INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

DOCTORADO EN CIENCIAS DE LA INGENIERÍA

TESIS

MÉTODOS HSA PARA DOBLADO DE PROTEÍNAS CON MODELACIÓN DE GRANO GRUESO

Que para obtener el grado de
Doctor en Ciencias de la Ingeniería

Presenta
M.I.E. Diego Arturo Soto Monterrubio
D05071339
389515

Director de Tesis
Dr. Juan Frausto Solís
31308

Co-director de Tesis
Dr. Juan Paulo Sánchez Hernández

Cd. Madero, Tamaulipas Marzo 2023

Instituto Tecnológico de Ciudad Madero
Subdirección Académica
División de Estudios de Posgrado e Investigación

Ciudad Madero, Tamaulipas, 25/noviembre/2022

OFICIO No. : U.154/22
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. DIEGO ARTURO SOTO MONTERRUBIO
No. DE CONTROL D05071339
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su Examen de Grado de Doctorado en Ciencias de la Ingeniería, se acordó autorizar la impresión de su tesis titulada:

"MÉTODOS HSA PARA DOBLADO DE PROTEÍNAS CON MODELACIÓN DE GRANO GRUESO"

El Jurado está integrado por los siguientes catedráticos:

PRESIDENTE:	DR.	JUAN FRAUSTO SOLÍS
SECRETARIO:	DR.	RUBÉN SALAS CABRERA
PRIMER VOCAL:	DR.	JUAN JAVIER GONZÁLEZ BARBOSA
SEGUNDO VOCAL:	DR.	ULISES PÁRAMO GARCÍA
TERCER VOCAL:	DR.	PEDRO MARTÍN GARCÍA VITE
DIRECTOR DE TESIS:	DR.	JUAN FRAUSTO SOLÍS
CO-DIRECTOR:	DR.	JUAN PAULO SÁNCHEZ HERNÁNDEZ

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Por mi patria y por mi bien"®

MARCO ANTONIO CORONEL GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

c.c.p.- Archivo
MACG 'LFCS'

DECLARACIONES DE ORIGINALIDAD, PROPIEDAD INTELECTUAL, CESIÓN DE DERECHOS Y/O CONFIDENCIALIDAD

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares. Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones. Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

Marzo de 2023, Cd. Madero, Tamaulipas

MIE Diego Arturo Soto Monterrubio

DEDICATORIA

Dedico este trabajo a mi familia, mi esposa, hijos, hermanos, padre, madre, mi director, co-director y miembros del comité de tesis. A todos ellos dedico el presente trabajo, porque han fomentado en mí, el deseo de superación y de triunfo en este camino llamado vida. Lo que ha contribuido a la consecuencia de este importante logro. Espero contar siempre con su valioso apoyo.

AGRADECIMIENTOS

Agradezco a Dios primeramente por darme la oportunidad de tener una gran familia y permitirme continuar en esta vida para poder realizar mis metas, propósitos y principalmente lograr terminar este trabajo.

Agradezco a mi esposa, hermanos, mi padre y madre por todo su apoyo.

Agradezco a mis hijos por ser una gran inspiración para concluir este trabajo.

Gracias a mi director, co-director por apoyarme y guiarme en este trabajo.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico brindado durante la realización de este proyecto.

Agradezco al Instituto Tecnológico de Ciudad Madero, la cual es una institución de enorme calidad, que me brindó todo el apoyo durante mi estancia.

Métodos HSA para doblado de proteínas con modelación de grano grueso.

Diego Arturo Soto Monterrubio.

Resumen

El problema de doblado de proteínas o PFP (de sus siglas en inglés, Protein Folding Problem) es un reto enorme en diferentes áreas de conocimiento como biología molecular, biología computacional y ciencias de la computación. Este problema consiste en obtener la estructura nativa o terciaria funcional de una proteína o péptido a partir de su secuencia de aminoácidos (estructura primaria). Durante el proceso de obtención a partir de la estructura primaria hasta la estructura terciaria los péptidos pueden obtener una gran cantidad de formas diferentes en su estructura tridimensional de átomos que la conforman y representa la energía libre de Gibbs más baja. Existen métodos que solo utilizan la secuencia de aminoácidos de la proteína y métodos que no solo utilizan está secuencia para resolver el PFP. En este documento se presentan métodos híbridos de recocido simulado (SA, Simulated Annealing) para la solución del PFP basados en ab initio se ha implementado un algoritmo muy eficaz denominado GRSA que es del tipo HSA (Hybrid Simulated Annealing) el cual ha dado lugar a varios algoritmos de ese tipo como GRSA2. Los algoritmos GRSA son sintonizados analíticamente y pueden teóricamente utilizar al menos tres tipos de estrategias de mejora en el área de HSA: a) Los que exploran nuevas soluciones a partir de una solución inicial basada en una predicción de ángulos de torsión basada en el conocimiento, siendo los más prometedores los que obtienen soluciones a partir de la estructura secundaria y los que las obtienen de otros algoritmos, b) Métodos novedosos para mejorar el proceso completo de GRSA a partir de optimización combinatoria y/o basados en la física; estos métodos se emplean para perturbar soluciones y generar otras; y c) Métodos de optimización para la mejora del proceso, a partir de subdividirlo en fases, recomenzarlo o detenerlo por equilibrio dinámico métodos de recalentamiento. Con estas estrategias se han obtenido buenos resultados y estas técnicas, con las estrategias tipo a, b y c se han publicado en la literatura. Cabe señalar que los algoritmos GRSA en particular los denominado GRSA2 han obtenido las mejores soluciones de PFP en el caso de péptidos de forma que su investigación es muy prometedora. Está propuesta de tesis busca desarrollar algoritmos tipo GRSA utilizando estrategias basadas en el conocimiento o tipo a) antes descritas, además de buscar perfeccionar los otros dos tipos de técnicas. El desarrollo de dichas estrategias es un gran reto pues se deben investigar métodos de diferentes áreas del conocimiento particularmente de la física, biotecnología y optimización combinatoria. Para realizar la predicción de una proteína o péptido (instancia) se realiza una sintonización analítica en los algoritmos de HSA. En este documento, se presentan resultados de un conjunto de instancias probadas en los algoritmos HSA y se realiza un análisis de resultados para este conjunto de instancias, mostrando un buen desempeño para GRSA2. Los resultados de GRSA2 son comparados con algoritmos del estado del arte. Posterior al GRSA2, se describe el método GRSA2-SSP con estructuras secundarias y GRSA2-SSPR con la estrategia de selección por ruleta para el refinamiento de cadenas laterales. Por ultimo se presenta el método GRSA2-FCNN, el cual es basado en fragmentos generados por una red neuronal convolucional, aplicando este método a un conjunto de instancias similar al GRSA2-SSP. Se presentan resultados de los métodos GRSA2-SSP, GRSA2-SSPR y GRSA2-FCNN los cuales son comparados con los servidores del estado del arte.

HSA methods for protein folding with coarse-grained modeling.

Diego Arturo Soto Monterrubio.

Abstract

The Protein Folding Problem (PFP) is a huge challenge in different areas of knowledge such as molecular biology, computational biology and computer science. This problem consists of obtaining the native or functional tertiary structure of a protein or peptide from its amino acid sequence (primary structure). During the process of obtaining from the primary structure to the tertiary structure, peptides can obtain a large number of different forms in its three-dimensional structure of atoms that make it up and represents the lowest Gibbs free energy. There are methods that only use the amino acid sequence of the protein and methods that do not only use this sequence to solve the PFP. In this document we present hybrid simulated annealing (SA) methods for the solution of PFP based on ab initio a very efficient algorithm called GRSA has been implemented which is of the HSA (Hybrid Simulated Annealing) type which has given rise to several such algorithms such as GRSA2. GRSA algorithms are analytically tuned and can theoretically use at least three types of improvement strategies in the HSA area: (a) those that explore new solutions from an initial solution based on a knowledge-based prediction of torsion angles, the most promising being those that obtain solutions from secondary structure and those that obtain them from other algorithms, (b) novel methods for improving the entire GRSA process from combinatorial and/or physics-based optimization; these methods are used to perturb solutions and generate others; and c) Optimization methods for process improvement, from subdividing it into phases, restarting it or stopping it by dynamic equilibrium reheating methods. Good results have been obtained with these strategies and these techniques, with strategies type a, b and c have been published in the literature. It should be noted that GRSA algorithms in particular those called GRSA2 have obtained the best PFP solutions in the case of peptides so that their investigation is very promising. This thesis proposal seeks to develop GRSA-type algorithms using the knowledge-based or type a) strategies described above, in addition to seeking to refine the other two types of techniques. The development of such strategies is a great challenge since methods from different areas of knowledge, particularly physics, biotechnology and combinatorial optimization, must be investigated. To perform the prediction of a protein or peptide (instance), an analytical tuning is performed in the HSA algorithms. In this paper, results are presented for a set of instances tested on HSA algorithms and an analysis of results for this set of instances is performed, showing good performance for GRSA2. The results of GRSA2 are compared with state-of-the-art algorithms. After GRSA2, the GRSA2-SSP method with secondary structures and GRSA2-SSPR with the roulette selection strategy for sidechain refinement are described. Finally, the GRSA2-FCNN method is presented, which is based on fragments generated by a convolutional neural network, applying this method to a set of instances similar to GRSA2-SSP. Results of the GRSA2-SSP, GRSA2-SSPR and GRSA2-FCNN methods are presented and compared with the state-of-the-art servers.

Índice General

Índice Tablas

Índice de Figuras

Capítulo 1

1 Introducción

Las proteínas desempeñan una serie de funciones biológicas vitales, entre las que se encuentran: la síntesis y mantenimiento de tejidos, ayudan a transportar gases como el oxígeno y el dióxido de carbono a través de la sangre. Se han hecho investigaciones sobre las propiedades estructurales y funcionales de las proteínas, estas tienen diferentes estructuras que van desde la primaria hasta la cuaternaria. En consecuencia, comprender el proceso del cambio de la estructura primaria a la terciaria es un gran reto de investigación para el cual se han realizado estudios sobre este proceso desde la primera proteína propuesta por L. Pauling [Mckee & Mckee, 2014].

La investigación realizada por L. Pauling en 1951 de la existencia de un estado termodinámicamente estable con una estructura conformada por una cadena de aminoácidos para ciertos tipos de proteínas, establece por primera vez un puente entre los campos de la Biología Molecular y la Física. Las observaciones experimentales hechas primero por Watson y Crick para la molécula de ADN en 1953 y posteriormente por Kendrew para la molécula de mioglobina en 1958, establece simultáneamente la relación existente entre función y estructura. Esta relación se puede observar en el problema del plegamiento o doblado de una proteína (PFP, de sus siglas en inglés Protein Folding Problem), el cual se refiere a la explicación de los procesos a través de los cuales una proteína adquiere una configuración tridimensional termodinámicamente estable. La existencia del estado nativo de las proteínas es importante para que pueda ejercer sus funciones biológicas [Olivarez Quiroz & García Colín, 2004].

1. Introducción

El problema de doblado de proteínas es un gran reto en diversas áreas del conocimiento, tales como biología molecular, biofísica, biología computacional y ciencias de la computación. Una proteína puede tomar diferentes formas desde su estructura primaria hasta su estructura nativa la cual corresponde con la estructura tridimensional de los átomos que conforman la proteína y la que presenta la energía libre de Gibbs más baja. Se ha considerado en la literatura que el plegamiento incorrecto de proteínas es la causa primaria de la enfermedad de Alzheimer, la enfermedad de Parkinson y otras enfermedades [Chaudhuri & Paul, 2006].

Existen diversas clasificaciones de los métodos que buscan resolver el PFP; se pueden clasificar en dos grupos: métodos que solo utilizan la secuencia de aminoácidos de la proteína o estructura primaria y métodos que no solo utilizan esta secuencia. En dichos métodos se ha buscado determinar el algoritmo que determine el camino que debe seguir una proteína desde su secuencia de aminoácidos hasta su estructura nativa.

El método que utiliza solo la secuencia de aminoácidos es denominado "*ab initio*" que consiste en determinar la estructura nativa solo conociendo la estructura primaria. Otro método, es el basado en fragmentos, que ocupa fragmentos cortos de estructuras de proteínas conocidas para obtener la estructura tridimensional o nativa. Los métodos ab initio y ensamble de fragmentos son con frecuencia combinados para formar métodos híbridos que permitan encontrar mejores soluciones. Al implementar estos métodos para resolver PFP se pretende encontrar la solución óptima (estructura nativa) o bien cercana a la óptima, la cual tiene un valor mínimo de energía libre de Gibbs. En este trabajo de investigación se pretende utilizar pequeñas proteínas de hasta 100 aminoácidos y poder contribuir en la mejora de algoritmos que tratan de resolver este problema mejorando la calidad de resultados.

1. Introducción

1.1 Planteamiento del problema

El problema de doblado de proteínas tiene varios años investigándose, es un problema en el que varias áreas del conocimiento humano son combinadas tales como la biología molecular, biofísica, biología computacional y ciencias de la computación. En este problema se busca encontrar la estructura tridimensional en la cual la proteína puede realizar su función biológica. En la naturaleza este proceso de pasar de una estructura primaria a terciaria es alrededor de los nanosegundos y en una computadora los tiempos de resolución aún son grandes que pueden tardar meses. Realizar la predicción de la estructura terciaria de una proteína tiene una complejidad computacional NP duro como en [Hart & Istrail, 1997]. Particularmente el método ab initio que solo utiliza la estructura primaria tiene un buen funcionamiento para predecir estructuras de proteínas más pequeñas, pero para la predicción de proteínas más grandes ya no. Por tal motivo se desarrollan estrategias que también tengan un buen funcionamiento en proteínas más grandes, uno de los métodos es utilizando ensamble de fragmentos mencionado en la sección 2.7.4 para obtener predicciones de estructuras de proteínas con mayor tamaño.

