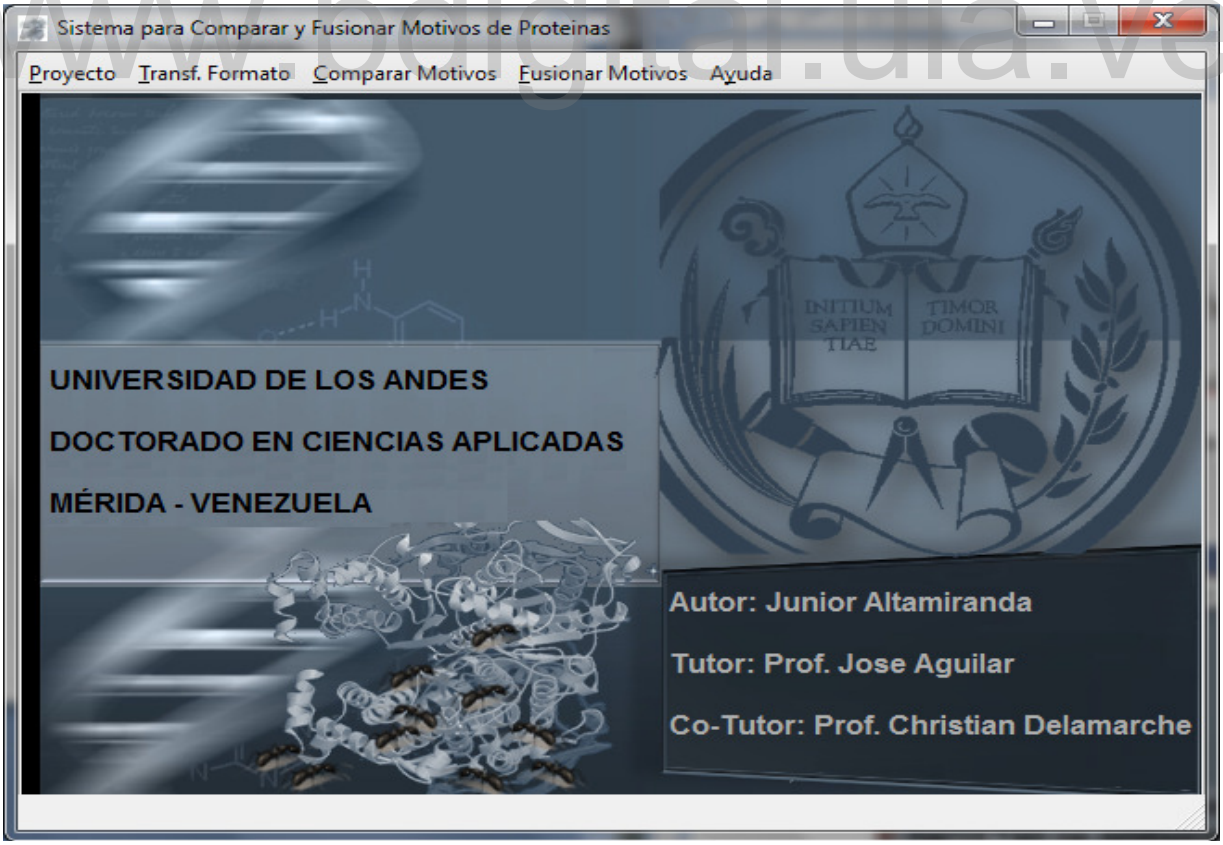


Figura A.1 Ventana principal del sistema

En el menú desplegable “Proyecto” (ver figura A.2), se pueden seleccionar 4 opciones:

1. Abrir: abre la interfaz abrir proyecto, la cual por defecto se enlazara a la carpeta proyectos y mostrara todos los trabajos que se han realizado previamente con el sistema, en esta carpeta se encontrara un archivo de texto plano por cada carpeta creada, para abrir el proyecto, se debe seleccionar el archivo de texto.

2. Nuevo: conduce a una interfaz similar a la de abrir proyectos, pero en este caso nos permite crear un trabajo nuevo que será almacenado en la carpeta proyectos.

3. Guardar: el sistema permite que se realice la comparación y fusión de patrones sin la necesidad de haber creado un proyecto, en ese caso los resultados del trabajo realizado se almacenaran en la carpeta temp, y serán eliminados una vez que se haya cerrado el sistema, por eso se agrego esta opción, para poder guardar un trabajo especifico en aquellos casos en los que se obvio la previa creación de un proyecto, al darle clic sobre guardar, se abre una ventana que pedirá el nombre del proyecto, y trasladara toda la información de la carpeta temp a la carpeta proyectos con el nombre que se le haya indicado.

4. Salir: permite salir del sistema y borrar todos los datos almacenados en la carpeta temp.

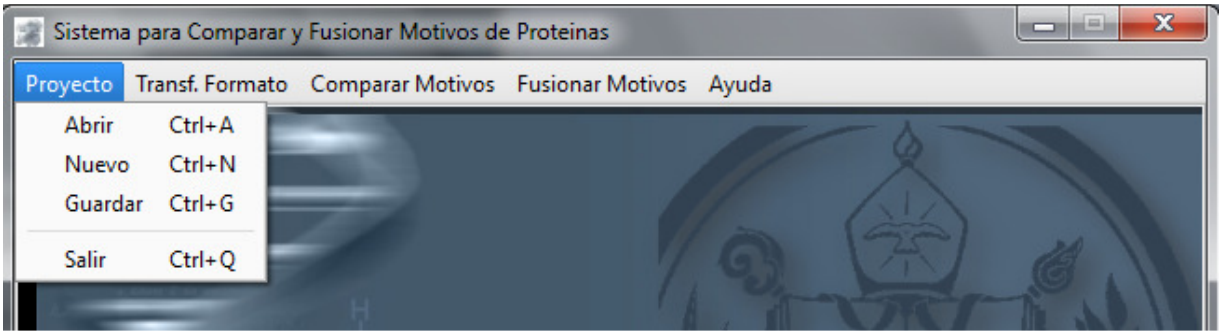


Figura A.2 Opciones del menú Proyecto

En el menú desplegable “Transf. Formato” (ver figura A.3), se puede cambiar los motivos de otros formatos a PROSITE, existen 2 opciones:

1. Fasta a Prosite: convierte un motivo del formato Fasta a Prosite (ver figura A.4).

2. Tres letras a Prosite: convierte un motivo de formato 3 letras a Prosite (ver figura A.5).

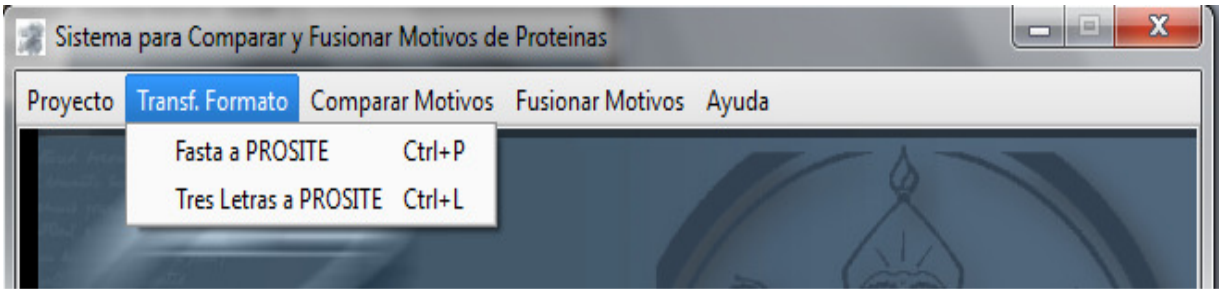


Figura A.3 Opciones del menú Transf. Formato

Figura A.4 Transformar cadena de FASTA a una expresión en PROSITE

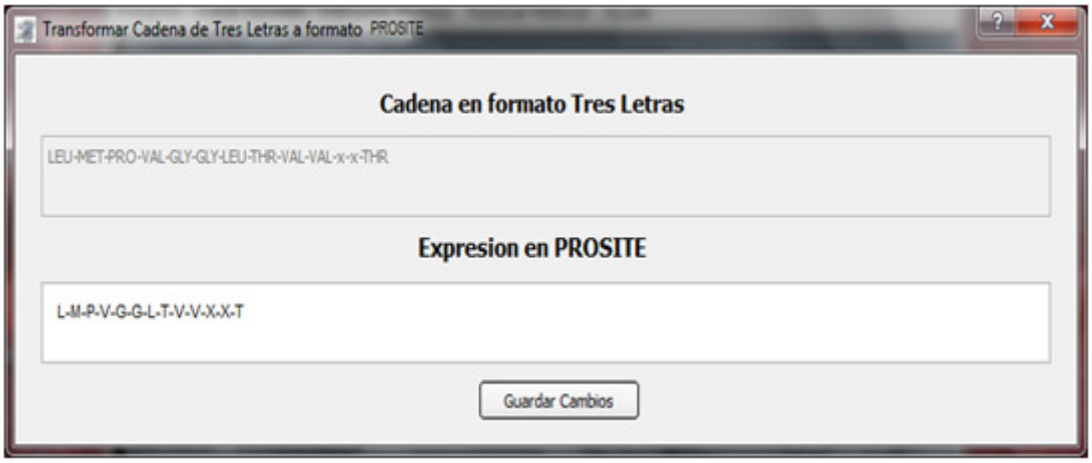


Figura A.5 Transformar cadena de Tres Letras a una expresion PROSITE

De vuelta al menú principal se localiza el menú "Comparar Motivos" (ver figura A.6**¡Error! No se encuentra el origen de la referencia.**).

