Universidad Central "Marta Abreu" de Las Villas

Facultad Química-Farmacia

Laboratorio CAMD-BIR

&

Facultad Matemática-Física-Computación

Centro de Estudios de Informática

En opción del título:

Máster en Computación Aplicada

Tema:

Comparación de Metaheurísticas aplicadas a la predicción de la estructura terciaria de proteínas

Autor:

Lic. Elizabeth Martínez Pérez (elizabethmp@uclv.cu)

Tutores:

Dr. Yasser B. Ruiz Blanco (yasserrb@uclv.edu.cu)

Dr. Gladys Casas Cardoso

2016

ANEXO ACTA DE CONFORMIDAD PARA ESTUDIANTES DE POSTGRADO

Universidad Central "Marta Abreu" de Las Villas

ACTA DE CONFORMIDAD

Por una parte:

Elizabeth Martínez Pérez estudiante de la maestría en: Computación Aplicada en la facultad de Matemática-Física-Computación, en lo adelante **El ESTUDIANTE.** Con número de identidad permanente: 90053132272.

Y por otra parte:

Carlos Morell Pérez Jefe del Laboratorio de: Inteligencia Artificial y Bioinformática en la ya mencionada facultad, en lo adelante **EL JEFE DEL LABORATORIO,** y Yasser B. Ruiz Blanco y Gladys Casas Cardoso profesores encargados de tutorar el Trabajo de Maestría **DEL ESTUDIANTE**, en lo adelante **EL TUTOR.**

Reconocen que:

I. A **EL ESTUDIANTE** se le ha aprobado como tema de investigación para su Trabajo de Diploma el titulado: Comparación de Metaheurísticas aplicadas a la predicción de la estructura terciaria de proteínas.

II. **EL ESTUDIANTE** no divulgará información concerniente a la investigación, tanto durante el desarrollo como tras la culminación de esta sin la debida autorización **DEL TUTOR** o **EL JEFE DE DEPARTAMENTO**.

III. Que el Trabajo de Maestría fruto de la labor investigativa **DEL ESTUDIANTE** y la asesoría **DEL TUTOR**, resulta de **TITULARIDAD EXCLUSIVA** de la Universidad Central “Marta Abreu” de las Villas.

IV. **El ESTUDIANTE** una vez aprobada su tesis para la defensa, depositará una copia electrónica de la misma en el Repositorio Digital Institucional de la Universidad Central “Marta Abreu” de las Villas.

V. A partir de la defensa y aprobación del Trabajo de Maestría, la publicación total, parcial o la elaboración de cualquier obra que se derive de esta investigación por parte **DEL ESTUDIANTE**, contará con la coautoría **DEL TUTOR** y viceversa, resultando de referencia obligada esta obra en cualquier otra que se elabore. El incumplimiento de esta cláusula, puede llevar consigo el inicio de procesos de plagio. Todo lo anterior de acuerdo a la normativa de Derecho de Autor vigente en Cuba.

Y para que así conste se firma la presente en la Universidad Central “Marta Abreu” de Las Villas, a los _____días del mes de _____del año 20____.

_____	_____
Lic. Elizabeth Martínez Pérez	Dr. Carlos Morell Pérez
Estudiante	J´ del Laboratorio
_____	_____
Dr. Yasser B. Ruiz Blanco	Dr. Gladys Casas Cardoso
Tutor	Tutora

PUBLICACIONES

E. Martínez-Pérez, J. A. Castillo-Garit & Y. B. Ruiz-Blanco. 2015. Big-Datasets Manager: Una herramienta libre para la manipulación de ficheros de datos con número elevado de instancias y atributos. Nereis. Vol 7. Pages 59-66. Valencia. ISBN 1888-8550.

Elizabeth Martínez Pérez, Yasser B. Ruiz Blanco y Ernesto Contreras Torres. 2015. Algoritmos Genéticos para Predecir la Estructura Terciaria de Proteínas. XIII Congreso Nacional de Reconocimiento de Patrones (RECPAT2015). ISBN: 978-959-207-540-5

Fernando Wong Delgado, **Elizabeth Martínez Pérez** y Yasser B. Ruiz Blanco. 2015. Algoritmo Recocido Simulado aplicado a la Predicción de Estructura Terciaria de Proteínas. XIII Congreso Nacional de Reconocimiento de Patrones (RECPAT2015). ISBN: 978-959-207-540-5

EVENTOS

Ponencia Oral: Algoritmos Genéticos para Predecir la Estructura Terciaria de Proteínas. XIII Congreso Nacional de Reconocimiento de Patrones. Santiago de Cuba. Octubre 2015. **Elizabeth Martínez Pérez**.

Ponencia Oral: Algoritmo Recocido Simulado aplicado a la Predicción de Estructura Terciaria de Proteínas. XIII Congreso Nacional de Reconocimiento de Patrones. Santiago de Cuba. Octubre 2015. **Elizabeth Martínez Pérez**.

Poster: 10th Seminars of Advance Studies on Molecular Desing and Bioinformatics (SEADIM). Habana-Varadero, Cuba. Junio 23 - 28, 2015. Título: Folding Proteins with Meta-heuristics. **Elizabeth Martínez Pérez**, Fernando Wong Delgado y Yasser B. Ruiz Blanco.

Poster: A hybrid meta-heuristic approach to search protein conformational space. Data Day: Big data. Carleton University. Ottawa, Canadá. Abril 24, 2014. Yasser B. Ruiz Blanco, Yovani Marrero-Ponce, **Elizabeth Martínez Pérez** & James Green.

TUTORÍA DE TESIS

Tesis de grado: Fernando Wong Delgado. Algoritmo Recocido Simulado aplicado a la Predicción de Estructura Terciaria de Proteínas. Julio 2015. Universidad Central “Marta Abreu” de Las Villas. Supervisores: Lic. **Elizabeth Martínez Pérez**, Lic. Yasser B. Ruiz-Blanco y Dr. Gladys Casas Cardoso.

REGISTRO DE SOFTWARE

Big-Dataset Manager (BDM). 2 de julio de 2015. **Elizabeth Martínez Pérez**, Juan Alberto Castillo Garit y Yasser Bruno Ruiz Blanco. Centro Nacional de Derecho de Autor (CENDA). Número de Registro: 2407-07-2015. Facultad de Química – Farmacia de la Universidad Central “Marta Abreu” de Las Villas.

Agradecimientos

Quisiera agradecer a:

Mi familia por el apoyo incondicional en el tiempo dedicado a los estudios y a las tantas horas de ausencia por trabajo investigativo.

Mis tutores Dr. Yasser Bruno Ruiz Blanco y a Dr. Gladys Casas Cardoso por sus seguimientos y revisiones de la tesis.

Los profesores de los que he aprendido tanto en los postgrados cursados.

A mis amigas Lic. Maricel Meneses Gómez e Ing. Sandra Romero Molina y compañeros de trabajo Dr. Juan Alberto Castillo Garit y MsC. Yudith Cañizares Carmenate por sus apoyos con la tesis.

Por las ayudas prestadas en la obtención de los cálculos presentadas en la tesis:

- Dr. Cristina Marino Buslje y a Lic. Franco Simonetti.
- Dr. Yovani Marrero Ponce, MsC. Jairo Enrique Serrano Castañeda y MsC. William Alejandro Caicedo Torres.
- Dr. Miriel Martín Mesa y su equipo de trabajo en función del Clúster de la Universidad (HPC).
- Dr. Reinaldo Molina Ruiz por su apoyo con el clúster del Centro de Bioactivos Químicos de la Universidad.

Resumen

La predicción experimental de la estructura terciaria de una proteína es una tarea compleja, que implica considerables costos en tiempo y tecnología, además el número de secuencias conocidas y almacenadas en bases de datos como Genbank ha aumentado constantemente durante los últimos 15 años sin un incremento similar en el número de estructuras 3D elucidadas. En consecuencia, la búsqueda de métodos computacionales de predicción estructural ha sido uno de los campos más activos en Biología y Química Computacional. Algunos de los métodos utilizados emplean metaheurísticas como algoritmos de optimización, sin embargo no se describen en la literatura estudios comparativos de su desempeño en una misma base de casos. Siendo por consiguiente el objetivo principal de la tesis: evaluar el desempeño de tres metaheurísticas: Optimización basada en Mallas Variables (VMO), Algoritmos Genéticos (GA) y Recocido Simulado (SA). Estos algoritmos fueron implementados en un software de fácil uso. Con una base de casos de 26 proteínas [100-300 aminoácidos], se comparan las metaheurísticas respecto a la calidad de las soluciones y desempeño computacional, donde GA es seleccionada como la mejor metaheurística. GA respecto a predictores internacionales, en promedio, logra ubicarse en por encima del 50% de los predictores en el 33.66% de las proteínas. La mejor solución de GA logra ubicarse en por encima del 50% de las predicciones en el 70% de las proteínas. GA permite obtener soluciones, que en promedio, superan el 50% de las predicciones de contemporáneos en la métrica GDT (calculada en el servidor LGA).

Abstract

Experimental prediction of the tertiary structure of a protein is a complex task, which involves considerable costs in time and technology, besides the number of known sequences in databases like Genbank has grown increasingly over the past fifteen years, without the same increment in elucidated 3D structures. Hence the search for computational structural prediction methods has been one of the most active fields in Biology and Computational Chemistry. Some methods have used metaheuristics as optimization algorithms; however it has not been described in the literature any comparative studies of their performances in a given data set. It is therefore the objective of this thesis: to evaluate the performance of three metaheuristics: Variable Mesh Optimization (VMO), Genetic Algorithms (GA) and Simulated Annealing (SA). These algorithms were implemented in single user-friendly software. A data set of 26 proteins [with 100 to 300 amino acids each] was selected for conducting the experiments. The metaheuristics were assessed according the quality of the solutions and computational performance, the results show that GA compares favorably with the other two algorithms. The GA was ranked with international predictors resulting, on average, over 50% of them in 33.66% of the proteins. The best solution of GA was positioned over 50% of all the predictions in 70% of proteins. GA allows obtaining solutions, on average, over 50% of the solutions of contemporary predictors, according GDT metric (calculated on LGA server).

Tabla de Contenidos

Índice de Tablas

Índice de Figuras

Índice de Gráficos

Índice de Ecuaciones

I. Introducción

Las proteínas son esenciales para la biología molecular y juegan un rol vital en casi todos los procesos biológicos. Las proteínas son las responsables de la formación y reparación de los tejidos, e intervienen en el desarrollo corporal e intelectual. La organización de una proteína viene definida por cuatro niveles estructurales denominados: estructura primaria, secundaria, terciaria y cuaternaria. Cada una de estas estructuras informa de la disposición de la anterior en el espacio. Esta investigación sólo tendrá en cuenta la estructura terciaria de las proteínas. La estructura terciaria de las proteínas determina su funcionamiento (1-4). Además se conoce que proteínas cuyo plegamiento dio lugar a estructuras anómalas son causa de enfermedades, como la fibrosis quística, Alzheimer y la enfermedad de Creutzfeldt-Jakob o su variante en bovinos conocida como "enfermedad de las vacas locas" (5). Por ello la predicción de la estructura terciaria de proteínas empleando únicamente información de la secuencia de aminoácidos ha sido uno de los más importantes e interesantes objetivos de la biología molecular y la biofísica.

El estudio de la estructura terciaria de proteínas brinda utilidad en diferentes campos de la ciencia (6): (*i*) *Medicina*: Con el objetivo de ayudar a comprender las funciones biológicas, ya que la unión de proteínas con ligandos moleculares, ácidos nucleicos, carbohidratos, lípidos y otras proteínas, determinan gran parte de la actividad celular. (*ii*) *Diseño de fármacos:* Para logar cribar bibliotecas de dianas farmacológicas. (*iii*) *Agricultura*: Para optimizar la ingeniería genética de los cultivos más productivos y resistentes. (*iv*) *Industria Química*: En la obtención de biocatalizadores (7).

Sin embargo, determinar experimentalmente la estructura de una proteína es una tarea compleja, que implica considerables costos asociados a la tecnología, el personal y el tiempo requerido. Evidencia de ello es que bases de datos de secuencias de proteínas como UniProt (8) y GenBank (para secuencias traducidas literalmente de genomas)(9), cuentan con decenas de millones de secuencias almacenadas. En tanto, el Banco Mundial de Datos de Proteínas (PDB) (10) posee solo cerca de cien mil estructuras tridimensionales determinadas experimentalmente. En consecuencia, la búsqueda de métodos computacionales

de predicción estructural de proteínas ha sido uno de los campos más activos en Biología y Química Computacional en los últimos veinte años (11, 12). Los mejores resultados se han obtenido con estrategias de modelación comparativas, basadas en el uso de información contenida en bases de datos estructurales. La gran diversidad de las secuencias de aminoácidos en los genomas caracterizados hasta la fecha, conlleva a la existencia de secuencias con muy baja homología (<30%) con respecto a las secuencias de aminoácidos de proteínas con estructura conocida, lo cual constituye la principal limitante para los métodos comparativos. la modelación por homología donde las soluciones se construyen tomando como plantilla de una estructura conocida.

Han quedado rezagados, en cuanto a su aplicabilidad, los métodos de predicción *ab initio* (13) y particularmente, el empleo de potenciales energéticos basados en principios físicos. Tal situación responde, por un lado, a elevados costos computacionales y por otro, a que no ha habido un consenso en una aproximación (nivel de teoría) que logre alcanzar el mejor compromiso entre calidad y desempeño en la evaluación de la estabilidad de proteínas (14-16).

Los métodos *ab-initio* al no depender del conjunto de estructuras conocidas; permiten, en principio, explorar todo el espacio conformacional de una determinada proteína, dando lugar a la posibilidad de generar nuevos plegamientos. Estos métodos siguen la llamada *hipótesis termodinámica* que plantea que la estructura funcional, y óptima, de una proteína debe corresponder al mínimo global de la energía libre del sistema conformado por la proteína y su entorno en condiciones fisiológicas (17). De esta forma los métodos *ab initio* están conformados esencialmente por una función que aproxime efectivamente la energía libre del sistema proteína-entorno y un método de búsqueda para explorar el espacio conformacional de la proteína. En este marco de métodos ab-initio es en donde se desarrollan los estudios resumidos en el presente informe de tesis. Los modelos estructurales predichos computacionalmente permiten, en primer término, brindar una aproximación a la estructura de una proteína; además facilitan la interpretación de datos experimentales y por tanto la elucidación de nuevas estructuras (6).

Los estudios relacionados a el problema de predicción de estructura terciaria de proteínas han tratado de simplificar algunos aspectos como: la función de energía mediante la definición de potenciales empíricos, el espacio de posibles conformaciones a través de la simplificación de la representación estructural de la proteína cambiando la representación atomística por modelos donde cada aminoácido es considerado como un único centro de interacción. A su vez se han aplicado técnicas de optimización heurísticas (llamadas metaheurísticas) como los algoritmos: Recocido Simulado (18-20), Monte Carlo (18) y Algoritmos Evolutivos (18, 21), en respuesta al declarado carácter NP-completo que posee el problema de predicción de la estructura de una proteína. A pesar de tales esfuerzos, la predicción ab-initio de estructura de proteínas sigue representando uno de los mayores retos en la Biología y Química Computacional moderna.

Enmarcado en el contexto de metaheurísticas para la solución de problemas complejos, en el grupo de investigación de Descubrimiento de Fármacos y Bioinformática de la Facultad de Química-Farmacia se adaptaron e implementaron tres metaheurísticas: Optimización basada en Mayas Variables (VMO) (22), Recocido Simulado (SA) (23, 24) y Algoritmos Genéticos (GA) (25) para ser aplicadas a la predicción de estructura terciaria de proteínas. Debe destacarse que GA y SA han sido los métodos más empleados hasta el momento en el estudio del plegamiento de proteínas. Las implementaciones anteriores tienen características en común por lo que se desea tenerlas en un mismo software, y así facilitar su uso. Las implementaciones anteriores no han sido comparadas entre ellas y en la literatura no hemos encontrado comparaciones entre varias metaheurísticas aplicadas a este problema con una misma base de casos. Es necesario conocer cuál de las metaheurísticas obtiene los mejores resultados para utilizarlas en futuras investigaciones.

Teniendo en cuenta lo antes mencionado nos hemos trazado como **objetivo general**: Evaluar el desempeño de las metaheurísticas: Optimización basada en mallas Variables (VMO), Algoritmo Genético (GA) y Recocido Simulado (SA) en la predicción de estructura terciaria de proteínas.

Para desarrollar el objetivo general se ha subdividido en los siguientes **objetivos específicos**:

1. Implementar un software en Java que unifique las tres metaheurísticas.
2. Realizar un diseño de experimentos para seleccionar una configuración óptima de parámetros por metaheurística.
3. Determinar el desempeño de la mejor metaheurística respecto a predictores internacionales.

Para dar solución a los objetivos específicos se plantean las **tareas** siguientes:

1. Combinar la implementación de las metaheurísticas en un software.
2. Realizar un estudio de los parámetros resultantes de la evaluación de los experimentos en cada meta-heurística.
3. Realizar la comparación estadística en cuanto a calidad y desempeño computacional entre las distintas metaheurísticas (considerando los parámetros previamente optimizados).
4. Realizar la comparación estadística entre la calidad de las predicciones de la mejor metaheurística con las soluciones de predictores internacionales.

Aporte

Como **aporte práctico** se introduce una herramienta que permite la predicción ab-initio de estructura de proteínas. La misma esta implementada en Java por los que es multiplataforma, emplea programación en paralelo (memoria compartida) para optimizar su desempeño en ordenadores multi-núcleos. Esta herramienta unifica las implementaciones de las tres metaheurísticas: VMO, GA y SA. La herramienta se puede extender implementando otras metaheurísticas.

Como **aporte metodológico** se identifica a Algoritmo Genético como la mejor metaheurística, entre las comparadas, para la predicción de estructura de proteínas. Se muestra además un estudio de parámetros con evaluación estadística que permite identificar las mejores configuraciones en los algoritmos empleados.

Como **aporte teórico** se adaptó una metaheurística lineal (Recocido Simulado) a un esquema en paralelo. En el cual se introduce un parámetro de relajación de la

temperatura inicial (cuyo valor es distinto para cada hilo) a fin de aportar diversidad entre los descensos de la temperatura.

Este trabajo consta de tres capítulos y está estructurado de la forma siguiente:

En el Capítulo “Marco Teórico”, se describe el plegamiento de proteínas y se realiza una revisión general sobre modelos meta heurísticos y de otra índole que aborda el plegamiento proteico y finalmente se muestra un resumen de los mejores métodos y estrategias aplicados en esta área de la Biología computacional. Además se describen de forma general las metaheurísticas utilizadas en la investigación y se realiza una comparación entre ellas.

En el Capítulo “Materiales y Métodos”, se presentan los aspectos computacionales para representar el problema de predicción de proteínas. Se analizan las implementaciones realizadas de las metaheurísticas, además de la función objetivo utilizada en las optimizaciones. Se conforma un software que acopla las metaheurísticas implementadas para un fácil uso. Se diseña un experimento de los parámetros a estudiar de cada metaheurística. Se presentan los aspectos utilizados para comparar dos estructuras tridimensionales de una proteína. Se describe la base de datos utilizada para la obtención de los resultados.

En el Capítulo “Resultados” se detallan los experimentos realizados con el fin de evaluar los métodos propuestos, y se presentan los resultados obtenidos realizando 3 experimentos. El primer experimento está dedicado a seleccionar la mejor configuración de parámetros por metaheurística. El segundo emplea varios aspectos para seleccionar la mejor metaheurística. El tercer experimento describe el comportamiento de la mejor metaheurística respecto a predictores internacionales.

La tesis culmina con conclusiones y recomendaciones para trabajos futuros.

II. Marco Teórico

En este capítulo introduce los temas referentes al problema del plegamiento de proteínas. Se realiza una revisión general sobre métodos de predicción de estructura de proteínas. Se describen las tres metaheurísticas que en el presente trabajo se comparan para decscribir su desempeño como métodos de optimización estructural de proteínas: Optimización basada en Mallas Variables (VMO), Algoritmos Genéticos (GA) y Recocido Simulado (SA). Se introducen las herramientas para comparar dos estructuras de proteínas mediante superposiciones (alineamiento estructural), las cuales suelen ser entre las soluciones obtenidas y las geometrías reales (nativas). Además se presentan las métricas para medir la calidad de dichos alineamientos y las pruebas estadísticas a realizar.

II. 1. ¿Qué son las proteínas?

Las proteínas son biomoléculas formadas básicamente por carbono, hidrógeno, oxígeno y nitrógeno. Pueden además contener azufre y en algunos tipos de proteínas, fósforo, hierro, magnesio y cobre entre otros elementos. Pueden considerarse polímeros lineales de pequeñas moléculas denominadas aminoácidos (26).

Existen 20 tipos de aminoácidos proteogénicos comunes (ver Anexo VIII. 1); aunque se conoce que el código genético codifica otros dos aminoácidos, sin embargo la frecuencia de aparición de estos últimos considerablemente menor. Los aminoácidos están compuestos por un grupo amino y uno carboxilo unidos por un carbono (llamado carbono Alfa), del que parte una cadena lateral corta que les confiere propiedades químicas específicas (como hidrofobicidad, aromaticidad o carga) (27). Los aminoácidos se encadenan entre sí mediante enlaces peptídicos, combinándose en distinto orden y número (pudiendo encontrar hasta miles en una única cadena). Un ejemplo de concatenación de tres aminoácidos se muestra en la Figura II-1 (28).

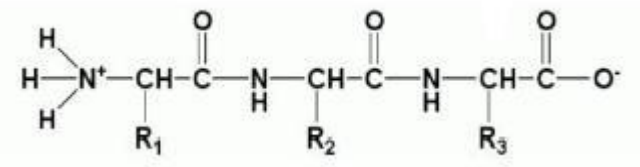


Figura II-1: Concatenación de tres aminoácidos. CH corresponde al carbono Alfa y R_i a las cadenas laterales.

Aunque no es un criterio estándar, comúnmente se refiere como péptido a una cadena no mayor a 10 residuos de aminoácidos, en tanto de 10 a 50 se les llama oligopéptidos, un número superior se denomina polipéptido y si el número es superior a 100 aminoácidos se habla de proteína (26). Los enlaces peptídicos se forman por la interacción covalente entre dos aminoácidos liberando una molécula de agua, de forma que la cadena resultante se dice estar formada por residuos de aminoácidos, refiriéndose al producto de la perdida de las moléculas de agua.

Tabla II-1: Características de los 20 aminoácidos naturales

Aminoácido	Código 3 letras	Código 1 letra
Alanina	ALA	A
Cisteina	CYS	C
Áspartato	ASP	D
Glutamato	GLU	E
Fenilalanina	PHE	F
Glicina	GLY	G
Histidina	HIS	H
Isoleucina	ILE	I
Lisina	LYS	K
Leucina	LEU	L

Aminoácido	Código 3 letras	Código 1 letra
Metionina	MET	M
Asparragina	ASN	N
Prolina	PRO	P
Glutamina	GLN	Q
Arginina	ARG	R
Serina	SER	S
Treonina	THR	T
Valina	VAL	V
Triptófano	TRP	W
Tirosina	TYR	Y

II. 1. 1. Estructuras de las proteínas

El nivel más básico de la estructura proteica, llamada *estructura primaria*, es la secuencia lineal de aminoácidos (29). Considerando el número posible de aminoácidos, la conformación o secuencia primaria de una proteína puede ser abstraída como una cadena de caracteres sobre un alfabeto de tamaño 20 §=[A;C;D;E;F;G;H;I;K;L;M;N;P;Q;R;S;T;V;W;Y], donde cada símbolo representa la letra que corresponde al aminoácido, según la Tabla II-1.

Las fuerzas como: los enlaces de hidrógenos, los puentes disulfuro, la atracción entre cargas positivas-negativas y las interacciones hidrofóbicas (repelentes al agua) e hidrofílicas (afines al agua) hacen que la cadena se pliegue y adopte una estructura tridimensional en la cual se reconocen motivos estructurales comunes entre todas las proteínas, denominados *estructuras secundarias* (30-32). Dentro de los tipos de estructuras secundarias se destacan la hélice alfa y la hoja beta como los más frecuentes y por tanto los que mayor inciden en la estabilidad de la estructura global denominada *estructura terciaria*. La estructura secundaria de las proteínas se puede codificar de manera similar a la secuencia primaria, asignando a cada residuo una letra que identifica el estado de estructura secundaria en que se encuentra. Se suele identificar a los residuos de una -hélice (ver Figura II-2) con H, los de una lámina (ver Figura II-3Figura II-4) con E y los demás con C (27).

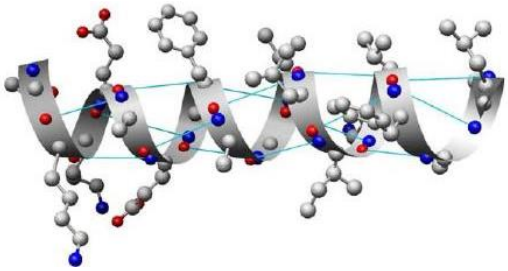


Figura II-2: Hélice-alfa mostrando cadenas laterales y puentes de hidrógeno

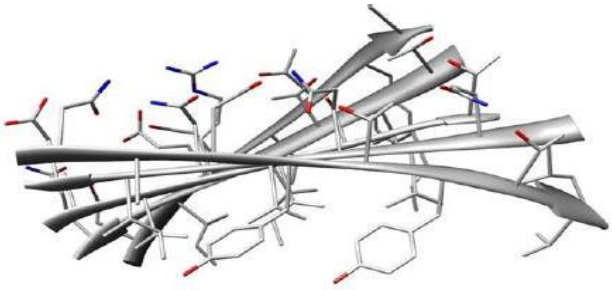


Figura II-3: Diagrama de láminas beta antiparalelas mostrando la disposición de las cadenas laterales

Cuando las fuerzas provocan que la molécula se vuelva todavía más compacta, como ocurre en las proteínas globulares, se constituye una estructura terciaria (ver Figura II-4) donde la secuencia de aminoácidos adquiere una conformación tridimensional estable y funcional, denominada estado NATIVO.

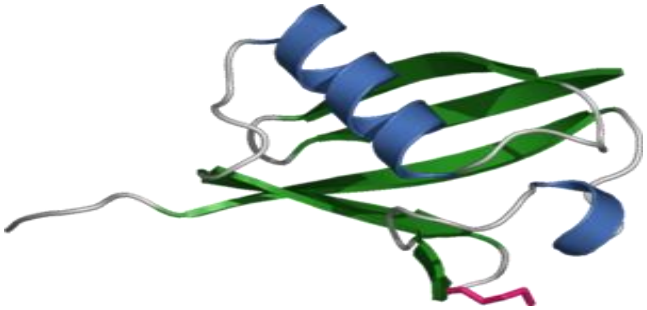


Figura II-4: Imagen de una proteína. Las α-hélice están marcadas de color azul y las hojas-β de color verde.