1.2 Objetivos

Esta parte describe el objetivo general y objetivos específicos de esta investigación.

1.2.1 Objetivo general.

Desarrollar estrategias algorítmicas basadas en la información biológica de la proteína y los algoritmos HSA aplicadas a la predicción de proteínas de cinco a cien aminoácidos.

1.2.2 Objetivos específicos.

1. OE-1: Implementar una base de datos de fragmentos de ángulos de aminoácidos cortos de entre 3 y 10 aminoácidos. Identificar adecuadamente ángulos de torsión para determinar los puntos de ensamble.

2. OE-2: Desarrollar estrategias basadas en conocimiento a partir de la estructura secundaria de las proteínas objetivo. Integrar el algoritmo de ensamble de fragmentos para la simulación de grano grueso.

3. OE-3: Desarrollar estrategias de refinamiento de algoritmos HSA para mejorar el proceso completo de GRSA a partir de optimización combinatoria y/o basados en la física; estos métodos se emplean para perturbar soluciones y generar otras nuevas

4. OE-4: Integrar métodos de refinamiento para la mejora del proceso de los algoritmos HSA, mediante la subdivisión de fases, reinicio (métodos de recalentamiento) o detección de la convergencia por equilibrio dinámico.

5. OE-5: Determinar la calidad de las estrategias desarrolladas mediante métricas de desempeño para los algoritmos HSA con un conjunto de péptidos proteínas hasta cien aminoácidos.

1.3 Hipótesis

Desarrollar estrategias algorítmicas basadas en la información biológica de la proteína y los algoritmos HSA aplicadas a la predicción de proteínas de cinco a cien aminoácidos, mejorará la calidad de las predicciones.

1.4 Justificación del estudio

Como se mencionó el PFP es un gran reto dado que una proteína puede tomar diferentes conformaciones desde su estructura primaria hasta su estructura nativa la cual corresponde con la estructura tridimensional de los átomos que conforman la proteína y la que presenta la energía libre de Gibbs más baja. Pero el incorrecto doblado de algunas proteínas es la causa primaria de la enfermedad de Alzheimer, la enfermedad de Parkinson y otras enfermedades [Chauduri and Paul, 2006]. Comprender el PFP puede ayudar para la

manipulación proteínas con lo cual se pueden prevenir estas enfermedades y también elaborar fármacos.

Encontrar métodos de solución para el PFP es un reto importante, por lo que se han creado competencias para que se involucren en ese tema poder solucionarnos, como la competencia conocida CASP (Critical Assessment of Structure Prediction), en la cual los investigadores presentan sus avances de investigación sobre este problema [Kinch, 2021]. Crear métodos de solución que mejoren calidad y tiempo es valorado en esta competencia. Por lo que es importante desarrollar estos métodos y contribuir a la solución.

Una estrategia que puede ayudar a mejorar este problema es la combinación del método ab initio con el método basado en fragmentos para aumentar la calidad de las estructuras tridimensionales que se obtienen utilizando un solo método. En el CASP se ha demostrado que la combinación de métodos genera buenos resultados. Por lo que estos son también llamados métodos híbridos que tratan de mejorar calidad y tiempo de solución.

1.5 Organización de la tesis

La tesis esta organizada de la siguiente manera en el capítulo 2 se presenta el marco teórico en el cual se describe algunos conceptos para la comprensión del tema, y se muestran los trabajos más relevantes del estado del arte, en el capítulo 3 se muestra y explica la metodología utilizada del proyecto, se describen los algoritmos utilizados, posteriormente en el capítulo 4 se realiza un análisis de los resultados obtenidos de la experimentación y en el capítulo 5 se tienen las conclusiones.

Capítulo 2

2 Antecedentes/Marco Teórico

Esta sección se presentan conceptos para la compresión del tema del proyecto, se describen conceptos como estructuras de proteínas y doblado de proteínas.

2.1 Dogma central de la biología molecular.

El ácido desoxirribonucleico (ADN) es un ácido nucleico que contiene la información genética usadas en el desarrollo y funcionamiento de todos los organismos vivos y de algunos virus, es la molécula responsable también de la transmisión hereditaria [Encina, 2013].

El dogma central de la biología molecular es un concepto que muestra el proceso de transmisión de información del ADN al ARN (ácido ribonucleico) por medio de una transcripción y este a su vez por la traducción hacia las proteínas, esta información es información genética contenida en los genes del ADN, este proceso fue formulado por Francis Crick en 1970 (véase Figura 2.1.1). Posteriormente hubo modificaciones en la cual la información genética es bidireccional, en esta modificación la transcripción es inversa y pasa de ARN a ADN, proceso que es utilizado por retrovirus para multiplicarse en las células infectadas. El ADN puede también duplicarse, generando una nueva copia de ADN. Además, algunas proteínas son capaces de duplicarse y proliferar en ausencia de ADN, como en el caso de los priones, el cual es un agente infeccioso formado por una proteína denominada priónica, capaz de formar agregados moleculares anormales [Encina, 2013] [Crick, 1970].

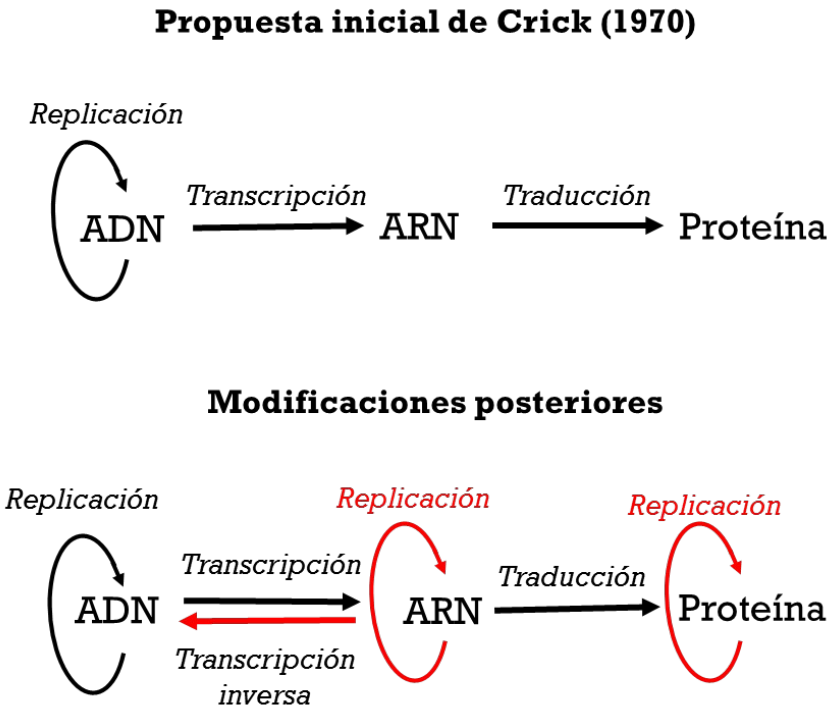


Figura 2.1.1 Dogma central de la biología molecular. Fuente: [Crick, 1970]

2.2 Estructura de ADN y ARN.

Oswald Avery, Colin MacLeod y Maclyn McCarty en 1944, demostraron que el ADN es la molécula que contiene la información genética. En los siguientes años se determinaron las estructuras de los nucleótidos, y en 1953, James D. Watson y Francis H. C. Crick propusieron su famoso modelo de la estructura de ADN de doble hélice o doble hebra [Horton, 2008].

Watson y Crick en 1953 propusieron un modelo de ADN que se basó en las estructuras conocidas de los nucleótidos, sobre figuras de difracción de rayos X que obtuvieron Rosalind Franklin y Maurice Wilkins de fibras de ADN. El modelo de Watson-Crick explicó las cantidades iguales de purinas y pirimidinas al plantear que el ADN tiene doble hebra (doble cadena) y que las bases en una hebra se apareaban en forma específica

con las bases de la otra: A (Adenina) con T (Timina) y G (Guanina) con C (Citosina) [Horton, 2008].

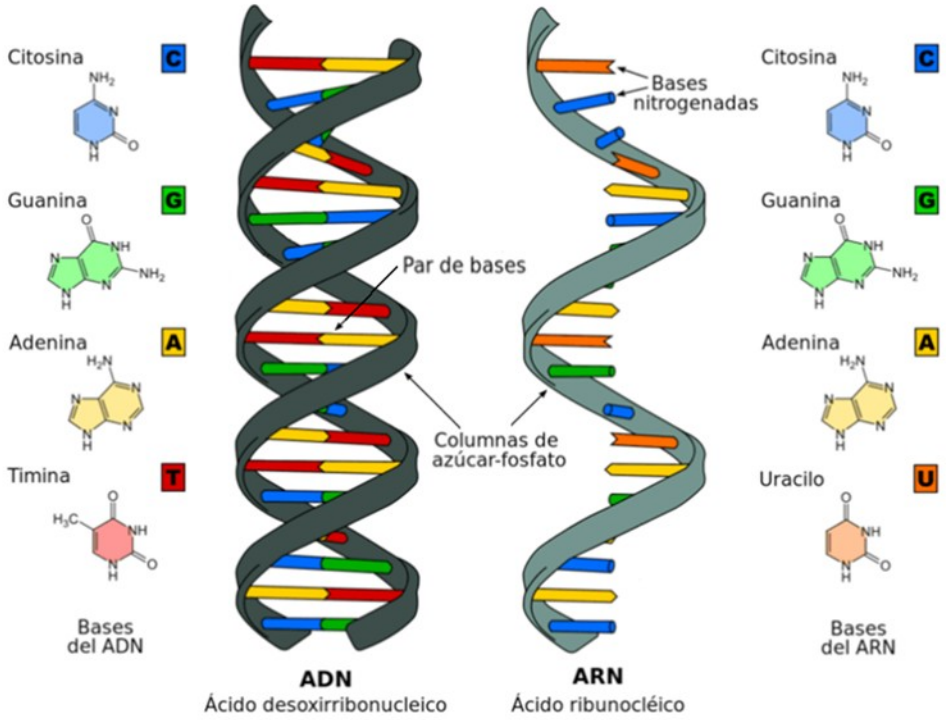


Figura 2.2.1 Estructura de ADN y ARN. Fuente: [Horton, 2008]

2.3 Estructuras de los aminoácidos.

Las proteínas son macromoléculas que están conformadas por cadenas de aminoácidos, los cuales están unidas por enlaces peptídicos. Los aminoácidos tienen una estructura conformada por un átomo de un carbono central (carbono alfa o alpha “Cα”) que se encuentra unido a un grupo amino (-NH2), un grupo carboxilo (-COOH), un átomo de hidrógeno (H) y un grupo residuo R (cadena lateral), el residuo hace la diferencia entre un aminoácido y otro, se muestra la estructura general de aminoácidos en la Figura 3. Existen 20 aminoácidos que aparecen en las proteínas (ver Tabla 2.3.1), son llamados aminoácidos estándar, los cuales se pueden clasificar en cuatro clases: apolares (no polares), polares, ácidos y básicos [Mckee & Mckee, 2014].

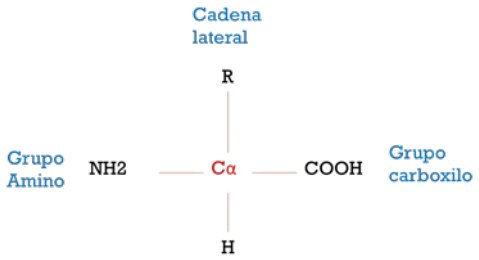


Figura 2.3.1 Estructura general de los aminoácidos. Fuente: [Mckee & Mckee, 2014]

Tabla 2.3.1 Nombre de los aminoácidos estándar y sus abreviaturas.

	Aminoácido	Abreviatura de tres letras	Abreviatura de una letra
NO POLARES	Alanina	Ala	A
	Cisteína	Cys	C
	Fenilalanina	Phe	F
	Glicina	Gly	G
	Isoleucina	Ile	I
	Leucina	Leu	L
	Metionina	Met	M
	Prolina	Pro	P
	Triptófano	Trp	W
	Valina	Val	V
POLARES	Asparagina	Asn	N
	Glutamina	Gln	Q
	Serina	Ser	S
	Tirosina	Tyr	Y
	Treonina	Thr	T
AC.	Ácido aspártico	Asp	D
	Ácido glutámico	Glu	E
BÁSICOS	Arginina	Arg	R
	Histidina	His	H
	Lisina	Lys	K

Los enlaces peptídicos se forman por la unión de dos o más aminoácidos, en un aminoácido el grupo amino (-NH2) reacciona con el grupo carboxilo (-COOH) de otro aminoácido, dentro de esta reacción se libera una molécula de agua, el nitrógeno del grupo amino queda enlazado con el carbono del grupo carboxilo, el resultado de la unión de estos dos aminoácidos se llama dipéptido, cuando se forma la unión de tres aminoácidos se denomina tripéptido, para cuatro recibe el nombre de tetra-péptido, y así pueden formarse enlaces mucho más grandes los cuales se denominan polipéptidos. En la Figura 2.3.2 se ilustra el enlace de aminoácidos donde se muestra la liberación de una molécula de agua [Mckee & Mckee , 2014].