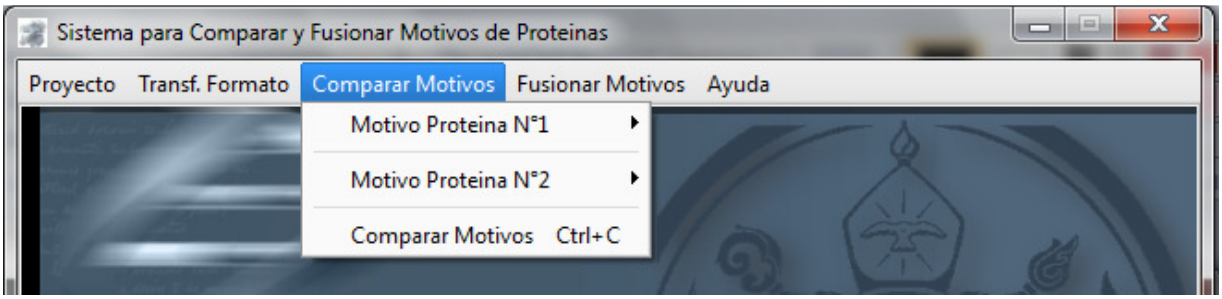


Figura A.6 Opciones del menú Comparar Motivos

Este menú está dividido en 3 sub-menús:

1. Motivo Proteína Nº 1: es un motivo que se utiliza como objeto de estudio. Este menú contiene (ver figura A.7):

   a. Nuevo Motivo: Permite escribir un nuevo motivo.

   b. Abrir Motivo: Permite seleccionar un motivo existente.

   c. Editar Motivo: Permite modificar un motivo existente.

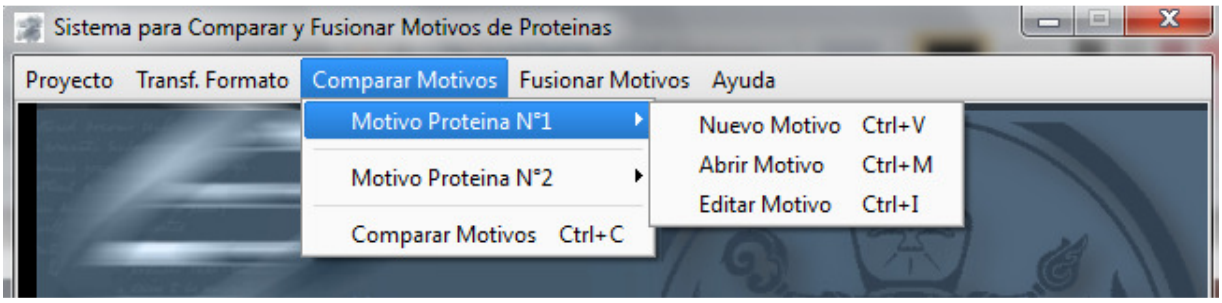


Figura A.7 Opciones del sub-menú Motivo Proteína Nº1

2. Motivo Proteína Nº 2: Permite escoger el motivo a comparar. Este menú contiene los mismos elementos descritos en el punto anterior.

3. Comparar Motivos: permite comenzar proceso de comparación de los motivos (ver figura A.8).

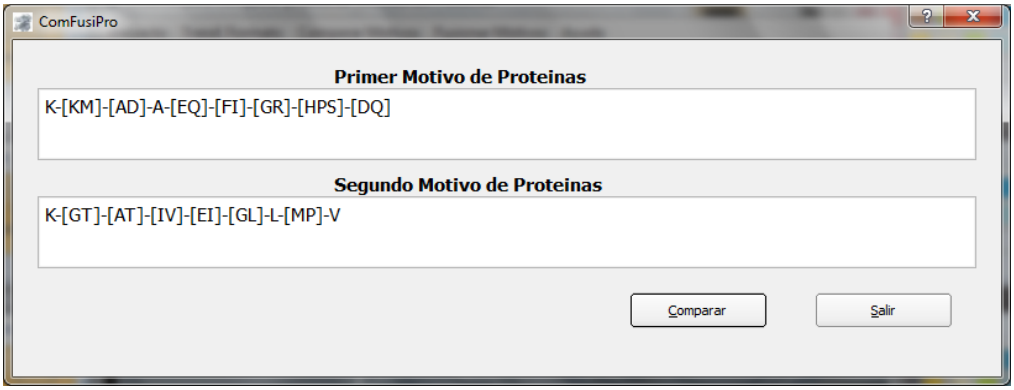


Figura A.8 Ventana donde se muestran los motivos a comparar

Al presionar el botón "Comparar" en la figura A.8 el sistema muestra el motivo utilizado como objeto de estudio y las secuencias que se pueden construir con el motivo Nº 1 (ver figura A.9). Al presionar el botón "salir" regresa a la ventana principal del sistema (ver figura A.1)

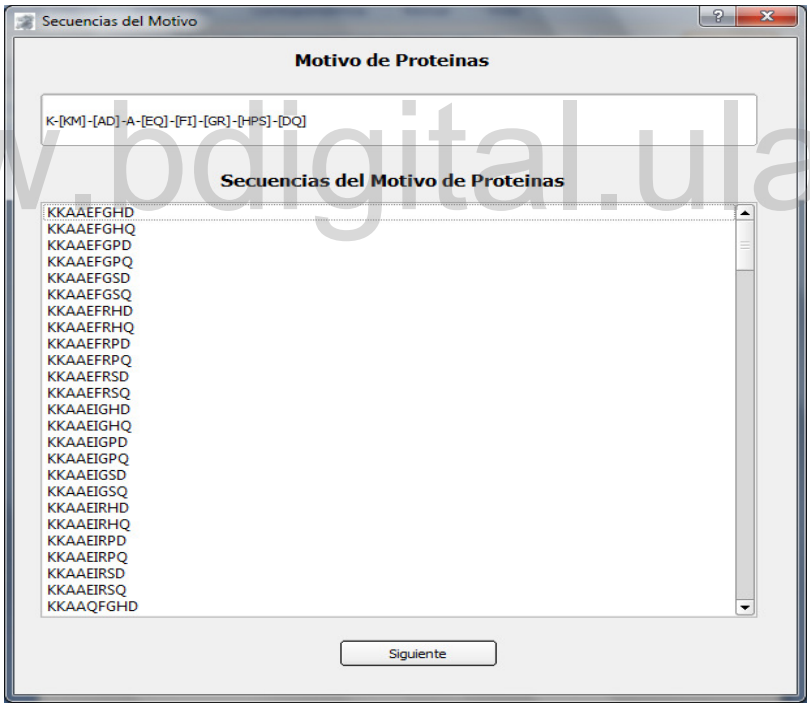


Figura A.9 Ventana donde se muestra el motivo seleccionado como objeto de estudio y el conjunto de secuencias que se pueden construir a partir de él.

Al presionar el botón "siguiente" en la figura A.9 el sistema se dirige a la ventana para seleccionar el tamaño de la muestra del motivo a aprender (ver figura A.10). Con el botón "abrir" se puede abrir un archivo donde se encuentran los valores del error estándar y la

fiabilidad deseada, seleccionar un valor específico para estos parámetros directamente y con el botón “guardar” se puede almacenar estos valores en un archivo.

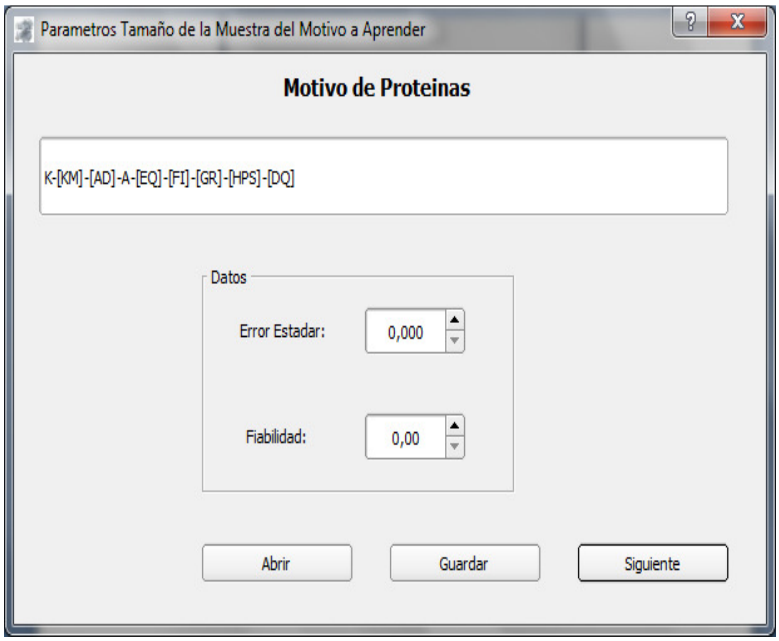


Figura A.10 Ventana para los parámetros para el tamaño de la muestra del motivo a aprender.

Al presionar el botón “siguiente” en la figura A.10 el sistema se dirige a la ventana donde se muestra el tamaño de la muestra obtenido (ver figura A.11) utilizando los parámetros seleccionados en la ventana anterior, si el valor no es el esperado por el usuario podemos presionar el botón “atrás” para ir a la ventana anterior.

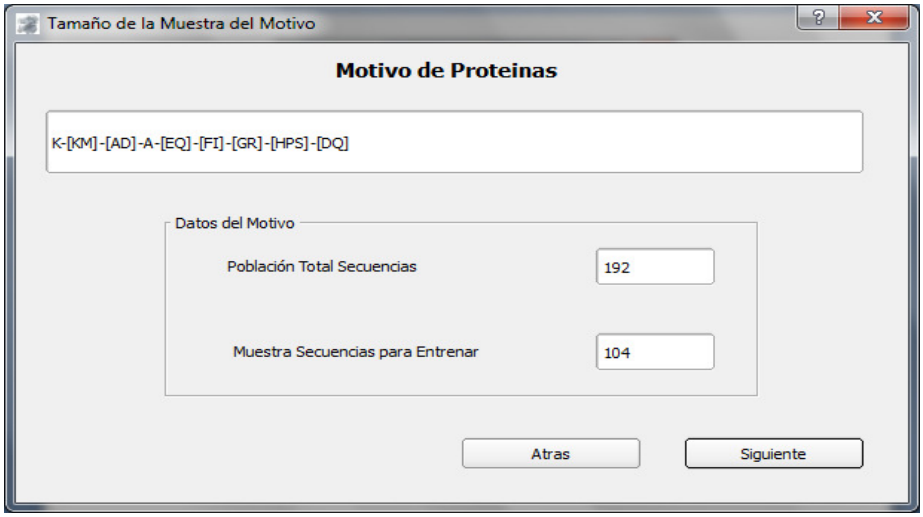


Figura A.11Ventana del tamaño de la muestra del motivo.