II. 1. 2. Ángulos de torsión

La estructura tridimensional de una proteína puede ser descrita mediante los grados de libertad conformacionales asociados a la cadena central de la proteína. Estos grados de libertad también son llamados ángulos de torsión o diedros. Una proteína cuenta con tres ángulos de torsión por aminoácido phi (φ), psi (ψ) y omega (ω), en la Figura II-5 (33) se muestra la disposición de los ángulos. Los ángulos φ y ψ pueden adoptar valores en el rango continuo [-180°;180°], en tanto el ángulo ω presenta mucha mayor rigidez estando fijado alrededor de los 180° (34) y con muy baja frecuencia en 0°. El primer aminoácido de la proteína solo cuenta con dos de los ángulos: ψ y φ.

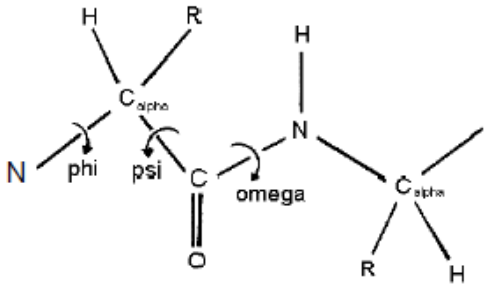


Figura II-5: Disposición de los ángulos en un fragmento de proteína

II. 2. El problema del plegamiento de proteínas

“El Plegamiento de Proteínas” es el término utilizado para describir el proceso complejo en el que las cadenas poli-peptídicas adoptan su conformación tridimensional “nativa”. Para llevar a cabo sus funciones deben plegarse con

rapidez y a un estado lo suficientemente estable bajo condiciones fisiológicas (35). El dogma central de la Biología Molecular sugiere que la secuencia de aminoácido de la proteína determina su estructura única global (36).

Las diferencias en las secuencias dan lugar a diferencias en la estructura secundaria y terciaria. Hasta ahora, las estructuras tridimensionales de aproximadamente 116085 proteínas han sido determinadas por cristalografía de rayos X y espectroscopia de RMN (Resonancia Magnética Nuclear), almacenadas en el Banco de Datos de Proteínas (PDB). Los dominios de estas proteínas pueden ser agrupados en aproximadamente 350 familias de pliegues, que consisten en secuencias con estructuras similares (37).

Es decir, la evolución ha seleccionado las secuencias con un mínimo de energía profundo para el estado nativo, lo que elimina las estructuras mal plegadas o parcialmente desplegadas a temperaturas fisiológicas. El proceso de plegamiento de proteínas es uno de los procesos biofísicos fundamentales. Dicho proceso resulta de interés debido al importante papel que desempeña en los mecanismos y controles de una amplia variedad de procesos celulares. Estos incluyen la regulación de eventos complejos durante el ciclo celular y la translocación de proteínas a través de las membranas de los órganos (38).

Por otra parte, la ocurrencia de plegamientos incorrectos en determinadas proteínas es asociada con enfermedades como el Alzheimer y la enfermedad de Creutzfeldt-Jakob, las cuales se originan por la conversión de un estado nativo soluble en una conformación con características hidrofóbicas que da lugar a la formación de placas amiloideas y agregados de fibrillas (39). Otras como la fibrosis quística, por ejemplo, son el resultado de mutaciones que dificultan el plegamiento normal y la secreción de proteínas específicas (40). Por ello, en la actualidad, la predicción de estructuras de proteínas, se ha convertido en uno de los objetivos principales de ramas de la ciencia como la Bioinformática y, la Química Teórica, etc. (41, 42). Hoy en día aun el proceso de plegamiento posee incognitos en relación a la relación a los parámetros cinéticos y energéticos que determinan su desarrollo (43-45). Como consecuencia no existe una función energética definitiva con la cual caracterizar este proceso y que a la vez sirva

unívocamente para guiar un proceso de optimización hasta la estructura nativa. Por ello se definen funciones empíricas con las cuales aproximar soluciones a este problema.

II. 3. Métodos Computacionales para la predicción estructural de proteínas

El problema de la predicción de estructura de proteínas ha sido catalogado como NP-Completo (46-50), debido a la cantidad de variables a optimizar, donde cada variable es continua, obteniendo un espacio de búsqueda muy grande.La predicción de estructura de proteínas continúa siendo un reto para la comunidad científica. Las optimizaciones basados en métodos físicos como la Dinámica Molecular emplean las relaciones de fuerza interatómicas, representadas por la segunda ley de Newton, para seguir el gradiente de energía potencial y alcanzar la conformación de mínima energía, que se asume como el estado nativo de la proteína (51-53). Estos métodos dinámicos cuya principal tienen como principal limitación el costo computacional.

Las metaheurísticas suelen ser utilizados en problemas donde no existe un algoritmo para hallar soluciones en tiempo polinomial (problemas NP) (54), donde haciendo uso de heurísticas se explora el espacio de búsqueda. Al problema de la predicción de estructura terciaria de proteínas se han aplicado metaheurísticas como: redes neuronales recurrentes, máquinas de vectores de soporte de múltiples capas, algoritmos bioinspirados o combinación de clasificadores (55) obteniendo resultados satisfactorios. En las metaheurísticas destinadas a la optimización para determinar si un elemento del espacio de búsqueda es mejor que otro utiliza una función objetivo, en el caso de las proteínas la función objetivo debe medir parámetros físico-químico-conformacionales de la estructura terciaria de proteínas. En este acápite se expondrán los aspectos esenciales de tres de las metaheurísticas mencionadas anteriormente ya que son parte esencial del objetivo de investigación de la tesis.

II. 3. 1. Optimización basada en Mallas Variables

La esencia del método Optimización basada en Mallas Variables (VMO por sus siglas en inglés) (56). Se crea un conjunto de elementos, denominada malla de

puntos, del espacio de búsqueda, donde cada punto tiene m dimensiones El proceso de optimización viene dada por una función $FO(x_1, x_2, \dots, x_m)$. La malla se mueve por el espacio de búsqueda mediante un proceso de expansión hacia otras regiones del espacio.

El proceso de generación de nodos está comprendido por varios pasos como: generación de la malla inicial la cual consta de *Ni* nodos, los cuales en la primera iteración son generados de forma aleatoria o por otro método que garantice obtener soluciones diversas. La generación de nodos en dirección a los extremos locales constituye otro tipo de exploración que se realiza en VMO se lleva a cabo en las vecindades de cada uno de los nodos de la malla inicial. Al igual que en el paso anterior, la generación de nodos en dirección hacia el extremo global tiene como propósito realizar una exploración global hacia el nodo que mejor calidad ha tenido hasta el momento (extremo global).

La generación de nodos a partir de los nodos más externos de la malla es un proceso más de expansión de nuevos nodos, que tiene lugar con el objetivo de explorar el espacio de búsqueda en dirección a las fronteras de cada dimensión.

II. 3. 1. 1. Seudocódigo general de VMO

Generación semi-aleatoria los elementos (n_i) de la malla inicial (MI).
Inicializar la malla total (MT) con los elementos de la malla inicial MI.
Evaluar los nodos de la MI con la función objetivo
Seleccionar el mejor n_b.
Repetir:
    Para cada nodo n_i **hacer**
        Encontrar sus k nodos más cercanos dentro de MI
        Determinar el mejor de los vecinos de n_i (n_k).
        Si n_k es mejor que n_i, **entonces**
            Generar nuevo nodo *h* entre n_i y n_k.
            Adicionar *h* a la MT
    Para cada nodo n_i en la malla inicial del ciclo **hacer**
        Generar nuevo nodo *p* entre n_i y n_b.
        Adicionar *p* a la MT

Mientras no se hayan alcanzado tres veces el tamaño de la MI
Seleccionar un nodo n_i aleatoriamente.
Generar a partir de n_i un nuevo nodo q hacia los bordes del dominio
Adicionar q a la MT
Ordenar los nodos de la MT según su calidad.
Seleccionar el mejor nodo g de la MT.
Si g es mejor que n_b, **entonces** n_b recibe g
Contraer la MT a un 1/3 de su tamaño de manera elitista (obtener nueva MI)
Hasta N ciclos
Retornar n_b

II. 3. 2. Recocido Simulado

El Recocido Simulado (SA por sus siglas del inglés Simulated Annealing), también llamado Enfriamiento Simulado, Método Descendiente Probabilístico o Relajación Estocástica, es un algoritmo de búsqueda metaheurística de optimización global. Las ideas en la que se basa el algoritmo fueron introducidas por Metropolis en el 1953 (57), en un algoritmo para simular el proceso de enfriamiento de metales. Posteriormente, fue desarrollado independientemente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983 (58), y por Vlado Černý en 1985 (59) que propusieron que este algoritmo podía ser empleado para resolver problemas de optimización.

El SA busca en el espacio de soluciones mediante un iterador estocástico. En cada iteración el algoritmo escoge un vecino de la solución y se calcula la energía (función a minimizar) del vecino. Con este valor se calcula la probabilidad de transición (siempre que el sucesor sea mejor que su origen lo sustituye si no se utiliza una función de probabilidad). El valor de esta función (que toma en cuenta además de la energía, la temperatura) determina si la nueva solución es tomada en cuenta o no. Esta función permite que vecinos con peores soluciones sean tomados en cuenta cuando se comienza a iterar y disminuye luego la probabilidad de que peores soluciones sean escogidas cuando ya la solución va más avanzada (60). El parámetro más importante del algoritmo SA es la generación de vecinos. La selección del vecino puede ser aleatoria pero dirigida en cierta forma (54).

El algoritmo comienza con un valor de Temperatura (*T)* muy alto, que va decreciendo en cada iteración siguiendo un cierto *protocolo de recocido*, que puede ser diferente para cada problema, pero que siempre debe terminar con *T=0*. Así el sistema será libre inicialmente de explorar una gran porción del espacio de búsqueda, ignorando pequeñas variaciones de la energía entre los estados vecinos evaluados, para más tarde centrarse en regiones con estados de baja energía y, al final, cambiar solo a estados con energía menor que la inicial, hasta alcanzar un mínimo. Protocolos de recocido es las distintas funciones utilizadas para descender el valor de la Temperatura, los más utilizados en la literatura son: Boltzman, Cauchy, Geométrico y Lundi-Mess.

II. 3. 2. 1. Seudocódigo general de RS

```
Seleccionar Solución inicial So
Seleccionar estado = So;
Seleccionar tope
Repetir:
    Seleccionar it = 0;
    Repetir:
        Seleccionar siguiente= sucesor_aleatorio (estado);
        DE = valor (siguiente) – valor (estado);
        Si DE < 0 entonces estado = So = siguiente;
        Sino
            Seleccionar q = min (1, e^(-DE/T));
            Si aleatorio (0,1) < q entonces estado = siguiente;
        it = it + 1;
    Mientras it != tope;
    To = enfriar (To, it);
Mientras To =~ 0
Retornar So;
```

II. 3. 3. Algoritmos Genéticos

La teoría de la evolución fue desarrollada por Charles Darwin (1859) en *El Origen de las Especies por medio de la Selección Natural (61)*. La idea central es: las variaciones (conocidas como mutaciones) ocurren en la reproducción y serán conservadas por generaciones sucesivas aproximadamente en la proporción de su efecto sobre la idoneidad reproductiva. La teoría de Darwin fue desarrollada sin el conocimiento de cómo los rasgos de los organismos se pueden heredar y modificar (33). Las leyes probabilísticas que gobiernan estos procesos fueron identificadas primero por Gregor Mendel en 1866 (28), un monje que experimentó con guisantes dulces usando lo que él llamó la fertilización artificial.

Los Algoritmos Genéticos son una de las metaheurísticas más empleadas para la solución de problemas NP-Completos. Esto se debe principalmente al éxito que han tenido en la solución de problemas duros donde el espacio de posibles soluciones es muy grande. En general, los algoritmos genéticos consisten de una población de individuos donde la selección del mejor individuo está basada en su aptitud respecto a algún ambiente. El algoritmo procede en pasos llamados generaciones. Durante cada generación, una nueva población de individuos es creada a partir de la aplicación de operadores genéticos (cruza, mutación, etc.), y evaluada como solución al problema dado. Debido a la presión efectuada por la selección, la población se adapta al ambiente a través de generaciones donde se garantiza que la aptitud promedio de estas mejora; característica que provoca una evolución hacia mejores soluciones (62). Durante la simulación computacional de la evolución, la calidad de la población incrementa y por consiguiente también lo hacen las soluciones finales brindadas al problemas de optimación (63).

II. 3. 3. 1. Seudocódigo general de GA

```
Seleccionar Población P
Repetir:
    Seleccionar nueva Población p_new = conjunto vacío
    Para i desde 1 hasta tamaño de p
        Individuo x = selección_aleatoria (P);
        Individuo y = selección_aleatoria (P);
```

```
            Individuo hijo = Reproducir(x,y);
            Si probabilidad pequeña entonces hijo = Mutar(hijo);
            Añadir hijo a p_new;
        P = p_new;
Mientras no se encuentre un individuo adecuado
Retornar Mejor individuo de p;
```

II. 3. 4. Comparación entre las tres metaheurísticas

Las metaheurísticas VMO y GA se caracteriza por ser algoritmos poblacionales. VMO genera una malla inicial que se expande y contrae, realizando una gran exploración, mientras que GA haciendo uso de los operadores genéticos (cruce y mutación) obtiene nuevas poblaciones. A diferencia de GA y VMO, el SA no es un algoritmo poblacional. Existen estudios realizados de como descender la temperatura y su principal ventaja es que garantiza poder salir de óptimos locales (64).

Una característica que tienen en común estas metaheurísticas es que utilizan las probabilidades para recorrer el espacio de búsqueda. A estos algoritmos se les puede configurar como condición de parada la cantidad de evaluaciones de la función objetivo de manera que se puedan comparar respecto a la calidad de las soluciones y al tiempo promedio de ejecución.

Estas metaheurísticas han sido aplicadas al problema del plegamiento de proteínas en tesis anteriores de nuestro centro y en entidades internacionales. Gabi Escuela en el 2006 ha implementado GA para el problema del replegado de las proteínas como parte de su tesis de maestría (21). Iosvani More Quintero en el 2011 ha implementado VMO para la predicción de la estructura terciaria de las proteínas como parte de la investigación para su tesis de grado (22). Fernando Wong Delgado en 2015 (23) se implementa el SA aplicado a la predicción de estructura terciaria de proteínas. Existen muchos artículos que tratan el tema de las metaheurísticas aplicadas a este problema.

II. 4. Alineamiento Local – Global (LGA) de las estructuras de proteínas

Las estructuras tridimensionales obtenidas a partir de una secuencia de aminoácidos se pueden comparar contra la estructura nativa (si se conoce su nativa) para determinar la calidad de las soluciones. Para dichas comparaciones se utilizan métricas y funciones de similitud, algunas de estas se pueden obtener en el servidor LGA (65). LGA es un servidor web (66) diseñado para comparar estructuras de proteínas buscando la mejor superposición (alineamiento) estructural de dos proteínas o de fragmentos estructurales. El alineamiento estructural se sostiene sobre el principio de que puede definirse una función objetivo (función de ajuste) con su valor óptimo correspondiente a la superposición estructural más significativa. LGA emplea en su función de ajuste dos parámetros, LCS (*Longest continuous segment*) y GDT (*Global Distance Total*), introducidos por Zemla (65), los cuales han mostrado gran éxito en la detección de similitudes estructurales locales y globales entre dos proteínas (67, 68). El procedimiento LCS localiza el conjunto continuo más largo de residuos que puede ajustarse bajo un umbral de RMSD predeterminado. Por otro lado el algoritmo de GDT está diseñado para complementar las evaluaciones realizadas con LCS, en este caso se busca el mayor conjunto de residuos (no es necesario un conjunto continuo) que no se desvíen por más de una distancia de corte máxima seleccionada (65, 69).

II. 4. 1. Métricas calculadas por el servidor LGA

El alineamiento estructural, entre cada modelo creado y la estructura nativa de cada proteína, es generado por el método LGA. Este alineamiento es evaluado mediante cuatro métricas, provistas por la propia aplicación, las cuales se describen a continuación:

N: Número máximo de pares de residuos *correspondientes* alineados satisfactoriamente de acuerdo a la distancia de corte seleccionada.

RMSD: Raíz de la media cuadrática de las desviaciones entre los N pares de residuos *correspondientes* satisfactoriamente alineados. Donde v_i^2 y v_i^1 representa las coordenadas de la estructura predicha y la estructura nativa respectivamente.

$$RMSD = \sqrt{\sum_{i=1}^{N}\left(v_i^2 - v_i^1\right)^2 \Big/ N}$$

Ecuación II-1Métrica RMSD

LGA_S: Función de puntuación del ajuste, definida como:

$$\mathrm{LGA\_S} = w * GDT\_S + (1 - w) * LCS\_S$$

Ecuación II-2: Métrica LGA_S

Donde w varía entre 0.0 y 1.0, LCS_S es el porcentaje de residuos *correspondientes* que se ajusta por debajo de un valor de corte de RMSD denominado v_{LCS}, GDT_S se asocia al porcentaje de residuos *correspondientes* que se ajustan por debajo de un valor de corte de distancia v_{GDT}. Estos últimos parámetros, son estimados como F_S:

$$\begin{aligned} &foreach\ vi\ (v1, v2, \ldots, vk)\{ \\ &\quad Y = (k - i + 1)/k; \\ &\quad X = X + Y(F\_S_{vi}); \\ &\} \\ &F\_S = X/((1+k)k/2); \end{aligned}$$

Donde para LCS, vi = 1, 2,…, $\mathbf{v_{LCS}}$ y para GDT, vi = 1, 2,…, $\mathbf{v_{GDT}}$. Los valores $\mathbf{v_{LCS}}$ y $\mathbf{v_{GDT}}$ corresponden a las distancias (en Å) empleados en el alineamiento. En los cálculos realizados en el presente trabajo $\mathbf{v_{LCS}}$ = 4 Å, $\mathbf{v_{GDT}}$ = 10 Å y w = 0.5.

GDT_TS = Distancia total global, se define como (P1 + P2 + P4 + P8)/4, donde $\mathbf{P_d}$ son los porcentajes de residuos que pueden ajustarse bajo los valores de distancia de corte d = 1, 2, 4 y 8 Å.

II. 5. Predictores Internacionales

En la tesis se desea conocer como son las soluciones obtenidas por nuestros algoritmos y para ello se necesita de soluciones de otros algoritmos para comparar. El CASP y CASP-ROLL son competencias (que se describen más adelante) donde compiten predictores (algoritmos) con 5 predicciones (5 soluciones) aplicadas a la predicción de estructura terciaria de proteínas. Las soluciones que brindan los competidores son publicadas para realizar comparaciones futuras. Estas dos competencias utilizan muchas métricas para

evaluar la calidad de las soluciones brindadas por los competidores, entre ellas se encuentran las cuatro métricas que brinda el servidor LGA.

El CASP surgió en el año 1994 (70-72) en respuesta al marcado desbalance que se produjo entre el número de secuencias conocidas de proteínas y el número de estructuras elucidadas. Debido a las dificultades experimentales y el tiempo requerido para determinar la estructura de una proteína es que ya en aquella fecha se apostaba por el desarrollo de métodos computacionales de predicción de estructura de proteínas. La competencia ha brindado el marco apropiado para que los principales investigadores de la rama confrontasen sus métodos en una prueba a ciegas y se pudiese determinar que algoritmos lograban los mejores resultados en este problema. Dando lugar a un espacio para el intercambio entre los investigadores y la publicación de los mejores métodos. En la actualidad la competencia sigue vigente y cuenta con una gran reputación científica y diversidad de predictores participantes, siendo evidencia de las dificultades del problema del plegamiento de proteínas y los avances que aún son necesarios hacer en este campo.

La competencia se organiza de modo que los grupos que en distintos laboratorios en el mundo se dedican a la elucidación experimental de estructuras de proteínas envían a los organizadores la secuencia de las proteínas que están próximos a publicar. Los organizadores dan entonces un plazo limitado para que los distintos predictores computacionales envíen un máximo de cinco modelos estructurales para cada proteína ofertada. Una vez los experimentalistas hacen llegar las estructura reales, los organizadores evalúan las predicciones y publican los resultados junto con la estructura de las proteínas y los distintos modelos propuestos por los diferentes predictores.

El evento CASP-ROLL (Rolling Critical Assessment of Structure Predictions) es una competencia continua de predicción de estructuras de proteínas, derivada del evento más general CASP cuya frecuencia es bienal (73). El CASP_ROLL surgió hace seis años como un espacio donde se agrupan aquellas proteínas más complejas de modelar por poseer una secuencia con baja homología con secuencias existentes y donde por tanto los métodos basados en conocimiento no

brindan los mejores resultados. De aquí que en este evento participan mayormente métodos *ab initio*, los cuales básicamente realizan una optimización de la estructura partiendo de candidatos aleatorios y empleando una determinada función de ajuste y un método de exploración apropiado.

II. 6. Análisis de test estadísticos para comparar los resultados de las metaheurísticas

Ya implementadas las metaheurísticas es necesario comparar sus resultados respecto a la convergencia de la función objetivo y la calidad de las soluciones generadas. El uso de pruebas estadísticas ayuda a interpretar los datos; en los subepígrafes siguientes se describen algunas pruebas estadísticas. Al comparar dos muestras pareadas y tener dudas sobre la normalidad de los datos se utiliza la prueba no paramétrica de Wilcoxon. La prueba de Friedman es utilizada para comprar si existen diferencias significativas entre varias (más de dos) variables pareadas. Cuando se encuentran diferencias significativas con la prueba de Friedman se utilizan pruebas post-hoc para determinar las diferencias, donde de las pruebas post-hoc más utilizadas es Holm.

II. 6. 1. Prueba de rangos con signos de Wilcoxon

La prueba de rangos con signos de Wilcoxon (74) es una prueba no paramétrica para comparar la mediana de dos muestras relacionadas y determinar si existen diferencias entre ellas. Se utiliza como alternativa a la prueba t de Student de muestras pareadas cuando no se puede suponer la normalidad de dichas muestras (75). Para *n* pares de observaciones, denominados (x_i, y_i), el objetivo del test es comprobar si puede dictaminarse que los valores x_i e y_i son o no iguales. Si $z_i = y_i - x_i$, entonces los valores z_i son independientes y tienen una misma distribución continua y simétrica respecto a una mediana común θ.

La hipótesis nula es H_0: $\theta = 0$. Para verificar la hipótesis, en primer lugar, se ordenan los valores absolutos (z_n) y se les asigna su rango R_i. Luego se calcula los parámetros S^+ y S^- correspondientes a la suma de los rangos (R_i) positivos y negativos respectivamente. La menor de ambas sumas es tipificada (Z) considerando una distribución normal. El valor-p bilateral resulta de multiplicar por

dos la probabilidad de encontrar valores menores o iguales que Z. Valores-p menores que el nivel de significación (α) rechazan la hipótesis nula de igualdad de las medianas.

II. 6. 2. Prueba de Friedman

La prueba de Friedman pertenece a las pruebas no paramétricas de comparación de tres o más muestras relacionadas, es decir: es libre de la curva normal, se usa la distribución de chi-cuadrado y la variable dependiente presenta al menos nivel ordinal. Se utiliza para comparar tres o más grupos de rangos (medianas) relacionados y determinar que las diferencias no se deban al azar (que las diferencias sean estadísticamente significativas) (76).

El Test de Friedman (77, 78), trabaja asignando rankings r_{ij} a los resultados obtenidos por cada algoritmo j en cada problema i. Esto es, para cada problema, se asigna un ranking $1 \leq r_{ij} \leq k$, donde k es el número de algoritmos a comparar. Estos rankings se asignan de forma ascendente, es decir, 1 al mejor resultado, 2 al segundo, etc. (en caso de haber empates, se asignan rankings medios). El Test de Friedman requiere el cálculo de los rankings medios de los algoritmos sobre los n problemas,

$$R_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}$$

Ecuación II-3: R_j de Friedman

La hipótesis nula que indica que todos los algoritmos se comportan similarmente, por lo que sus rankings R_j deben ser similares. Siguiendo esta hipótesis, el estadístico de Friedman

$$F_f = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]$$

Ecuación II-4: Estadístico de Friedman

se distribuye de acuerdo a una distribución chi-cuadrado con k- 1 grados de libertad.

II. 6. 3. Prueba de Iman-Davenport

Este estadístico de Friedman fué mejorado a su vez por Iman y Davenport (79), quienes mostraron que el estadístico de Friedman presenta un comportamiento demasiado conservativo. Para evitar este problema, propusieron otro estadístico más ajustado que

$$F_{ID} = \frac{(n-1)F_f}{n(k-1)F_f}$$

Ecuación II-5: Estadístico de Friedman ajustado por Iman-Davenport

se distribuye de acuerdo a una distribución F con k-1 y (k-1)(n-1) grados de libertad.

II. 6. 4. Prueba post-hoc Holm

Las pruebas post-hoc son utilizadas cuando es rechazada la hipótesis nula de los test y se desea encontrar diferencias significativas a posteriori (80). Uno de los métodos Post-hoc más utilizados es el Holm (81). En este método $H_0$1,$H_0$2,..., H_0k, son las k hipótesis nulas que se desean contrastar y P1, P2,..., Pk, sus respectivos p-valores, el procedimiento de ajuste propuesto por Holm es el siguiente:

A1. Sean P(1), P(2),..., P(k) los p-valores ordenados, esto es, P(1) ≤ P(2) ≤⋯≤ P(k), y sean $\mathbf{H}_0(1)$, $\mathbf{H}_0(2)$,⋯, $\mathbf{H}_0(k)$ sus respectivas hipótesis nulas.

A2. Calcular: **j** = máximo (i Є1,...,k, tal que (k - l +1)·P(l) < α, para todo 1 ≤ l ≤ i).

A3. Se rechazan las hipótesis correspondientes a los p-valores: P(1) ,..., P(l) y no se rechazan las hipótesis correspondientes a los p-valores P(l+1) ,..., P(k). Esto es, se rechaza $\mathbf{H}_0(1)$,..., $\mathbf{H}_0$(l) y no se rechaza $\mathbf{H}_0$(l+1),...,$\mathbf{H}_0(k)$.

II. 7. Conclusiones Parciales

Se realiza un recorrido sobre los temas básicos referentes al problema de plegamiento de proteínas, donde se expone que existen 20 aminoácidos esenciales, los cuales al concatenarse forman las proteínas que pueden estar compuestas ente 100 y 1500 aminoácidos. Este problema está considerado como NP-completo. Existen muchos métodos computacionales para obtener estructuras 3D de proteínas a partir de la secuencia de aminoácidos y uno de los más utilizados son las metaheurísticas. Existen muchas metaheurísticas utilizadas en

este problema donde algunas de las más frecuentes son el algoritmo genético y el de recocido simulado. Los dos algoritmos mencionados anteriormente han sido implementados para este problema al igual que Optimización basada en mallas variables. Siempre que simulemos la obtención de una proteína hay que evaluar la calidad de las soluciones obtenidas contra las soluciones reales para ello se utilizan los alineamientos: local o global y el servidor LGA provee métricas para ello. La comparación de los resultados haciendo uso de las pruebas estadísticas ayuda a determinar la veracidad de las conclusiones obtenidas en los estudios.

III. Materiales y Métodos

En el presente capítulo se describe la metodología desarrollada para realizar la predicción de la estructura de proteínas, así como algunas implementaciones y consideraciones tomadas durante el análisis y diseño. Se realiza un análisis de la representación computacional empleada para una proteína. Se plantean las restricciones del problema y el espacio de búsqueda. Se analiza la Función de Aptitud (Función Objetivo) que se utilizará para evaluar las soluciones. Se presentan consideraciones sobre el diseño e implementación de las tres metaheurísticas. Se realiza un diseño de experimento para ajustar los parámetros en las metaheurísticas. Además se describe la base de casos a utilizar.