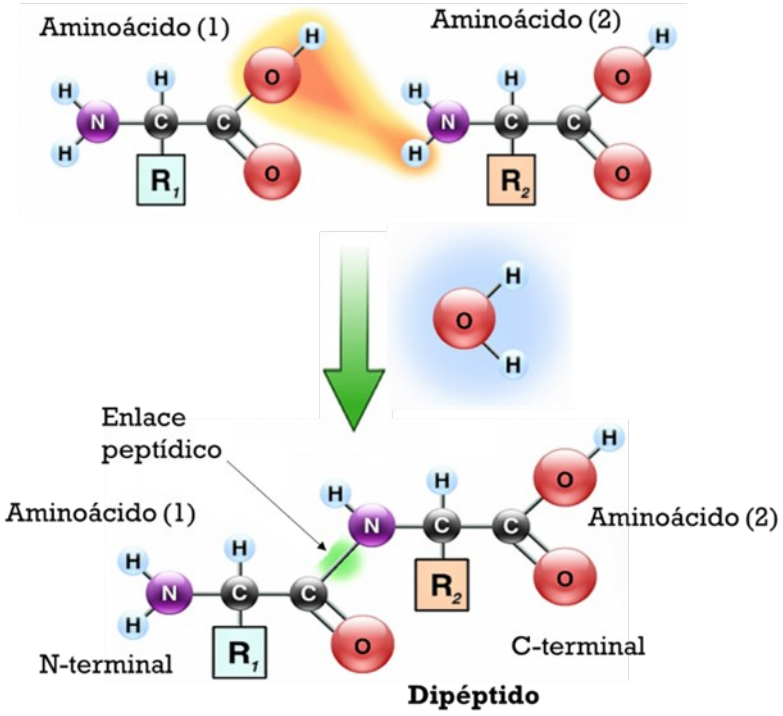


Figura 2.3.2 Enlace de aminoácidos (peptídico). Fuente: [Mckee & Mckee, 2009]

2.4 Proteínas escala de tiempo.

Las proteínas pueden pasar de una estructura primaria a otra en periodos de tiempo pequeños los cuales pueden ir desde los picosegundos ($1x10^{-12}$s) hasta los microsegundos ($1x10^{-6}$s), las primeras conformaciones de este proceso hay rotaciones no impedidas de cadenas laterales, posteriormente se van cerrando lazos por medio de interacciones de Van der Waals, en los microsegundos se van formando estructuras hélices α y láminas β (horquillas β) de las proteínas, aún en microsegundos hay formación del plegado o doblado de proteínas que obtienen su estructura tridimensional (véase Figura 2.4.1), pero existen proteínas que tienen un doblado más lento que se forman después de los milisegundos [Santos, 2009].

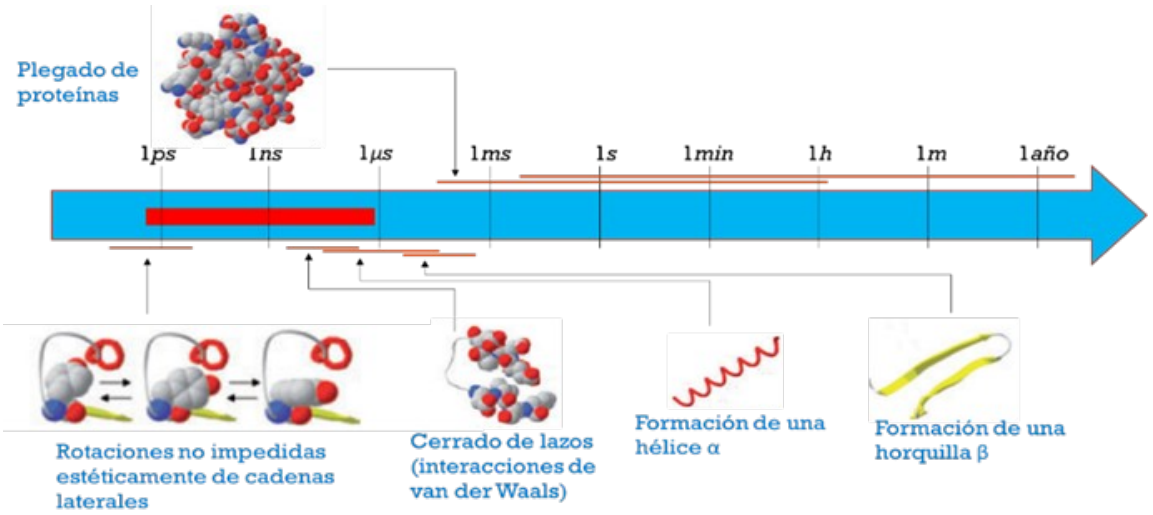


Figura 2.4.1 Escala de tiempo de las proteínas.
Fuente: [Santos, 2009]

2.5 Estructuras de las proteínas.

La estructura final de una proteína es tridimensional formada por enlaces péptidos. Sin embargo, una proteína tiene diferentes niveles de organización por su estructura clasificada como primaria, secundaria, terciaria y cuaternaria, las cuales tienen características diferentes de una a otra estructura. A continuación, se describen las diferentes estructuras [Mckee & Mckee, 2014].

2.5.1 Estructura primaria.

La estructura primaria de las proteínas esta formada por la secuencia lineal de aminoácidos, los cuales están unidos por enlaces peptídicos. El orden de esta secuencia de aminoácidos proviene de la información del material genético, cuando se traduce el ARN se obtiene el orden de estos aminoácidos los cuales forman la proteína, la estructura primaria es la secuencia de aminoácidos de la proteína Met-encefalina como se aprecia en la Figura 2.5.1.1 [Santos, 2009].

TYR – GLY – GLY - PHE - MET

Figura 2.5.1.1 Estructura primaria.
Fuente: https://www.rcsb.org/structure/2LWC.

2.5.2 Estructura secundaria.

La estructura secundaria de una proteína esta formada por puentes de hidrógeno entre los grupos carbonilo (N-H), estas estructuras secundarias son observadas mayormente en las hélices α y las láminas plegadas β, aunque existen otras formaciones de estructura secundaria estas son las de más frecuencia. La Figura 2.5.2.1 muestra hélices α las cuales son estructuras rígidas en forma de varilla que se genera cuando una cadena polipeptídica se enrolla en una conformación helicoidal dextrógira (forma de hélice)

Las láminas β se forman cuando dos o más segmentos de la cadena polipeptídica se alinean uno al lado de otro, cadena segmento se denomina cadena β, en lugar de enrollarse como las hélices α, las cadenas β se extienden como una lámina. Las láminas β forman puentes de hidrógeno (N-H) con el carbonilo de la estructura polipeptídica de cadenas adyacentes, estas laminas pueden ser paralelas o antiparalelas debido a la dirección de las cadenas, las cuales se muestran en la Figura 2.5.2.1 [Mckee & Mckee, 2014].

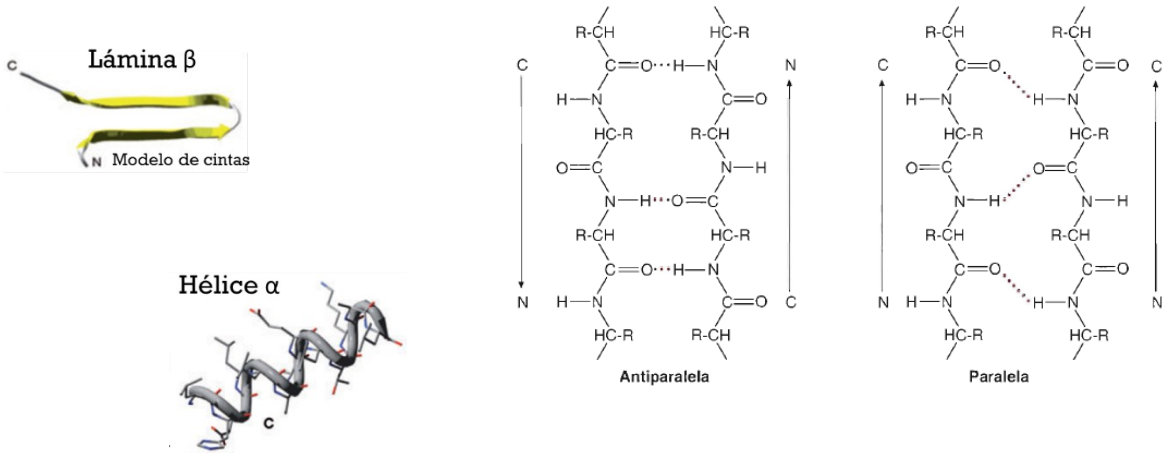


Figura 2.5.2.1 Estructura secundarias, hélice α y làmina β.
Fuente: [Mckee & Mckee, 2014]

2.5.3 Estructura terciaria.

La estructura terciaria se produce por el plegamiento de un polipéptido el cual contiene estructuras secundarias de hélice α y lamina β las cuales forman una estructura tridimensional (véase Figura 2.5.3.1). Estas estructuras pueden ser globulares, las estructuras globulares tienen forma esférica y una característica es que son solubles, que se pueden disolver al mezclar con líquidos, en su estructura terciaria las proteínas pueden ejercer sus funciones biológicas [Mckee & Mckee, 2014].

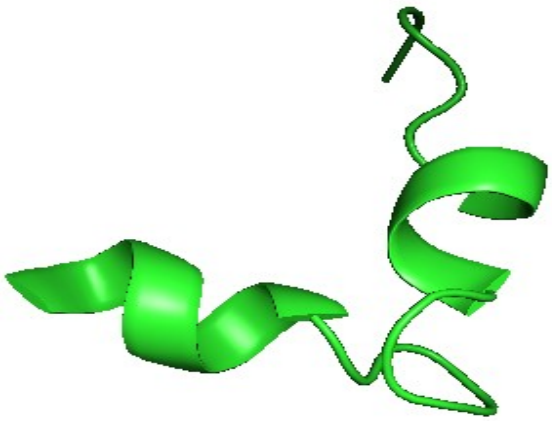


Figura 2.5.3.1 Estructura terciaria de una proteína.
Fuente: software SMMP [Eisenmeger, 2001]

2.5.4 Estructura cuaternaria.

La estructura cuaternaria esta formada por varias subunidades de cadenas polipeptídicas, las subunidades pueden ser idénticas o no serlo, ver Figura 2.5.4.1 Un ejemplo es la hemoglobina, compuesta por cuatro subunidades que unen oxígeno en forma cooperativa. Y otro es la frataxina en el que las subunidades que se unen son idénticas, en este caso son 24 subunidades [Santos, 2009].

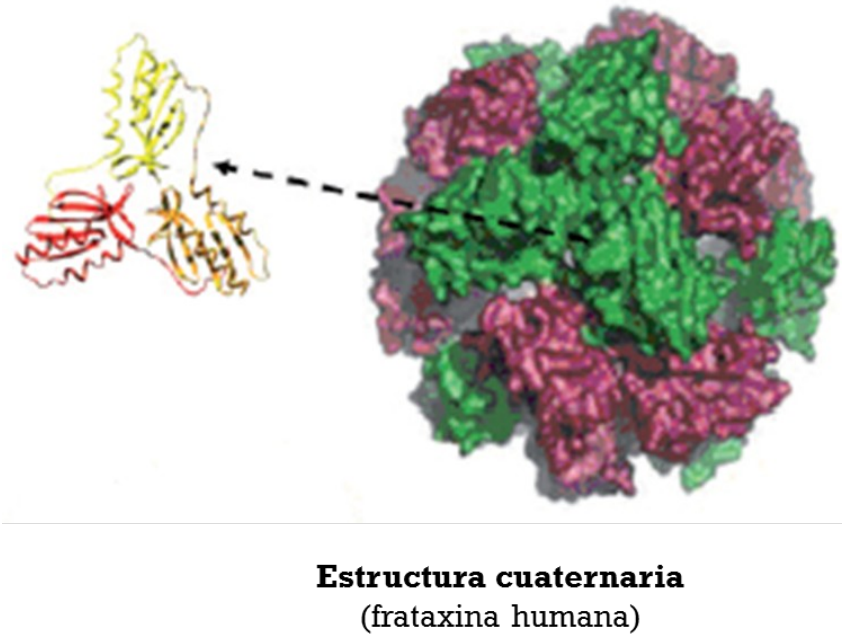


Figura 2.5.4.1 Estructura cuaternaria, frataxina humana. Fuente: [Santos, 2009]

2.6 Problema de doblado de Proteínas.

J. Kendrew en 1958 realizó análisis de la estructura de la molécula de mioglobina lo que permitió revisar una propiedad de estructuras terciarias, observó que el núcleo de la mioglobina estaba formado mayormente por aminoácidos hidrofóbicos que no interactúa con moléculas de agua, por lo que la superficie expuesta al solvente estaba formada por aminoácidos polares [Kendrew, 1958]. Esta particularidad, se observó posteriormente para un gran número de estructuras terciarias, permitió una idea sobre los procesos que utiliza la proteína para producir una estructura termodinámicamente estable. A partir de esta idea, se propuso que el proceso principal para el plegamiento de doblado de proteínas es de carácter físico y se debe a la interacción entre los aminoácidos hidrofóbicos e hidrofílicos (aminoácidos que captan el agua con facilidad) con las moléculas de agua, los aminoácidos hidrofóbicos se juntan en el interior de la molécula y los hidrofílicos ocupan la estructura exterior, los que se exponen a la interacción con las moléculas de solvente. Para esta unión interior los átomos de carbono alfa (o Cα) deben agruparse en el centro, para estabilizar la interacción entre la cadena de átomos de Cα con las moléculas de agua se generan puentes de hidrógeno lo que forman las estructuras secundarias y a partir de ellas se forman las estructuras terciarias. Estas observaciones las realizaron con base en la caracterización por difracción de rayos X sobre las moléculas globulares como la mioglobina (véase Figura 2.6.1) [Kendrew, 1958], [Olivarez Quiroz & García Colín, 2004].