Al presionar el botón “siguiente” en la figura A.11 el sistema se dirige a la ventana donde se seleccionan los valores de los parámetros utilizados para el aprendizaje de la Red Neuronal: (ver figura A.12) tasa de aprendizaje, el factor de momento, el error de la red neuronal y el número de iteraciones. Con el botón “abrir” se puede abrir un archivo donde se encuentran los valores de parámetros anteriores, se puede seleccionar un valor específico para estos parámetros directamente y con el botón “guardar” se puede almacenar estos valores en un archivo.

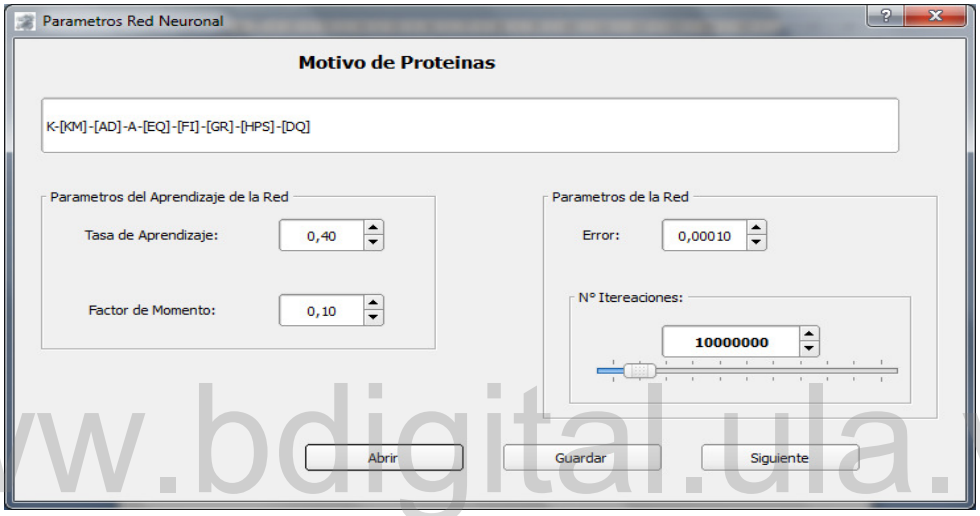


Figura A.12 Ventana para los parámetros de la red neuronal para el aprendizaje del motivo.

Al presionar el botón “siguiente” en la figura A.12 el sistema se dirige a la ventana donde se muestra el proceso de aprendizaje del motivo por la Red Neuronal (ver figura A.13), se observa el motivo utilizado, todos los parámetros seleccionados anteriormente y el valor de error en cada iteración realizada por la red neuronal hasta alcanzar el error dado por el usuario. Con el botón “cancelar” se cancela el proceso de aprendizaje de la red, con el botón “reiniciar” se reinicia el proceso de aprendizaje, con el botón “atrás” se regresa a la ventana que se muestra en la figura A.10, con el botón “guardar” se almacena en un archivo todos los datos obtenidos de la red neuronal en el aprendizaje del motivo.

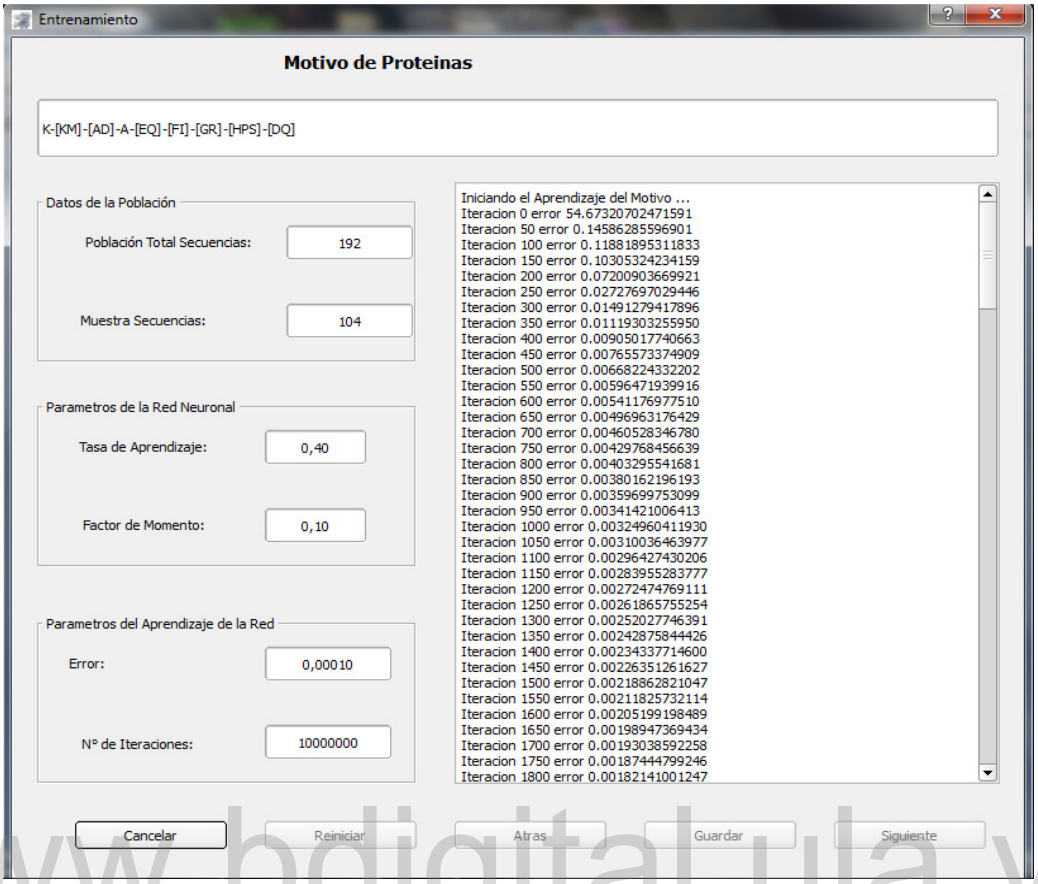


Figura A.13 Ventana de aprendizaje del motivo por la red neuronal.

Al presionar el botón “siguiente” en la figura A.13 el sistema se dirige a la ventana donde se muestra los parámetros para la similitud: (ver figura A.14) número de generaciones, número de individuos y el valor de los índices de similitud de los aminoácidos. Con el botón “abrir” se puede abrir un archivo donde se encuentran los valores de parámetros anteriores, se puede seleccionar un valor específico para estos parámetros directamente y con el botón “guardar” se puede almacenar estos valores en un archivo.

Figura A.14 Ventana para los parámetros de similitud.

Al presionar el botón "siguiente" en la figura A.14 el sistema se dirige a la ventana donde se muestra el proceso de aprendizaje del motivo por la Red Neuronal (ver figura A.15), se observa el motivo utilizado como objeto de estudio, el motivo a comparar, los parámetros de similitud, y los valores de similitud cuando se realiza la comparación de los individuos del motivo, los valores de similitud de cada generación y la similitud total. Con el botón "cancelar" se cancela el proceso de comparación, con el botón "reiniciar" se reinicia el proceso comparación, con el botón "atrás" se regresa a la ventana que se muestra en la figura A.14, con el botón "guardar" se almacena en un archivo los datos de la comparación.

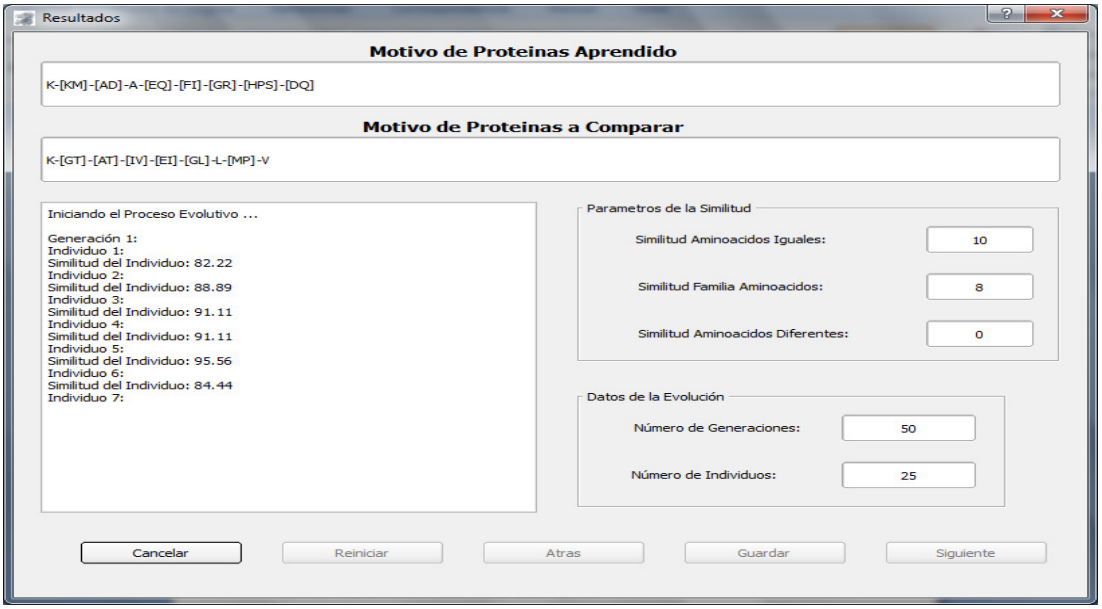


Figura A.15 Ventana de comparación de los motivos.

Al presionar el botón “siguiente” en la figura A.15 el sistema se dirige a la ventana de resultados donde se muestra los motivos utilizados y el valor de similitud entre ambos (ver figura A.16).