III. 1. Representación computacional de las proteínas

Los algoritmos de predicción *ab-initio*, clase en la que se enmarca el estudio realizado en la presente tesis, utilizan la secuencia de aminoácidos para obtener una estructura tridimensional, y por tanto constituye la entrada fundamental de las metaheurísticas empleadas. En la representación de las proteínas a nivel de código se distingue al primer residuo de la cadena del resto de los aminoácidos de la misma debido a que el mismo presenta un único ángulo de torsión asociado a la cadena central (variable de la metaheurística) en tanto los demás residuos poseen dos ángulos de torsión (ver epígrafe II. 1. 2).

III. 1. 1. Coordenadas internas

En química computacional los sistemas de N partículas ligadas (como es el caso de una molécula) pueden especificarse completamente con menos de 3N coordenadas. La “armazón” molecular se puede mover como un todo, o puede rotar sin que con ello cambien las distancias internas y los ángulos entre los átomos que lo conforman. Por consiguiente es posible eliminar tres variables que fijen la traslación como un todo de la molécula, así como tres que determinen la rotación global de la estructura si esta es no lineal y aun así dar toda la información para construir la molécula si se fija el origen en el centro de masas o en un átomo determinado y además, si impedimos que el sistema de ejes de referencia rote libremente, fijándolo con lo cual podemos descartar tres

coordenadas. Al conjunto resultante se denomina matriz de coordenadas internas (82, 83).

La Z-Matriz es un formato muy común coordenadas internas moleculares empleándose indistintamente los términos “matriz Z” o “matriz de coordenadas internas”. Una geometría molecular se puede especificar mediante vías de una Z-Matriz quedando expresada mediante una matriz de N filas (una por cada átomo) con tres columnas de coordenadas correspondientes a la distancias de enlace, ángulo de enlace y ángulos de torsión empleado para definir la posición del átomo. Aparejada a cada una de las anteriores columnas existe otra columna de referencia en la cual se ubica el número de orden del átomo (punto espacial) con respecto al cual se especifica la coordenada interna dada, los valores de las columnas de referencia deben cumplir por tanto con la regla que: en una determinada fila las referencias situadas deben corresponder a filas anteriores, correspondientes a átomos previamente localizados en el espacio (84).

Computacionalmente la construcción de la Z-Matriz depende en gran medida de la secuencia de aminoácidos ya que cada uno posee un número de átomos (filas) diferente. Puesto que en muchas optimizaciones es posible restringir determinados rasgos o motivos estructurales, como una porción de la molécula o determinados tipos de enlaces o grupos de átomos, en la gran mayoría de las ocasiones el uso de una Z-Matriz permite realizar optimizaciones geométricas con mayor velocidad de cómputo respecto del uso de coordenadas cartesianas.

A continuación se presenta a modo de ejemplo una matriz en coordenadas cartesiana y coordenadas internas del metano (no es una proteína). El metano presenta 5 elementos químicos: un carbono y cuatro hidrógenos, donde los hidrógenos se encuentran a la misma distancia del carbono, en los vértices de un tetraedro regular. En la matriz de coordenadas cartesianas presenta los valores x, y, z de cada átomo presente en la sustancia. Las coordenadas internas de la sustancia se ven reflejadas con 6 valores (átomo1, distancia hacia el átomo1, átomo2, ángulo de enlace, átomo3, ángulo de torsión)

1	C 0.000000 0.000000 0.000000	C
2	H 0.000000 0.000000 1.089000	H 1 1.089000
3	H 1.026719 0.000000 -0.363000	H 1 1.089000 2 109.4710
4	H -0.513360 -0.889165 -0.363000	H 1 1.089000 2 109.4710 3 120.0000
5	H -0.513360 0.889165 -0.363000	H 1 1.089000 2 109.4710 3 240.0000

Figura III-1: De izquierda a derecha: numeración de los átomos, coordenadas cartesiana y coordenadas internas del metano

En el presente trabajo se adoptaron como fijas las distancias de enlace, los ángulos de enlace y parte de los ángulos de torsión (aquellos correspondientes al enlace peptídico, ver Marco Teórico). Además son optimizados por separado los ángulos diedros correspondientes a la cadena central de la proteína (variables reales de las metaheurísticas) y los correspondientes a las cadenas laterales que son explorados estocásticamente en determinados puntos de la optimización. De esta forma las metaheurísticas operan con vectores de variables de una longitud de 2N – 1, donde N es el número de aminoácidos de la cadena.

La evaluación de la Función Objetivo se realiza en coordenadas cartesianas, mientras que la optimización ocurre en coordenadas internas. El método utilizado para convertir de coordenadas internas a cartesianas es el algoritmo Natural Extension Reference Frame (NERF) (85) implementado en la plataforma Chemistry Development Kit (CDK) (86-88) de Java.

III. 2. Restricciones del problema

Existen dos restricciones fundamentales a lo largo de la optimización. La primera de ellas, realizada a nivel de la Z-Matriz, efectúa un conteo de la cantidad de ángulos ψ y φ cuyos valores están ubicados en rangos poco favorables (irregulares). Las soluciones cuyo conteo supere el máximo permitido por la restricción son descartadas. Esta restricción se evalúa antes de seleccionar la mejor solución y al generar las soluciones iniciales de la optimización. La segunda restricción se aplica a nivel de la matriz de coordenadas cartesianas (necesaria para evaluar la función objetivo) y busca identificar estructuras con átomos superpuestos, para lo cual fija un valor mínimo de distancia espacial entre un par de átomos.

III. 3. Espacio de búsqueda

Las metaheurísticas persiguen optimizar la columna asociada a los ángulos diedros de la Z-Matriz de la proteína. En sus tres columnas de coordenadas reúne los valores de distancias de enlace, ángulos de enlace y ángulos diedros necesarios para ubicar a un determinado átomo en un punto del espacio cartesiano común conociendo las posiciones de los átomos que le preceden.

La optimización asume que la columna de las distancias y ángulos de enlace son fijos y por tanto no constituyen variables para el proceso de exploración. De los ángulos diedros que conforman la tercera columna de coordenadas internas se hace distinción de tres clases:

i) los ángulos omega (ω) de la cadena central son igualmente considerados fijos con valor igual a 180°, este ángulo está asociado al enlace peptídico que es un elemento estructural plano y considerablemente rígido por lo que su variación es 180±10º entre los distintos residuos de la cadena. Esto constituye una aproximación usual en los métodos de predicción estructural de proteínas.

ii) Los demás ángulos diedros de la cadena central (ψ_i y φ_i) cuyos valores determinan la forma tridimensional de la estructura y constituyen por tanto las variables principales de la metaheurística se exploran en el rango [-180,180].

iii) El resto de los ángulos diedros de la estructura, correspondientes a las cadenas laterales de cada residuo. Las cadenas laterales por su corta extensión no determinan la forma global de la estructura sino que sus conformaciones generalmente dependen de la forma adoptada por la proteína. No obstante la posición de los átomos de las cadenas laterales afecta la función objetivo. En las metaheurísticas las conformaciones de las cadenas laterales se exploran aleatoriamente cada vez que se encuentra una mejor solución a lo largo de la optimización, manteniendo aquella configuración de valores que minimice la función objetivo de la mejor solución encontrada.

En correspondencia a lo anterior la implementación de la optimización debe continuamente convertir la Z-Matriz de la proteína, con los valores de los diedros de una determinada solución, en una matriz de coordenadas cartesianas; que es empleada para evaluar la función objetivo.

III. 4. Función de Aptitud

Siempre que utilicemos un método heurístico para optimizar, necesitamos una función de evaluación para comparar las posibles soluciones y obtener la "más" factible respecto a dicha función, esta función también se denomina función objetivo o función a optimizar. En esta tesis se utiliza la formulación general del **modelo de estabilidad** propuesta por Ruiz-Blanco y colaboradores (6) como **función objetivo**, el modelo se define como sigue:

$$\Delta G_f = \omega_H \left(\Delta G_{el} + \Delta G_{VdW} + \Delta G_{tor} + \Delta G_{HBd} \right) + \omega_S \left(\Delta G_w + \Delta G_{conf} \right)$$

Ecuación III-1: Función Objetivo

Donde los valores de los coeficientes son: ω_H = 0.01024 y ω_S = -0.08016. Para representar las **interacciones de Van der Waals** se utiliza ΔG_{VdW}. El potencial de torsión de la cadena central es considera por ΔG_{tor}. Los **puentes de hidrógeno** son penalizados con ΔG_{HBd}. Las **interacciones electrostáticas** se representan por ΔG_{el}. El **efecto hidrofóbico** es representado por ΔG_w. El **cambio en la energía libre conformacional** se representa por ΔG_{conf}.

Debido a la gran complejidad de este problema se decide implementar otras dos funciones que complementen la función objetivo. Estas otras funciones son utilizadas por las metaheurísticas VMO y GA cuando se realiza la reducción de la población (obtención de una muestra). VMO utiliza las dos funciones en el proceso de contracción para obtener una nueva malla inicial. GA utiliza las funciones en el proceso de selección del 40% de la población (población intermedia) para luego realizar los cruzamientos y mutaciones. El tipo de muestreo donde se seleccionan los individuos donde se utilizan varias características (estratos) es llamado muestreo estratificado y si el porciento que se selecciona por estrato no es proporcional a la población recibe el nombre de muestreo estratificado

desproporcional (89). La forma de seleccionar los mejores por estrato recibe el nombre de selección simple sin remplazo (90).

La primera de las funciones de complementarias ayuda a garantizar la presencia de aminoácidos en conformaciones de hoja beta. La hoja beta es una estructura secundaria importante en la estabilización de una proteína, debido a los puentes de hidrógeno que forma. La función objetivo pondera en forma similar la magnitud de las contribuciones de este tipo de estructuras y de las hélices alfa, sin embargo las primeras contrario a las hélices no son motivos locales lo que hace que su muestreo (y por tanto prevalencia) a lo largo de la optimización sea muy inferior que las hélices. Tal desbalance en el muestre unido a la similitud de sus contribuciones (en términos de FO) conlleva a una subestimación de la frecuencia de aparición de estos motivos en las soluciones finales. Es por ello que se decide introducir una función ad hoc diseñada con el fin aumentar la prevalencia de los motivos de hojas beta en las poblaciones de cada metaheurística.

La fórmula se representa en la Ecuación III-2. Donde S representa la cantidad de bandas (fragmentos) que contienen más de 3 aminoácidos en conformaciones beta. N_B es el número total de aminoácidos que se encuentran en conformación de hoja beta. N_{HB} representa el número de puentes de hidrógeno en los que participan los aminoácidos de N_B.

$$F1 = \left[1 + S * \left(2 + \frac{N_{HB}}{2N_B}\right)\right] * \Delta G_{tor}$$

Ecuación III-2: Primera función de Apoyo (F1)

Uno de los aspectos fundamentales en el proceso de plegamiento de una proteína es que ella tiende a minimizar el área accesible al solvente. La segunda función complementaria a implementar ayuda a garantizar que la prevalencia de estructuras compactas en las poblaciones. En esta función (ver Ecuación III-3) se utiliza el valor de la función objetivo dividido por el área de la superficie de la proteína tridimensional. . La necesidad de la inclusión de esta función en las optimizaciones surge por el hecho que al muestrearse estructuras compactas la probabilidad de colapsos atómicos es alta conllevando a la eliminación de la estructura por la restricción de choques. En consecuencia las poblaciones

intermedias de los procesos de optimización poseen mayor probabilidad de estar constituidas por soluciones asociadas a conformaciones abiertas de la proteína. Además al muestrearse una conformación compacta, sin colapso, existe igualmente la posibilidad de que no se encuentre entre las mejores soluciones de la población y por tanto tal solución puede no ser seleccionada por el criterio de la FO. Esta segunda función permite ponderar el valor de la FO con el inverso del área superficial de modo que aquellas estructuras no colapsadas y compactas adquieran mayor posibilidad de prevalecer a lo largo de la optimización.

$$F2 = \frac{FO}{Area}$$

Ecuación III-3: Segunda función de Apoyo (F2)

III. 5. Descripción del paquete de metaheurísticas

En este epígrafe se describen las características esenciales para el diseño e implementación del software con las tres metaheurísticas. Se toman como precedentes de la tesis los esquemas de VMO propuestos por Puris y colaboradores (56), utilizados en la tesis de More-Quintero (22). El diseño e implementación de SA son los presentados en la tesis de Wong-Delgado (23) y en el evento RECPAT 2015 (24). La formalización del GA ha sido expresada en el evento RECPAT 2015 (25).

Cada elemento a optimizar representa una estructura tridimensional de la proteína. Este elemento en la metaheurística VMO es llamado “puntos de la malla”, en GA es llamado “individuo” y en SA es llamado “estado”. Cada elemento está compuesto por un arreglo de tamaño 2N, donde N representa la cantidad de aminoácidos en la proteína, cada elemento del arreglo representa el valor del ángulo de torsión ψ y φ de la cadena central. Todos los elementos obtenidos durante la ejecución de los algoritmos no necesariamente tiene que ser válidos, pues se pueden generar estructuras colisionadas (dos o más átomos se encuentran muy próximos entre sí). Un elemento puede ser seleccionado como el mejor si no está colisionado y es mejor que un que un elemento elegido anterior como mejor.

III. 5. 1. Optimización basada en Mallas Variables

Recordando el diseño de VMO (ver epígrafe II. 3. 1), se puede decir que VMO se basa en un proceso de expansión-compresión. El proceso de expansión que se presenta en esta metaheurística logra alcanzar el triplo del tamaño de la población inicial (llamada malla inicial o MI), esta nueva población formada se llama malla total (MT). En la implementación realizada de la metaheurística, la MT se inicializa con los puntos de la MI, por lo que es necesario generar el doble de puntos presentes en la MI para alcanzar el tamaño fijado de la MT (3*MI). Esta expansión ocurre en tres fases de generación:

1. atendiendo a la vecindad (hacia un mínimo local)
2. hacia el mejor global
3. mutación

La cantidad máxima de puntos a generar en la primera fase equivale a MI-1 (solo se alcanza este máximo si todos los puntos tienen un vecino con mejor FO que él). En el proceso de generar puntos hacia un mínimo local se utilizan dos puntos donde: uno el punto del cual obtendremos su mejor vecino y su mejor vecino. Para obtener el nuevo punto se cruzan y mutan determinadas zonas ente los puntos. La cantidad de posiciones a cambiar recibe el nombre de porciento de variables a cambiar. Los segmentos a cruzar se encuentran entre dos posiciones de cambio o entre el inicio de la proteína y el primer punto de cambio o entre el último punto de cambio y el final de la proteína. En la segunda fase se generan como máximo MI-1 puntos, el proceso se realiza igual, pero cambiando el mejor vecino por el mejor global.

En la tercera fase como mínimo se generan dos puntos pero se generan tantos puntos como se necesiten para completar la MT. En la última fase se escoge de la MI un punto aleatoriamente y en cada punto se cambian los valores de los dos diedros de los aminoácidos seleccionados a través del porciento de variables a cambiar.

Un aspecto a considerar en VMO es la forma en que se reduce la malla total a 1/3 de su tamaño para conformar una nueva la malla inicial. Este proceso se hace depender de dos parámetros %FO y %F1 que corresponde al porciento de puntos

a seleccionar por función objetivo y porciento de puntos a seleccionar por la primera función complementaria (ver epígrafe III. 4). Si con los valores que se define estos parámetros no se llega a obtener el 100% de los puntos de la MI, los puntos que falten se seleccionan con la segunda función complementaria.

III. 5. 2. Recocido Simulado

La metaheurística SA, no es poblacional, es decir, comienza con un estado aleatorio y este se va optimizando. Con el objetivo de obtener un mejor rendimiento computacional (explotando la capacidad de cómputo existente), se decide modificar el algoritmo para que trabaje en paralelo. En la implementación en paralelo se decide ubicar un nodo diferente en cada hilo, con temperaturas iniciales perturbadas con respecto a la inicial fijada, y ejecutar en cada hilo un SA lineal. Para que cada nodo comience con una temperatura inicial diferente se introduce un parámetro de relajación de dicha temperatura. el cual garantiza valores diferentes de temperatura durante su descenso entre los distintos hilos aumentando las posibilidades de exploración del algoritmo.

Sobre los métodos de enfriamientos propuestos para este algoritmo existen diversas investigaciones (19, 54, 91, 92). Estos métodos son los encargados de regular el decrecimiento de la temperatura cuyo valor determina en qué medida pueden ser aceptadas soluciones vecinas. Mientras más lento sea el descenso de la temperatura hay mayor posibilidad de que converge al óptimo global. A continuación se exhiben los métodos implementados. Aclaraciones Ti: Temperatura en la iteración actual y To: Temperatura inicial.

Descenso Geométrico: $Ti+1 = \alpha * Ti; \alpha \in [0.8, 0.99]$

Criterio de Boltzmann: $Ti = To/ (\log (i))$

Esquema de Cauchy: $Ti = To/ (1+i)$

Lundi-Mess: $Ti+1 = Ti / (1+\beta*Ti)$ con β muy pequeña

En la tesis de Wong-Delgado (23) se realiza un estudio de el método de enfriamiento a utilizar aplicado al problema de predicción de proteínas, donde se obtuvo que Lundi-Mess con un $\beta=0.004$ brindó mejores resultados. En la implementación el descenso de la temperatura se realiza en cada iteración del algoritmo, siendo esta la forma clásica de implementarlo.

A partir de un nodo actual, es necesario encontrar un sucesor que puede ser aceptado o no como nuevo nodo actual. De cada nodo se obtienen hasta cinco sucesores, el primero que no se encuentre colisionado es considerado el sucesor, si los cinco sucesores obtenidos se encuentran colisionados es devuelto uno de ellos. Para obtener cada sucesor se realizan mutaciones en los ángulos de rotación (la cantidad de cambios a efectuar se denomina diedros a mutar) de los aminoácidos que conforman la proteína. En cada aminoácido a mutar durante el proceso de generar un sucesor tienen igual probabilidad de modificarse cualquiera de los dos ángulos diedros ψ y φ.

III. 5. 3. Algoritmo Genético

El Algoritmo Genético implementado posee algunas características específicas las cuales se describen a continuación. Para ello es necesario ilustrar el ciclo de vida de un AG, ver Figura III-2:

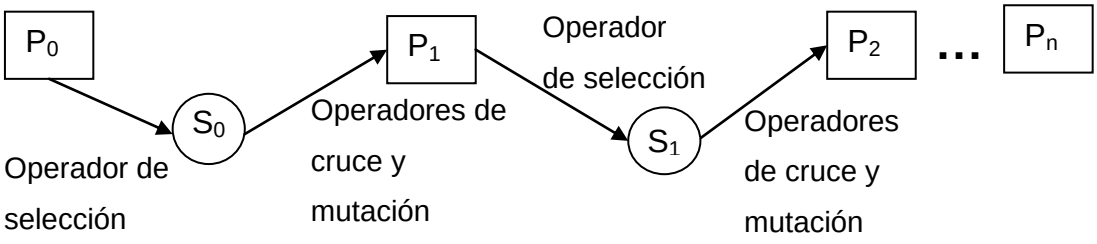


Figura III-2: Ciclo de Vida de un Algoritmo Genético

Donde P_i ($0 < i <= n$) son las poblaciones a obtener y S_j ($0 < j < n$) son las poblaciones intermedias. En la aplicación que se realiza en este trabajo del método GA se adoptan las siguientes características (25):

i) Cada individuo cuenta con una cantidad de genes equivalentes al doble de la cantidad de aminoácidos.
ii) Cada par de genes son los ángulos diedros de un mismo aminoácido (que determinan su conformación tridimensional).
iii) La población P_0: se realiza mediante una siembra de individuos
iv) Para obtener una nueva población P_j se selecciona un subconjunto de los individuos y se ubican en la población intermedia S_{j-1}.

La selección de los individuos para la población intermedia (S_j) cuenta con dos parámetros modificables en el algoritmo los cuales representan el porciento de los individuos seleccionados: por mejor función objetivo, por la primera función complementaria. Lo que falte para llegar al 100 % de la suma de los dos parámetros es el porciento de individuos a seleccionar por la segunda función complementaria (25). Se seleccionan dos individuos de S_j y se cruzan con cierta probabilidad, al cruzarlos se generan dos individuos y son agregados a la nueva población. Los nuevos individuos son mutados con cierta probabilidad, los mutados se agregan a la nueva población. Para mutar un individuo se seleccionan los diedros correspondiente con el 1% de los aminoácidos presentes en la proteína, pero de cada aminoácido se muta con igual probabilidad uno de los dos diedros. Este proceso de cruce y mutación acurre hasta que la nueva población alcanza la cantidad máxima de individuos.

III. 5. 4. Generación de las soluciones iniciales

Desde principios de los años 60 una vez que fueron obtenidas las primeras estructuras con alta resolución de la mioglobulina (primera proteína caracterizada estructuralmente mediante difracción de rayos-X, por John Kendrew) (93) se logró identificar que casi todos los aminoácidos que constituían el interior de la proteína eran de naturaleza hidrofóbica (94). Tal evidencia dio lugar al primer principio general sobre los factores que estabilizaban la estructura compacta de las proteínas globulares, en el que la proteína tiende a agrupar sus cadenas laterales para formar un núcleo hidrofóbico y una superficie hidrofílica. Este problema es resuelto mediante la formación de estructuras secundarias regulares, de las cuales se reconocen dos tipos fundamentales, las hélices-a y las hojas-ß, ambos tipos están caracterizados por formar puentes de hidrógeno entre los grupos de la cadena central.

Estas estructuras se forman cuando un número de aminoácidos consecutivos tienen los valores de sus ángulos psi y phi dentro de una rango muy estrecho valores (hélice-α: phi = -60, psi = -45 a -50; hoja-β: phi = -129, psi = 124)(31). Las hojas beta además deben cumplir con la orientación paralela o antiparalela de diferentes segmentos de la cadena, los cuales se unen por interacciones de

puentes de hidrogeno. La coincidencia de los valores de estos ángulos en un segmento de una cadena polipeptídica es un evento matemáticamente muy poco probable al ocurrir mediante una generación aleatoria de los valores de estos ángulos en un método de optimización. Por tal motivo estrategias que induzcan el muestreo de tales estructuras es necesario para generar soluciones óptimas que posean estructuras secundarias.

En la implementación de las metaheurísticas se asegura la representatividad de las estructuras secundarias mediante la introducción aleatoria de las posiciones dentro de cada individuo donde se asignarán inicialmente estas estructuras con el fin de permitir que el propio modelo de optimización durante la evolución reagrupe, reproduzca o elimine dichas estructuras de acuerdo a la estabilidad de las estructuras intermedias generadas con segmentos de estructuras secundarias. Tal criterio está en correspondencia con uno de los modelos de plegamientos aceptados en la actualidad conocido como modelo jerárquico o de difusión – colisión (95) el cual supone la formación de la mayor parte de los motivos de estructuras secundarias en etapas iniciales del proceso de plegamiento.

Para introducir elementos de estructuras secundarias en las optimizaciones la población inicial de los métodos de exploración se genera con mitad de las conformaciones con segmentos en hélice y la otra mitad con segmentos en hoja beta. Se emplean los parámetros %Helice y %Beta correspondientes a los porcientos de aminoácidos con valores propios de cada estructura secundaria por solución, de forma que en la parte de las soluciones generadas con hélices cada conformación tendrá como mínimo un número de aminoácidos asociados a %Helice con valores típicos de hélices en tanto el resto son generados aleatoriamente. Lo mismo se realiza para la mitad de soluciones iniciales con segmentos tipo hoja beta pero empleando el parámetro %Beta. Los valores aproximados de estas estructuras secundarias en las proteínas catalogadas como “todo alfa” (no tienen conformaciones de hoja beta) y “todo beta” (no tienen conformaciones en hélice alfa) se encuentran entre el 60 y el 75% de las conformaciones.

En el epígrafe II. 3. 4 se plantea que VMO y GA son poblacionales, por lo que necesitan generar una población inicial. SA no es poblacional pero como se explica en el epígrafe III. 5. 2 está implementado para trabajar en paralelo entonces necesita generar los nodos iniciales. En las tres metaheurísiticas los nodos iniciales se generan de igual forma. El algoritmo para generar los nodos iniciales se muestra en el seudocódigo siguiente.

```
Inicializar K con la cantidad de nodos a generar
Inicializar Nodos como arreglo de tamaño K
Inicializar i = 0
nodoTodoHelice = Generar_nodo_todoHelice()
nodoTodoBeta = Generar_nodo_todoBeta()
Mientrar i < K/2 hacer
        Nodos[i] = Generar_nodo_helice( nodoTodoHelice, %Helice)
Mientrar i < K hacer
        Nodos[i] = Generar_nodo_beta( nodoTodoBeta, %Beta)
Retornar Nodos
```

La mitad de los nodos se generan con un porciento mínimo de aminoácidos en hélice (método Generar_nodo_helice) y la otra mitad con un porciento como mínimo de aminoácidos en conformación de hoja beta (método Generar_nodo_beta). Para generar un nodo con un porciento como mínimo de conformaciones en hélice se utiliza un nodo que solo tiene conformaciones en hélice y se mutan aleatoriamente el porciento resultante de 100-%Helice. El proceso de generar nodos con un mínimo de porciento en beta es igual al de generar con un porciento mínimo en hélice explicado anteriormente. De esta forma se obtienen K nodos iniciales que contienen las estructuras secundarias.

III. 5. 5. Parámetros comunes y no comunes entres las metaheurísticas

Del estudio de las implementaciones de las tres metaheurísticas se determina que tienen algunos parámetros en común. Los parámetros %Helice y %Beta son utilizados para generar los nodos iniciales en cada algoritmo. El parámetro “Replicas” corresponde a la cantidad de soluciones obtenidas con la misma configuración. “Hilos” responde a la cantidad de hilos de ejecución que utilizaran

los algoritmos. La cantidad de evaluaciones máxima de la función objetivo está representada por “Eval FO”. Las metaheurísticas optimizan la cadena central de aminoácidos pero, las cadenas laterales también son importantes en la obtención de las conformaciones tridimensionales de las proteínas; el parámetro “Rotamer” representa la cantidad de optimizaciones aleatorias a realizar en la cadena lateral cada vez que se encuentra una solución mejor.

Tabla III-1: Parámetros comunes entre las tres metaheurísticas

Parámetro	**Parámetro**	**Parámetro**
%Helice	Replicas	Eval FO
%Beta	Hilos	Rotamer

Del estudio de las implementaciones se tiene que cada algoritmo tiene algunos parámetros propios los cuales influyen en la calidad de las soluciones a obtener. Estos parámetros son:

Tabla III-2: Parámetros específicos por metaheurística

Optimización Basada en Mallas Variables	**Recocido Simulado**	**Algoritmo Genético**
% var cambiar	T. inicial	% cruc
% FO	Diedro	% mut
% F1	Delta Temp.	Prob. selección FO
	Enfriamiento	Prob. selección F1

En Optimización Basada en Mallas Variables *% var cambiar* representa el porciento de variables a cambiar en cada fase del proceso de expansión. En la selección de la población se utiliza el *% FO* que consiste en el porciento de selección mediante la función objetivo y *% F1* representa el porciento de selección por la segunda función.