El doblado de proteínas es el proceso en el que una proteína alcanza su estado tridimensional el cual es funcional biológicamente, o también nombrada estructura nativa que es la estructura más estable de todas las posibles. La información de la estructura primaria genera la estructura tridimensional. Una proteína doblada incorrectamente no cumpliría su función biológica, lo que puede ocasionar enfermedades como el Alzheimer, la enfermedad de Parkinson y otras.

Figura 2.6.1 Mioglobina en resolución de 6 Å (angstrom).
Fuente: [Kendrew, 1958].

2.6.1 Paradoja de Levinthal.

En 1968, C. Levinthal mostró que el problema de hallar la configuración terciaria de una proteína necesariamente implica un proceso evolutivo. Este investigador observó que para explorar todas las posibles configuraciones tridimensionales hasta encontrar la más estable podrían requerir un tiempo demasiado grande ya que este depende del número de posibles configuraciones (C) definido por la ecuación (2.1) para cualquier proteína [Levinthal, 1968]:

$$C = v^N \quad \text{-------(2.1)}$$

C = Posibles configuraciones tridimensionales.
v = Configuraciones espaciales.
N = Aminoácidos de una proteína.

De acuerdo a la ecuación 2.1, para mostrar lo grande que puede ser el número de posibles configuraciones, consideremos que se tienen solamente 3 configuraciones espaciales para una proteína con 100 aminoácidos. Al reemplazar estos valores en la ecuación 2.1, se obtienen $3^{100} = 5.15 \times 10^{47}$ posibles configuraciones. Si consideramos una

computadora paralela como la Blue gene/L de IBM con una velocidad para procesar 280 x 10^{12} de operaciones por segundo, el tiempo requerido para examinar este simple caso llevaría a un número gigantesco alrededor de 5.83 x 10^{25} años, comparando este valor con la edad aproximada del universo 13.7 x 10^{9} años, es mucho más grande el tiempo de procesamiento por la computadora. En consecuencia, no es posible utilizar un método determinístico que examine todas las posibles configuraciones.

2.6.2 Hipótesis de la termodinámica.

En 1973 realizando estudios sobre la renaturalización de la ribonucleasa, Anfinsen demostró que los enlaces disulfuros de la ribonucleasa A se forman de manera espontánea mediante una reacción de oxidación en presencia de aire, lo que lo llevo a declarar la llamada hipótesis de la termodinámica [Anfinsen, 1973]:

"La estructura nativa de una proteína en sus condiciones fisiológicas normales es la configuración más estable termodinámicamente hablando, pues su estructura es aquella en la cual la energía libre de Gibbs es la menor".

Las proteínas llevan un proceso para su doblado, el cual bajo la termodinámica que estudia las reacciones de energía, es la energía más pequeña del sistema denominada “energía libre de Gibbs” que se presenta en este proceso cuando la proteína alcanza su estructura nativa, la cual es la configuración más estable en la que sus propiedades fisiológicas normales están presentes (véase Figura 2.6.2.1). A continuación, se muestra la ecuación de la segunda ley de la termodinámica, en la que se mide la variación de energía libre de Gibbs, (véase ecuación 2.2) [Anfinsen, 1973].

$$\Delta G = \Delta H - T\Delta S \qquad \text{--(2.2)}$$

ΔG: Variación de la energía libre entre reactivos y productos. Se mide con el Potencial termodinámico que da las condiciones de equilibrio en la reacción del sistema.

ΔH: Variación de la entalpia libre entre reactivos y productos y, mide la variación de la energía calorífica en la reacción.

T: Temperatura.

ΔS: Variación de la entropía entre reactivos y productos. Mide el número de microestados compatibles con el macro-estado de equilibrio; Calcula el grado de organización del sistema.

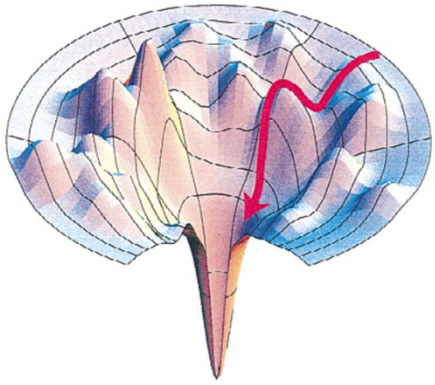


Figura 2.6.2.1 Gráfico genérico de la energía libre de una macromolécula.
Fuente: [Dill, 1999].

2.6.3 Complejidad del problema de doblado de proteínas.

El problema de doblado de proteínas es considerado como un gran desafío en diversas áreas de investigación. La estructura de la proteína contiene ángulos que conforman el esqueleto enlazado por los aminoácidos Psi (ψ), ángulo para el grupo carboxilo, Chi(χ) ángulo de la cadena lateral, Phi(ϕ) ángulo entre grupo amino, carbón α y Omega(ω) ángulo entre aminoácidos, estos son ángulos de torsión que se van repitiendo en las uniones de cada aminoácido (véase Figura 2.6.3.1). Determinar los ángulos de la estructura es un problema complejo ya que hay un gran número de combinaciones posibles [Ngo, 1994].

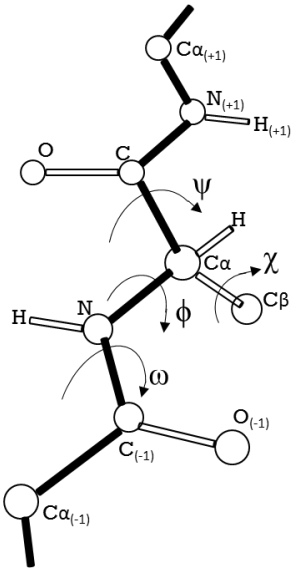


Figura 2.6.3.1 Configuración con sus ángulos diedros de una proteína.
Fuente: [Santos, 2009].

2.6.4 Función de energía en el problema de doblado de proteínas.

La estructura nativa de la proteína corresponde a la mínima energía, en la cual se alcanza un equilibrio térmico. Las fuerzas de cada átomo en la estructura están relacionadas con la energía, las cuales están determinadas por un campo de fuerza. Para analizar esto se estudian las fuerzas de Van der Waals que son fuerzas de estabilización molecular que se forman por un enlace no covalente que producen fuerzas de dispersión (atracción) y las fuerzas de repulsión entre las capas electrónicas de 2 átomos contiguos [Horton, 2008]. La ecuación 2.3 de Van der Waals es la siguiente:

$$\left(p+\frac{a'}{v^2}\right)(v-b')=kT \qquad \text{--(2.3)}$$

Donde:

p es la presión del fluido.

v es el volumen molar en el que se encuentran las partículas dividido por el número de partículas (litro/mol).

k es la constante de Boltzmann.

T es la temperatura en kelvin.

a’ es la atracción de las partículas.

b’ es el volumen excluido de v por cada partícula.

La ecuación 2.3 es una ecuación de estado de un fluido compuesto de partículas que involucra fuerzas intramoleculares de Van der Waals, esta ecuación sirvió para precisar el comportamiento de los gases reales considerando su tamaño no nulo y atracción entre partículas.

2.7 Estrategias computacionales para el doblado de proteínas.

La comunidad científica ha definido varias estrategias computacionales (métodos) para abordar el problema de doblado de proteínas, estas estrategias se pueden clasificar en los siguientes grupos [Dorn, 2014].

1. Método por homología.
2. Método Threading.
3. Método ab initio.
4. Método basado en fragmentos.

2.7.1 Método por homología.

El método predictivo de modelado por homología o comparativa permite predecir la estructura terciaria de una proteína objetivo conociendo solo su secuencia de aminoácidos y la estructura terciaria resuelta experimentalmente. Este método inicia con la identificación de proteínas homológas (plantillas) dada una secuencia de aminoácidos, las cuales son buscadas en el banco de datos de proteínas PDB (Protein Data Bank) que contiene plantillas de estructuras tridimensionales de proteínas. Posterior a esta identificación se lleva a cabo un alineamiento entre la secuencia objetivo y estas plantillas homológas, enseguida se debe realizar una optimización del modelo para llegar a la estructura tridimensional final de la proteína, se muestra este método en la Figura 2.7.1.1 [Maldonado & Frausto, 2018].

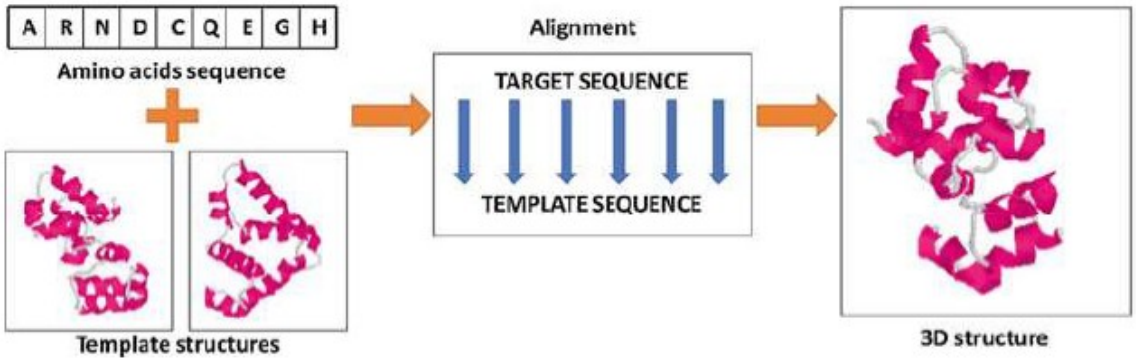


Figura 2.7.1.1 Modelado por homología.
Fuente: [Maldonado, 2018].

2.7.2 Método Threading.

El método predictivo de modelado por Threading busca predecir la estructura terciaria de una proteína objetivo conociendo su secuencia de aminoácidos y una familia de secuencias que comparten una estructura terciaria particular, eligiendo la plantilla con mayor relación. Este método consiste en dos etapas, se tiene una secuencia de aminoácidos y se selecciona una plantilla estructural de una librería o PDB. Posteriormente la plantilla con mayor relación a la estructura del modelo de la proteína se elige para reemplazar los espacios de la estructura (ver Figura 2.7.2.1) [Maldonado & Frausto, 2018].

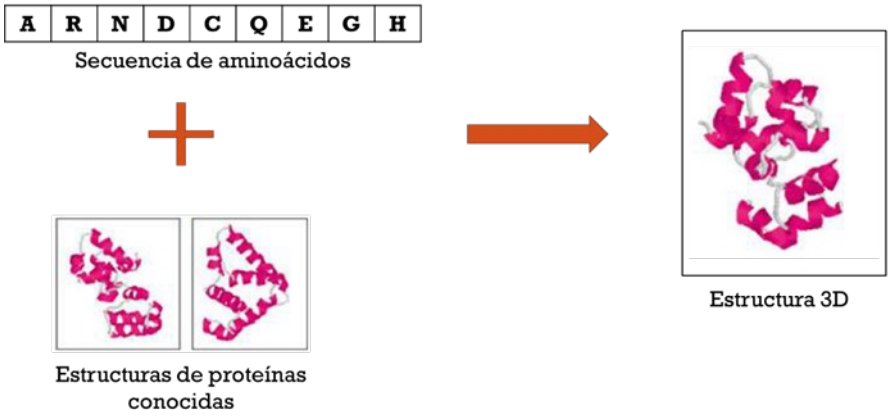


Figura 2.7.2.1 Método Threading.
Fuente: [Maldonado & Frausto, 2018].

2.7.3 Método ab initio.

La técnica predictiva de modelado por el método ab initio busca predecir la estructura terciaria de una proteína objetivo conociendo solo su secuencia de aminoácidos y no requiere ninguna otra información adicional (véase Figura 2.7.3.1). Este método se considera como un problema de optimización, donde se identifican los valores de las variables (ángulos), los cuales describen la mínima energía de la proteína. Se usan tres componentes para esta estrategia: una representación geométrica, una función de energía, y una búsqueda de técnica de optimización [Maldonado & Frausto, 2018].

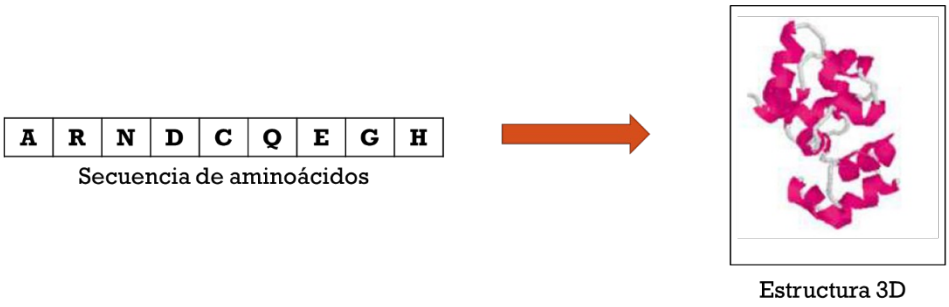


Figura 2.7.3.1 Método ab initio.
Fuente: [Maldonado & Frausto, 2018].

2.7.4 Método basado en fragmentos.

El método basado en fragmentos compara fragmentos cortos, es decir una sub-secuencia corta de aminoácidos de un fragmento objetivo contra fragmentos de estructuras de proteínas conocidas, entonces estos fragmentos pueden ser usados para construir el modelo estructural 3-D de la proteína [Dorn, 2014]. Este método primero, divide la secuencia objetivo en fragmentos, posteriormente lleva a cabo la búsqueda de secuencias similares de cada fragmento, en una base de datos de estructuras conocidas, luego clasifica los fragmentos, después construye la estructura tridimensional a partir de la plantilla de fragmentos utilizando una técnica de combinación y por último hace un proceso de refinamiento y validación para presentar la estructura final de la proteína, en la Figura 2.7.4.1 se muestra este proceso.