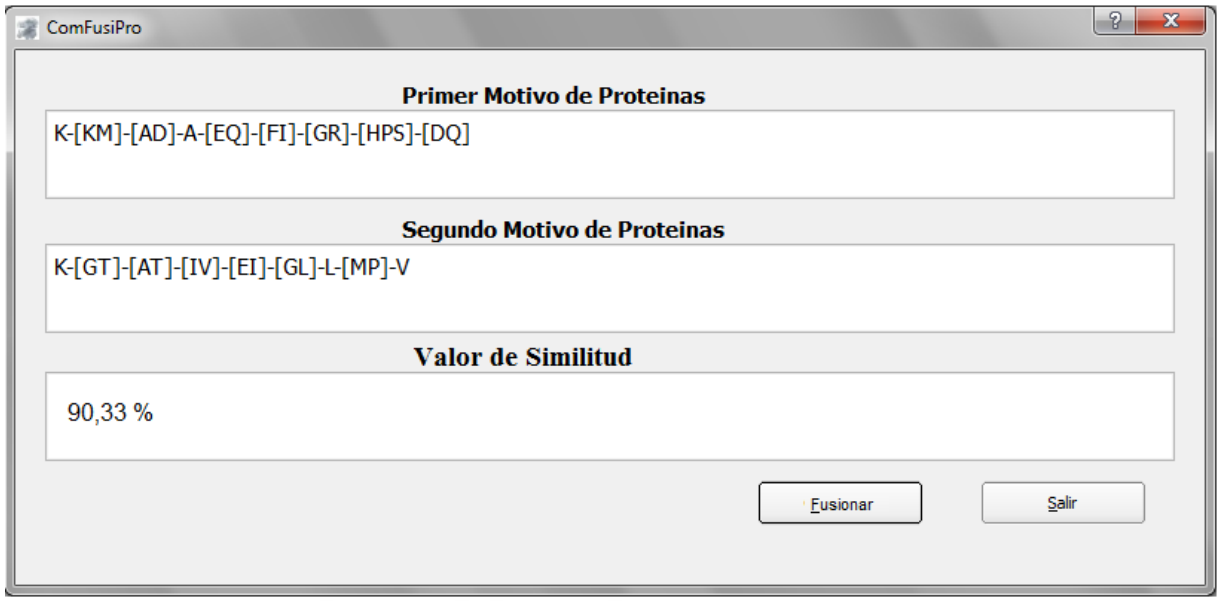


Figura A.16 Ventana de resultados de la comparación de los motivos.

Al presionar el botón “Fusionar” en la figura A.16 el sistema se dirige al fusionar los motivos para ello muestra los parámetros para la fusión: (ver figura A.17) población de hormigas, ciclos de la colonia, índice de similitud, ajuste de los agentes, ajustes de la feromona. Se tienen cuatro botones que permitirán, “verParametros” le permite al usuario ver los parámetros guardados en un archivo, igualmente podremos reiniciar los parámetros con el botón de “limpiar”, si se desea salvar los parámetros suministrados por el usuario se presiona el botón “guardar”, el botón “cargar” permite suministrar al sistema los parámetros seleccionados para la fusión.

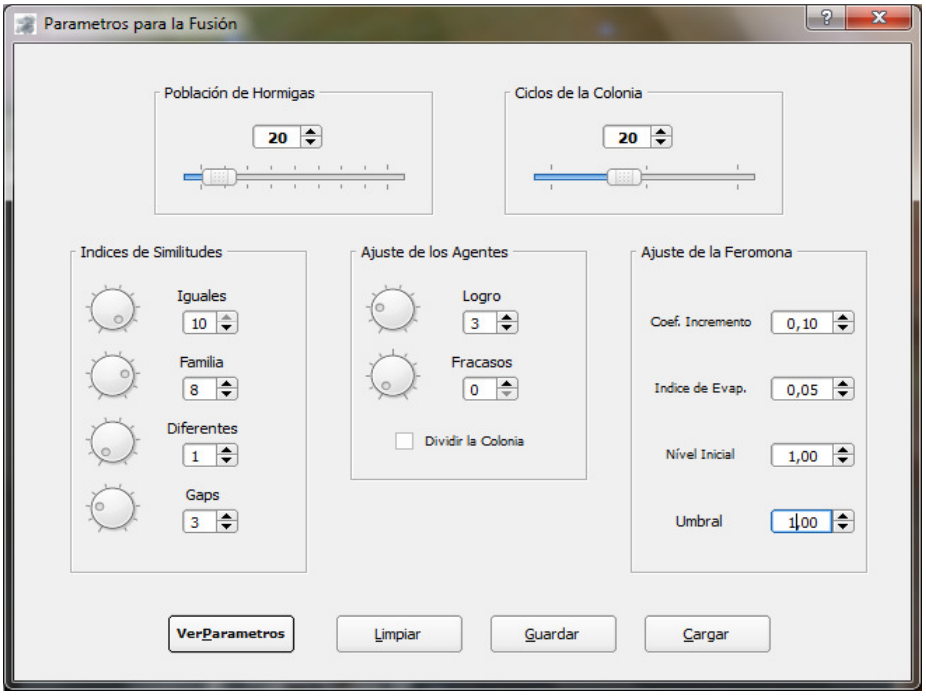


Figura A.17 Ventana de parámetros para la fusión.

Luego se realiza la fusión de los motivos (ver figura A.18).

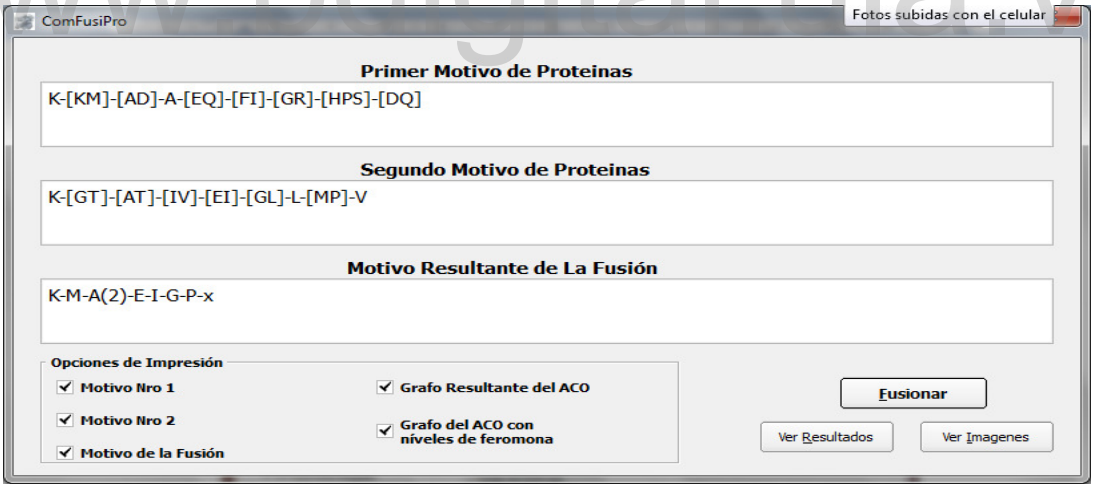


Figura A.18 Ventana para realizar la fusión de los motivos.

En esta ventana se muestra los motivos seleccionados, el motivo resultante de la fusión, además de cinco opciones de chequeo donde se podrán elegir los tipos de grafos y patrones que serán representados gráficamente, y al final se dispone de tres botones; "FUSIONAR**"** es el que inicia la fusión de los motivos, una vez se haya obtenido el motivo resultante se podrán acceder a los restantes dos botones:

1. Ver Resultados: muestra una ventana con la información de la fusión realizada (ver figura A.19), donde se puede observar los motivos seleccionados, el motivo resultante de la fusión y los parámetros seleccionados.

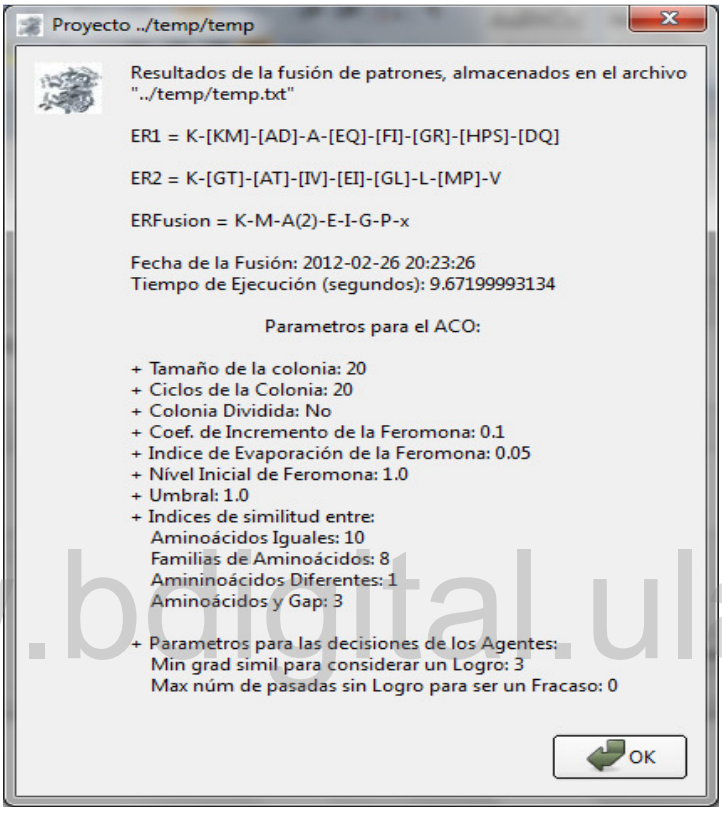


Figura A.19 Ventana de resultados de la fusión de motivos.

2. Ver Imágenes**:** este botón enlaza directamente a un visor de imágenes en formato '.png' (ver figura A.20), y que permitirá al usuario observar gráficamente los grafos y motivos seleccionados en la pantalla anterior, dentro de este visor se escoge el archivo de la imagen que se desee abrir, también se puede maximizar o reducir la imagen utilizando el menú disponible en el visor.

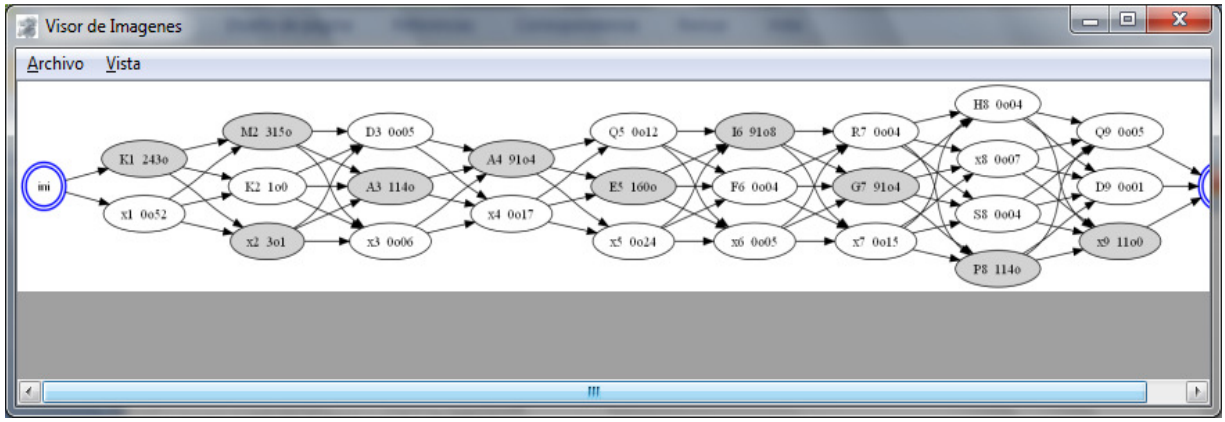


Figura A.20 Ventana del visor de imágenes.