En Recocido Simulado, la temperatura inicial se especifica con *T. inicial*. La cantidad de diedros a rotar en el proceso de selección del sucesor es el parámetro *Diedro*. Para obtener una relajación de la temperatura se utiliza un valor mayor a cero en el parámetro *Delta Temp*. El método de enfriamiento de la temperatura se expresa con *Enfriamiento*, donde: 1 es Boltzmann; 2 es Cauchy; 3 es Geométrico; 4 es Lundi-Mess.

En Algoritmo Genético, *% cruc* representa el porciento de variables a cruzar y *% mut* representa el porciento de variables a mutar para obtener una nueva población a partir de una población intermedia. En GA al igual que en VMO se

realiza una selección de la población, en este caso para obtener la población intermedia, en este algoritmo intervienen los parámetros, *prob. selección FO* y *prob. selección F1,* que representan, el porciento de la población a seleccionar, la probabilidad de seleccionar un individuo respecto al función objetivo y respecto a la segunda función respectivamente.

III. 5. 6. Diagramas de clases

Descritas las características de restricciones del problema, espacio de búsqueda, función objetivo y metaheurísticas se decide implementar un paquete llamado *heurística*. Este paquete consta con dos clases principales: **Metaheuristica** y **Node**. En la Figura III-3 se presenta el diagrama de clase del paquete.

La clase **Metaheurística** es una clase abstracta que cuenta con los atributos y métodos mínimos necesarios para implementar cualquier metaheurística. Entre los atributos básicos se encuentran: Cantidad de evaluaciones realizadas de la FO, cantidad máxima de evaluaciones de la FO, mejor nodo encontrado, Fichero de salida, cadena de aminoácidos de la proteína, etc. Los métodos que tiene implementados son comunes para cualquier metaheurística los cuales son: impresión del estado del algoritmo, impresión del mejor nodo, realizar una optimización de las cadenas laterales (método **rotamers**) en el mejor nodo y devolver la cantidad de evaluaciones de la función objetivo realizadas hasta el momento.

En este proceso de homogenizar las tres implementaciones, se decide realizar una clase llamada **Proteins_Prediction** encargada de leer el fichero de entrada y atendiendo a los parámetros de ejecución ejecutar la metaheurística apropiada.

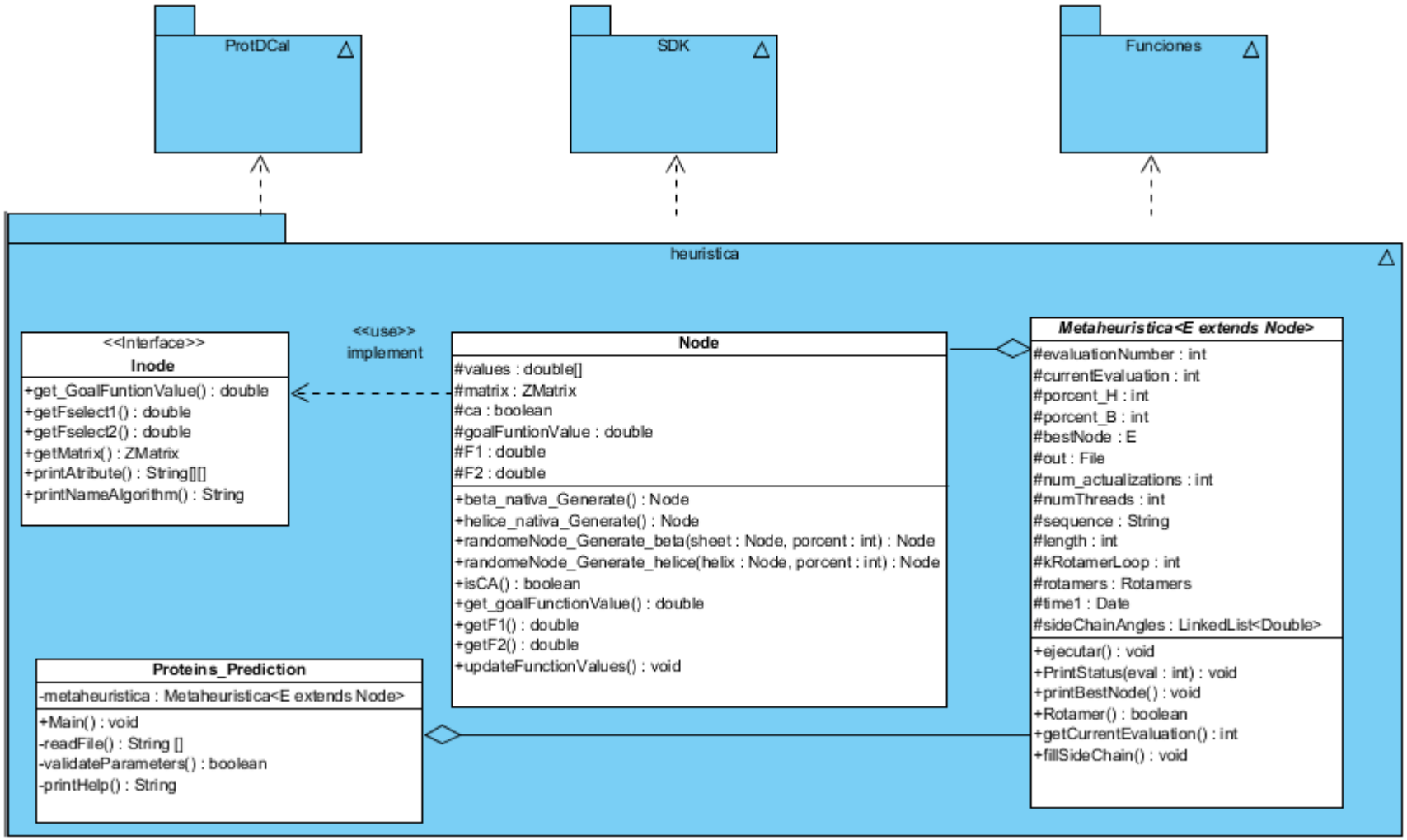


Figura III-3: Descripción del paquete heuristica

Un objeto de la clase **Node** representa una solución potencial del problema real. La clase **Node** está formada por un conjunto de ángulos de torsión que son almacenados en un arreglo llamado **values**. Esta clase contiene los métodos para generar los nodos aleatorios con un porciento de la estructura secundaria. Tiene dos métodos encargados de generar nodos con todos los aminoácido en una u otra conformación (hélice o en hoja beta). El arreglo que almacena dichos ángulos depende, en gran medida, de la longitud de la secuencia de aminoácido (N) de la cadena que se analiza y viene dada por la Ecuación III-4:

$$Dim_{Val} = 2 * N - 1$$

Ecuación III-4: Cantidad de ángulos diedros en un nodo

En la clase **Node** hay un atributo con el valor calculado de la función objetivo (**goalFunctionvalue**), otro con el valor de la primera función complementaria (**F1**) y otro con la segunda función complementaria (**F2**). El método **updateFunctionValues** es el encargado de recalcular las funciones mencionadas anteriormente cuando se realiza algún cambio en los ángulos de torsión.

Luego de implementar las clases básicas es necesario crear un paquete por cada metaheurística (Variable_Mesh_Optimization, Simulating_Annealing y Genetic_Algoritm). En cada paquete aparecen dos clases: una con el nombre de la metaheurística que extiende de la clase abstracta Metaheurística y define la implementación; la otra define las características del nodo asociado y extiende de Node. En la Figura III-4 se muestra la estructura de los paquetes tomando como ejemplo el paquete asociado a Recocido Simulado. Cada metaheurística implementa el método abstracto **ejecutar**, este método utiliza objetos de la clase Thread para realizar la ejecución en paralelo atendiendo a la cantidad de hilos presente en el atributo **numThread** de la superclase Metaheurística.

En la implementación el nodo de Recocido Simulado (**NodeSA**) extiende Node y adiciona los atributos de temperatura (Temperatura Inicial relajada y temperatura en la cual se obtiene el nodo).

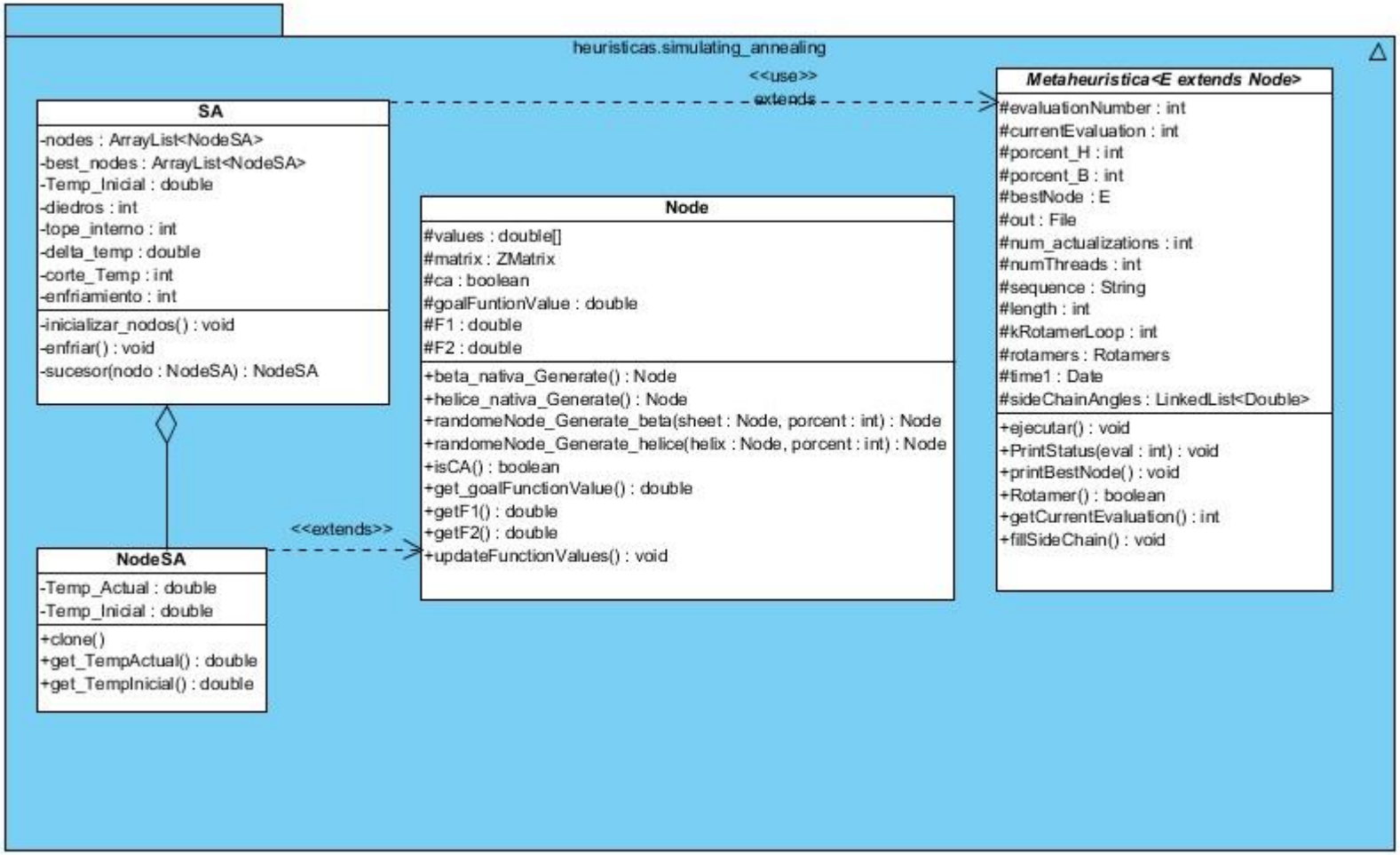


Figura III-4: Descripción del paquete de recocido simulado

III. 5. 7. Entrada y Salida de las metaheurísticas

Entre las tres metaheurísticas existían diferencias entre los ficheros de entrada de datos. Como ejemplo tenemos que VMO y SA tenían un solo fichero de entrada donde aparecían las secuencias y seguidamente los parámetros, pero entre ellas no coincidían los nombres de los parámetros comunes. GA tenía dos ficheros de entrada uno donde se especificaban las secuencias de aminoácidos y otra con los parámetros a utilizar en la ejecución. Además existían diferencias en los ficheros de salida, tanto en las especificaciones del pdb y en informar la evolución de los algoritmos en la salida estándar. Debido a esto se decide homogenizar los datos de entrada y salida de las implementaciones.

El fichero de entrada está compuesto por el nombre de cada proteína y su secuencia de aminoácidos. En la Tabla III-3 se presenta un ejemplo del fichero de entrada. Las líneas que comienzan con dos barras (//) se consideran comentarios y el software las omite. Los parámetros para ejecutar cada metaheurística son pasados por parámetro. En la se describen los parámetros.

Tabla III-3: Descripción de los parámetros a utilizar en la línea de comandos

Parámetro	Descripción	Valores posibles	Valor por defecto	Disponible para
-a	Algoritmo a ejecutar	1: VMO, 2: GA ó 3: SA	No tiene. Este es obligatorio	Todos
-f	Nombre del fichero con las proteínas	Cadena de texto	data.in	Todos
-h	% de hélice, expresado en probabilidad	(0; 1)	0.6	Todos
-b	% de beta, expresado en probabilidad	(0; 1)	0.75	Todos
-t	Cantidad de hilos	[2,+∞]	Núcleos de la PC	Todos
-e	Cantidad de evaluaciones de la FO	> 0	20 000	Todos
-s	Cantidad de réplicas	>= 1	1	Todos
-r	Cantidad de optimizaciones de las cadenas laterales	>= 0	10	Todos
-x	%FO, expresado en	(0; 1)	0.5	VMO, GA

	probabilidad			
-y	%F1, expresado en probabilidad	(0; 1)	0.3	VMO, GA
-m	% variables a cambiar, expresado en probabilidad	(0; 1)	0.01	VMO
	Probabilidad de mutar	(0; 1)	0.2	GA
	Cantidad de diedros a mutar	> 1	1	SA
-c	Probabilidad de cruce	(0; 1)	0.7	GA
-i	Temperatura inicial	>0	200	SA
-d	Delta de relajación de la temperatura	>= 0	10	SA
-e	Método de enfriamiento	1: Cauchy 2: Boltzman 3: Geométrico 4: Lundi-Mess	4	SA

En las metaheurísticas implementadas se sigue obteniendo dos tipos de salidas. La primera es un archivo de extensión pdb, por cada réplica a obtener. En la segunda salida se obtiene la evolución del algoritmo mostrando todas las actualizaciones que se realizan hasta obtener el mejor nodo, además aquí se imprimen los tiempos de ejecución. El texto resultante de la salida estándar es automáticamente redireccionado a un fichero, que se puede procesar para obtener las estadísticas. En la Figura III-5 se muestra un ejemplo se salida de la metaheurística GA.

```
Folding protein: R0001_GA_80_25
Generating initial population 274
0 FO(DGsc) -718.164484982283 | Bsht_DGtor: -147.9632350690746 | #FO 274 | Time(s): 98.459
Update Rotamers
1 FO(DGsc) -750.9334150121102 | Bsht_DGtor: -147.9632350690746 | #FO 274 | Time(s): 110.055
2 FO(DGsc) -760.190606908653 | Bsht_DGtor: -147.54194725258864 | #FO 548 | Time(s): 230.291
3 FO(DGsc) -760.798303816269 | Bsht_DGtor: -148.18153644143817 | #FO 822 | Time(s): 332.735
Update Rotamers
4 FO(DGsc) -761.1857031701267 | Bsht_DGtor: -148.18153644143817 | #FO 822 | Time(s): 344.802
5 FO(DGsc) -780.838728438667 | Bsht_DGtor: -149.07570968030467 | #FO 1096 | Time(s): 471.041
```

Figura III-5: Fichero resultante de la Salida estándar

En la Figura III-6 se presenta un ejemplo de fichero con extensión pdb resultante de la metaheurística SA. Esta salida muestra el algoritmo empleado y los parámetros con que se ejecutó. Este fichero puede ser procesado por softwares destinados a la visualización computacional de estructuras químicas como son Chimera, RasMol2, SPDBV, etc. Las tres últimas columnas representan las

posiciones de cada átomo en el sistema de coordenadas cartesianas, obtenidas por el SDK a partir de las coordenadas internas.

```
REMARK Structure Generated using
REMARK Simulating Annealing
REMARK and ProtDCal's Folding Scoring Function
REMARK Parameters
REMARK -a 3 -h 0.6 -b 0.75 -t 4 -r 10 -e 1000
REMARK  -m 1 -i 100 -k 4 -d 10
REMARK Score (interfacial + close-packing free energy):
REMARK -516.953
REMARK Beta-Sheet-Weighted torsion potential:
REMARK -79.60639889511033
REMARK Temperature:
REMARK 29.840848806366047 (90.0)
REMARK # of Evaluations (Total): 996 (1000)
REMARK Time: 0.4326552777777778
ATOM      1  N   MET A   1       0.000   0.000   0.000
ATOM      2  CA  MET A   1       1.438   0.000   0.000
ATOM      3  CB  MET A   1       1.946   1.436   0.000
ATOM      4  C   MET A   1       1.940  -0.711   1.233
ATOM      5  CG  MET A   1       3.469   1.436  -0.045
ATOM      6  SD  MET A   1       4.025   1.121  -1.744
ATOM      7  CE  MET A   1       3.019   1.684  -3.147
ATOM      8  O   MET A   1       1.276  -0.712   2.242
ATOM      9  H   MET A   1      -0.477  -0.439   0.761
ATOM     10  N   ARG A   2       3.245  -1.380   1.132
ATOM     11  CA  ARG A   2       3.785  -2.077   2.268
```

Figura III-6: PDB resultante de la metaheurística SA

III. 6. Análisis de los parámetros a estudiar

Para comparar algoritmos es necesario realizar un estudio de los parámetros, pues estos determinan la calidad de las soluciones. En esta tesis se decide fijar algunos parámetros que son comunes en las metaheurísticas. Los valores asociados a los parámetros %Helice y %Beta fueron estudiados anteriormente para GA (25), debido a la similitud de generar la población inicial se tomaron los mismos valores para las tres metaheurísticas. En estudios preliminares se utiliza una cantidad de 10000 evaluaciones de la FO, considerando que el valor de las mejores soluciones obtenidas, se encontraban distantes del mínimo valor a alcanzar, se duplicó este valor. La cantidad de hilo se ha fijado en 12 con el

objetivo de utilizar eficientemente los recursos del clúster y reducir el tiempo de ejecución.

Tabla III-4: Parámetros comunes ente las tres metaheurísticas. Valores fijados

Parámetro	Valor
%Helice	60
%Beta	75

Parámetro	Valor
Replicas	3
Hilos	12

Parámetro	Valor
Eval FO	20 000
Rotamer	10

III. 6. 1. Configuraciones para Optimización basada en Mallas Variables

De los parámetros propios de VMO, se decide estudiar el porciento de variables a cambiar en cada punto durante el proceso de expansión (%var cambiar). Se obtienen las 4 configuraciones siguientes donde en la proteína de 100 aminoácidos (200 diedros) el 0.5% corresponde a cambiar un diedro y 2% corresponde a cambiar 4 diedros:

Tabla III-5: Configuraciones a estudiar en VMO

Configuración	% var cambiar
VMO_0-5	0.5
VMO_1	1

Configuración	% var cambiar
VMO_1-5	1.5
VMO_2	2

Se mantienen comunes los parámetros %FO y %F1 en las 4 configuraciones. Seleccionados para evaluar la calidad de la FO, con una utilidad del 50% para reducir la MT, y las otras dos funciones complementarias con utilidades menores (primera Función complementaria con 30% y segunda con 20%).

Tabla III-6: Parámetros comunes en las configuraciones de VMO

Parámetro	Valor
% FO	50
%F1	30

III. 6. 2. Configuraciones para Recocido Simulado

En el SA se estudian dos parámetros: “valor de la Temperatura inicial” y “cantidad de diedros a rotar” para encontrar un sucesor, obteniendo 16 configuraciones. Se decide estudiar estos parámetros por ser los más importantes ya que la temperatura inicial debe garantizar la aceptación de nodos para poder salir de óptimos locales y la cantidad de diedros a mutar es imprescindible en el proceso de selección de un sucesor. La temperatura inicial 200 fue utilizada en la tesis de Wong-Delgado (23) y en esta tesis se decide estudiar algunas temperaturas superiores con el objetivo de incrementar la probabilidad de salir de óptimos

locales, se utiliza una temperatura inferior de estudiar el comportamiento a temperaturas más bajas. La cantidad de diedros a mutar sean desde 1 hasta 4 diedros. Mientras más diedros se muten a la vez mayor probabilidad existe de obtener estructuras colisionadas.

Tabla III-7: Configuraciones a estudiar en RS

Configuración	T_inicial	Diedros
SA_1_100	100	1
SA_1_200	200	1
SA_1_400	400	1
SA_1_600	600	1
SA_2_100	100	2
SA_2_200	200	2
SA_2_400	400	2
SA_2_600	600	2

Configuración	T_inicial	Diedros
SA_3_100	100	3
SA_3_200	200	3
SA_3_400	400	3
SA_3_600	600	3
SA_4_100	100	4
SA_4_200	200	4
SA_4_400	400	4
SA_4_600	600	4

Los demás parámetros se mantienen comunes en las 16 configuraciones como se muestra en la Tabla III-8. El valor del parámetro enfriamiento fue objeto de estudio de la tesis de grado de Wong-Delgado Fernando (23) y presentado los resultados en el evento RECPAT 2015 (24). El parámetro delta de la temperatura introducido en esta implementación se le asignó un valor que no permita solapamiento entre las temperaturas a estudiar.

Tabla III-8: Parámetros comunes en las configuraciones de RS

Parámetro	Valor
Enfriamiento	4
Delta Temp.	50

III. 6. 3. Configuraciones para Algoritmos Genéticos

Se pretende estudiar dos parámetros: “el porciento de mutación” y “el porciento de cruce”, obteniendo las 9 configuraciones que se listan a continuación. Esta metaheurística depende de estos parámetros para cambiar la población inicial, por ello son los de mayor interés para estudiar.

Tabla III-9: Configuraciones a estudiar en GA

Configuración	prob cruc	prob mut
GA_70_15	0.70	0.15
GA_70_20	0.70	0.20
GA_70_25	0.70	0.25
GA_75_15	0.75	0.15

Configuración	prob cruc	prob mut
GA_75_25	0.75	0.25
GA_80_15	0.80	0.15
GA_80_20	0.80	0.20
GA_80_25	0.80	0.25

GA_75_20	0.75	0.20

Los demás parámetros se mantienen comunes en las 9 configuraciones. Seleccionados para evaluar la calidad de la FO, con una utilidad del 50% para reducir la MT, y las otras dos funciones complementarias con utilidades menores (primera Función complementaria con 30% y segunda con 20%).

Tabla III-10: Parámetros comunes en las configuraciones de GA

Parámetro	**Valor**
Prob. selección FO	0.5
Prob. selección F1	0.3

III. 7. Descripción de la base de casos

Características para la selección de las proteínas:

1. Que tienen reportada su estructura Nativa en el PDB. De esta forma se puede comparar las soluciones obtenidas en las metaheurísticas con la estructura tridimensional real.
2. Que pertenezcan a la competencia CASP-ROLL. En esta competencia se brindan públicamente las soluciones de los competidores para cada proteína donde compitieron, obteniendo así una fuente importante de algoritmos con los cuales comparar las soluciones de las metaheurísticas en cuanto a la calidad.
3. Que tengan en su secuencia entre 100 y 300 aminoácidos. Debido al costo computacional de generar soluciones de proteínas grandes (> 300 aminoácidos) se decide trabajar con las que presentan una cantidad de aminoácidos inferior a 300, pero con una cantidad mínima de aminoácidos para considerarlas proteínas.

Con las características anteriores se seleccionaron 26 proteínas (ver VIII. 2 VIII. 2). En la Tabla III-11 se muestran algunas características de estas proteínas, donde N1 es el nombre de la proteína propuesto por el CASP, N2 es el nombre indexado por el PDB y Longitud es la cantidad de aminoácidos en la proteína.

Tabla III-11: Descripción de las proteínas a utilizar

N1	**N2**	**Longitud**
R0001	4A0U	183
R0002	4BQ6	197

N1	**N2**	**Longitud**
R0023	4IAJ	100
R0025	4HSP	172

R0006	4E0E	196
R0007	4DKC	161
R0008	4EW7	128
R0009	4DMI	176
R0013	4ECN	127
R0014	4ECN	136
R0015	4E9K	261
R0018	4EBG	124
R0020	4HLB	138
R0021	4HWM	144
R0022	4JGL	191

R0026	4D0V	115
R0027	3ZPE	158
R0029	4BJM	235
R0031	4IXJ	283
R0033	4LG3	211
R0034	4L3U	138
R0038	4UW7	271
R0040	3ZCJ	205
R0043	4Q0Y	163
R0044	4QE0	204
R0046	4PWU	128

III. 8. Conclusiones Parciales

La estructura computacional de las proteínas se representa como un arreglo con los valores de los ángulos de torsión, a esta representación se le llama sistema de coordenadas internas. Para evaluar algunas restricciones y crear los ficheros pbd es necesario convertir de este sistema al de coordenadas cartesianas. Se describe la función objetivo a utilizar para evaluar los elementos y se formalizan dos funciones complementarias para estudiar el espacio de búsqueda. Se enuncian detalles de la implementación de las tres metaheurísticas y de la homogenización para contruir un software que permita ejecutar de forma fácil las metaheurísticas. Se describen los parámetros a ajustear de las metaheurísticas y la base de casos donde se evaluarían los resultados.

IV. Resultados

Haciendo uso de la base de casos se realizan una serie de experimentos para analizar el comportameinto de las tres metaheurísticas y evaluar la calidad de las soluciones. Se detallan los experimentos y los resultados obtenidos en los mismos. En el análisis de los resultados se utilizan pruebas estadísticas de Friedman, Iman-Davenport y Holm implementadas en el software KEEL (96, 97) y la prueba de Wilcoxon del SPSS 16.0 (98).

IV. 1. Experimento 1: Estudio de parámetros en las metaheurísticas

El esquema general de este experimento consiste en los siguientes pasos:

1. Ejecutar los algoritmos con las configuraciones (generando 3 soluciones por configuración)
2. Por cada metaheurística
   2.1. Por cada proteína, seleccionar la configuración que obtiene el mínimo valor promedio de la función objetivo.
   2.2. Determinar la(s) mejor(es) configuración(es).

IV. 1. 1. Estudio de parámetros en VMO

Como resultados de aplicar este experimento se obtiene que para VMO: la mejor configuración resultó VMO_1 con 11 proteínas (representa el 42.31%) con el menor valor promedio de FO (ver Gráfico IV-1).