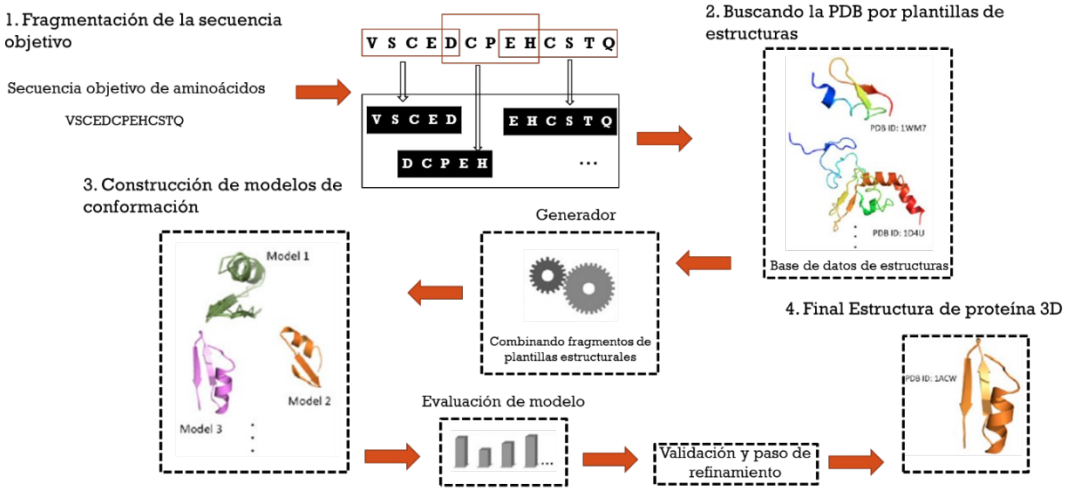


Figura 2.7.4.1 Método basado en Fragmentos. Fuente: [Dorn, 2014]

2.8 Estado del arte.

Se muestran a continuación los principales proyectos que se han realizado a nivel internacional dentro de la competencia CASP en la cual se presentan estrategias computacionales para resolver el problema de doblado de proteínas. Gran parte de estos proyectos están basado en los métodos de homología, Threading, ab initio, basado en fragmentos.

2.8.1 Aplicación de estrategias computacionales.

En la competencia CASP se presentan investigaciones de varias partes del mundo, en las cuales se emplean diferentes estrategias para la predicción de las estructuras terciarias de proteínas. Esta competencia se realiza cada dos años desde el primer CASP en 1994, se toma en cuenta la calidad de las estructuras que predicen los participantes con sus estrategias propuestas. Se valoran las investigaciones que obtienen mejor calidad. A continuación, se describen algunos métodos sobresalientes de estas competencias. En dichos proyectos de los investigadores se denotan con el nombre de sus servidores que utilizan para ejecutar sus estrategias propuestas (ver Tabla 2.8.1.1)

Tabla 2.8.1.1 Estrategias más destacadas de los diferentes CASP.

Servidor	Estrategia	Competencia
AlphaFold2	Utiliza aprendizaje profundo y redes convolucionales con estrategia de fragmentos. Este trabajo es una mejora de AlphaFold [Senior, 2020]	CASP-14
BAKER	Utiliza su algoritmo Rosetta mediante predicciones de contactos basados en la co-evolución. [Ovchinnikov, 2018]	CASP-14
A7D	Utiliza aprendizaje profundo y redes convolucionales con estrategia de fragmentos.[Senior, 2020].	CASP-13
Zhang	Utiliza los algoritmos de I-TASSER y QUARK con modelado basado plantillas. [Zhang, 2018]	CASP-13
Raptor X	Utiliza Homología. Como mejora desarrollan un algoritmo de redes neuronales para lidiar con proteínas para las cuales la cantidad de proteínas homólogas es menor. [Wang, 2017]	CASP-12
QUARK	En este proyecto se utiliza Threading. Se implementa el método Monte Carlo y ensamble de fragmentos, (CASP10) [Zhang, 2009], (CASP11) [Zheng, 2009; Zhang, 2016].	CASP-10 Y 11
I-TASSER/Zhang-server	Este proyecto utiliza la técnica Threading. Ensamble de fragmentos y refinamiento. [Zhang, 2009]	CASP-8
BAKER-ROSETTA	Utiliza la técnica de Fragmentos de proteínas y fragmentos de la PDB, técnica de ensamble de Rosetta. Utiliza Recocido Simulado y Modelos de Markov [Raman, 2009]	CASP-8
pro-sp3-TASSER	Se emplea la técnica Threading, Ensamble de fragmentos y refinamiento [Zhou, 2009]	CASP-8

2.8.2 Métodos basados fragmentos.

En los métodos basados en fragmentos y combinación de estrategias para la predicción de fragmentos un trabajo relevante es [Oliveira, 2017], donde utilizan un método híbrido para predecir estructuras terciarias de proteínas. Teniendo una estructura de proteínas de novo realizan un muestreo aleatorio del espacio conformacional para identificar la energía mínima, el muestreo se la realizan por una búsqueda Monte-Carlo en la cual se hace un criterio de aceptación, se generan estructuras secundarias y predicen ángulos de torsión para realizar la unión de estas, se extraen fragmentos de estructuras de una librería (método basado en fragmentos) para construir estructuras terciarias objetivo y seleccionar la de mejor resultado.

Como se observa en los algoritmos que se combinan dos métodos lo que da una hibridación de ellos, han obtenido resultados de mejor calidad.

En la Tabla 2.8.1.2 se muestra los diez mejores servidores del CASP14 donde los resultados más sobresalientes han sido obtenidos por el AlphaFold2, y se muestran otros

servidores como BAKER, Zhang quienes han participado en CASP previos obteniendo buenos resultados, estos servidores se basan en el método de ensamble de fragmentos. Por lo tanto en el CASP14 los mejores servidores de la competencia están basados en el método basado en fragmentos.

Tabla 2.8.1.2 Clasificación de los mejores proyectos del CASP14.

Servidor
AlphaFold2
BAKER
BAKER-experimental
FEIG-R2
Zhang
tFold_human
MULTICOM
QUARK
Zhang-Server
tFold-IDT_human

Fuente: https://predictioncenter.org/casp14/zscores_final.cgi

Sin embargo, incluso con los métodos del estado del arte, no ha sido posible obtener la estructura nativa para proteínas o péptidos. Por ello, aún en la actualidad, estos métodos siguen mejorando sus estrategias de predicción. Las estrategias que han tenido resultados destacados utilizan técnicas de aprendizaje profundo como Alphafold [Mirdita, 2022]

Capítulo 3

3 Metodología

En este capítulo se presenta la metodología general de solución propuesta que se realizará para alcanzar los objetivos del proyecto. Además, en base a esta metodología general se han realizado experimentaciones previas durante su desarrollo, las cuales también son descritas en este capítulo.

La Metodología general para la predicción de estructuras tridimensional se puede observar en la Figura 3.1, la cual contiene 5 etapas y se describen cada una de ellas. Cada etapa de la metodología tiene un ejemplo de la instancia y como se va desarrollando durante el proceso hasta obtener la predicción final de la estructura tridimensional lo que es el resultado o salida del método.

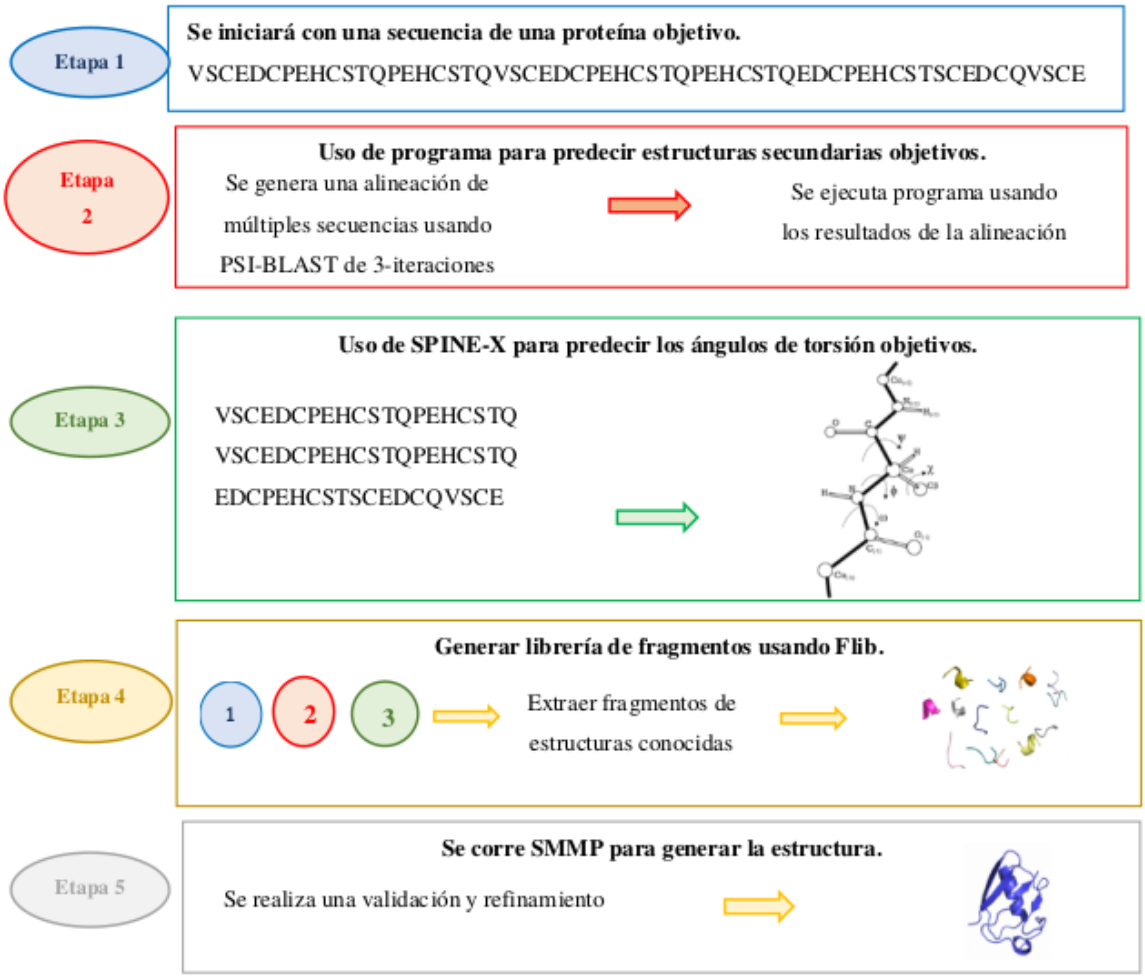


Figura 3.1 Etapas del método.

En la primera etapa se lee una secuencia de aminoácidos que puede ser descargada de la PDB, la cual es la secuencia objetivo de la proteína que se desea predecir su estructura terciaria.

En la segunda etapa se generan estructuras secundarias partiendo de la información de la secuencia de aminoácidos usando software PSIPRED (PSIBlast-PREDiction).

En la tercera etapa se identifican los ángulos de torsión objetivo para generar una plantilla inicial y es la base para la selección de estructuras cortas que se unirán para formar la estructura tridimensional.

En la cuarta etapa se extraen los fragmentos mediante la librería Flib (Fragment library) que crea fragmentos a partir de la estructura secundaria y ángulos de torsión generados en la plantilla inicial con los cuales se realiza el armado de la estructura final.

En la quinta etapa se construye la estructura tridimensional en la cual se realiza una validación y un refinamiento por algoritmos de optimización híbridos del recocido simulado (HSA, siglas en ingles), para obtener una solución la energía de Gibbs más baja.

Una vez obtenida la estructura tridimensional se compara contra la estructura nativa mediante métricas de RMSD (Root Mean Square Deviation), TM-score [Zhang, 2004] y GDT-TS [Zemla, 2021]. El RMSD es la medida de la distancia promedio entre los átomos de dos proteínas superpuestas. El TM-score también es una medida de similaridad entre dos estructuras superpuestas. Y el Global Distance Total – Total score se utiliza para cuantificar la similitud entre una estructura de proteína predicha y una estructura de referencia.

Durante el desarrollo de la metodología general de solución propuesta se han realizado experimentaciones en las cuales se consideran de acuerdo al método algunas etapas de la metodología general de solución:

1) Método con algoritmos HSA, usando solamente la secuencia de aminoácidos (etapas 1 & 5). [Morales, 1991; Frausto, 2007&2015; Maldonado, 2016; Frausto, 2019]
2) Método con algoritmos HSA, utilizando la secuencia de aminoácidos y la predicción de estructura secundaria (etapas 1, 2, 3 & 5) [Sánchez, 2021].

3) Método con algoritmo GRSA2, la secuencia de aminoácidos, predicción de estructura secundarias y aplicación de estrategias de selección de ruleta para el refinamiento de las regiones de las cadenas laterales de aminoácidos [Soto, 2021].
4) Método con algoritmo GRSA2 usando fragmentos, mediante la predicción de fragmentos en una red neuronal convolucional, ensamble de fragmentos y refinamiento.