De vuelta al menú principal se localiza el menú "Fusionar Motivos" (ver figura A.21). Éste permite realizar solo la fusión de motivos, sin realizar el proceso de comparación.

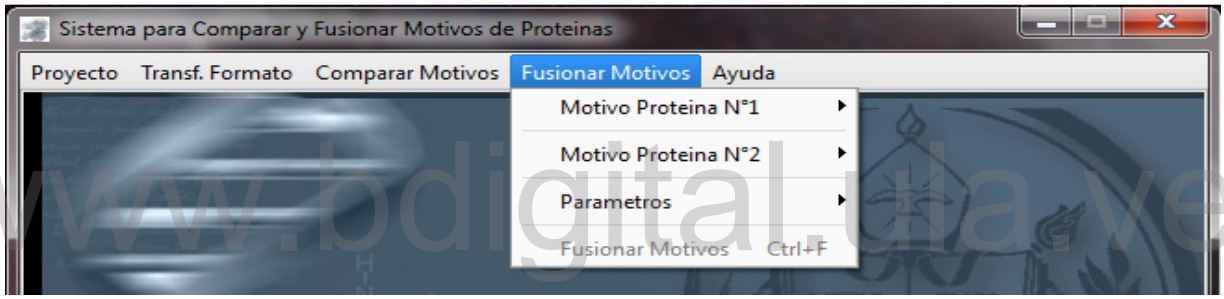


Figura A.21 Opciones del menú Fusionar Motivos

Este menú está dividido en 3 sub-menús:

1. Motivo Proteína Nº 1: contiene las mismas características de su homologo en el menú de Comparar motivos (ver figura A.7).

2. Motivo Proteína Nº 2: contiene las mismas características de su homologo en el menú de Comparar Motivos.

3. Parámetros: Representa los parámetros utilizados para la fusión descrito anteriormente (ver figura A.17).

4. Fusionar Motivos: permite fusionar motivos, este proceso fue descrito anteriormente (ver figura A.18)

APENDICE B: IMPLANTACION DEL SISTEMA

B.1. DESCRIPCION DEL LENGUAJE UTILIZADO

En la codificación del sistema desarrollado se utiliza el lenguaje PYTHON. Esta elección se baso principalmente en las bondades establecidas por este lenguaje para el tratamiento de la memoria dinámica y de la programación paralela (necesidad que surge debido a que se debe optimizar el tiempo de ejecución). Adicionalmente, el lenguaje Python cumple con las normas GNU y le da al sistema desarrollado la capacidad de ser ejecutado en varias plataformas.

Por otro lado, para darle una presentación más limpia y acorde con los software de la actualidad se decidió dotar al sistema de una interfaz de usuario gráfica, que agilice la manipulación del mismo por parte de los usuarios finales, razón por la cual se utilizo el framework PyQt4, que así como Python permite la adaptabilidad del sistema a diferentes arquitecturas.

B.2. CASOS DE USO DEL SISTEMA

El sistema contiene los siguientes actores (ver figura B.1):

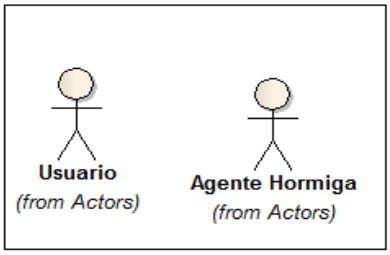


Figura B.1 Actores del Sistema.

A continuación se presentan los casos de uso del sistema. El caso de uso de la figura B.2 muestra que puede hacer el usuario en el sistema: operar proyectos, transformar formato, comparar motivos y fusionar motivos.

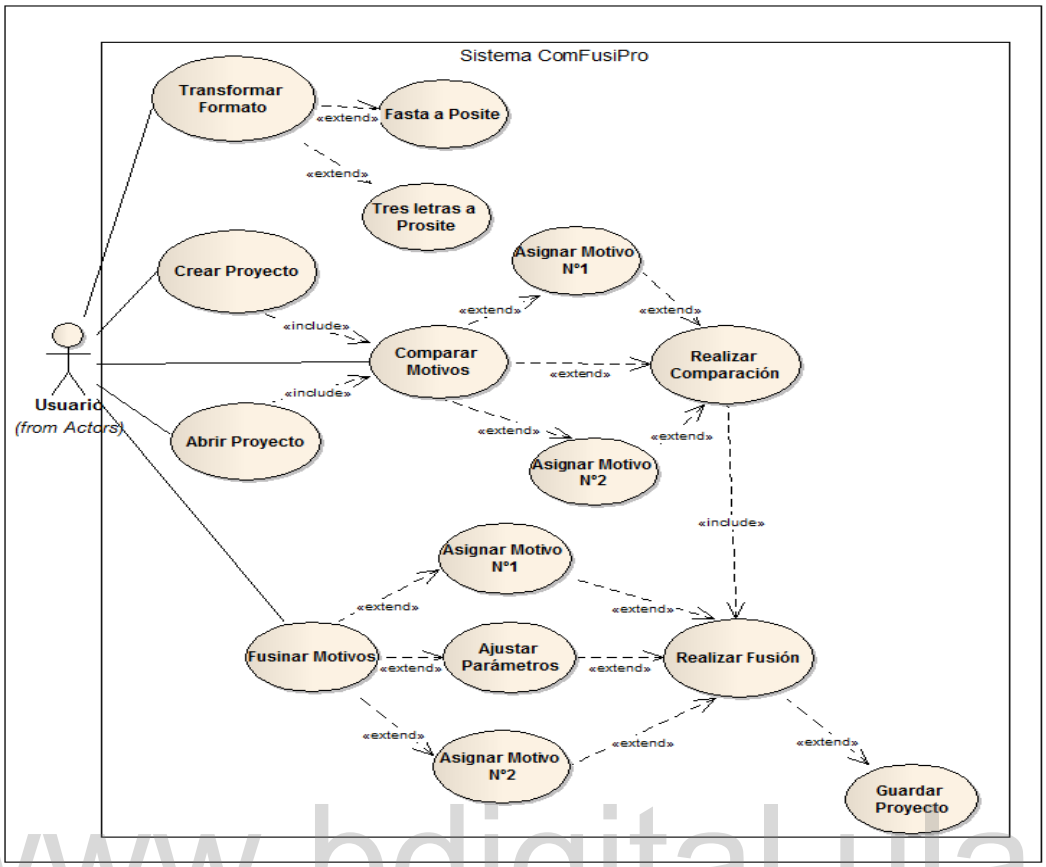


Figura B.2 Inicio del Sistema

La figura B.3 muestra el caso de uso donde se establecen los parámetros del tamaño de la muestra del motivo a comparar.

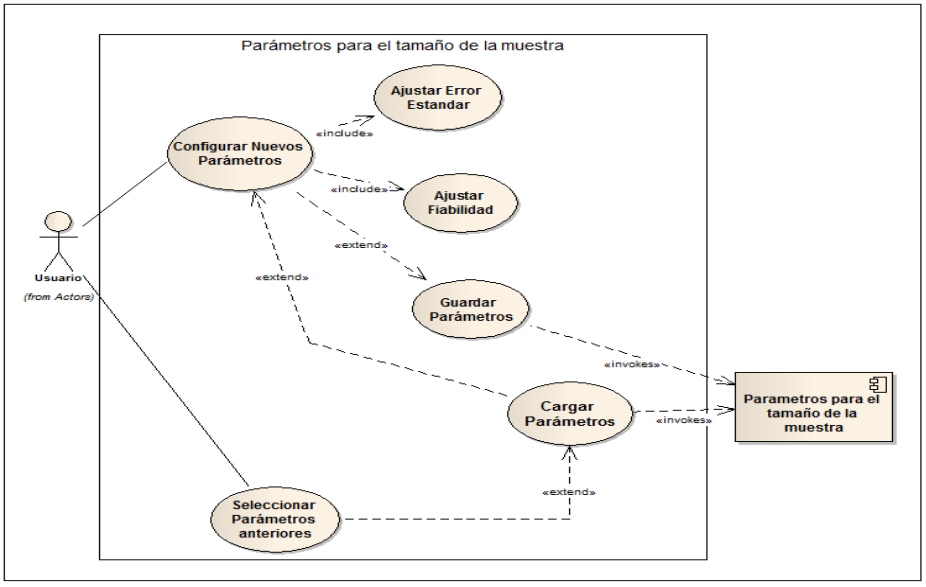


Figura B.3 Ajuste de los Parámetros para el tamaño de la muestra.

La figura B.4 muestra el caso donde se establecen los parámetros de la red neuronal.

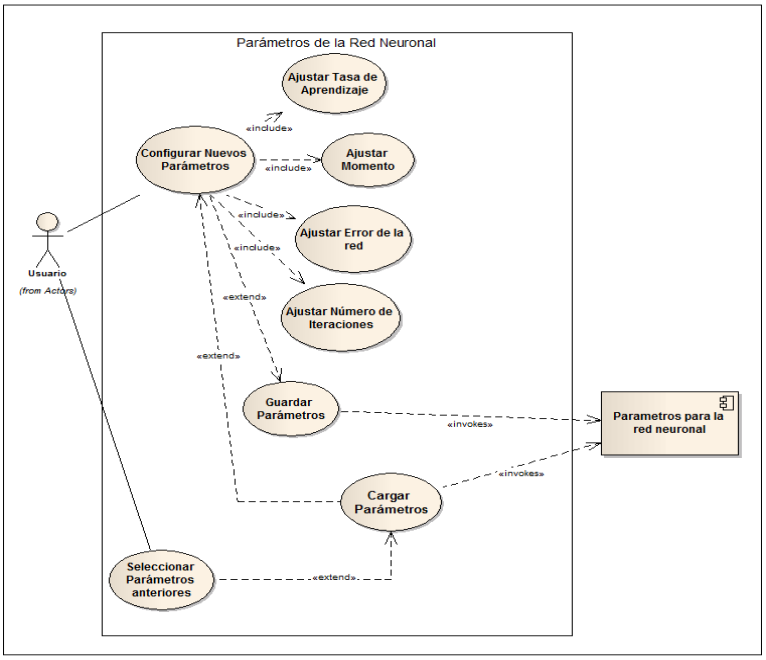


Figura B.4 Ajuste de los Parámetros para la red neuronal.