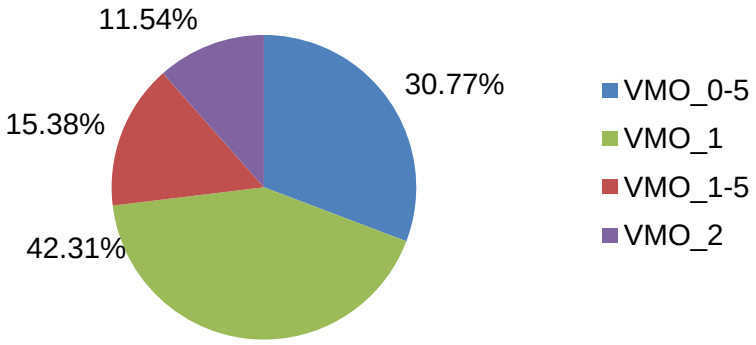


Gráfico IV-1: Distribución de las proteínas en VMO

Esto ha sido corroborado con la prueba estadística Friedman e Iman-Davenport (ver Tabla IV-1) donde se evidencia que entre las cuatro configuraciones de VMO se presentan diferencias estadísticas (ver Tabla IV-1).

Tabla IV-1: Prueba de Friedman e Iman-Davenport para VMO

	Friedman	Iman-Davenport
Grados de libertad	3	3 y 75
Chi-cuadrado	15.3692	6.1348
Significación	**0.0015**	**0.0009**

Algoritmo	Rank Friedman
VMO_0.5	2.3077
VMO_1	1.9615
VMO_1.5	2.4231
VMO_2	3.3077

Para evaluar las diferencias significativas se realiza la prueba post-hoc de Holm (ver Tabla IV-2) con los resultados de Friedman 1xN. Se evidencia que VMO_1 presenta diferencias significativas con VMO_2. Mientras mayor sea la cantidad de diedros a mutar las probabilidades de colisión aumentan, evidenciando que rotar el 2% de los ángulos presenta diferencias significativas en cuanto al valor de la FO.

Tabla IV-2: Prueba de Holm a partir de Friedman 1xN para VMO

	p	Holm	Hipótesis
VMO_2	0.0002	0.0167	Rechazada
VMO_1.5	0.1974	0.025	Aceptada
VMO_0.5	0.3337	0.05	Aceptada

IV. 1. 2. Estudio de parámetros en SA

Luego de realizar el paso 2.1 del Esquema general del experimento para la metaheurística SA (ver Anexo VIII. 4), no se encontraron resultados en las configuraciones donde la temperatura es 100, 400 y 600. De las cuatro configuraciones posibles se obtuvo que: la mejor resulta ser SA_1_200 con 10 proteínas (38.46%). Para más detalles consultar el Gráfico IV-2.

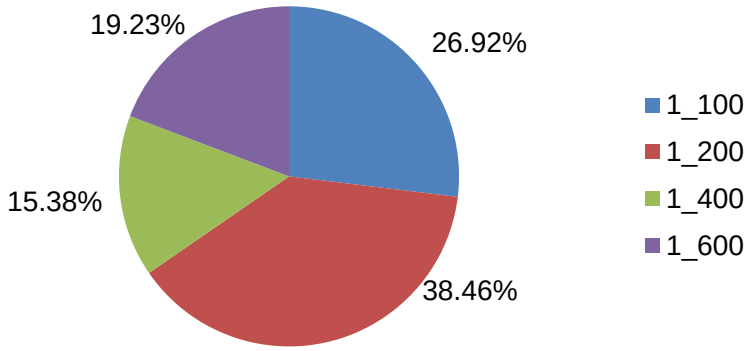


Gráfico IV-2: Distribución de las proteínas en SA

Esto ha sido corroborado con las pruebas estadísticas de Friedman e Iman-Davenport (ver Tabla IV-3) donde se evidencia que entre las 16 configuraciones de SA se presentan diferencias estadísticamente significativas.

Tabla IV-3: Prueba de Friedman e Iman-Davenport para SA

	Friedman	Iman-Davenport
Grados de libertad	15	15 y 375
Chi-cuadrado	345.6813	194.9973
Significación	**0.0000**	**0.0000**

Algoritmo	Rank Friedman
SA_1_100	2.4615
SA_1_200	2.0385
SA_1_400	2.7692
SA_1_600	2.8462
SA_2_100	6.5
SA_2_200	6.3077
SA_2_400	6.9615
SA_2_600	6.7308
SA_3_100	10.6154
SA_3_200	10.4423
SA_3_400	10.9038
SA_3_600	11.3846
SA_4_100	13.6538
SA_4_200	14
SA_4_400	14
SA_4_600	14.3846

Para evaluar las diferencias significativas se realiza la prueba post-hoc de Holm (ver Tabla IV-4) con los resultados de Friedman 1xN. Se evidencia que las 4 mejores seleccionadas (las que se presentaron en el Gráfico IV-2), no presentan diferencias significativas. Se selecciona SA_1_200 como la mejor configuración para la metaheurística SA.

Tabla IV-4: Prueba de Holm a partir de Friedman 1xN para SA

	p	Holm	Hipótesis
SA_4_600	0	0.0033	Rechazada
SA_4_200	0	0.0036	Rechazada
SA_4_400	0	0.0038	Rechazada
SA_4_100	0	0.0042	Rechazada
SA_3_600	0	0.0045	Rechazada
SA_3_400	0	0.005	Rechazada
SA_3_100	0	0.0055	Rechazada
SA_3_200	0	0.0062	Rechazada
SA_2_400	0.0002	0.0071	Rechazada
SA_2_600	0.0004	0.0083	Rechazada
SA_2_100	0.0007	0.01	Rechazada
SA_2_200	0.0012	0.0125	Rechazada
SA_1_600	0.5408	0.0167	Aceptada
SA_1_400	0.58	0.025	Aceptada
SA_1_100	0.7487	0.05	Aceptada

IV. 1. 3. Estudio de parámetros de GA

Luego de realizar el paso 2.1 del Esquema general del experimento para la metaheurística GA (ver Anexo VIII. 5), se obtiene que la mejor configuración es GA_70_20 con seis contiene la mayor cantidad de proteínas (23.08%). Para más detalles ver el Gráfico IV-3.

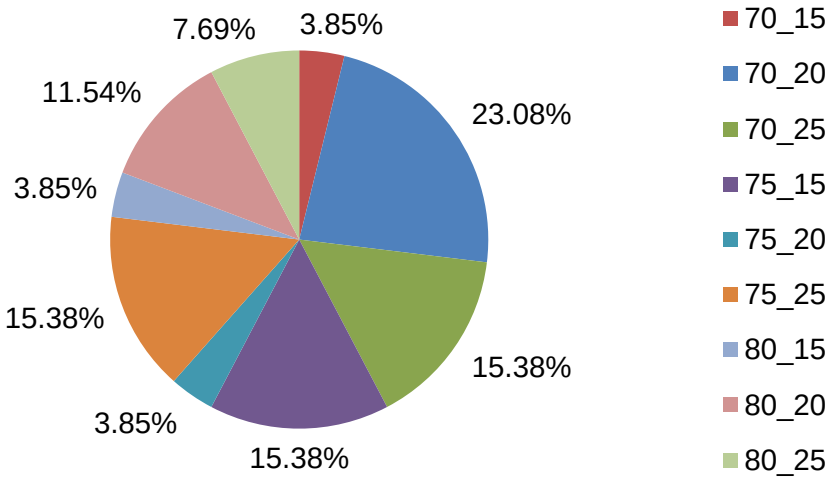


Gráfico IV-3: Distribución de las proteínas en GA

Las configuraciones de GA han sido analizadas con las pruebas estadísticas de Friedman e Iman-Davenport (ver Tabla IV-5) donde se evidencia que no hay diferencias significativas entre ellas. Se selecciona GA_70_20 como la mejor.

Tabla IV-5: Prueba de Friedman e Iman-Davenport para GA

	Friedman	Iman-Davenport
Grados de libertad	8	8 y 200
Chi-cuadrado	14.4513	1.8666
Significación	**0.0707**	**0.0671**

Algoritmo	Rank Friedman
GA_70_15	5.9231
GA_70_20	4.6154
GA_70_25	3.8846
GA_75_15	4.3846
GA_75_20	5.6538
GA_75_25	4.5
GA_80_15	5.2692
GA_80_20	5.8846
GA_80_25	4.8846

IV. 1. 4. Perturbación en los datos para el estudio de los parámetros de las metaheurísticas

El porciento de proteínas que coinciden con las configuraciones resultantes del experimento 1 (42.31%, 23.08%, 38.46% en VMO_1, GA_70_20 y SA_1_200

respectivamente) se encuentran por debajo del 50%. Debido a ello se decide utilizar un esquema de validación cruzada interna de 7-pliegues para confirmar la robustez de la selección de las configuraciones. En este esquema el modelo es cada metaheurística y lo que se pretende estimar son los valores mínimos de FO, para cada proteína atendiendo a los parámetros de las metaheurísticas. El esquema general de este experimento consiste en los siguientes pasos:

1. Conformar 7 grupos de proteínas mediante un análisis de conglomerados empleando descriptores estructurales de proteínas.
2. Por metaheurística, con un esquema 7-pliegues:
   2.1. Utilizar 6 grupos para entrenar (seleccionar la configuración que mayor cantidad de veces minimiza el valor promedio de FO) y predecir en el grupo restante con la configuración seleccionada.
   2.2. Evaluar la calidad de las soluciones, en las configuraciones, con las métricas brindadas por el servidor LGA.
3. Determinar la metaheurística con mejores resultados en su mejor configuración.

En el paso 1 se utilizaron 26 proteínas nativas (ver epígrafe III. 7). Se calculan los descriptores que utilizan los ángulos de torsión de las proteínas en el software ProtDCal (99), obteniendo un total de 644 descriptores. Luego se utiliza un esquema de formación de conglomerados empleando el coeficiente de correlación de Spearman (100, 101) para formar los clústeres empleando un valor de corte de 0.9, de modo que cada atributo (descriptor) de un conglomerado tiene al menos otro vecino con una correlación superior a este valor. Finalmente los centroides de cada conglomerado son elegidos como atributos representativos del conjunto de datos original. En dicho proceso se obtuvo 278 clústeres, resultando 278 descriptores, equivalente a una reducción del 56.83% de los descriptores. Luego se procede a utilizar el software IMMAN (102) para ordenar los atributos por su entropía de Shannon (103) en el conjunto de proteínas. El valor máximo de entropía para el conjunto de 26 proteínas es 4.7, decidimos fijar como valor de corte a 4.0 correspondiente a un 85% de la entropía total del conjunto de datos, quedándonos de esta forma con 6 descriptores que corresponden a los más

variables (entre todos los casos) de los 278 descriptores no redundantes elegidos previamente. En todos los procesos anteriores, la selección de los descriptores, se realiza automáticamente con el software Big-DataSet Manager (BDM por sus siglas) (104). Con el software BDM a partir de una lista de descriptores a seleccionar (obtenida con Spearman e IMMAN en cada proceso) se realiza la selección de los descriptores involucrados (105). En la Figura IV-1 se muestra el proceso de selección de descriptores abordado entre los pasos uno y dos.

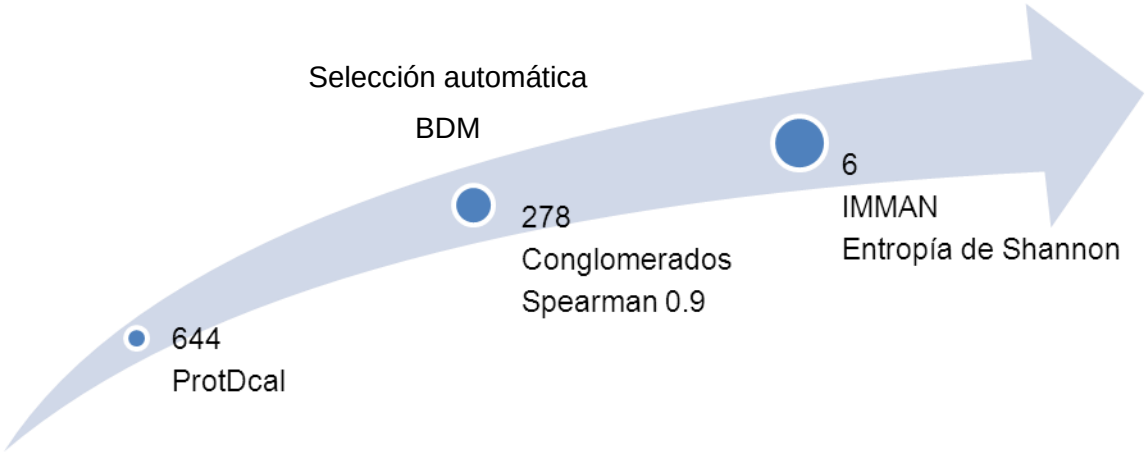


Figura IV-1: Proceso de selección de descriptores

Se utilizan los 6 descriptores y una configuración de clúster de enlace completo (Complete Linkage) para obtener 3 clúster (ver Dendograma en el Anexo VIII. 6). De los dos clúster más pequeños (cada uno de 7 proteínas) de forma aleatoria se extrae una proteína por clúster y del clúster más grande (12 proteínas) se extraen aleatoriamente dos para conformar un grupo con las cuatro proteínas. Este proceso se realiza hasta obtener los 7 grupos (ver Tabla IV-6).

Tabla IV-6: Descripción de los 7 grupos

Grup / Clust	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	R0008	R0046	R0007	R0015	R0031		
	R0025	R0038	R0022	R0043	R0044	R0018	R0021
2	R0033	R0009	R0026	R0001	R0002	R0020	R0023
3	R0034	R0006	R0029	R0013	R0040	R0014	R0027
Proteínas Entren.	22	22	22	22	22	23	23
Proteínas Predicc.	4	4	4	4	4	3	3

En cada pliegue se cuenta la cantidad de proteínas que obtienen menor promedio de FO por configuración y la configuración donde más proteínas tienen menor valor de FO es seleccionada.

En la Tabla IV-7 se muestra el resultado del esquema del experimento 2 para VMO. Se observa que siempre se obtuvo VMO_1 como la mejor configuración.

Tabla IV-7: Selección de configuraciones mediante 7-pliegues para VMO

1	**2**	**3**	**4**	**5**	**6**	**7**
VMO_1	VMO_1	VMO_1	VMO_1	VMO_1	VMO_1	VMO_1
9	9	10	8	10	11	9

En la Tabla IV-8 se muestra el resultado del esquema del experimento 2 para SA, la primera fila representa el grupo dejado fuera durante el entrenamiento, la segunda fila es la configuración resultante y la última es la cantidad de proteínas que obtuvieron esa configuración. Se observa que siempre se obtuvo SA_1_200.

Tabla IV-8: Selección de configuraciones mediante 7-pliegues para SA

1	**2**	**3**	**4**	**5**	**6**	**7**
SA_1_200	SA_1_200	SA_1_200	SA_1_200	SA_1_200	SA_1_200	SA_1_200
8	7	10	8	8	9	10

En la Tabla IV-9 se muestra el resultado del esquema del experimento 2 para GA. Cuando se deja el grupo tres fuera para el entrenamiento en el 7-fold se obtienen dos configuraciones con la misma cantidad de proteínas, estas son GA_70_25 y GA_75_25.

Tabla IV-9: Selección inicial de configuraciones mediante 7-pliegues para GA.

1	**2**	**3**	**4**	**5**	**6**	**7**
GA_70_20	GA_70_20	GA_70_25	GA_70_20	GA_70_20	GA_70_20	GA_70_20
		GA_75_25				
6	5	4	5	5	5	5

Se puede realizar una prueba de Wilcoxon (ver Tabla IV-10) donde no existen diferencias significativas entre las dos configuraciones (sin tener en cuenta los datos del grupo 3). Como mejor configuración se selecciona, la que gana una mayor cantidad de veces con menor valor de FO, siendo esta la GA_70_25.

Tabla IV-10: Prueba de Wilcoxon sin los casos del grupo 3

	GA_75_25 – GA_70_25
Z	-1.088[a]
Significación	**.277**

a. Basada en los rangos negativos.

	N	Media de rangos	Suma de rangos
Rangos Negativos	10[a]	9.30	93.00
Rangos Positivos	12[b]	13.33	160.00
Iguales	0[c]		
Total	22		

a.GA_75_25 < GA_70_25
b.GA_70_20 > GA_70_25
c.GA_70_20 = GA_70_25

Como resultado del esquema perturbado para GA se obtiene la Tabla IV-11.

Tabla IV-11: Selección inicial de configuraciones mediante 7-fold para GA

1	**2**	**3**	**4**	**5**	**6**	**7**
GA_70_20	GA_70_20	GA_70_25	GA_70_20	GA_70_20	GA_70_20	GA_70_20
5	5	6	5	5	6	5

Se obtuvo una perturbación en los datos ya que las proteínas del grupo 3 obtienen como mejor configuración la GA_70_25 y todas las demás proteínas la GA_70_20. Se analizan los valores de la FO con la perturbación y se comparan con la configuración obtenida para GA por el experimento 1 (GA_70_20). Al aplicar la prueba de Wilcoxon (ver Tabla IV-12) a la FO perturbada y a la configuración GA_70_20 no se observan diferencias significativas (significación de 0.273), ver Tabla IV-12.

Tabla IV-12: Prueba de Wilcoxon entre la FO perturbada y GA_70_20

	GA_70_20 - FO_perturbada
Z	-1.095[a]
Significación	**.273**

a. Basada en los rangos positivos.

	N	Media de rangos	Suma de rangos
Rangos Negativos	3[a]	2.67	8.00
Rangos Positivos	1[b]	2.00	2.00
Iguales	22[c]		
Total	26		

a.GA_70_20 < FO_perturbada
b.GA_70_20 > FO_perturbada
c.GA_70_20 = FO_perturbada

Se pudo determinar con este experimento la robustez del ajuste de los parámetros en las metaheurísticas.

IV. 1. 5. Conclusiones del Experimento 1.

Para realizar la selección de la mejor configuración por metaheurística se utilizaron tres técnicas: Seleccionar la configuración donde mayor cantidad de proteínas obtienen menor valor de FO, pruebas estadísticas: Friedman, Iman-Davenport y Holm y como tercer método la perturbación de los datos. Se concluye de este experimento que la mejor configuración de:

1. VMO resultó ser: VMO_1 en el 42.31% de las proteínas.
2. SA es: SA_1_200 con el 38.46% de las proteínas.
3. GA es: GA_70_20 en el 23.08% de las proteínas.

IV. 2. Experimento 2: Comparación entre las tres metaheurísticas

En la comparación de las metaheurísticas se tienen en cuenta varios aspectos: valor de la FO en el proceso de optimización, convergencia de la FO, tiempo de ejecución y calidad de las soluciones obtenidas. Los resultados de las comparaciones de la metaheurísticas en los criterios anteriormente mencionados son descritos en los siguientes epígrafes.

IV. 2. 1. Comparación respecto a valor de FO

A las tres mejores configuraciones obtenidas se les realizan las pruebas de Friedman e Iman-Davenport para comprobar si hay diferencias significativas entre las tres metaheurísticas. En la Tabla IV-13 se muestran los resultados de la prueba estadística donde se concluye que existen diferencias significativas entre las metaheurísticas.

Tabla IV-13: Prueba de Friedman e Iman-Davenport entre los valores promedio de FO de las tres metaheurísticas

	Friedman	Iman-Davenport
Grados de libertad	2	2 y 50
Chi-cuadrado	9.000	5.2326
Significación	0.0111	0.0086

Algoritmo	Rank Friedman
VMO	2.1154
GA	2.3462
SA	1.5385

Para evaluar las diferencias significativas se realiza la prueba post-hoc de Holm (ver Tabla IV-14) con los resultados de Friedman 1xN. Se evidencia que SA obtiene diferencias significativas con las otras dos metaheurísticas, es decir, SA obtiene los menores valores de la FO.

Tabla IV-14: Prueba de Holm a partir de Friedman 1xN entre los valores promedio de FO de las tres metaheurísticas

	p	Holm	Hipótesis
GA	0.0036	0.025	Rechazada
VMO	0.0375	0.05	Rechazada

Para explicar el motivo por el cual SA desciende más el valor de la FO hay que analizar cuantas iteraciones realiza SA, cuantas poblaciones obtiene GA y cuantas MI se generan con VMO, respecto a la cantidad de evaluaciones de la FO. La cantidad de iteraciones y de poblaciones en los algoritmos determina considerablemente la explotación. En SA las evaluaciones de la FO están distribuidas en 12 hilos (20 000 / 12) obteniendo 1666 iteraciones en el algoritmo. En el caso de una proteína pequeña de 100 aminoácidos, cada población de GA se genera con 150 individuos (20 000 /150) obteniendo 133 poblaciones. En VMO, para una proteína pequeña de 100 aminoácidos, la MI cuenta con 50 puntos y crece 100 puntos en el proceso de expansión ([20 000 - 50] /100) obteniendo 199 veces la MI. Como se puede apreciar en SA permite mejorar un mismo nodo más veces que VMO y este más veces que GA, explicando el orden en el rank de Friedman.

IV. 2. 1. 1. Convergencia de la Función Objetivo

En el experimento 1, en el epígrafe de perturbación de los datos se realiza un análisis de clúster; se decide analizar la convergencia de la FO de una muestra de tres proteínas tomada una de cada clúster. La selección de la proteína por clúster es de forma aleatoria en el clúster. La proteínas R0020, R0013 y R0031 presentan 127, 138 y 283 aminoácidos respectivamente.

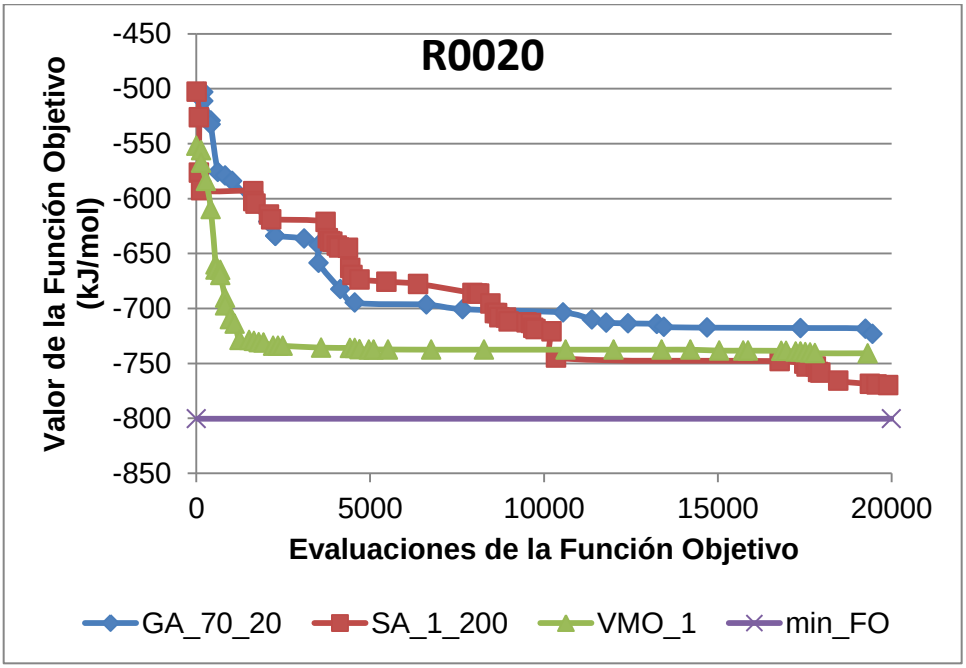


Gráfico IV-4: Convergencia de la FO en la proteína R0020

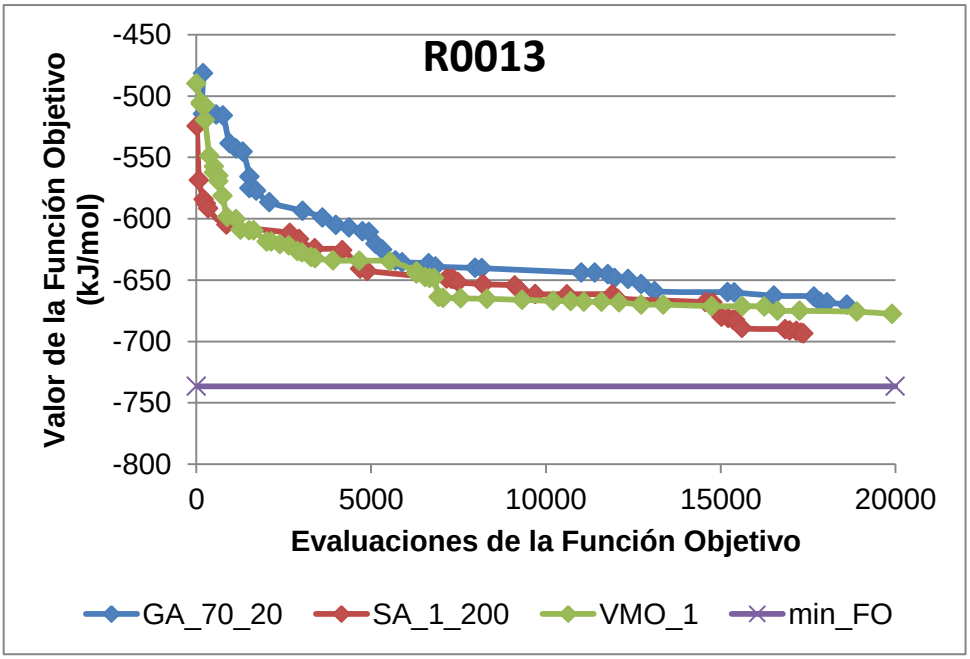


Gráfico IV-5: Convergencia de la FO en la proteína R0013

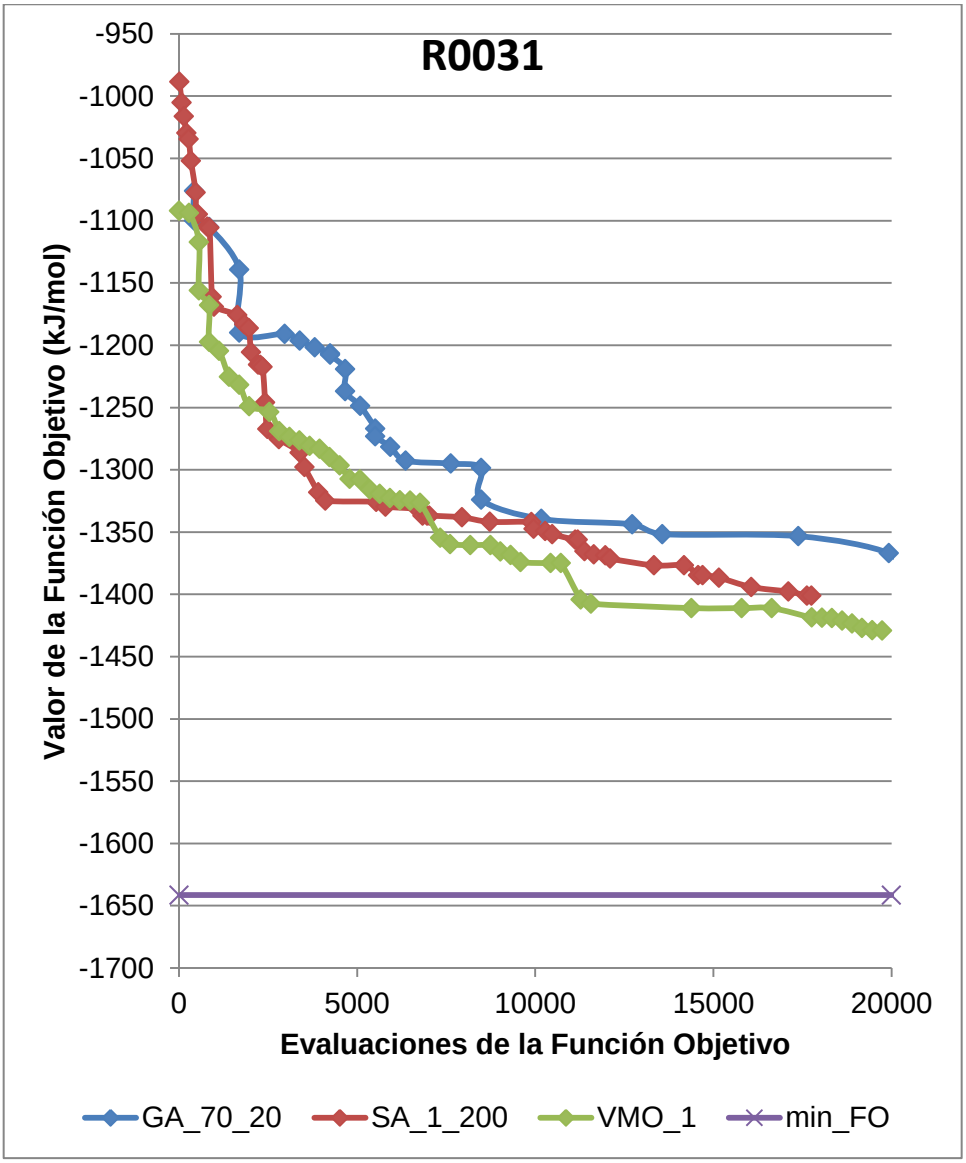


Gráfico IV-6: Convergencia de la FO en la proteína R0031

En los Gráficos Gráfico IV-4Gráfico IV-5Gráfico IV-6 se puede apreciar que en las metaheurísticas mientras mayores sean las proteínas (ver Tabla III-11), se logran minimizar menos la función objetivo. Esto puede estar dado por parámetro de condición de parada: cantidad de evaluaciones de la FO, que se empleó constante entre todas las proteínas haciendo que para aquellas con un mayor espacio conformacional, la exploración realizada pudo ser insuficiente.