Estos 4 métodos han sido evaluados mediante experimentaciones, las cuales se han aplicado a un conjunto de péptidos (instancias) y sus resultados han sido evaluados con métricas que miden la calidad de la estructura terciaria como el RMSD, TM-score y GDT-TS [Zemla, 2001]. Además, se realiza un análisis de resultados para ver el desempeño de los 4 métodos con sus respectivos conjunto de instancias, las cuales son péptidos o también llamadas proteínas pequeñas, estas instancias han sido procesadas en los métodos, programas o también denominados algoritmos del estado del arte para poder comparar su desempeño con nuestros métodos.

3.1 Algoritmos HSA.

Los algoritmos de optimización son utilizados para obtener y refinar la estructura terciaria de una proteína o péptido como el Simulated Annealing (SA) [Frausto, 2007], Golden Ratio Simulated Annealing (GRSA) [Frausto, 2015], Evolutionay GRSA (EGRSA) [Maldonado, 2016] y GRSA2 [Frausto, 2019]. Los algoritmos implementados SA, GRSA, EGRSA y GRSA2 son métodos híbridos del recocido simulado y se han denominado como Hybrid Simulated Annealing (HSA) para la experimentación de esta investigación.

El recocido simulado fue propuesto por Kirkpatrick [Kirkpatrick, 1983] y Cerny [Cerny, 1985] el cual esta inspirado en el proceso de recocido físico de los metales. Este algoritmo ha sido aplicado a problemas NP-hard como en el PFP [Frausto, 2007]. Tiene dos ciclos principales los cuales son el el ciclo de temperatura (esquema de enfriamiento) y el ciclo de Metrópolis como se ve en el pseudocódigo de SA en el algoritmo 1. En el ciclo de temperatura se tienen los parámetros T_i, T_f que son las temperaturas inicial con un valor alto (caliente) y final valor bajo de temperatura (frío) respectivamente, y el parámetro α (valor menor a 1) es el factor de enfriamiento que produce el decremento de la temperatura.

En el ciclo de Metrópolis se realiza una perturbación a partir de una solución aleatoria inicial que generara los ángulos de torsión de la proteína a predecir, este ciclo itera explorando nuevas soluciones vecinas y acepta una nueva solución si es mejor, y si no, aceptarlo determinado por un criterio de aceptación basado en la distribución de Boltzmann (línea 11-14). El algoritmo SA termina cuando la temperatura final (T_f) es alcanzada. Para determinar los valores de Temperatura inicial y final se ha realizado una sintonización analítica con lo cual para cada proteína se obtienen esos valores [Frausto, 2007].

Algorithm 1 SA algorithm Procedure

```
1: Data: T_i, T_f, α
2: T_k = T_i; α = 0.95
3: S_i = generateSolution()
4:     while T_k ≥ T_f do  //Temperature cycle
5:         while Metropolis length do  //Metropolis cycle
6:             S_j=perturbation(S_i)
7:             ΔE=Energy(S_j) – Energy(S_i)
8:             if ΔE ≤ 0 then
9:                 S_i=S_j
10:                E = Energy(S_i)
11:            else if e^(-ΔE/Ti) < random [0-1] then
12:                S_i=S_j
13:                E = Energy(S_i)
14:            end if
15:        end while  //End Metropolis cycle
16:        T_k = T_k*α
17:    end while  //End Temperature cycle
18: end Procedure
```

Debido a que el espacio de soluciones es muy amplio, el algoritmo SA toma un tiempo grande para obtener soluciones optimas. Por lo que se han utilizado técnicas para mejorar el tiempo de búsqueda. El algoritmo de GRSA (véase algoritmo 2) utiliza un número áureo (dorado) [Frausto, 2019] con el cual se modifica el esquema de enfriamiento y con esto se reduce el costo de exploración o búsqueda de soluciones del recocido simulado. Este esquema utiliza el número áureo (ϕ) con el que se realizan cortes al ciclo de temperatura lo que genera secciones y para cada sección se utiliza un parámetro α que incrementa su valor (en δ=0.05) y va del rango de 0.70 a 0.95, por lo que con un α de 0.70 se realiza un decremento rápido de temperatura y al cambiar de sección el α aumenta (se actualiza) por lo que en la siguiente sección el decremento es más lento que la sección previa (línea 17), se utilizan 5 secciones por lo que la última sección el decremento de temperatura es el más lento. Por lo tanto en el ciclo de temperatura se realiza un descenso

rápido en temperatura altas lo que realiza una exploración de soluciones más rápida hasta llegar a temperaturas bajas donde el descenso es más lento (exploración de soluciones más lento) y la aceptación de soluciones es más restrictiva por el criterio de aceptación. En este algoritmo se utiliza un criterio de paro de mínimos cuadrados entre las líneas 22 y 27 del pseudocódigo.

Algorithm 2 GRSA algorithm Procedure

```
1: Data: T_i, T_fp, T_f, E, S, α, ϕ, δ
2: α=0.70; ϕ=0.618; δ = 0.05
3: T_fp = T_i; T_k = T_i; E = 0
4: S_i=generateSolution()
5:     while T_k ≥ T_f do //Temperature cycle
6:         while Metropolis length do //Metropolis cycle
7:             S_j=perturbation(S_i)
8:             ΔE=Energy(S_j) – Energy(S_i)
9:             if ΔE ≤ 0 then
10:                S_i=S_j
11:                E = Energy(S_j)
12:            else if e^(-ΔE/Ti) < random [0-1] then
13:                S_i=S_j
14:                E = Energy(S_j)
15             end if
16:        end while //End Metropolis cycle
17:        T_fp = T_fp*ϕ //Golden ratio section (five cuts recommended)
18:        if T_k ≤ T_fp then                          } Update cooling speed
19:            α_new=α + δ
20:            T_k = α_new * T_k
21:        else
22:            T_k = T_k*α
23:        end if
24:        if T_k ≤ T_fpn then                         } Stop criterion
25:            m= Equilibrium(E)
26:            if m ≈ ε then
27:                T_K = T_f
28:            end if
29:        end if
30:    end while //End Temperature cycle
31: end Procedure
```

Algorithm 3 Equilibrium Function

```
1: Equilibrium(E)
2: i = 1; CE=i*E; Kmax=5; SumE=E; m=0
3:     if i < Kmax then
4:         CE=CE+i*E
5:         SumE=SumE+E
6:         i = i+1
7:     end if
8:     if i==Kmax then
9:         m =  (((12 * CE)-(6*(i-1)*SumE))/(i³ - i))
10:    end if
11:    return m
12: end Function
```

En el algoritmo EGRSA (ver algoritmo 4) se utiliza un algoritmo evolutivo [Maldonado, 2016] en el cual se realiza una perturbación mediante cruza y operadores de mutación a una población conformada por individuos (posibles soluciones) en la que se van generando nuevos individuos para obtener una nueva población con mejores soluciones (véase algoritmo 5). Este algoritmo utiliza un esquema de enfriamiento y un criterio de paro similar al algoritmo GRSA, la principal diferencia es la perturbación que es por medio un algoritmo evolutivo.

Algorithm 4 EGRSA algorithm Procedure

```
1: Data: T_i, T_fp, T_f, E, S, α, ϕ
2: α=0.70; ϕ=0.618; δ = 0.05
3: T_fp = T_i; T_k = T_i; E = 0
4: S_i=generateSolution()
5:     while T_k ≥ T_f do  //Temperature cycle
6:         while Metropolis length do  //Metropolis cycle
7:             S_j = EGRSApert(S_i)
8:             ΔE = Energy(S_j) – Energy(S_i)
9:             if ΔE ≤ 0 then
10:                S_i = S_j
11:                E =Energy(S_i )
12:            else if e^(-ΔE/Ti) < random [0-1] then
13:                S_i = S_j
14:                E =Energy(S_i )
15:            end if
16:        end while  //End Metropolis cycle
17:        T_fp = T_fp*ϕ  //Golden ratio section (five cuts recommended)
18:        if  T_k ≤ T_fp then
19:            α_new=α + δ
20:            T_k = α_new * T_k        } Update cooling speed
21:        else
22:            T_k = T_k*α
23:        end if
24:        if T_k ≤ T_fpn then
25:            m= Equilibrium(E)
26:            if m ≈ ε then
27:                T_K = T_f            } Stop criterion
28:            end if
29:        end if
30:    end while  //End Temperature cycle
31: end Procedure
```

Algorithm 5 EGRSApert Function

```
1:EGRSApert(S_i)
2: n= numGen, bestSol[ ], bestEnergy
3: pob=initialPob()
4:     while gen ≤ n do
5:         population = tournament()
6:         population = crossPopulation()
7:         population =  mutatePopulation()
8:     end while
9:     pop* = bestIndividual()
10:    return(bestSol[ ], bestEnergy)
11: end Function
```

El algoritmo GRSA2 [Frausto, 2019] (véase algoritmo 6) es una modificación de GRSA con una hibridación de Chemical Reaction Optimization (CRO) [Lam, 2012] aplicada en la perturbación del algoritmo (ver algoritmo 7), CRO se basa en el proceso químico donde se simulan la reacción química entre dos o más sustancias, en el algoritmo se generan colisiones (perturbaciones), de descomposición o colisión suave, las cuales genera nuevas soluciones que si son mejores que las previas se queda con las mejores, repitiendo este ciclo hasta que termina el esquema de enfriamiento. Este algoritmo utiliza un esquema de enfriamiento y criterio de paro similar al algoritmo GRSA.

Algorithm 6 GRSA2 algorithm Procedure

```
1: Data: T_i, T_fp, T_f, KE, E, S, α, ϕ
2: α=0.70; ϕ=0.618; δ = 0.05
3: KE=0;T_fp = T_i; T_k = T_i; E = 0
4: S_i=generateSolution()
5:     while T_k ≥ T_f do  //Temperature cycle
6:         while Metropolis length do  //Metropolis cycle
7:             Eold=Energy(S_i)
8:             S_j = GRSA2pert(S_i)
9:             EP = Energy(S_j)
10:            if (EP ≤ Eold + KE) then
11:                S_i = S_j
12:                E=Energy(S_i )
13:                KE = ((Eold+KE)-EP)*random[0,1]
14:            end if
15:        end while  //End Metropolis cycle
16:        T_fp = T_fp*ϕ  //Golden ratio section (five cuts recommended)
17:        if  T_k ≤ T_fp then
18:            α_new=α + δ
19:            T_k = α_new* T_k        } Update cooling speed
20:        else
21:            T_k = T_k*α
22:        end if
23:        if T_k ≤ T_fpn then
24:            m= Equilibrium(E)
25:            if m ≈ ε then
26:                T_K = T_f           } Stop criterion
27:            end if
28:        end if
29:    end while  //End Temperature cycle
30: end Procedure
```

Algorithm 7 GRSA2pert Function

```
1: GRSA2pert(S_i)
2: moleColl, b
3:     if b > moleColl then
4:         Randomly select one particle Mω
5:         if Decompositioncriterionmet
6:             S_j = Decomposition(S_j)
7:         else if
8:             S_j = SoftCollition(S_j)
9:         end if
10:    end if
11:    return S_j
12: end Function
```

3.2 Método con algoritmos HSA y solo la secuencia de aminoácidos.

Para este método con algoritmos HSA se ha realizado la experimentación con las etapas 1 y 5 para predecir la estructura terciaria, empleando los algoritmos SA, GRSA, EGRSA y GRSA2 donde a partir de la secuencia de aminoácidos se realiza un proceso de refinamiento para obtener la estructura terciaria o tridimensional para una proteína o péptido [Frausto, 2007], [Frausto, 2015], [Maldonado, 2016] y [Frausto, 2019] .

Las instancias de pruebas elegidas son un conjunto de 45. El número de aminoácidos es de 9 a 50 residuos como se muestra en la Tabla 3.1 donde se aprecia el nombre los péptidos de cada instancia con su número de aminoácidos o residuos, y la cantidad de variables (ángulos diedros) entre mayor es el número de residuos la instancia es

más grande es el tamaño del péptido y también por la cantidad de variables que contiene cada una.

Tabla 3.1 Instancias de prueba

Number	Instance (PDB code)	Amino acids	Number of variables
1	1egs	9	49
2	1uao	10	47
3	1l3q	12	62
4	2evq	12	66
5	1le1	12	69
6	1in3	12	74
7	1eg4	13	61
8	1rnu	13	81
9	1lcx	13	81
10	3bu3	14	74
11	1gjf	14	79
12	1k43	14	84
13	1a13	14	85
14	1dep	15	94
15	2bta	15	100
16	1nkf	16	86
17	1le3	16	91
18	1pgbF	16	93
19	1niz	16	97
20	1e0q	17	109
21	1wbr	17	120
22	1rpv	17	124
23	1b03	18	109

Number	Instance (PDB code)	Amino-acids	Instance (PDB code)
24	1pef	18	124
25	1l2y	20	100
26	1du1	20	134
27	1pei	22	143
28	1wz4	23	123
29	1yyb	27	160
30	1by0	27	193
31	1t0c	31	163
32	2bn6	33	200
33	1wr4	36	206
34	1yiu	37	206
35	1bhi	38	216
36	1i6c	39	218
37	1bwx	39	242
38	2ysh	40	213
39	1wr7	41	222
40	2dmv	43	229
41	2p81	44	295
42	1f4i	45	276
43	1dv0	47	279
44	1pgy	47	304
45	1ify	49	290
-	-	-	-

Para la ejecución de los algoritmos se utilizó el clúster Ehécatl del Instituto Tecnológico de Ciudad Madero, con Procesador Intel® Xeon® a 2.30 GHz, Memoria: 64gb (4x16gb) ddr4-2133, sistema operativo Linux CentOS. En la implementación se utiliza el SMMP (Simple Molecular Mechanics for Proteins) en lenguaje Fortran.