La figura B.5 muestra el caso de uso donde se establecen los parámetros de similitud.

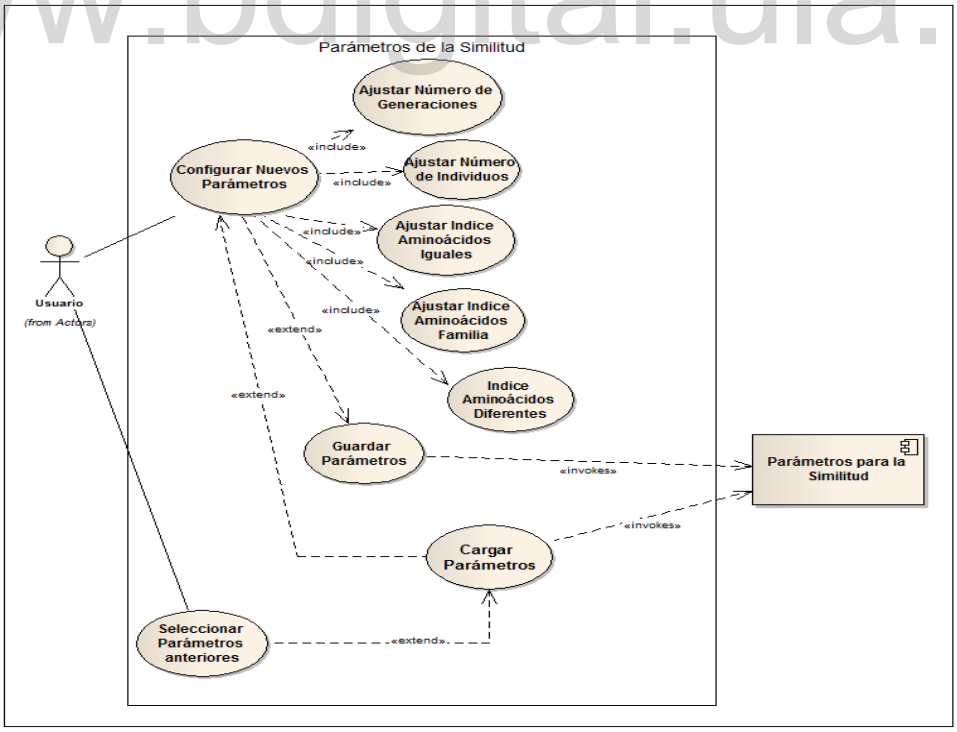


Figura B.5 Ajuste de los Parámetros para la Similitud.

La figura B.6 muestra el caso de uso donde se establecen los parámetros de similitud.

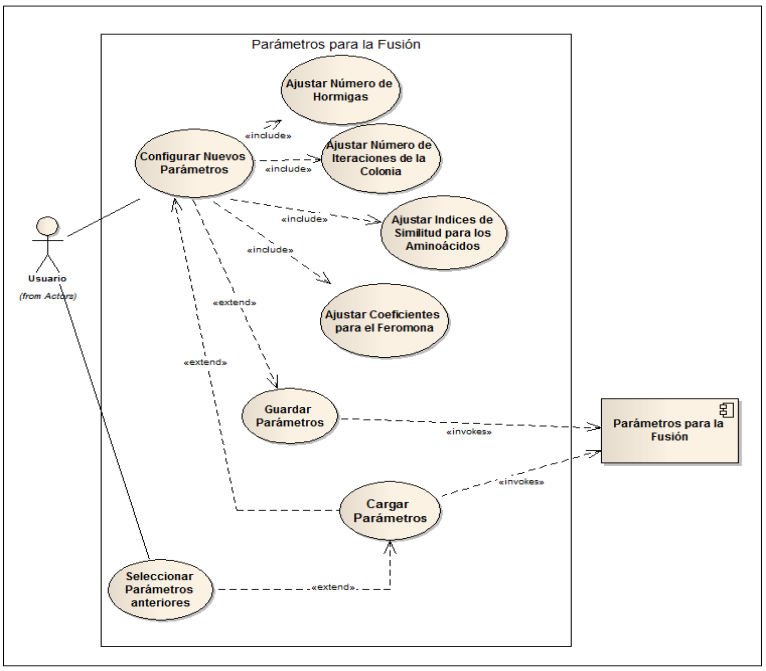


Figura B.6 Ajuste de los Parámetros para la Fusión.

La figura B.7 muestra el caso de uso para manipular un motivo.

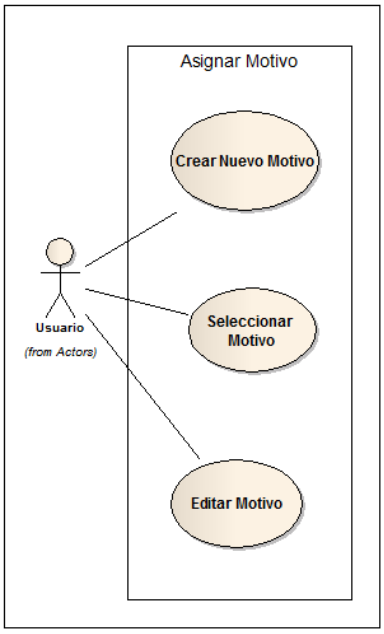


Figura B.7 Asignar un Motivo.

La figura B.8 muestra el caso de uso para realizar la comparación.

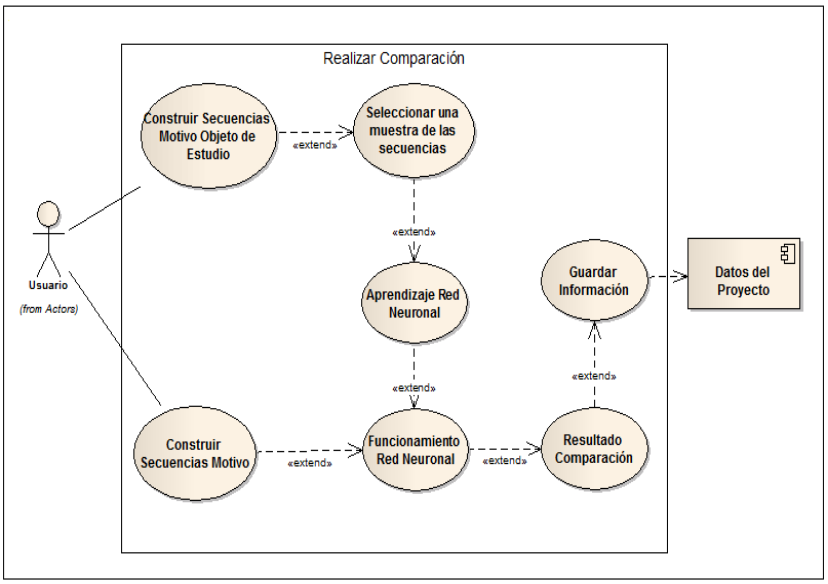


Figura B.8 Realizar Comparación.

La figura B.9 muestra el caso de uso para realizar la fusión.

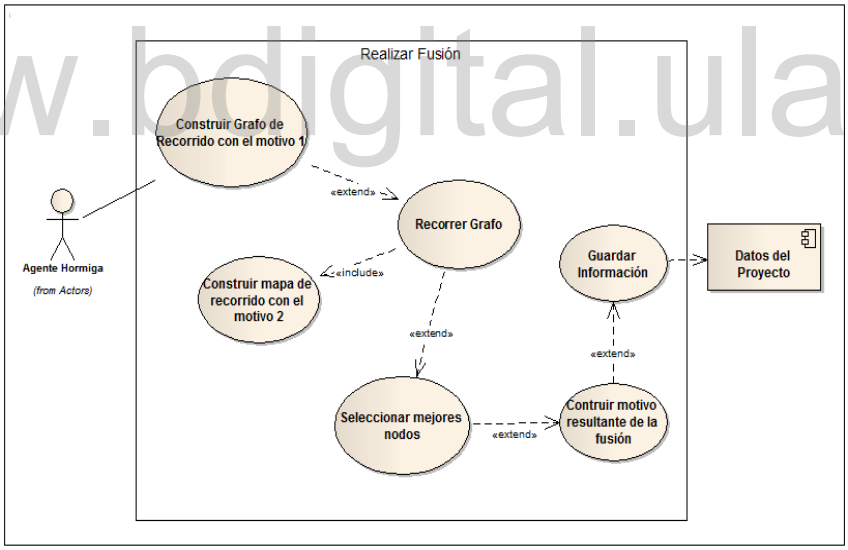


Figura B.9 Realizar Fusión.

B.3. DIAGRAMA DE CLASES

Para la implementación del Sub-Sistema de Comparación de Motivos se utilizaron tres clases Gramática, PG, RNB (ver figura B.10). Así, la clase Gramática contiene la información gramatical que permite construir las secuencias validas de los motivos utilizados. La clase PG contiene la información sobre la Programación Genética que permite generar el conjunto de individuos del motivo objeto de estudio, calcular la muestra de la población de individuos del motivo objeto de estudio utilizados para el aprendizaje de la red neuronal, crear la población de individuos del motivo a comparar utilizados en cada una de las generaciones del proceso evolutivo, y asignar un valor de similitud entre los motivos de cada generación y al finalizar el proceso. La clase RNB contiene la información sobre la Red Neuronal de Retropropagación para realizar el aprendizaje de un conjunto de secuencias del motivo objeto de estudio, y realizar la comparación entre secuencias y asignar un valor de similitud (reconocimiento).