IV. 2. 2. Comparación respecto al Tiempo de Ejecución

El tiempo de ejecución es otro aspecto importante a analizar para ello se ordenan las proteínas por cantidad de aminoácidos (de menor a mayor). En el Gráfico IV-7 se muestra el comportamiento promedio del tiempo de ejecución de cada metaheurística por proteína. Se aprecia que generalmente GA genera las soluciones en un menor tiempo.

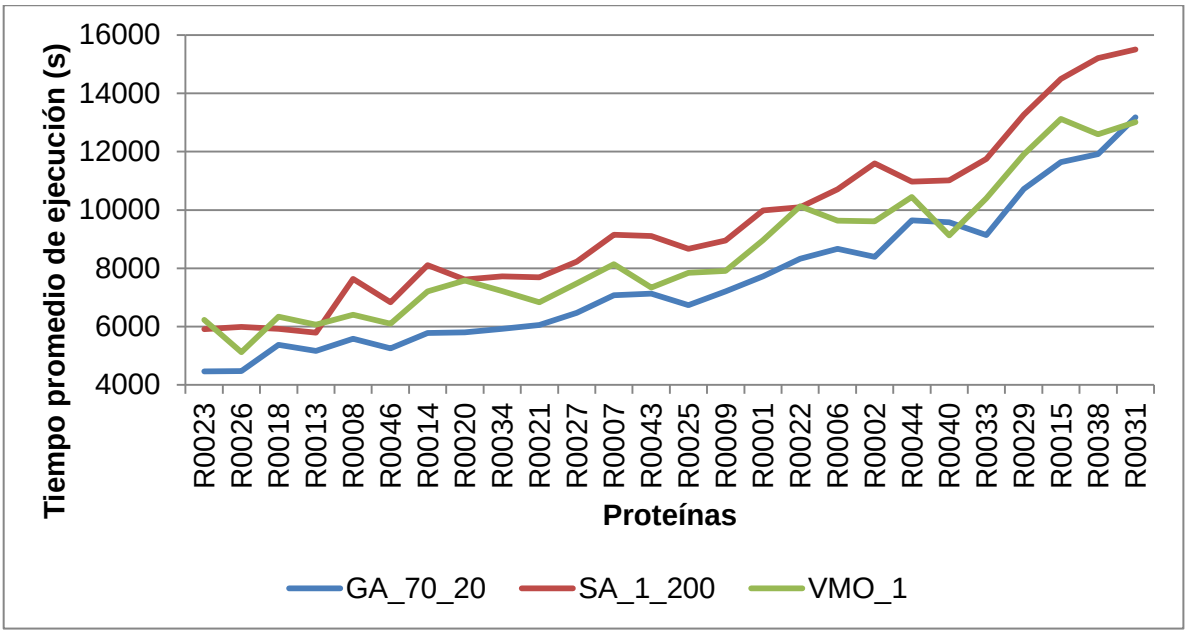


Gráfico IV-7: Promedio en tiempo total de ejecución en las metaheurísticas

En el gráfico se aprecia que el promedio del tiempo de ejecución máximo lo alcanza la proteína R0031 en SA_1_200 con 15509.67 segundos, equivale a generar una solución (réplica) en 4 horas y media. El menor tiempo de ejecución en promedio lo obtiene R0023 (la proteína más corta) con GA_70_20 y es de 4466.33 segundos, equivale a 1 hora y 15 minutos. En promedio GA_70_20 genera soluciones cada 7594.359 segundos, VMO_1 cada 8568.974 y SA_1_200 en 9536.692.

Para comprobar si hay diferencias significativas, entre los valores promedio del tiempo total de ejecución de las metaheurísticas, se decide realizar las pruebas estadísticas de Friedman e Iman-Davenport (ver Tabla IV-15). Se observa que existen diferencias significativas entre las tres metaheurísticas en cuanto al tiempo de ejecución. Para más detalle ver los resultados en la Tabla IV-15.

Tabla IV-15: Prueba de Friedman e Iman-Davenport en cuanto a los Tiempos promedio de ejecución de las tres metaheurísticas.

	Friedman	Iman-Davenport
Grados de libertad	2	2 y 50
Chi-cuadrado	40.9231	92.3611
Significación	**0**	**0**

Algoritmo	Rank Friedman
GA	1.0769
SA	2.8462
VMO	2.0769

Para evaluar las diferencias significativas se realiza la prueba post-hoc de Holm (ver Tabla IV-16) con los resultados de Friedman 1xN. Se evidencia que GA obtiene las soluciones en un menor tiempo con diferencias significativa respecto a las otras dos metaheurísticas.

Tabla IV-16: Prueba de Holm a partir de Friedman 1xN entre los Tiempos promedio de ejecución de las tres metaheurísticas

	p	Holm	Hipótesis
SA	0	0.025	Rechazada
VMO	0.0003	0.05	Rechazada

Estos resultados vienen dado por el costo de aplicar los operadores y operaciones involucradas en los algoritmos. La metaheurística GA realiza operaciones básicas sobre arreglos, las cuales están optimizadas en java. VMO utiliza operadores costos que involucran el cálculo de la distancia entre los puntos, cabe recordar que para una proteína pequeña de 100 aminoácidos se cuenta con 200 variables éntrelas cuales hay que computar la distancia. SA al obtener un sucesor realiza menos operaciones que GA tan básicas como este, pero en este proceso de obtener un sucesor se generan a lo sumo 5 posibles nodos aumentando la cantidad de operaciones, además en GA si el sucesor encontrado es peor en cuanto a valor de FO que al nodo que le dio origen hay que evaluar una operación compleja ($e^{-\Delta FO/T}$). Por lo anterior dicho se evidencia que GA presenta la menor complejidad en cuanto a operaciones y operadores, por consiguiente es el más rápido.

IV. 2. 3. Comparación en cuanto a la calidad de las soluciones

En la comparación respecto a la calidad utilizaremos las soluciones resultantes de la mejor configuración por cada metaheurística. Estas soluciones son comparadas con su *nativa* haciendo uso de las métricas propuestas en el servidor LGA.

Se calculan las 4 métricas presentes en el servidor LGA a las soluciones de las metaheurísticas (solo en la mejor configuración). Las métricas obtenidas se promedian en cada metaheurística por proteína (ver Anexo VIII. 7). A estos datos se les realizan las pruebas de Friedman e Iman-Davenport por cada métrica (ver Tabla IV-17) donde se observa que solo hay diferencias significativas en la métrica LGA_S.

Tabla IV-17: Prueba de Friedman e Iman-Davenport en cuanto a la calidad de las tres metaheurísticas en cuatro métricas del servidor LGA

Pruebas para la métrica N					
	Friedman	Iman-Davenport		Algoritmo	Rank Friedman
Grados de libertad	2	2 y 50		GA	1.8654
Chi-cuadrado	4.75	2.5132		SA	1.7885
Significación	**0.0930**	**0.0911**		VMO	2.3462
Pruebas para la métrica RMSD					
	Friedman	Iman-Davenport		Algoritmo	Rank Friedman
Grados de libertad	2	2 y 50		GA	2.0385
Chi-cuadrado	1.2885	0.6352		SA	2.1346
Significación	**0.5251**	**0.5341**		VMO	0.8269
Pruebas para la métrica GDT					
	Friedman	Iman-Davenport		Algoritmo	Rank Friedman
Grados de libertad	2	2 y 50		GA	1.6346
Chi-cuadrado	5.4423	2.9223		SA	2.1154
Significación	**0.0658**	**0.0631**		VMO	2.25
Pruebas para la métrica LGA_S					
	Friedman	Iman-Davenport		Algoritmo	Rank Friedman
Grados de libertad	2	2 y 50		GA	1.6154
Chi-cuadrado	6.0769	3.3082		SA	2.1154
Significación	**0.0479**	**0.0447**		VMO	2.2692

Para evaluar las diferencias significativas se realiza la prueba post-hoc de Holm (ver Tabla IV-18) con los resultados de Friedman 1xN en la métrica LGA_S. Se evidencia que GA presenta diferencias significativas con VMO, pero no con SA.

Tabla IV-18: Prueba de Holm a partir de Friedman 1xN entre las tres metaheurísticas en la métrica de calidad LGAS

	p	Holm	Hipótesis
VMO	0.0184	0.025	Rechazada
SA	0.0714	0.05	Aceptada

Aunque GA y SA no presenten deferencias estadísticamente significativas con los test Friedman y Iman-Davenport se decide realizar la prueba de Wilcoxon entre estas dos metaheurísticas (ver Tabla IV-19), para detectar diferencias entre ellas. GA presenta diferencias significativas con SA, por lo que GA es seleccionada como la mejor ya que contiene mayor cantidad de casos que sobrepasan el valor de LGAS de SA.

Tabla IV-19: Prueba de Wilcoxon entre las metaheurística GA y SA en cuanto a la métrica de calidad LGAS

	LGAS_SA – LGAS_GA
Z	-2.095[a]
Significación	**.036**

a. Basada en los rangos positivos.

	N	Media de rangos	Suma de rangos
Rangos Negativos	18[a]	14.33	258.00
Rangos Positivos	8[b]	11.62	93.00
Iguales	0[c]		
Total	26		

a. LGAS_SA < LGAS_GA
b. LGAS_SA > LGAS_GA
c. LGAS_SA = LGAS_GA

IV. 2. 4. Conclusiones sobre la comparación de las tres metaheurísticas

En cuanto al valor de FO obtenido en las optimizaciones de la metaheurísticas, SA presenta los mejores resultados. La marcada diferencia entre SA y las otras metaheurísticas viene dada por el consenso de exploración-explotación presente en las implementaciones.

En cuanto a la convergencia de la FO, se pudo apreciar que en todas las metaheurísticas se necesitan más iteraciones para que la FO tenga una convergencia hasta su valor mínimo.

En cuanto al tiempo de ejecución, existen diferencias significativas entre los valores de los tiempos de ejecución. En GA se obtienen en promedio las soluciones con mayor rapidez, debido a la sencillez de los operadores y operaciones.

En cuanto a la calidad de las soluciones SA y GA no presentan diferencias significativas al ser evaluadas en conjunto con VMO con los test de Friedman e Iman-Davenport. Cuando se comparan SA y GA con la prueba de Wilcoxon si se

presentan diferencias, evidenciando que GA obtiene soluciones con mayor calidad debido al operador de cruce y posiblemente a las funciones complementarias.

IV. 3. Experimento 3: Comparación con predictores internacionales

Es importante comparar las soluciones de los algoritmos con otros algoritmos para determinar la aplicabilidad de los algoritmos al problema. Se compara con las soluciones de predictores internacionales perteneciente a la competencia CASP-ROLL. Las soluciones presentadas en el CASP-ROLL por los predictores utilizan métodos *ab-initio* que constan fundamentalmente de tres etapas:

1. Exploración, donde es usual restringir el espacio de búsqueda empleando por ejemplo bibliotecas de fragmentos prediseñados y conformando las estructuras a modo de puzle.
2. Selección, se clusterizan decenas o centenares de soluciones obtenidas de múltiples optimizaciones en la primera etapa, seleccionando un subconjunto reducido de candidatos.
3. Refinamiento, esta etapa comprende la aplicación de métodos comúnmente muy costosos computacionalmente como las Dinámicas Moleculares para tunear localmente las conformaciones de las cadenas de los candidatos seleccionados anteriormente.

En este experimento se pretende comparar las soluciones de GA contra las soluciones de predictores de las competencias. Nuestras soluciones difieren de las presentadas en que:

1. Se Realiza una exploración no restringida del espacio de búsqueda.
2. La selección se las soluciones se realiza en una población total de solo 3 soluciones.
3. No se realiza clusterización de las soluciones.
4. No se realizan etapas de refinamiento.

Atendiendo a las diferencias anteriores se procede a calcular las métricas en el servidor LGA de los predictores contemporáneos a las 26 proteínas de la base de casos. En la Tabla IV-20 se muestra una relación por proteína, de la cantidad de predictores y la cantidad de predicciones comparadas. La cantidad de predictores

del CASP-ROLL por proteína corresponde al valor de la columna Predictores menos 1, en tanto la cantidad de predicciones presentadas por ellos es equivalente al valor de la columna Predicciones menos 3.

Tabla IV-20: Cantidad de predictores y predicciones por proteína

Proteína	Predictores	Predicciones	Proteína	Predictores	Predicciones
R0001	20	98	R0023	16	76
R0002	18	87	R0025	15	72
R0006	18	87	R0026	12	53
R0007	19	93	R0027	10	47
R0008	18	88	R0029	10	47
R0009	19	92	R0031	11	53
R0013	19	92	R0033	6	23
R0014	19	91	R0034	11	48
R0015	19	92	R0038	10	47
R0018	17	82	R0040	6	28
R0020	16	76	R0043	9	43
R0021	14	68	R0044	6	26
R0022	16	77	R0046	5	22

Se decide analizar el comportamiento de los resultados en las cinco métricas de dos formas:

1. Como se comportan en promedio nuestras soluciones con el resto de los predictores.
2. Como se comporta nuestra mejor solución en relación al total de soluciones

En el primer caso se promedian los resultados por predictor en cada métrica independientemente. Por cada proteína, se ordenan los promedios por cada métrica. Luego a cada predictor se le calcula el porciento de predictores que se encuentran con un promedio de soluciones por debajo de él con la Ecuación IV-1. En la ecuación T representa el total de predictores y P es la posición en que quedó el predictor cuando fue ordenado.

En el segundo caso de estudio, por proteína se ordenan las predicciones en cada métrica, obteniendo una posición. A cada predicción se le calcula el porciento de predicciones que quedan por debajo de ella (con la Ecuación IV-1, pero en este caso, T representa el total de predicciones y P la posición de la predicción a evaluar).

$$Rank\ (P,T) = \frac{T-P}{T} * 100$$

Ecuación IV-1: Cálculo de posiciones respecto al porciento de soluciones

IV. 3. 1. Comparación con los predictores, evaluado los datos en promedio

Los datos para este caso de estudio se analizan separándolos por métrica (ver Anexo VIII. 7). Al comparar las proteínas por predictores se puede observar que la R0022, R0033 y R0044 obtienen la primera posición entre los predictores en las métricas GDT y LGA_S. Las proteínas R0026, R0031, R0033 y R0044 obtienen la primera posición entre los predictores en la métrica N.

En el Gráfico IV-8 se muestra un resumen de los datos presentados en el Anexo VIII. 7, donde se aprecia el porciento de proteínas ubicadas en cada cuartil por métrica. En el primer cuartil de la métrica N, GA obtiene un 23.08% de las proteínas con soluciones en promedio por encima del 75% de los predictores, similar ocurre con las métricas GDT y LGA_S. En las métricas RMSD y LGA_S se encuentran tres proteínas (11.54%) en el primer cuartil. Se puede destacar que en la métrica GDT el 50% de las proteínas se encuentran ubicadas entre el primer y segundo cuartil, es decir, que en la mitad de las proteínas los resultados promedio de GA_70_20 son superiores al 50% de los predictores.

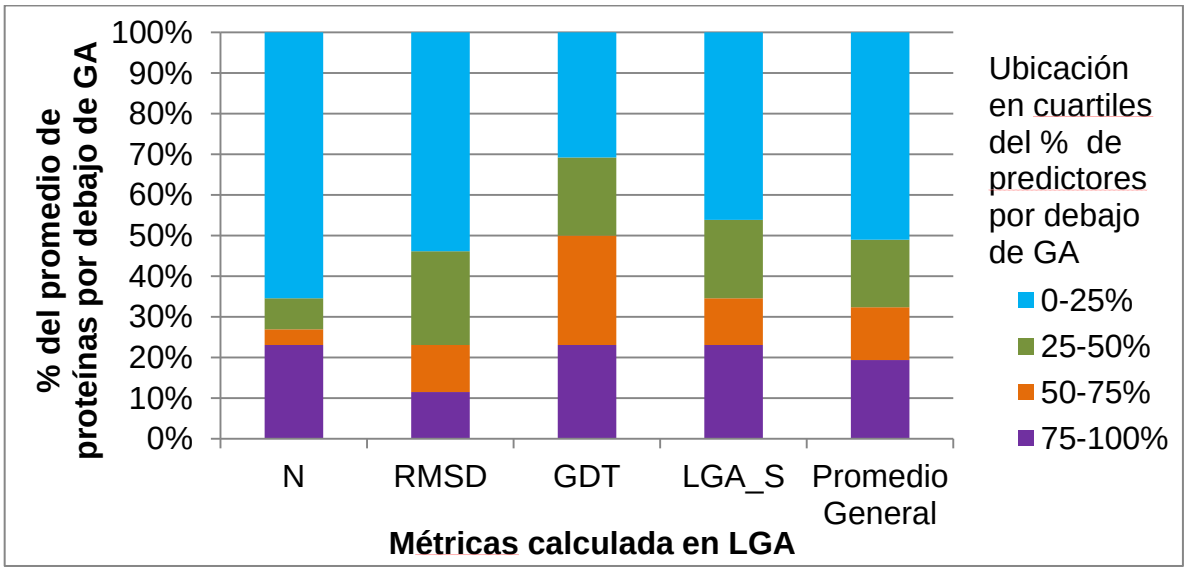


Gráfico IV-8: Agrupamiento en cuartiles del porciento de predictores que se encuentran en promedio por debajo de GA.

Como mínimo en el 11.54% de las soluciones en todas las métricas se obtienen soluciones que en promedio superan al menos al 75% de los predictores. Se obtiene que en promedio en algunas métricas por proteínas logra GA ubicarse como el primer predictor.

IV. 3. 2. Comparación con las predicciones evaluando la mejor solución

Estando en el marco de una competencia es de importancia conocer cómo se comporta nuestra mejor solución con las demás soluciones que compiten. En este caso de estudio se ordenan las predicciones por proteína, sin promediar los datos. A cada predicción se le calcula el porciento de predicciones superadas. Por cada métrica se selecciona la mejor predicción de GA por proteína (ver Anexo VIII. 9). En el Gráfico IV-9 se muestra un resumen de la evaluación de la mejor solución por métrica de GA. Se puede observar que se generan soluciones (más de un 30%, en el primer cuartil) con GA que llegan a superar como mínimo al 75% de las predicciones en todas las métricas.

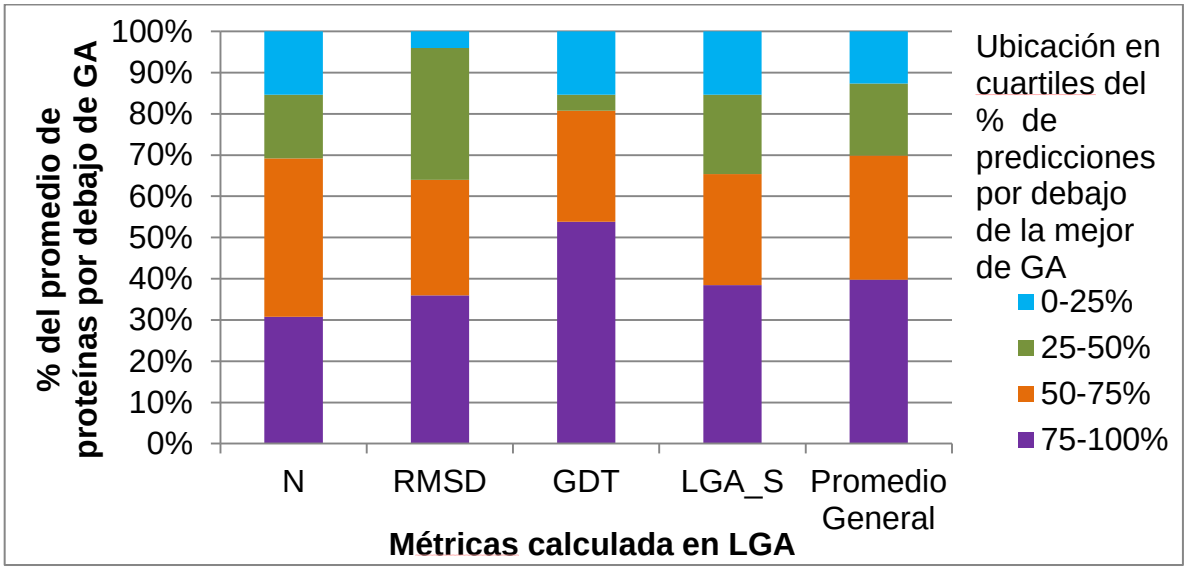


Gráfico IV-9: Agrupamiento en cuartiles del porciento de predicciones que se encuentran por debajo de la mejor solución de GA_70_20

V. Conclusiones

Como conclusiones tenemos que:

1. Se creó una herramienta que permite realizar predicciones de las estructuras terciarias de proteínas en lenguaje Java empleando tres metaheuristicas (GA, SA y VMO).
2. La mejor configuración de SA fue la variación de un ángulo diedro (variables) para generar un sucesor y una temperatura inicial de 200. La mejor configuración de GA fue una probabilidad de cruce de 0.7 y una probabilidad de mutación de 0.2. La mejor configuración de VMO se obtuvo al variar el 1% de los ángulos diedros. Al comparar las tres metaheurísticas (en las configuraciones seleccionadas) se determina que GA obtiene los mejores resultados en cuanto a rapidez de ejecución y calidad de las soluciones.
3. En promedio, GA logra ubicarse entre los dos primeros cuartiles de predicciones en aproximadamente un 33.66% de las proteinas considerando las cuatro metricas estudiadas. Al examinar la mejor solución de GA, se pudo observar que esta solución sale entre los dos primeros cuartiles en aproximadamente un 70% de las proteinas entre todas las metricas. La metrica GDT se destaca como la de mayor desempeño lo que responde a que las optimizaciones realizadas predicen mejor la conformación global de la estructura que los arreglos locales.

VI. Recomendaciones

Recomendamos para estudios posteriores:

1. Configurar que la condición de parada (cantidad de evaluaciones de la FO) de las metaheurísticas dependa del largo de la proteína.
2. Extender el estudio a un número mayor de proteínas.
3. Introducir la etapa de clusterización para la obtención de soluciones finales y participar en la competencia CASP este año.

VII. Referencias

1. Hagerman P, Jr I. From sequence to structure to function. Structural Biology. 1996:277-80.
2. Mirsky A, Pauling L. On the structure of native, denatured, and coagulated proteins. Proceedings of the National Academy of Sciences of the United States of America 1936:439-47.
3. Orengo CA, Todd AE, Thornton JM. From protein structure to function. Structural Biology. 1999:374-82.
4. Yi-Yao H, Chang-Biau Y, Kuo-Tsung T, Chia-Ning Y. Protein Folding Prediction with Genetic Algorithms.
5. Cheng-Jian L, Shih-Chieh S. Protein 3D HP Model Folding Simulation Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization. International Journal of Fuzzy Systems. 2011;13(2):140-7.
6. Ruiz Blanco YB. Desarrollo de modelos energéticos reduccionistas aplicados al plegamiento de proteínas globulares y su generalización en descriptores para proteínas. Havana: Universidad Central “Marta Abreu” de Las Villas; 2015.
7. Chaput JC, Woodbury NW, Stearns LA, Williams BAR. Creating protein biocatalysts as tools for future industrial applications. Expert Opinion on Biological Therapy. 2008;8(8):1087-98.
8. The-UniProt-Consortium. Activities at the Universal Protein Resource (UniProt). Nucleic Acids Research. 2014 January 1, 2014;42(D1):D191-D8.
9. Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. Nucleic Acids Research. 2014;42(D1):D32-D7.
10. Berman HM, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nature Structural & Molecular Biology. 2003;10(12):980.
11. Xu D, Zhang Y. Ab Initio structure prediction for Escherichia coli: towards genome-wide protein structure modeling and fold assignment. Sci Rep. 2013;3.
12. Baker D, Sali A. Protein Structure Prediction and Structural Genomics. Science. 2001;294(5540):93-6.
13. Hardin C, Pogorelov TV, Luthey-Schulten Z. Ab initio protein structure prediction. Curr Opin Struct Biol. 2002;12:176–81.
14. Shakhnovich E. Protein Folding Thermodynamics and Dynamics: Where Physics, Chemistry, and Biology Meet. Chem Rev. 2006;106:1559-88.
15. Khatun J, Khare SD, Dokholyan NV. Can Contact Potentials Reliably Predict Stability of Proteins? J Mol Biol. 2004;336:1223–38.
16. Fogolari F, Pieri L, Dovier A, Bortolussi L, Giugliarelli G, Corazza A, et al. Scoring predictive models using a reduced representation of proteins: model and energy definition. BMC Structural Biology. 2007 Mar 23;7.
17. Becker OM. Protein Folding: Computational Approaches. In: Becker OM, MacKerell Jr. AD, Roux B, Watanabe M, editors. Computational Biochemistry and Biophysics New York: Marcel Dekker Inc.; 2001.
18. Berman HM. The Protein Data Bank: a historical perspective. Acta Crystallographica Section A: Foundations of Crystallography. 2008;64:88-95.
19. Suman B, Kumar P. Asurvey of simulated annealing as a tool for single and multiobjective optimization Journal of the Operational Research Society 2006.
20. Aarts E, Korst J. Simulated Annealing and Boltzmann Machines. United States: New York: John Wiley and Sons Inc; 1988.