3.3 Método con algoritmos HSA y la predicción estructuras secundarias.

Las estructuras de las proteínas se van formando a partir de la estructura primaria pasando por una estructura secundaria y posteriormente a la estructura terciaria, por lo que tenemos tres estructuras de las cuales se tiene información para poder predecir una estructura a través de la otra. En esta experimentación se ha agregado una predicción de la

estructura secundaria basada en la estructura primaria (secuencia de aminoácidos). Las predicciones de estructuras secundarias son empleadas en métodos que utilizan fragmentos para obtener cortes de estructuras conocidas que coincidan con la información de la estructura secundaria predicha. Las estructuras secundarias son clasificadas principalmente en hélices y betas con las cuales se componen la estructura terciaria de una proteína o péptido. Las etapas utilizadas para esta experimentación son 1, 2, 3 y 5 de la metodología de la solución propuesta y se ha anexado una etapa de construcción. Esta metodología ha sido presentada en [Sanchez, 2021], donde se presenta la metodología GRSA-SSP para la predicción de péptidos (Figura 3.2), el termino GRSA refiere a la familia de algoritmos GRSA que pertenecen a los algoritmos HSA y el termino SSP se refiere a la predicción de estructura secundaria Secondary Structure Prediction (SSP). Para la experimentación se ha utilizado el mismo conjunto de instancias (45) de la tabla 3.1 y el equipo clúster Ehécatl del Instituto Tecnológico de Ciudad Madero para la ejecución de los algoritmos.

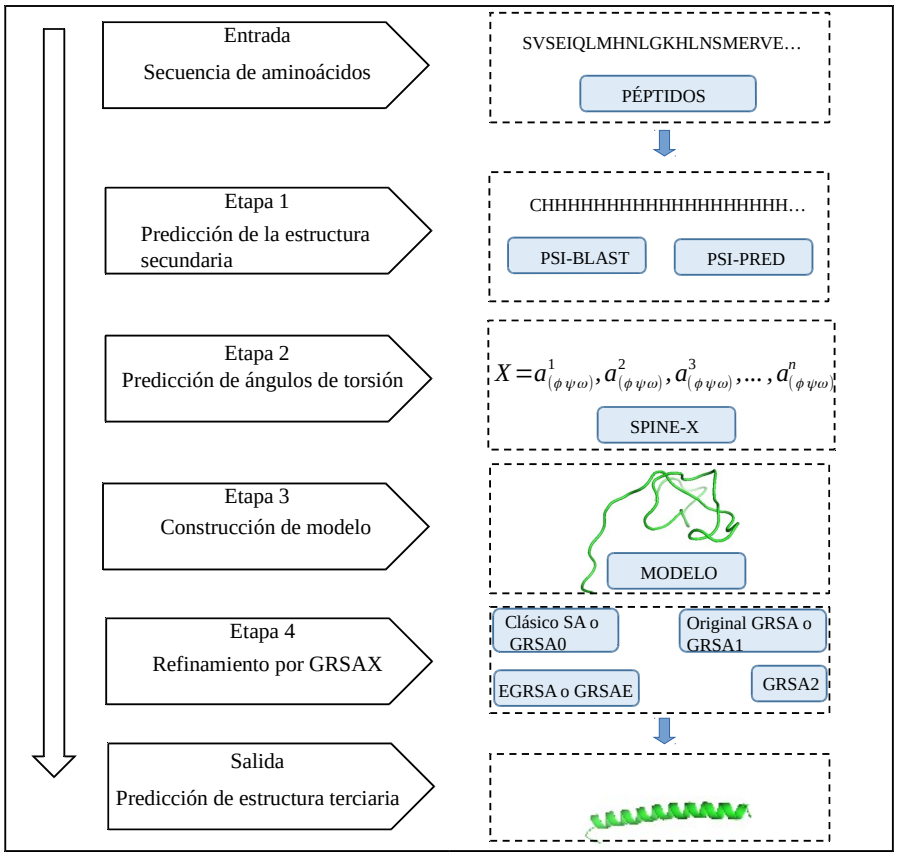


Figura 3.2 Metodología GRSA-SSP .

La entrada es la secuencia de aminoácidos la cual es la estructura primaria de una proteína o péptido. La etapa 1 es la predicción de la estructura secundaria la cual corresponde a la secuencia de aminoácidos, para la predicción de la estructura secundaria se utiliza PSI-PRED [Jones, 1999], el cual genera un archivo de secuencia con PSI-BLAST [Altschul, 1997] y desarrolla estados hélice (H), beta (E) y coil (C) para clasificar cada estructura secundaria; y a partir de estos estado PSI-PRED calcula la probabilidad de que estado tiene cada aminoácido de la estructura primaria. Para la etapa 2 se realiza la predicción de los ángulos de torsión mediante SPINE-X [Faraggi, 2017], con las predicciones de los estados en la etapa previa se obtienen los ángulos de torsión (ϕ, Ψ, and ω) para cada aminoácido, esta predicción de ángulos se realiza mediante redes neuronales artificiales. En la etapa 3 una vez obtenidos los ángulos de torsión o variables son utilizados para construir una plantilla inicial $S_i=[\phi_1, \Psi_1, X_1, \omega_1, \phi_2, \Psi_2, X_2, \omega_2, \ldots, \phi_n, \Psi_n, X_n, \omega_n]$ el cual representa el vector de solución inicial que está constituido por los ángulos torsionales de cada aminoácido donde cada subíndice representa cada aminoácido con su respectivo ángulo. En la etapa 4 se realiza el refinamiento de la solución inicial de ángulos de torsión por medio de los algoritmos HSA. Por último en la salida se obtiene la estructura terciaria o tridimensional de un proteína o péptido.

3.4 Método con algoritmo GRSA2 y la estrategia de selección de ruleta.

El método GRSA2-SSP es un método computacional para PFP, muy bueno para el caso de péptidos en el que ha obtenido buenos resultados. Este método utiliza la heurística de búsqueda áurea y también soluciones de estructura secundaria para buscar una estructura cercana al NS. Sin embargo, GRSA2-SSP se ha aplicado a péptidos pero en los más grandes no siempre da buenos resultados, por lo que una estrategia importante es optimizar las cadenas laterales de aminoácidos. En este trabajo, mejoramos GRSA2-SSP en varios aspectos importantes y lo aplicamos a un conjunto de datos con péptidos pequeños, medianos y grandes. Los resultados muestran que este método obtiene resultados estadísticamente equivalentes o mejores que los mejores algoritmos del área. Para lograr

esto aplicamos dos pasos: en primer lugar, se aplica el algoritmo GRSA2-SSP para mejorar la estructura general de la proteína, y en segundo lugar, se aplican estrategias de selección de ruleta [Zhou, 1996] que permitan una buena selección de las regiones de las cadenas laterales de aminoácidos para mejorar la búsqueda de la estructura. de la proteína con la energía más baja. Evaluamos el desempeño de nuestro método utilizando las métricas RMSD y TM-score [Zhang, 2004] y comparando la estructura inicial con la estructura refinada.

El mejoramiento de GRSA2-SSP con estrategias de selección de la ruleta para el refinamiento de las regiones de las cadenas laterales de aminoácidos se describe principalmente en dos pasos: 1) GRSA2-SSP se aplica para mejorar la predicción general de la estructura de la proteína, 2) Las estrategias de selección de la ruleta se utilizan para hacer un selección de regiones de la cadena lateral.

El método mejorado GRSA2-SSP con la estrategia de selección de la ruleta se ha nombrado método GRSA2-SSPR por el de Secondary Structure Prediction Roulette (SSPR). En la Figura 3.3 se muestra la metodología extendida del GRSA2-SSP incorporando la estrategia de ruleta en el refinamiento para mejorar las cadenas laterales de las estructuras de los péptidos.

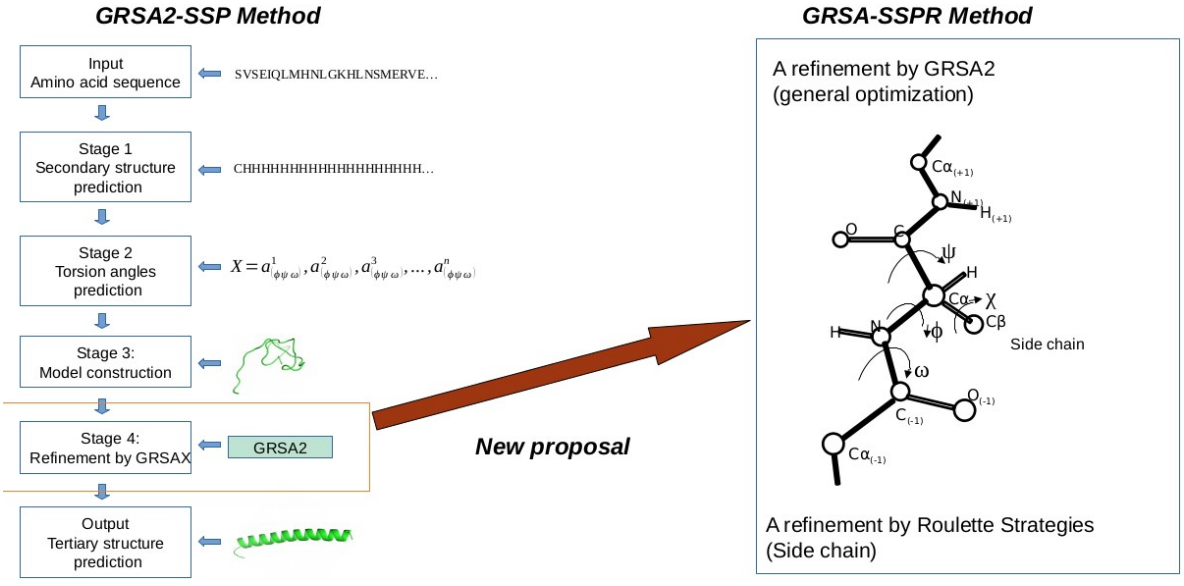


Figura 3.3 Metodología GRSA-SSPR.

Cuando se realiza el refinamiento se perturban los ángulos de torsión de la estructura basados en una selección de ruleta. Los ángulos de torsión laterales son seleccionados para ser perturbados mediante la estrategia de ruleta en la cual se da una puntuación para los ángulos seleccionados, si los ángulos seleccionados al ser perturbados y posteriormente al evaluar la estructura general con la función de energía, si mejora la energía de la estructura entonces se le asigna una puntuación mayor a los ángulos laterales. Al realizar un incremento dando una puntuación a los ángulos de torsión laterales, y basando la selección de estos ángulos en base a esa puntuación, incrementara la probabilidad de selección de estos ángulos de la cadena lateral como se puede mostrar en la Figura 3.4, donde la probabilidad de seleccionar ángulos laterales que ayuden a mejorar la energía aumentara y así mejorar la estructura general del péptido o proteína. Donde S_i es el vector de ángulos de torsión con sus respectivos ángulos (ϕ, Ψ, ω, X) para cada aminoácido (aa_1, …,aa_n). Los resultados de esta estrategia se pueden observar en la sección de análisis y resultados.

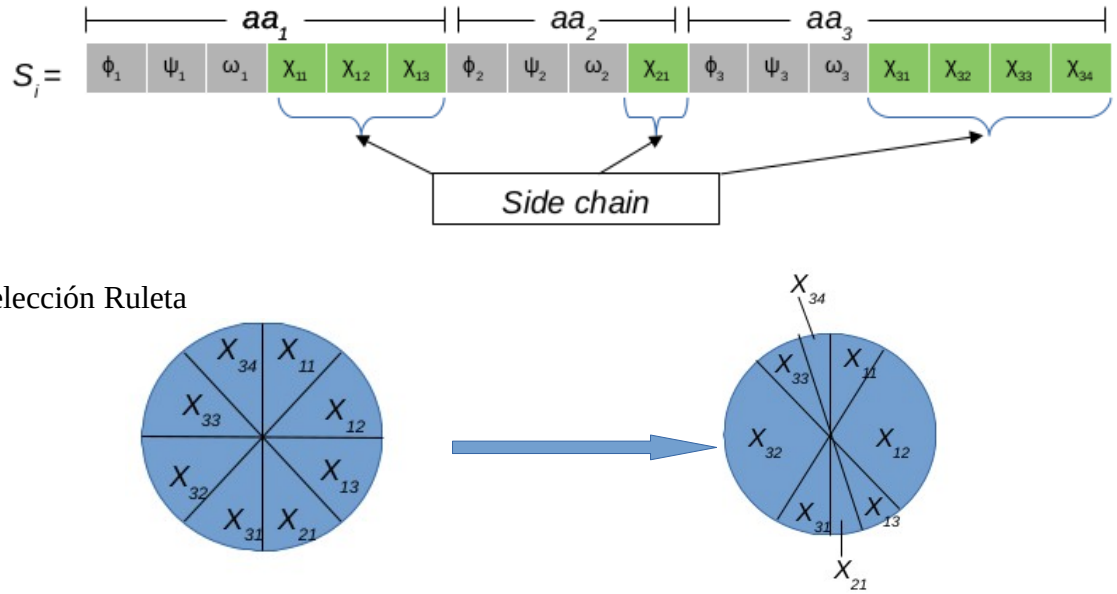


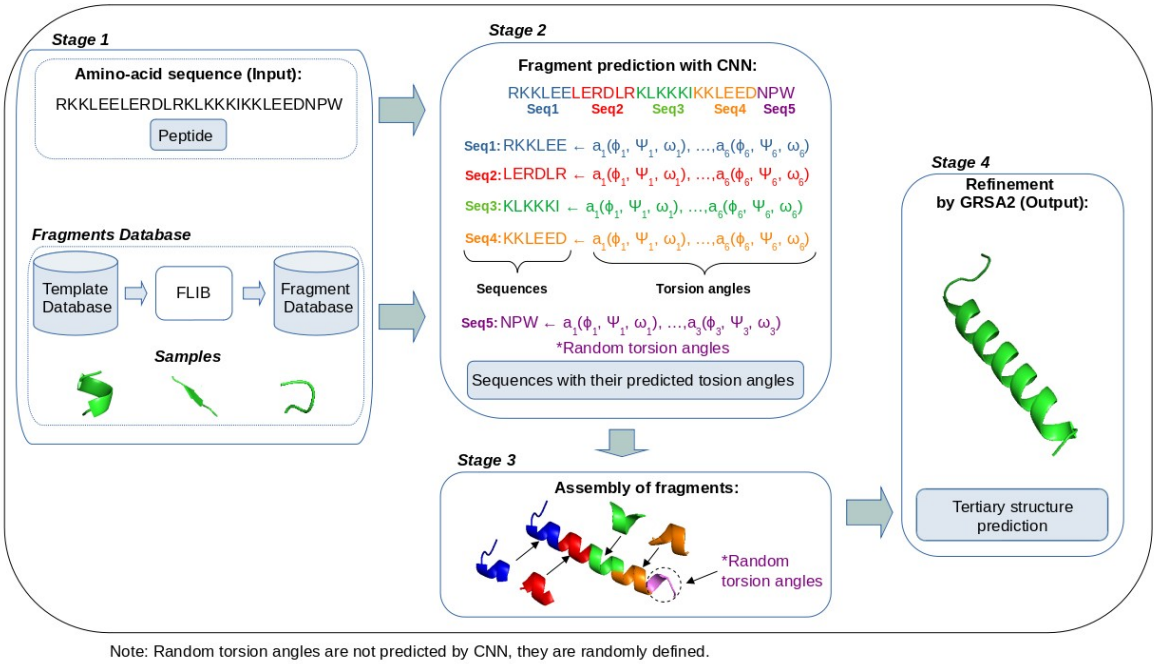
Figura 3.4 Selección ruleta.