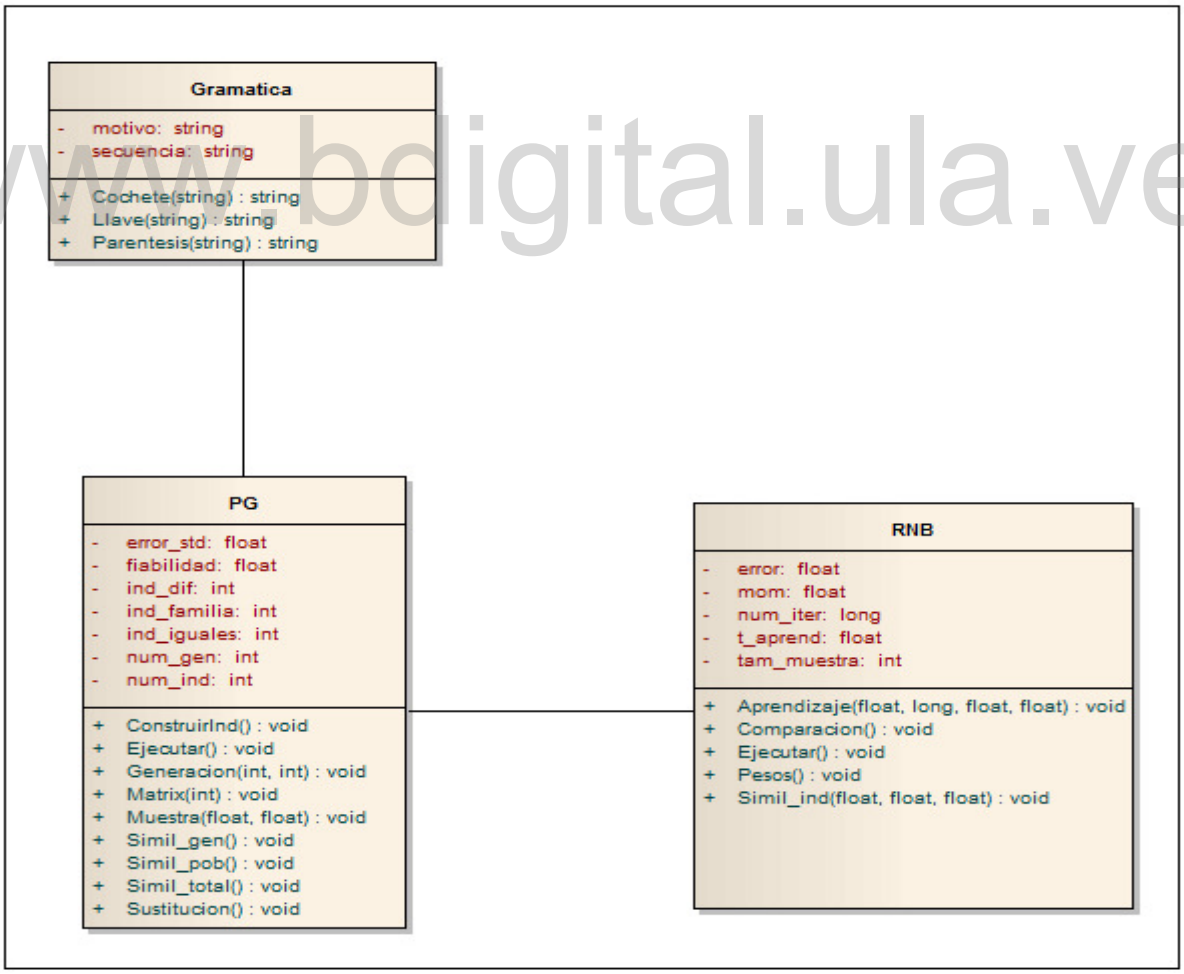


Figura B.10 Diagrama de Clases Sub-Sistema del Comparación de Motivos.

Para la implementación del sub-sistema de fusión de motivos se recurrió al uso de dos clases, Aminoácido y AgentAnt (ver figura B.11), las cuales se ocupan del manejo de la información contenida en el grafo de recorrido y en los Agentes de las colonias, respectivamente. Así, la clase Aminoácido contiene la información que se almacena en los nodos del grafo (nombre del aminoácido, familia a la que pertenece el aminoácido, y nivel de feromona), y la clase AgentAnt se ocupa de manipular la información que requieren las hormigas para sus recorridos (nivel de similitud aprobatoria, grafo de recorrido, mapa de recorrido, índices de similitud, índice de evaporación e índice de incremento de feromona). Adicionalmente, se desarrollo un grupo de bibliotecas que contienen una serie de funciones y métodos públicos:

1. FusionER.py: contiene funciones necesarias para llevar a cabo el recorrido y la construcción del grafo de recorrido.

2. Nodos.py: es una biblioteca que provee al sistema de métodos necesarios para la manipulación de los nodos del grafo de recorrido (Copiar nodo, construir cadenas a partir de listas, Invertir listas, etc.)

3. ImprimeGraf.py: le provee al sistema las funciones necesarias para la representación grafica de los patrones utilizados y generados por el sistema.

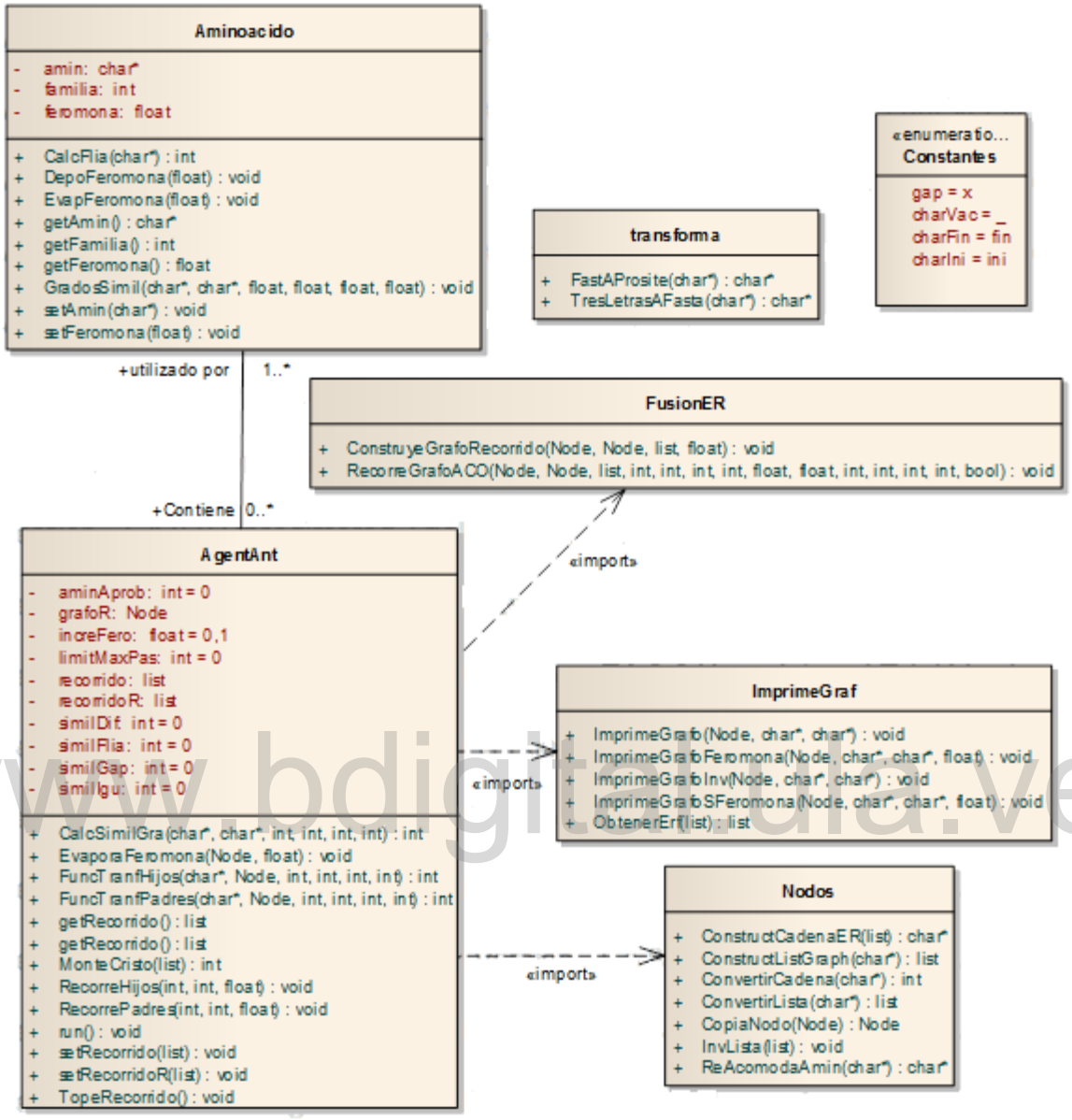


Figura B.11 Diagrama de Clases para el Sub-Sistema de Fusión de Motivos.

B.4. DIAGRAMA DE COMPONENTES SEGÚN SU ARQUITECTURA

Se escogió un diseño arquitectónico por capas para la implantación del sistema (ver figura B.12).