21. Escuela G. Algoritmos evolutivo con representación basada en sistemas-L para el problema del replegado de las proteínas.: Universidad Simón Bolívar; 2006.
22. More Quintero I. Optimizacion basada en Mallas Variables para Predecir el Plegamiento de Proteinas Universidad "Martha Abreu" de Las Villas 2011.
23. Wong Delgado F. Algoritmo Recocido Simulado para predecir la estructura terciaria de proteínas. Santa Clara: Universidad Central "Marta Abreu" de Las Villas; 2015.
24. Wong Delgado F, Martínez Pérez E, Ruiz Blanco YB, editors. Algoritmo Recocido Simulado aplicado a la predicción de estructura terciaria de proteínas. XIII Congreso Nacional de Reconocimiento de Patrones (RECPAT 2015); 2015; Santiago de Cuba.
25. Martínez Pérez E, Ruiz-Blanco YB, Contreras Torres E, editors. Algoritmos Genéticos para Predecir la Estructura Terciaria de Proteínas XIII Congreso Nacional de Reconocimiento de Patrones (RECPAT 2015); 2015; Santiago de Cuba.
26. Guillén L, Victoria M. Estructura y propiedades de las proteínas.
27. Contreras-Moreira B. Algoritmos en bioinformática estructural2015.
28. Mendel G. Versuche über Pflanzenhybriden. Verhandlungen des naturforschenden Vereines in Brunn. 1866;4.
29. Pillardy J, Czaplewski C, Liwo A, Lee J, Ripoll DR, Rajmund Kaz´mierkiewicz, et al. Recent improvements in prediction of protein structure by global optimization of a potential energy function. PNAS. 2001;98(5):2329–33.
30. Dill KA. The meaning of hydrophobicity. Science. 1990;250(4978):297-8.
31. Lehninger AL. Principles of Biochemistry. 1st ed. New York: Worth Pub; 1982.
32. Mathews C, van Holde K, Ahern K. Biochemistry. 3rd ed: Addison Wesley Longman; 1999.
33. Russell SJ, Norvig P. Inteligencia artificial. Un enfoque moderno. 2nd. ed. Madrid, España Pearson Educación; 2004.
34. Protein 3D-structure analysis: why and how. In: Bioinformatics SIo, editor.2012.
35. Levinthal C. Mossbauer Spectroscopy in BiologicalSystems, in Proceedings of a meeting held at Allerton House,. In: P Debrunner, E Munck, eds, Editor ed: Univ Illinois Press: Urbana. ; 1969.
36. Anfinsen CB. Principles that govern the folding of protein chains. Science. 1973;181(4096):223-30.
37. Chothia C. One thousand families for the molecular biologist. Nature. 1992;357(6379):543-4.
38. Riley ML, Flitsch SL, Booth PJ. Biochemestry. 1997;36:192-6.
39. Prusiner S. Prion protein biology Cell. 1998;93:337-48.
40. Hyde SC, Emsley P, Hartshorn MJ, Mimmack MM, Gileadi U, Pearce SR, et al. Structural model of ATP-binding proteing associated with cystic fibrosis, multidrug resistance and bacterial transport. Nature. [10.1038/346362a0]. 1990;346(6282):362-5.
41. Holley LH, Karplus M. Protein secondary structure prediction with a neural network. Biophysics. 1989;86:152-6.

42. Walsh I, Bau D, Martin AJM, Mooney C, Vullo A, Pollastr G. Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks. BMC Structural Biology. 2009;9(5):1-38.
43. Dill KA, MacCallum JL. The protein-folding problem, 50 years on. Science. 2012;338(6110):1042-6.
44. Conklin D. La Bioinformática: una perspectiva de la estructura de proteínas. 2000.
45. Rodríguez Sotres R, Gaytán Mondragón SA, Hernández Domínguez EE, Valencia Turcotte LG. Estructuras tridimensionales in silico, a partir de la secuencia de aminoácidos de una proteína ¿cómo saber que el modelo es realista? Mensaje Bioquímico. 2011;XXXV:143-56.
46. Crescenzi P, Goldman D, Capadimitriou C, Piccolboni A, Yannakakis M. On the complexity of protein folding. Journal of Computational Biology. 1998;5(1):409–22.
47. Fraenkel A. Complexity of protein folding. Bulletin of Mathematical Biology [serial on the Internet]. 1993.
48. Hart W, Istrail S. Robust proofs of NP-hardness for protein folding: general lattices and energy potentials. Journal of Computational Biology. 1997;4(1):1-22.
49. Unger R, Moult J. Finding the lowest free energy conformation of a protein is NP-hard problem: Proof and implications. Bulletin of Mathematical Biology [serial on the Internet]. 1993; 55(6).
50. Berger B, Leighton T, editors. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. RECOMB '98 Proceedings of the second annual international conference on Computational molecular biology 1998; New York, NY, USA.
51. Arkun Y, Erman B. Prediction of Optimal Folding Routes of Proteins That Satisfy the Principle of Lowest Entropy Loss : Dynamic Contact Maps and Optimal Control. PLoS ONE. 2010;5(10):1-11.
52. Greenwood GW, Shin JM, Lee B, Fogel GB. A survey of Recent Work on evolutionary approaches to the Protein Folding Problem. IEEE. 1999;99:488–95.
53. Zhang Z. An Overview of Protein Structure Prediction: From Homology to Ab Initio. Bioc. 2002;218:1-10.
54. Aponte I. Recocido Simulado y Búsqueda Tabú para el problema de predicción de estructura terciaria de proteínas. Sartenejas: Universidad Simón Bolívar; 2006.
55. Santiesteban-Toca CE, Casañola-Martin GM, Aguilar-Ruiz JS. Las técnicas de aprendizaje automático en la predicción de estructura de proteínas: un enfoque desde la bioinformática. AFINIDAD. 2014;LXXI(567):119-227.
56. Puris A, Bello R, Molina D, Herrera F. Variable mesh optimization for continuous optimization problems. Soft Comput. 2012:511-25.
57. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines. Chemical Physics. 1953;21(6):1087.
58. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science. 1983;220(4598):671–80.

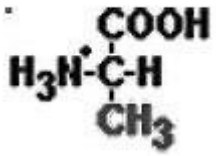
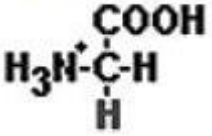
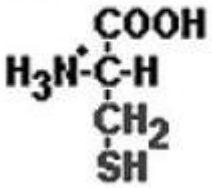
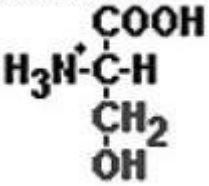
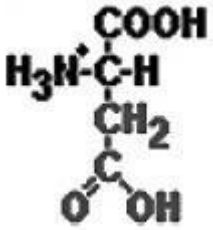
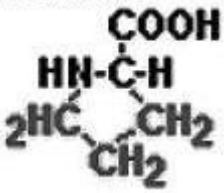
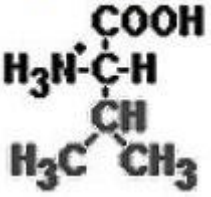
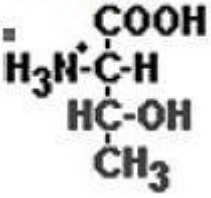
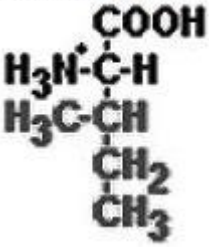
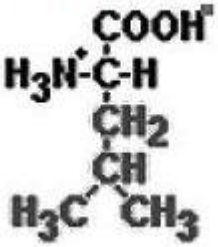
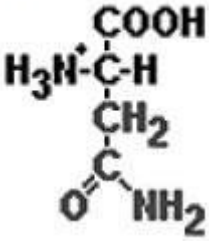
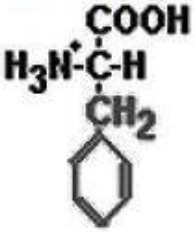
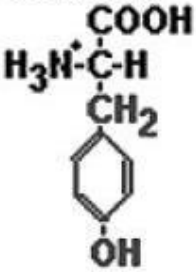
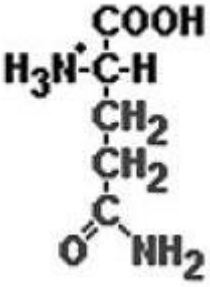
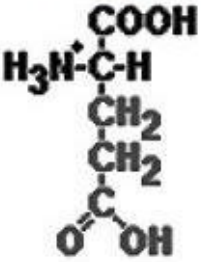
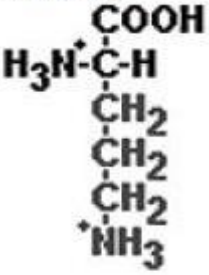
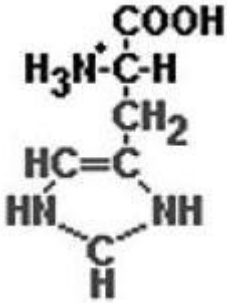
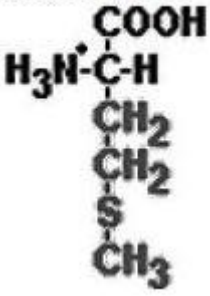
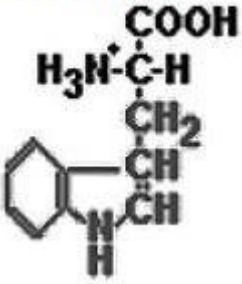
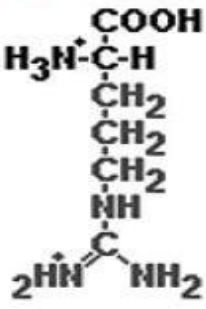
59. Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications. 1985;45(1):41-51.
60. Gallardo Cuberos M. Heurística de Recocido Simulado para la resolución del problema del acarreo terrestre: Escuela Técnica Superior de Ingeniería Universidad de Sevilla 2014.
61. Darwin C. On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. London: Murrey; 1859.
62. Goldberg DE, Holland JH. Genetic Algorithms and Machine Learning. Machine Learning. 1988;3(2-3):95-9.
63. Rabow AA, Scheraga HA. Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator. Prorein Science. 1996;5:180-1815.
64. García Pardo E. Optimización de Sistemas de Comunicación 2013.
65. Zemla A. LGA - A Method for Finding 3-D Similarities in Protein Structures. Nucleic Acids Research. 2003;31(13):3370-4.
66. Zemla A, Zhou Ce Fau - Slezak T, Slezak T Fau - Kuczmarski T, Kuczmarski T Fau - Rama D, Rama D Fau - Torres C, Torres C Fau - Sawicka D, et al. AS2TS system for protein structure modeling and analysis. Nucleic Acids Research. 2005;33 (supl. 2)(Web Server Issue):W111 - W5.
67. Zemla A, Venclovas C, Moult J, Fidelis K. Processing and evaluation of predictions in CASP4. PROTEINS: Structure, Function, and Genetics. 2001;45(S5):13-21.
68. Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A. A study of quality measures for protein threading models. BMC Bioinformatics. 2001;2.
69. Zemla A, Geisbrecht B, Smith J, Lam M, Kirkpatrick B, Wagner M, et al. STRALCP structure alignment-based clustering of proteins. Nucleic Acids Research. 2007;35(22).
70. Moult J, Pedersen JT, Judson R, Fidelis K. A Large-Scale Experiment to Assess Protein Structure Prediction Methods. PROTEINS Structure, Function, and Genetics. 1995;23:ii-iv.
71. Lattman EE. Protein Structure Prediction: A Special Issue. PROTEINS: Structure, Function, and Genetics. 1995;23:i.
72. Defay T, Cohen FE. Evaluation of Current Techniques for Ab Initio Protein Structure Prediction. PROTEINS: Structure, Function, and Genetics. 1995;23:431-45
73. Tai C-H, Bai H, Taylor TJ, Lee B. Assessment of template-free modeling in CASP10 and ROLL. PROTEINS Structure, Function, and Genetics. 2014;82(Suppl 2):57–83.
74. Wilcoxon F. Individual Comparisons by Ranking Methods. Biometrics Bulletin. 1945;1:80–3.
75. Sawilowsky S. Misconceptions leading to choosing the t test over the Wilcoxon Mann-Whitney U test for shift in location parameter. Journal of Modern Applied Statistical Methods. 2005;4(2):598-600.
76. Juárez García F, Villatoro Velázquez JA, López Lugo EK. Apuntes de Estadística Inferencial. México, D. F: Instituto Nacional de Psiquiatría Ramón de la Fuente; 2002.

77. Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association. 1937;32:674-701.
78. Friedman M. A comparison of alternative tests of signicance for the problem of m rankings. Annals of Mathematical Statistics. 1940;11:86-92.
79. Iman R, Davenport J. Approximations of the critical region of the friedman statistic. Communications in Statistics1980. p. 571-95.
80. Derrac J, García S, Molina D, Herrera F. Un tutorial sobre el uso de test estadísticos no paramétricos en comparaciones múltiples de metaheurísticas y algoritmos evolutivos. 2011.
81. Holm S. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics. 1979;6(2):60-5.
82. Simons J. An introduction to theoretical chemistry2003.
83. Field MJ. A practical introduction to the simulation of molecular systems2005.
84. Bell S, Dines TJ, Chowdry BZ, Withnall R. J Chem Educ. 2007;84:1364.
85. Parsons J, Holmes JB, Rojas JM, Tsai J, Strauss CEM. Practical Conversion from Torsion Space to Cartesian Space for In Silico Protein Synthesis. Journal of Computational Chemistry. 2005;36(10).
86. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. Journal of Chemical Information and Computer Sciences. 2003;42(2):493-500.
87. Steinbeck C, Hoppe C, Kuhn S, Floris M, Guha R, Willighagen. Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics. Current Pharmaceutical Design. 2006;12(17):2111-20.
88. O'Boyle NM, Hutchison GR. Chemistry Central. 2008;2.
89. Namakforoosh MN. Metodología de la Investigación. 2da ed. México: Lisuma; 2005.
90. Tillé Y. Sampling Algorithms. Springer, editor. Neuchatel. Switzerland2006.
91. Díaz A. Algoritmos de Enfriamiento Simulado Paraninfo1996.
92. Dowsland KA, Díaz A. Diseño de heurísticas y fundamentos del recocido simulado Revista Iberoamericana de Inteligencia Artificial2003.
93. Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. Nature. [10.1038/181662a0]. 1958;181(4610):662-6.
94. Branden C, Toose J. Introduction to Protein Structure. Second ed: Garland Publisher.
95. Islam SA, Karplus M, Weaver DL. Application of the Diffusion–Collision Model to the Folding of Three-helix Bundle Proteins. J Mol Biol. 2002;318:199–215.
96. AlcaláFdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing. 2008;13(3):307-18.
97. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, et al. KEEL data-mining software tool: Data set repository, integration of algorithms and

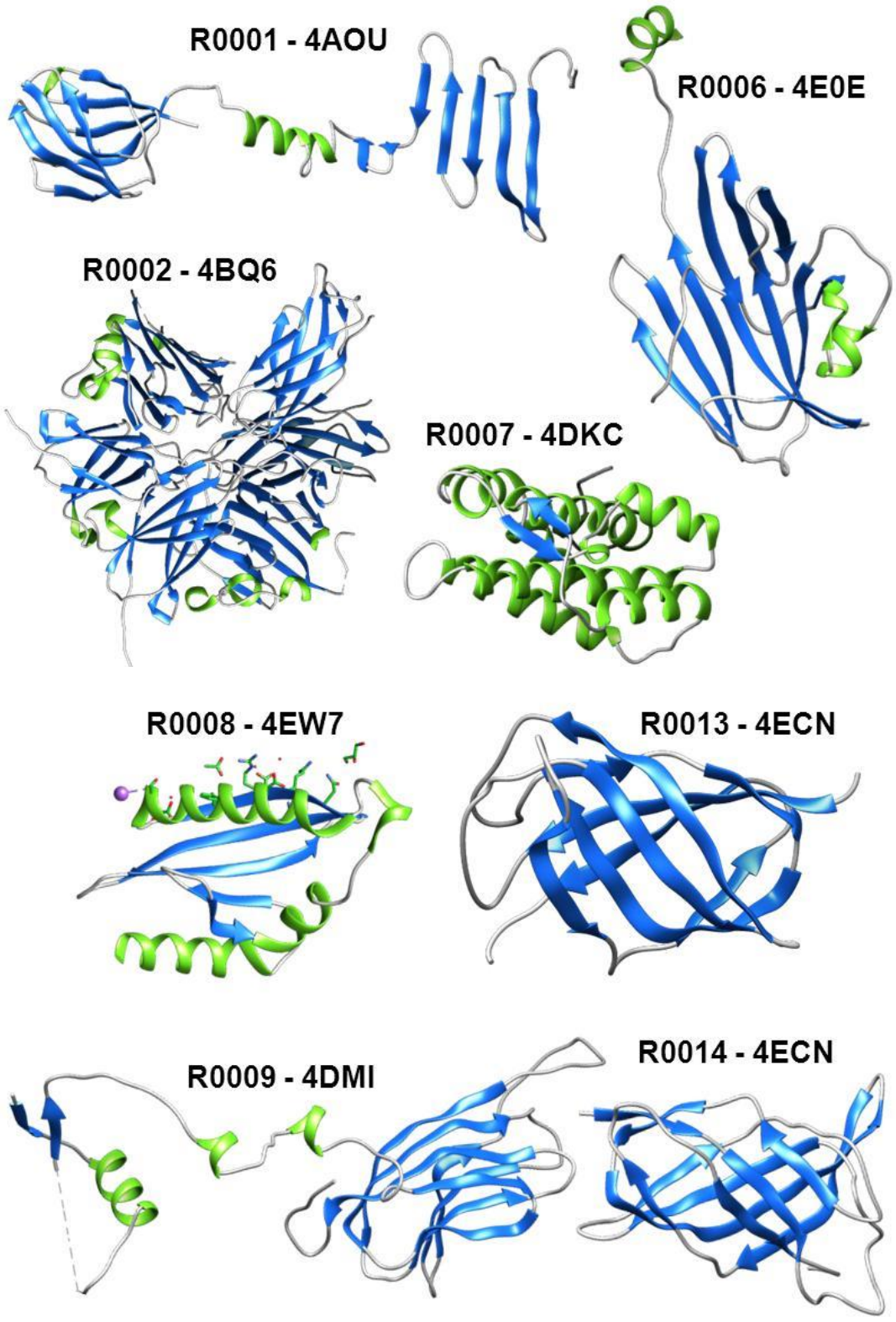
experimental analysis framework. Multiple-Valued Logic and Soft Computing. 2011;17(2-3):255-87.
98. Marques-de-Sà, Joaquim P. Applied Statistics, using SPSS, STATISTICA, MATLAB and R: Springer; 2007.
99. Paz W, Ruiz-Blanco YB, Marrero-Ponce Y, Quinteros IM. PROTDCAL (PROTein Descriptors CALculation program). Facultad de Química y Farmacia. Universidad central "Marta Abreu" de Las Villas: CENDA: 0066-01-2014; 2014.
100. Spearman CE. 'General intelligence' objectively determined and measured. American Journal of Psychology. 1994;5:201-93.
101. Spearman CE. Proof and measurement of association between two things. American Journal of Psychology. 1904;15:72-101.
102. Urias RP, Barigye S, Marrero-Ponce Y, García-Jacas C, Valdes-Martiní J, Perez-Gimenez F. IMMAN: free software for information theory-based chemometric analysis. Mol Divers. 2015 2015/01/26:1-15.
103. Shannon CE. A mathematical theory of communication. Bell System Technical Journal. 1948;27:623-56.
104. Martínez Pérez E, Castillo-Garit JA, Ruiz Blanco YB. Big-Datasets Manager: Una herramienta libre para la manipulación de ficheros de datos con número elevado de instancias y atributos. Facultad de Química y Farmacia. Universidad Central Marta Abreu de Las Villas: CENDA; 2015.
105. Martínez Pérez E, Castillo-Garit JA, Ruiz Blanco YB. Big-Datasets Manager: Una herramienta libre para la manipulación de ficheros de datos con número elevado de instancias y atributos. Nereis. 2015;7:59-66.

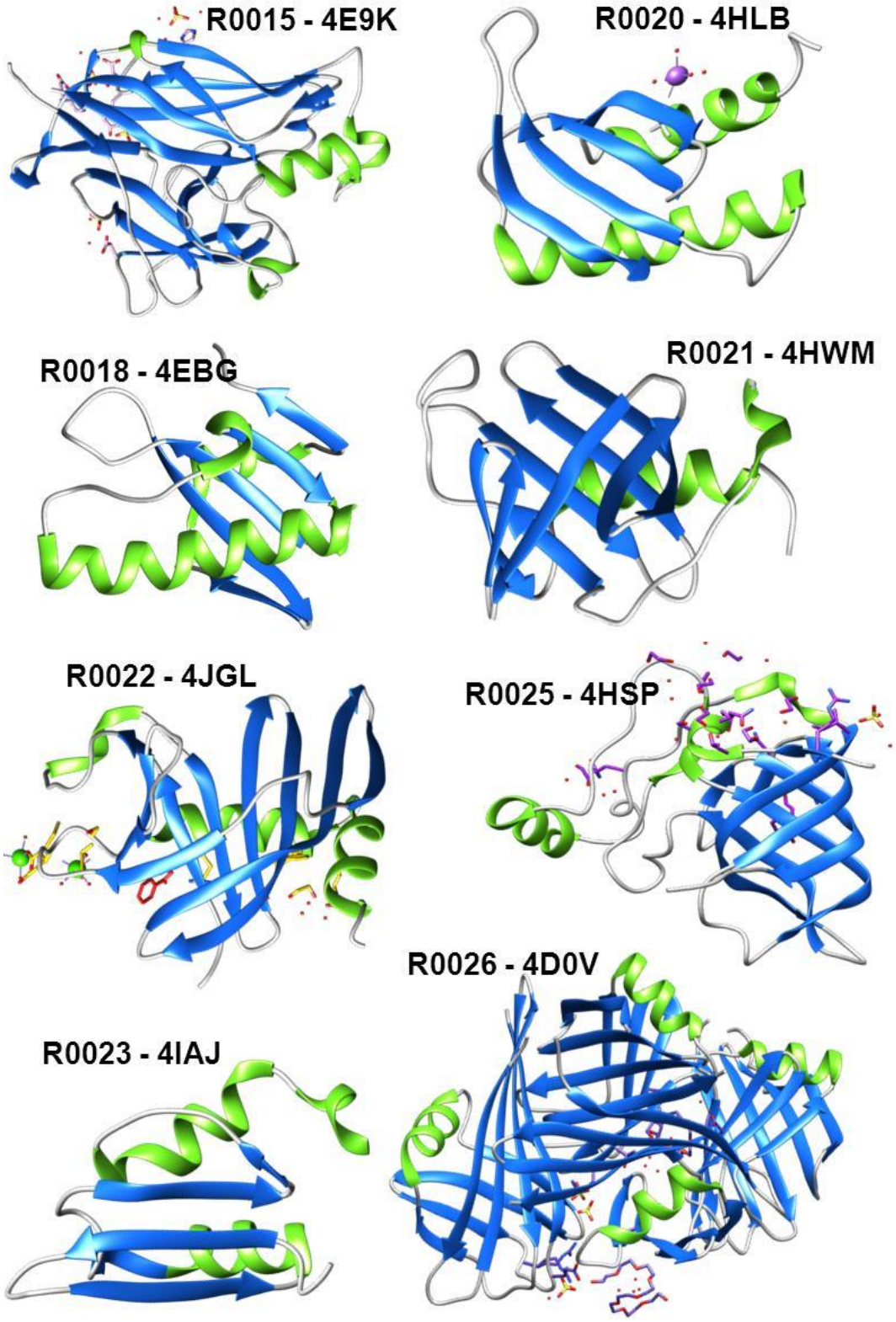
VIII. Anexos

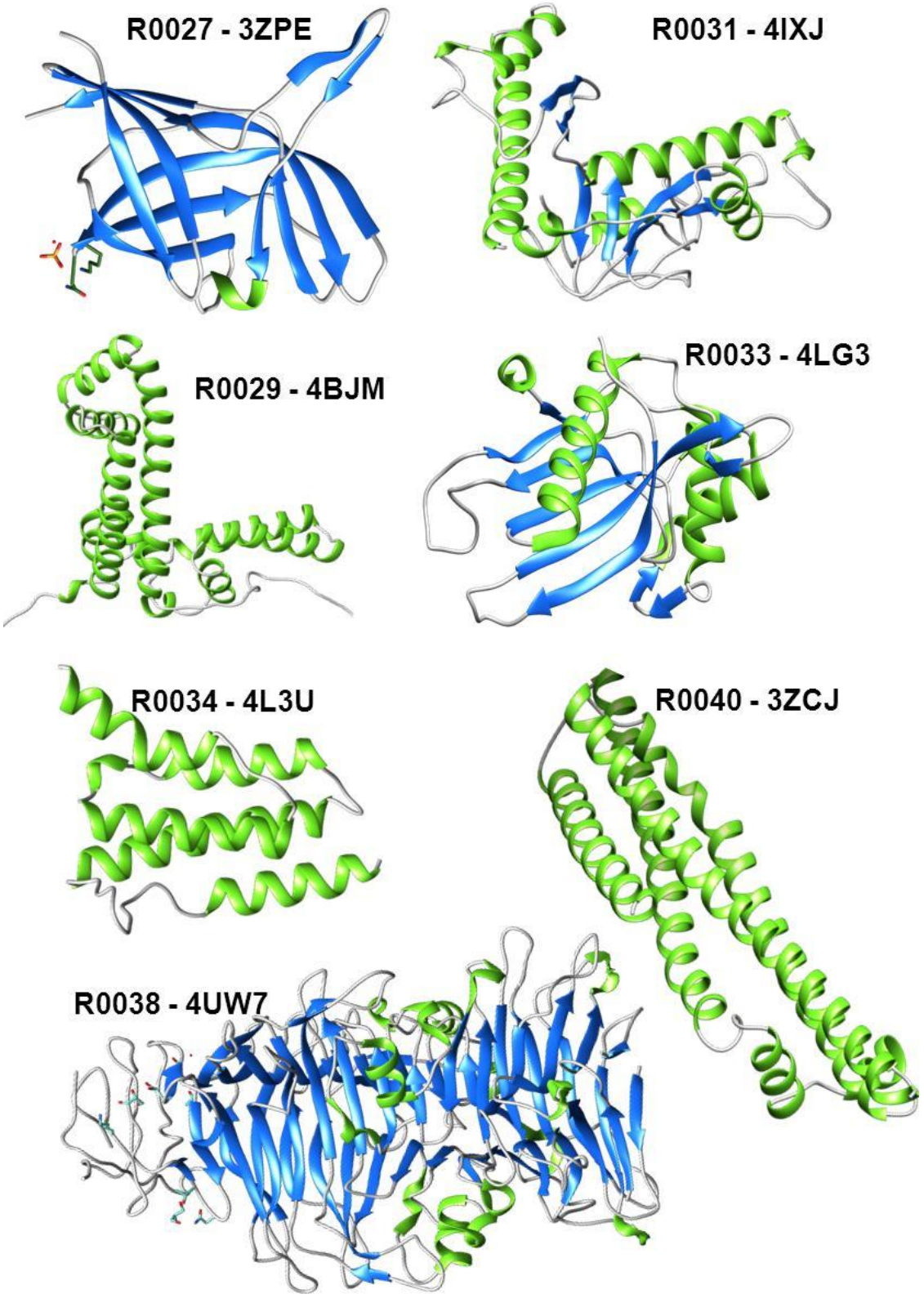
VIII. 1. Estructura Química de los Aminoácidos Naturales