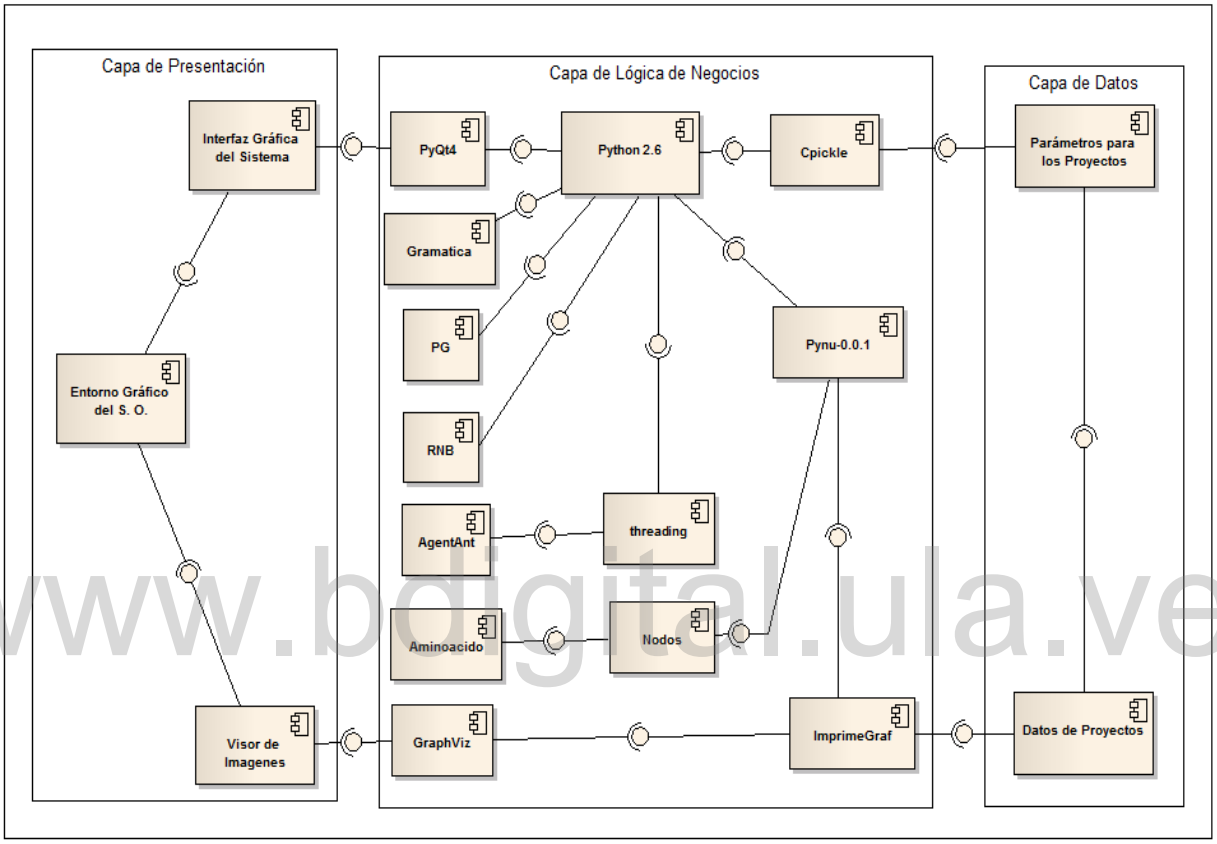


Figura B.12 Diagrama de Componentes.

De esta manera, tenemos:

1. La capa de presentación se compone de la interfaz grafica, que permite la interacción con el usuario, y del visor de imágenes, ambos desarrollados con el framework PyQt4.

2. En la capa del modelo del negocio se encuentran las clases Gramática, PG, RNB, Aminoácido y AgentAnt, así como el resto de paquetes que proveen las funciones necesarias para la ejecución del algoritmo: Python 2.6, Pynu-0.1.1 y threading (estas dos últimas facilitan el manejo de los nodos del grafo y los hilos de ejecución de los agentes hormiga, respectivamente).

3. Por último se encuentra la capa de datos, la cual está constituida por una serie de archivos de registros que hicieron innecesaria la utilización de una base de datos, debido a la simplicidad de los datos utilizados y a la funcionalidad del paquete Cpickle, el cual viene incorporado en el lenguaje Python y contribuye al manejo de los archivos de registros.

   Los directorios donde se guardan los archivos de nuestra herramienta se muestran en la figura B.13, y lo que contiene los archivos en la tabla B.1. La carpeta *aprender* contiene las sub-carpetas donde se almacenas los archivos del aprendizaje de un motivo (estos son: ah, ai, ao, ni, wh, wi, wo). La carpeta *ComFusiPro* contiene todos los archivos del Sistema de Comparación y Fusion de motivos de proteínas (éstos están almacenados en tres sub-carpetas, que son: formularios, herramientas e imágenes. La sub-carpeta *formularios* contiene los archivos python para la ejecución del sistema y para las distintas ventanas que se muestran, la sub-carpeta *herramientas* contiene los archivos de las clases creadas (Gramática, PG, RNB, Aminoácido y AgentAnt), la sub-carpeta *imágenes* contiene el archivo de la imagen que se muestra en el fondo de la ventana principal del sistema). La carpeta *parámetros* contiene los archivos donde se guardan los parámetros utilizados en la fusión de los motivos (tienen la extensión .p y son creados por los usuarios). La carpeta *pared* contiene los archivos donde se guardan los parámetros utilizados para el aprendizaje de la red neuronal (tienen la extensión .pr y son creados por los usuarios). La carpeta *parmuestra* contiene los archivos donde se guardan los parámetros utilizados para calcular el tamaño de la muestra del motivo objeto de estudio (tienen la extensión .pam y son creados por los usuarios). La carpeta *parsimil* contiene los archivos donde se guardan los parámetros utilizados para la comparación de los motivos (tienen la extensión .pas y son creados por los usuarios). La carpeta *patrones* contiene los archivos donde se guardan los motivos de proteínas (tienen la extensión .txt y son creados por los usuarios). La carpeta *temp* contiene los archivos de las imágenes de los grafos resultantes de la fusión (ACO; ACOFeromona, ER1, ER2).

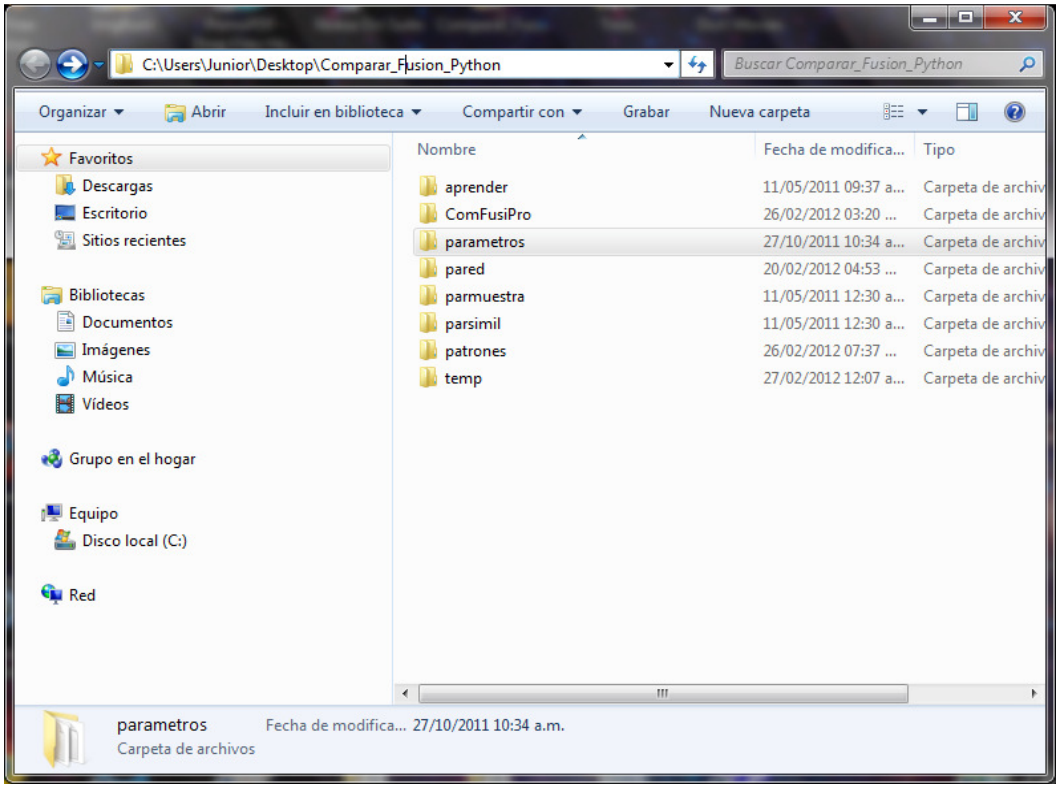


Figura B.13 Estructura de los directorios.

Los formatos utilizados son de dos tipos: imagen y texto. En la tabla B.1 se describen cada uno de archivos y sus respectivas extensiones.

Nombre	Descripción	Formato		Extensión
		Campo	tipo	
ACO	Contiene el grafo resultante de la fusión	Imagen		.png
ACOFeromona	Contiene el grafo resultante de la fusión con los rastros de feromona en cada nodo	Imagen		.png
ER1	Contiene el grafo de la expresión regular 1 (motivo nro. 1)	Imagen		.png
ER2	Contiene el grafo de la expresión regular 2 (motivo nro. 2)	Imagen		.png
Parámetros de la red	Parámetros utilizados para el aprendizaje de la red neuronal	Tasa aprendizaje	float	.pr
		momento	float	
		error	float	
		Número iteraciones	long	
Parámetros de la muestra	Parámetros utilizados para calcular el tamaño de la muestra	Error estandar	float	.pam
		Fiabilidad	float	
Parámetros de la similitud	Parámetros utilizados para realizar la similitud entre los motivos	Número de Individuos	int	.pas
		Número de Generaciones	int	
		Índice igualdades	int	
		Índice familia	int	
		Índice diferencias	int	
Parámetros	Parámetros para ser utilizados en la fusión	Población de Hormigas	int	.p
		Ciclos Colonia	int	
		Índice igualdades	Int	
		Índice familia	Int	
		Índice diferencias	Int	
		Índice Gaps	int	

		Similitud Aprobatoria	int	
		Fracaso	int	
		Incremento feromona	float	
		Índice de Evaporación	float	
		Nivel inicial	float	
		Umbral	float	
Patrón	Contiene un motivo	Motivo	string	.txt
ah	Contiene los valores de la neuronas de la capa oculta de la red neuronal después del proceso de aprendizaje	Valor de las Neuronas	Lista: float	.txt
ai	Contiene los valores de la neuronas de la capa de entrada de la red neuronal después del proceso de aprendizaje	Valor de las Neuronas	Lista: float	.txt
ao	Contiene los valores de la neuronas de la capa de salida de la red neuronal después del proceso de aprendizaje	Valor de las Neuronas	Lista: float	.txt
ni	Número de neuronas de la capa de entrada de la red neuronal	Número de neuronas	int	.txt
wh	Contiene los valores de los pesos de la capa oculta de la red neuronal después del proceso de aprendizaje	Pesos	Lista: float	.txt
wi	Contiene los valores de los pesos de la capa de entrada de la red neuronal después del proceso de aprendizaje	Pesos	Lista: float	.txt
wo	Contiene los valores de los pesos de la capa de salida de la red neuronal después del proceso de aprendizaje	Pesos	Lista: float	.txt

Tabla B.1 Formato y contenido de los archivos.