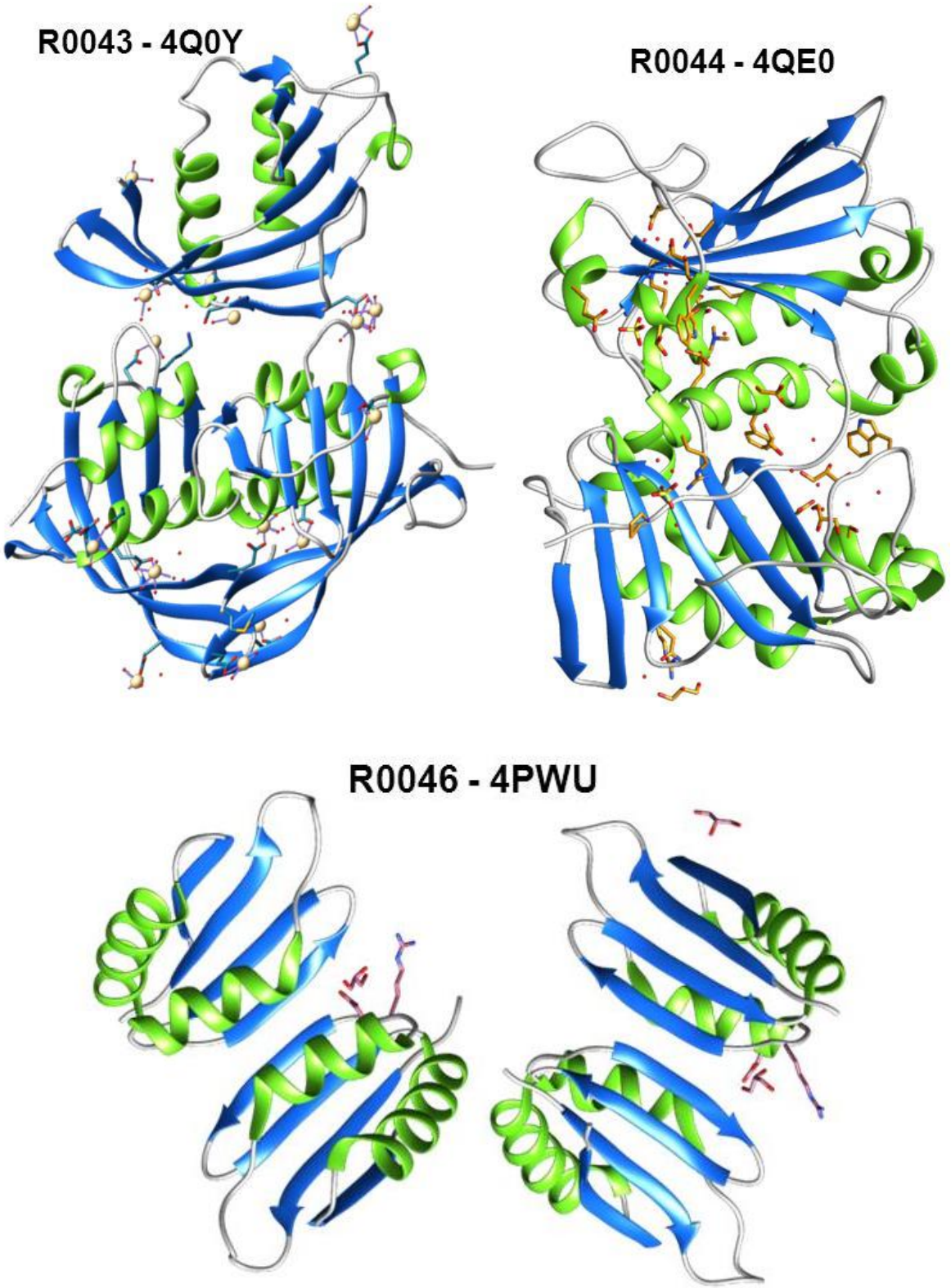
Alamina-ALA 	Glicina-GLY 	Cisteína-CYS 	Serina-SER 
Acido Aspártico-ASP 	Prolina-PRO 	Valina-VAL 	Treonina-THR 
Isoleucina-ILE 	Leucina-LEU 	Asparragina-ASN 	Fenilalanina-PHE 
Tirosina-TYR 	Glutamina-GLN 	Ácido Glutamico-GLU 	Lisina-LYS 
Histidina-HIS 	Metionina-MET 	Triptófano-TRP 	Arganina-ARG 

VIII. 2. Estructuras 3D de las 26 proteínas nativas




R0015 - 4E9K
R0020 - 4HLB
R0018 - 4EBG
R0021 - 4HWM
R0022 - 4JGL
R0025 - 4HSP
R0026 - 4D0V
R0023 - 4IAJ


R0027 - 3ZPE
R0031 - 4IXJ
R0029 - 4BJM
R0033 - 4LG3
R0034 - 4L3U
R0040 - 3ZCJ
R0038 - 4UW7


R0043 - 4Q0Y
R0044 - 4QE0
R0046 - 4PWU

VIII. 3. Función Objetivo promedio en las configuraciones de VMO

Proteína	VMO_0-5	VMO_1	VMO_1-5	VMO_2
R0001	-960.77	**-974.84**	-902.58	-919.43
R0002	**-1050.07**	-1017.24	-1017.72	-994.06
R0006	-1040.19	**-1060.94**	-1048.09	-1035.04
R0007	**-944.44**	-939.75	-931.69	-904.98
R0008	-716.97	**-745.94**	-731.37	-723.18
R0009	-844.79	**-882.82**	-858.27	-864.23
R0013	-639.75	**-665.69**	-644.19	-626.43
R0014	**-709.90**	-654.87	-694.80	-662.15
R0015	-1327.55	**-1345.96**	-1294.31	-1239.85
R0018	**-675.46**	-661.01	-645.13	-669.88
R0020	-706.99	-713.50	-707.54	**-729.64**
R0021	-769.15	**-776.47**	-776.24	-733.74
R0022	**-1056.31**	-1029.58	-1049.84	-1000.06
R0023	-592.66	-599.15	-588.00	**-608.11**
R0025	-913.91	**-931.87**	-914.16	-912.80
R0026	-606.12	-599.60	-602.90	**-620.62**
R0027	-873.88	**-876.25**	-850.29	-839.87
R0029	-1157.30	**-1173.31**	-1167.51	-1161.22
R0031	**-1397.72**	-1391.31	-906.60	-1299.39
R0033	-1062.65	-1050.44	**-1066.36**	-1001.19
R0034	-693.75	-726.38	**-744.96**	-679.95
R0038	-1293.67	-1294.78	**-1312.19**	-1249.69
R0040	-1063.87	-1032.90	**-1066.17**	-1010.40
R0043	**-913.40**	-891.09	-912.76	-890.47
R0044	-1086.87	**-1125.14**	-1098.95	-1074.46
R0046	**-754.54**	-742.31	-736.79	-727.90

VIII. 4. Función Objetivo en las configuraciones de SA

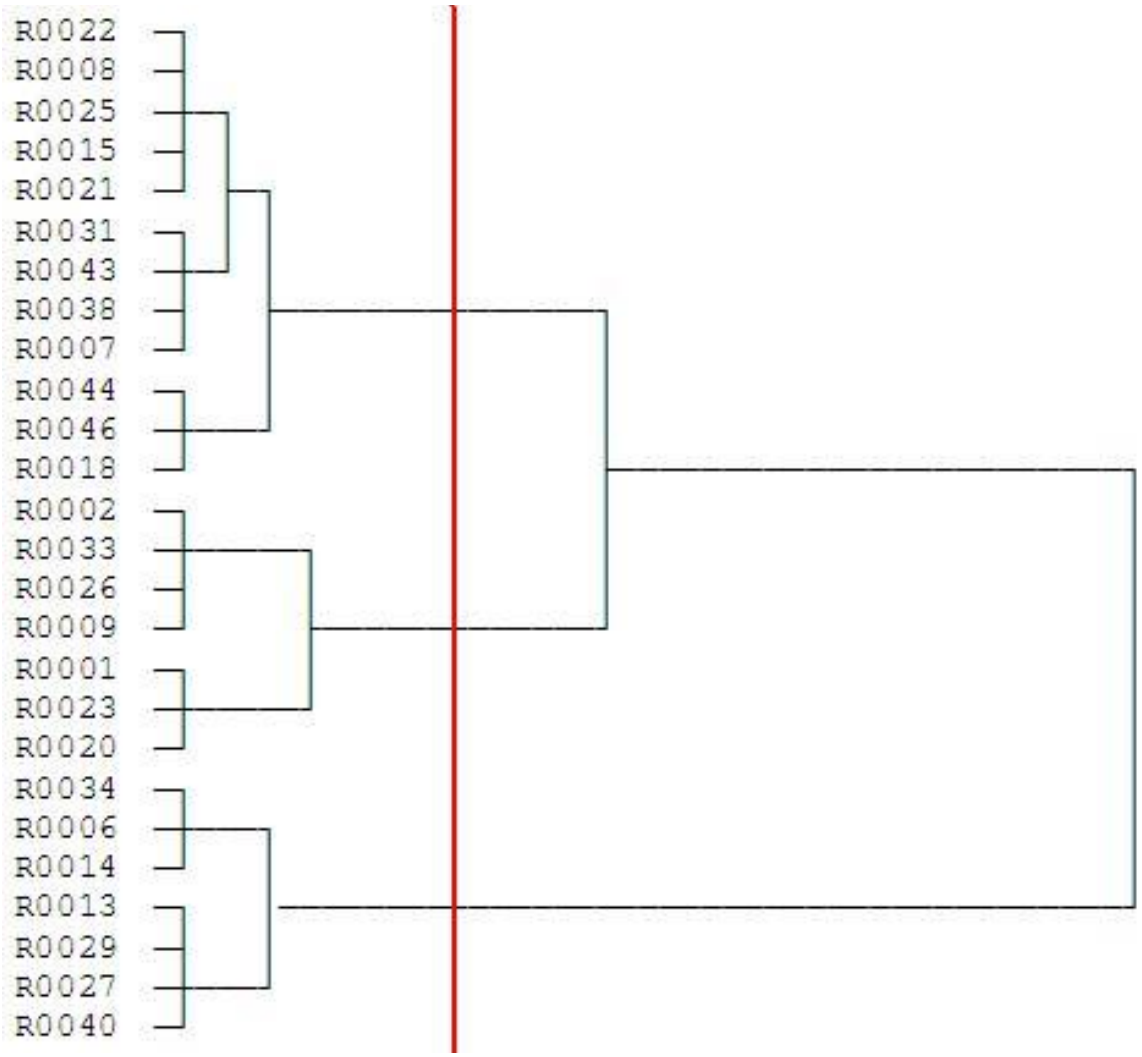
Proteína	1_100	1_200	1_400	1_600	2_100	2_200	2_400	2_100
R0001	-988.6	**-998.34**	-945.03	-957.19	-906.77	-910.29	-904.45	-906.77
R0002	-1054.2	**-1086.97**	-1077.31	-1034.09	-989.60	-995.46	-976.59	-989.60
R0006	-1051.2	**-1113.21**	-1067.05	-1045.99	-1004.96	-970.48	-994.04	-1004.96
R0007	-927.5	-958.77	**-970.03**	-963.65	-903.94	-917.26	-885.35	-903.94
R0008	-712.3	-734.72	-718.47	**-742.34**	-698.56	-703.52	-690.95	-698.56
R0009	**-860.8**	-858.73	-857.20	-837.20	-818.19	-787.57	-805.86	-818.19
R0013	-675.2	**-680.59**	-657.06	-676.77	-631.64	-628.76	-640.27	-631.64
R0014	**-732.7**	-725.97	-698.39	-696.59	-674.10	-659.59	-686.80	-674.10
R0015	-1374.3	-1319.56	-1309.27	**-1380.25**	-1242.94	-1264.14	-1224.43	-1242.94
R0018	-690.6	**-707.12**	-665.20	-686.22	-643.83	-649.35	-634.18	-643.83
R0020	-725.2	-733.88	**-749.02**	-732.43	-674.11	-687.33	-676.08	-674.11
R0021	-759.6	-756.84	-761.93	**-770.45**	-726.59	-715.67	-716.40	-726.59
R0022	-1029.3	-1030.67	-1014.35	**-1040.58**	-986.07	-983.35	-952.47	-986.07
R0023	-586.5	-594.82	**-594.87**	-591.99	-572.45	-586.02	-568.85	-572.45
R0025	-941.7	**-957.35**	-915.59	-948.64	-876.81	-892.46	-890.76	-876.81
R0026	-600.3	-606.58	**-617.80**	-615.74	-591.65	-579.17	-580.49	-591.65
R0027	**-890.9**	-861.39	-880.62	-879.94	-863.10	-857.20	-801.57	-863.10
R0029	**-1230.7**	-1180.13	-1176.88	-1209.81	-1133.25	-1132.21	-1098.05	-1133.25
R0031	-1388.1	-1363.92	-1385.62	**-1403.56**	-1326.92	-1268.08	-1290.24	-1326.92
R0033	**-1099.3**	-1092.88	-1077.35	-1064.93	-1014.01	-1004.82	-1041.01	-1014.01
R0034	-712.8	**-733.08**	-728.45	-703.14	-697.09	-701.37	-680.52	-697.09
R0038	-1321.5	**-1340.36**	-1317.22	-1296.91	-1229.13	-1262.99	-1257.12	-1229.13
R0040	-1043.8	**-1069.37**	-1050.57	-1021.54	-984.77	-1006.06	-1014.20	-984.77
R0043	**-915.1**	-913.10	-912.69	-898.73	-888.52	-847.87	-868.51	-888.52
R0044	**-1159.5**	-1146.40	-1130.40	-1126.63	-1080.42	-1081.92	-1064.18	-1080.42
R0046	-745.6	**-749.30**	-737.03	-739.96	-680.00	-719.26	-712.24	-680.00

Proteína	3_100	3_200	3_400	3_600	4_100	4_200	4_400	4_100
R0001	-920.31	-860.27	-887.02	-825.83	-848.83	-846.40	-810.86	-818.11
R0002	-973.68	-918.94	-957.46	-944.55	-958.62	-903.60	-909.11	-925.35
R0006	-962.87	-953.76	-949.42	-911.43	-929.23	-897.67	-923.86	-932.83
R0007	-896.12	-875.36	-846.11	-871.95	-867.69	-818.31	-825.28	-828.52
R0008	-686.20	-659.58	-666.15	-664.84	-672.44	-637.16	-627.69	-648.40
R0009	-801.33	-812.43	-798.68	-768.90	-781.20	-747.48	-726.37	-744.14
R0013	-644.86	-605.26	-606.78	-601.05	-608.57	-570.08	-600.94	-589.18
R0014	-674.16	-645.58	-647.85	-654.43	-639.67	-602.31	-627.28	-618.23
R0015	-1278.31	-1220.35	-1260.80	-1221.98	-1198.30	-1182.47	-1159.98	-1186.48
R0018	-632.26	-609.61	-617.53	-615.87	-610.68	-592.78	-602.36	-601.32
R0020	-697.70	-652.75	-648.81	-652.33	-648.95	-644.03	-653.56	-632.40
R0021	-710.86	-691.10	-674.78	-703.45	-685.40	-664.26	-686.91	-659.57
R0022	-966.98	-907.17	-945.53	-937.55	-950.45	-939.07	-905.40	-914.91
R0023	-566.91	-549.34	-558.88	-555.86	-533.99	-529.98	-528.51	-552.12
R0025	-883.30	-864.92	-862.05	-853.65	-883.25	-815.50	-848.81	-822.11
R0026	-594.30	-530.48	-556.45	-556.47	-545.14	-547.47	-533.43	-550.91
R0027	-833.79	-819.24	-774.00	-793.35	-809.40	-801.67	-774.31	-761.96
R0029	-1092.71	-1101.78	-1068.99	-1084.23	-1073.89	-1074.83	-1042.34	-1024.92
R0031	-1349.78	-1252.15	-1237.43	-1233.09	-1227.86	-1258.47	-1236.12	-1242.04
R0033	-977.72	-990.05	-941.60	-940.83	-928.39	-945.19	-931.30	-924.58
R0034	-714.62	-659.23	-648.79	-663.32	-638.60	-641.85	-634.82	-630.34
R0038	-1250.26	-1162.54	-1221.45	-1201.28	-1189.54	-1139.86	-1167.07	-1120.28
R0040	-989.83	-952.04	-982.65	-939.64	-976.89	-928.17	-893.60	-934.90
R0043	-894.25	-823.88	-817.41	-825.80	-817.28	-800.53	-787.52	-794.93
R0044	-1050.36	-1007.63	-1039.40	-1017.02	-1009.48	-991.60	-1006.24	-971.80
R0046	-710.39	-680.09	-681.03	-683.04	-655.15	-632.78	-654.44	-648.37

VIII. 5. Función Objetivo en las configuraciones de GA

Proteína	70_15	70_20	70_25	75_15	75_20	75_25	80_15	80_20	80_25
R0001	-936.31	-920.58	-931.41	**-950.51**	-947.93	-939.19	-947.63	-927.45	-946.25
R0002	-957.01	-1000.50	-1014.21	**-1014.85**	-987.48	-988.04	-999.66	-979.03	-981.79
R0006	-1047.29	-1033.12	**-1047.54**	-1014.19	-1043.29	-1033.42	-1025.84	-1028.99	-1044.62
R0007	-911.29	**-956.95**	-941.92	-942.59	-925.04	-933.28	-919.95	-938.24	-935.00
R0008	-709.79	-716.49	-707.61	-718.70	-702.37	-714.75	-718.27	-694.59	**-736.58**
R0009	-826.21	-833.83	**-864.76**	-846.71	-830.26	-859.38	-838.80	-833.03	-853.22
R0013	-645.00	-667.57	**-668.32**	-617.70	-631.68	-654.86	-643.49	-651.87	-662.85
R0014	-670.84	-687.39	-680.80	-696.40	-700.01	-688.33	-683.39	**-705.64**	-677.35
R0015	-1338.29	-1336.59	-1316.91	-1339.13	-1334.65	**-1340.75**	-1313.59	-1317.59	-1275.91
R0018	-669.79	-666.95	-667.99	-657.94	-660.58	-675.11	-644.62	**-679.18**	-659.97
R0020	-705.11	-695.92	-703.74	-713.31	**-729.10**	-698.02	-722.12	-700.51	-703.77
R0021	-760.70	-762.69	**-778.88**	-750.44	-751.68	-753.37	-760.51	-752.59	-773.08
R0022	-1009.48	**-1072.97**	-1010.18	-958.78	-1000.84	-1000.48	-1023.74	-984.04	-990.79
R0023	-586.12	-578.58	-600.14	**-603.95**	-588.23	-595.23	-586.82	-588.47	-592.77
R0025	-915.70	-927.06	-918.93	-930.22	-920.49	**-931.51**	-903.23	-914.82	-920.41
R0026	-594.98	-583.32	-602.76	**-613.41**	-595.14	-605.98	-596.47	-579.85	-606.69
R0027	-823.65	-867.30	-874.98	-840.78	-851.66	**-879.22**	-856.33	-840.22	-866.38
R0029	-1161.45	**-1225.60**	-1170.93	-1151.09	-1202.37	-1167.58	-1187.62	-1170.46	-1175.85
R0031	-1402.43	-1378.07	-1399.12	-1381.43	-1352.88	-1376.52	-1352.71	**-1410.27**	-1398.75
R0033	**-1073.77**	-1025.20	-1062.61	-1065.77	-1056.52	-1063.56	-1057.60	-1058.21	-1057.16
R0034	-703.11	-693.31	-694.21	-713.54	-710.45	-681.18	-704.77	-705.91	**-723.43**
R0038	-1310.99	**-1338.34**	-1315.75	-1306.69	-1275.44	-1305.85	-1319.09	-1285.40	-1263.86
R0040	-1059.91	**-1073.68**	-1055.42	-1065.49	-1037.45	-1038.89	-1053.79	-1049.94	-1037.88
R0043	-866.08	**-901.50**	-900.60	-877.22	-895.58	-879.65	-895.50	-877.80	-890.88
R0044	-1096.07	-1139.09	-1133.35	-1127.66	-1108.66	**-1174.08**	-1123.52	-1114.06	-1171.81
R0046	-716.98	-709.87	-721.63	-728.74	-719.71	-724.26	**-735.85**	-732.44	-710.33

VIII. 6. Dendograma de Clúster: Complete Linkage



VIII. 7. Promedio de la métricas en las tres metaheurísticas

Aclaraciones sobre las métricas:

- N, GDT y LGAS: mientras mayor sean sus valores, son mejores los resultados
- RMSD: mientras menores sean los valores, mejores son los resultados

	N			RMSD			GDT			LGAS		
Prot.	**GA**	**SA**	**VMO**	**GA**	**SA**	**VMO**	**GA**	**SA**	**VMO**	**GA**	**SA**	**VMO**
R0001	14.333	16.333	16.000	2.663	2.963	2.963	9.244	9.108	8.880	7.237	7.266	7.140
R0002	15.667	16.333	14.667	2.837	3.037	2.740	11.712	12.387	11.712	9.335	9.692	9.364
R0006	15.000	16.000	14.333	2.923	2.930	2.783	10.059	10.256	9.517	8.070	8.436	7.843
R0007	19.667	19.667	18.000	2.490	2.897	2.790	14.079	13.199	12.526	11.736	10.853	9.971
R0008	17.000	17.667	16.333	2.780	2.967	2.837	16.372	16.667	16.077	13.762	13.596	13.307
R0009	14.667	15.333	14.333	2.573	2.540	2.807	10.268	9.524	9.524	8.281	8.020	7.796
R0013	13.000	14.000	14.000	2.967	2.723	2.697	12.972	13.129	13.207	10.745	11.217	11.048
R0014	13.000	13.333	13.667	2.733	2.750	2.720	11.667	12.361	12.500	9.709	10.226	10.221
R0015	17.000	18.333	15.333	2.690	2.920	2.810	8.195	7.431	7.778	6.453	6.172	6.310
R0018	21.667	14.000	15.333	2.597	2.877	2.537	21.875	17.188	16.754	18.225	13.858	14.112
R0020	19.000	15.000	15.333	2.840	2.803	2.903	20.175	17.368	15.790	16.712	14.133	13.563
R0021	17.000	14.333	13.667	3.037	2.400	2.310	14.743	14.103	14.103	11.949	11.500	11.303
R0022	18.333	14.667	17.000	2.060	2.610	2.887	13.323	10.417	12.774	10.961	8.644	10.079
R0023	13.667	17.000	15.333	2.547	2.763	2.710	21.491	21.710	20.614	17.605	18.936	17.644
R0025	15.333	15.667	15.000	2.790	2.703	2.563	11.937	11.543	10.698	9.554	9.491	8.966
R0026	16.000	16.333	14.333	2.860	2.810	2.810	15.972	15.201	13.812	12.932	12.648	11.430

R0027	14.000	14.667	13.667	3.003	2.717	2.700	11.775	11.896	10.870	9.319	9.640	9.201
R0029	24.000	20.000	21.333	2.480	2.990	2.870	12.168	9.403	10.619	9.953	7.600	8.699
R0031	22.333	19.000	17.000	2.823	2.820	2.823	11.086	9.450	8.631	8.757	7.666	6.957
R0033	20.000	17.667	17.000	2.840	2.813	2.810	12.151	10.289	10.149	9.937	8.360	8.276
R0034	22.333	15.000	16.667	2.920	2.777	2.370	19.919	14.566	15.041	16.468	11.662	12.627
R0038	16.333	18.000	16.000	2.727	3.027	2.657	6.843	6.417	6.569	5.452	5.215	5.322
R0040	21.000	16.000	17.667	2.820	2.803	2.757	11.898	9.180	9.849	9.772	7.631	8.232
R0043	14.333	16.667	14.000	2.713	2.767	2.677	12.308	12.949	12.115	10.099	10.977	9.925
R0044	18.000	17.333	17.000	2.800	2.773	2.523	12.076	11.427	11.377	10.086	9.314	9.159
R0046	16.667	17.000	14.333	2.573	2.777	2.783	23.794	23.027	21.053	19.740	19.558	17.481

VIII. 8. Rank de GA en promedio, respecto a predictores internacionales, en las cuatro métricas

Proteína	N	RMSD	GDT	LGAS
R0001	0.00	80.00	5.00	0.00
R0002	16.67	27.78	50.00	16.67
R0006	16.67	0.00	72.22	50.00
R0007	0.00	78.95	0.00	0.00
R0008	16.67	16.67	44.44	38.89
R0009	10.53	47.37	52.63	26.32
R0013	5.26	0.00	5.26	0.00
R0014	0.00	21.05	0.00	0.00
R0015	31.58	57.89	68.42	31.58
R0018	29.41	23.53	29.41	29.41
R0020	18.75	0.00	0.00	0.00
R0021	78.57	0.00	71.43	64.29
R0022	87.50	**93.75**	**93.75**	**93.75**
R0023	18.75	68.75	81.25	81.25
R0025	20.00	33.33	60.00	26.67
R0026	**91.67**	16.67	83.33	75.00
R0027	10.00	0.00	20.00	20.00
R0029	10.00	0.00	40.00	10.00
R0031	**90.91**	0.00	81.82	81.82
R0033	**83.33**	16.67	**83.33**	**83.33**
R0034	54.55	0.00	63.64	72.73
R0038	10.00	30.00	40.00	10.00
R0040	16.67	0.00	33.33	16.67
R0043	0.00	44.44	22.22	11.11
R0044	**83.33**	33.33	**83.33**	**83.33**
R0046	20.00	60.00	20.00	20.00

VIII. 9. Rank de la mejor solución de GA, respecto a las soluciones de predictores internacionales, en las cuatro métricas

Proteína	N	RMSD	GDT	LGAS
R0001	10.204	90.816	12.245	8.163
R0002	55.172	54.023	58.621	45.977
R0006	60.920	43.678	74.713	54.023
R0007	17.204	91.398	16.129	12.903
R0008	76.136	88.636	85.227	80.682
R0009	36.957	73.913	72.826	53.261
R0013	53.261	35.870	52.174	45.652
R0014	8.791	58.242	5.495	4.396
R0015	64.130	95.652	77.174	65.217
R0018	71.951	56.098	30.488	39.024
R0020	42.105	48.684	13.158	10.526
R0021	88.235	27.941	89.706	89.706
R0022	81.818	98.701	97.403	96.104
R0023	60.526	67.105	85.526	81.579
R0025	61.111	90.278	87.500	63.889
R0026	81.132	43.396	98.113	92.453
R0027	21.277	97.872	68.085	25.532
R0029	42.553	29.787	76.596	65.957
R0031	96.226	47.170	96.226	98.113
R0033	95.652	73.913	91.304	91.304
R0034	87.500	35.417	81.250	87.500
R0038	72.340	63.830	97.872	85.106
R0040	42.857	17.857	60.714	39.286
R0043	69.767	86.047	79.070	72.093
R0044	96.154	80.769	96.154	96.154
R0046	54.545	81.818	50.000	50.000