3.5 Método con algoritmo GRSA2 y fragmentos.

El algoritmo GRSA2 híbrido de SA utilizado para el refinamiento de los modelos como en el método con estructura secundaria, refinando una estructura inicial a partir de los ángulos de las estructuras secundarias. El GRSA2 es también utilizado para el refinamiento pero para una estructura inicial formado por un ensamble de fragmentos predichos por una red convolucional neuronal.

Uno de los algoritmos de aprendizaje profundo (DL, Deep Learning) más populares son las redes neuronales convolucionales (CNN, Convolutional Neuronal Network), que utiliza el funcionamiento convolucional para la extracción automática de características de conjuntos de datos. La CNN se compone de etapas convolucionales, etapas de agrupación y capas totalmente conectadas. La CNN ha conseguido enfrentarse a varios retos, como [Dyrmann, 2016; Muhammad,2019; Frausto, 2021]. Hay tres aspectos importantes en una CNN: representaciones equivalentes, interacciones dispersas y reparto de parámetros [Goodfellow, 2016]. Existen varias arquitecturas de CNN [Alzubaidi, 2021], algunas de ellas son AlexNet, ZefNet, GoogLeNet y ResNet.

De acuerdo a la metodología de la Figura 3.1 se presenta una nueva variante la cual llamamos metodología GRSA2-FCNN, FCNN por sus sigla en inglés Fragments Convolutional Neuronal Network, en la cual se obtiene la estructura tridimensional de una proteína a partir de la secuencia lineal de amino ácidos (aa's). Se utiliza una CNN para predecir fragmentos de la secuencia objetivo de aminoácidos. Los fragmentos son pequeños cortes de tamaño de seis aa's y los fragmentos predichos se utilizan para ensamblar una estructura inicial para la proteína completa que luego se refina hasta obtener la estructura tridimensional final de la proteína. En la Figura 3.5 mostramos una vista general de GRSA2-FCNN.



Note: Random torsion angles are not predicted by CNN, they are randomly defined.

Figura 3.5 Método con fragmentos, GRSA2-FCNN para predicción de péptidos.

Las principales etapas de GRSA2-FCNN son las siguientes

- Secuencia de aminoácidos (etapa 1): La secuencia de aminoácidos de la proteína objetivo es la entrada para nuestro método. En esta etapa, la base de datos de fragmentos; contiene un conjunto de fragmentos, que se clasifican según sus estructuras secundarias predominantes alfa, beta y de bucle.

- Predicción de fragmentos con CNN (etapa 2): La base de datos de fragmentos de la etapa 1 se utiliza como entrada para el entrenamiento de una CNN que realiza la predicción de los fragmentos (alfa, beta y bucle) y sus ángulos de torsión; que son los ángulos internos de la columna vertebral de una proteína (phi, psi y omega). Una vez completado el entrenamiento de la CNN, se utiliza la secuencia objetivo de aminoácidos para predecir los ángulos de torsión en secuencias de seis aa denominadas cortes.

- Ensamblaje de los fragmentos (etapa 3): Los fragmentos predichos se ensamblan para construir un nuevo modelo de la secuencia objetivo. En este proceso, los ángulos de torsión de los fragmentos se ensamblan en cortes de seis aminoácidos basados en la secuencia del objetivo de aa. Si eventualmente, el tamaño de la secuencia objetivo no es

proporcional al tamaño de los fragmentos algunos ángulos no pudieron ser predichos, en ese caso, se utilizan valores aleatorios; este y otros problemas se resuelven en la siguiente etapa.

- Refinamiento mediante GRSA2 (etapa 4): El nuevo modelo formado por el ensamblaje de fragmentos de la etapa 3, se refina con el algoritmo GRSA2. El resultado de esta etapa es la estructura terciaria final de la secuencia proteica objetivo.

3.5.1 Predicción de fragmentos con CNN (FCNN)

En la etapa 2 utilizamos una CNN, que denominamos CNN de fragmentos (FCNN), que tiene como entrada un conjunto de fragmentos tomados de una base de datos generada por Flib [De Oliveria, 2015]. Esta última base de datos es una biblioteca de fragmentos de estructuras tridimensionales conocidas tomadas del Protein Data Bank [Bernstein, 1977]. Además, cada fragmento se realiza haciendo cortes en las estructuras conocidas; así, podemos conformar un conjunto de fragmentos alfa, beta y de bucle [De Oliveira, 2015]. Para obtener los fragmentos de nuestra base de datos, utilizamos 12,368 fragmentos de tipo alfa, 9,953 fragmentos de tipo beta y 3,576 fragmentos de tipo bucle; esta base de datos se utiliza como datos de entrada para nuestra CNN. Los fragmentos predichos por esta red se describen por su secuencia de aminoácidos y sus respectivos ángulos de torsión phi (ϕ), psi (Ψ) y omega (ω). El conjunto de datos se dividió en un 80% para el entrenamiento y un 20% para la validación; esto se hizo para cada tipo de fragmento. La arquitectura de la CNN (véase la figura 3.6) contiene cuatro capas unidimensionales (CNN 1D) con una configuración con un tamaño de núcleo de cuatro, y una función de activación ReLU seguida de un abandono con un valor de 0,1; a continuación, una capa de maxpooling con un tamaño igual a dos. Las características aprendidas se aplanan y pasan por dos capas totalmente conectadas de 128 y 256 neuronas, respectivamente, y una función de activación ReLU antes de la capa de salida de 18 neuronas que se utiliza para la predicción. La configuración de entrenamiento utilizada fue un optimizador Adam [Kingma, 2015], el error cuadrático medio como función de pérdida, 200 épocas y un tamaño de lote con un valor de 8. Partiendo de la secuencia de entrada de la etapa 1, realizamos seis cortes de aa para generar los fragmentos del modelo inicial construido en la siguiente etapa. La

configuración y los parámetros de nuestra CNN se determinaron por experimentación, mostrando el mejor rendimiento en la metodología propuesta.

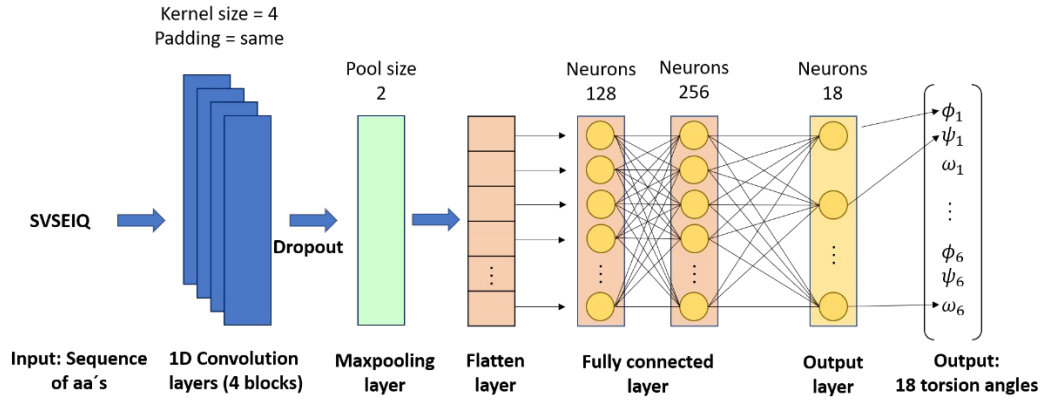


Figura 3.6 Arquitectura FCNN.

3.5.2 Ensamble de fragmentos

La construcción del nuevo modelo de proteína en la etapa 3 se basa en el ensamblaje de fragmentos. La FCNN predice los ángulos de torsión de la secuencia objetivo. Cada fragmento predicho por la FCNN se ensambla uno a uno según la posición de su secuencia de aminoácidos. Para ello, la FCNN utiliza la base de datos Flib para entrenar un modelo de predicción, que predice los ángulos de torsión para cada fragmento de la secuencia de aminoácidos objetivo. En otras palabras, estos ángulos de torsión representan un modelo inicial Si = [ϕ1, Ψ1, X1, ω1, ϕ2, Ψ2, X2, ω2,..., ϕn, Ψn, Xn, ωn], donde los ángulos correspondientes a cada aminoácido están determinados por el subíndice de 1 a n. Por ejemplo, en el caso de un péptido con 27 aa's, se construye con cuatro fragmentos cuya longitud es de seis aa's; los aa's restantes se inician con un valor aleatorio generado por el algoritmo GRSA2 durante la fase de refinamiento. La Figura 3.7 muestra dos ejemplos de los modelos iniciales con los fragmentos generados por FCNN, el péptido 1pef en (a) tiene una SS mayoritaria alfa y 1e0q (b) tiene una SS mayoritaria beta.

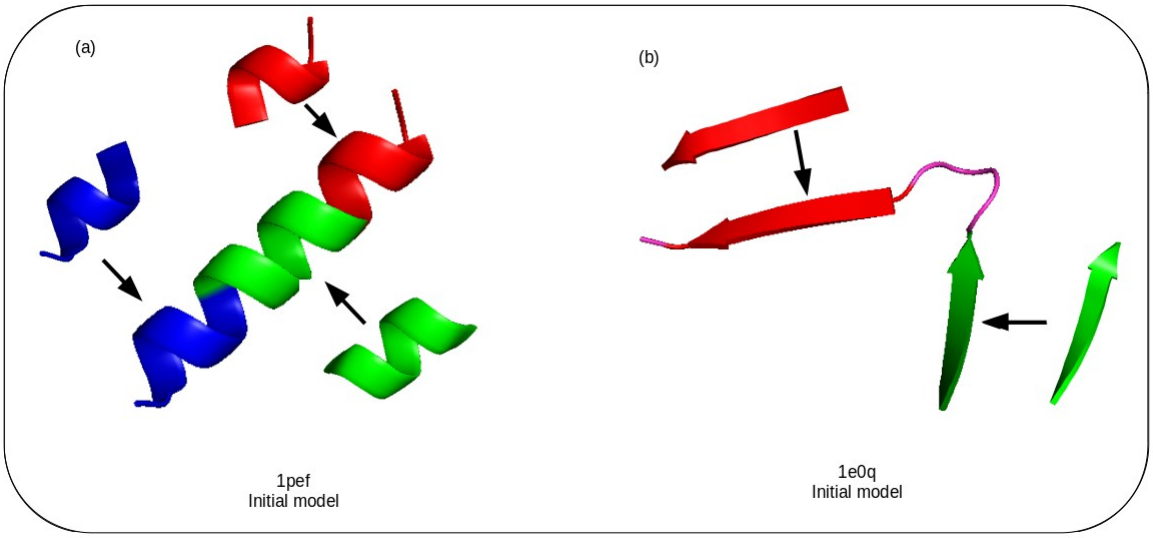


Figura 3.7 Modelos iniciales con los fragmentos, generados por FCNN.

3.5.3 Refinamiento por GRSA2.

El GRSA2 de la etapa 4 perfecciona el modelo obtenido en la etapa anterior. Las principales características de este algoritmo son: en primer lugar, se implementa un SA de enfriamiento más rápido. En el esquema de enfriamiento para bajar el valor de la temperatura, se utiliza el parámetro alfa en un rango de valores de 0,75 a 0,95 con cinco tramos de proporción áurea, que se determina por experimentación [Lamiable, 2016]; por último, se aplican diferentes estrategias de perturbación para explorar el espacio de soluciones. La búsqueda de soluciones se basa en la descomposición de la perturbación y la colisión suave para encontrar una nueva estructura con menor energía.

La figura 3.8 muestra cuatro modelos obtenidos mediante el refinamiento GRSA2 y la estructura nativa evaluada con las métricas TM-score y GDT-TS.

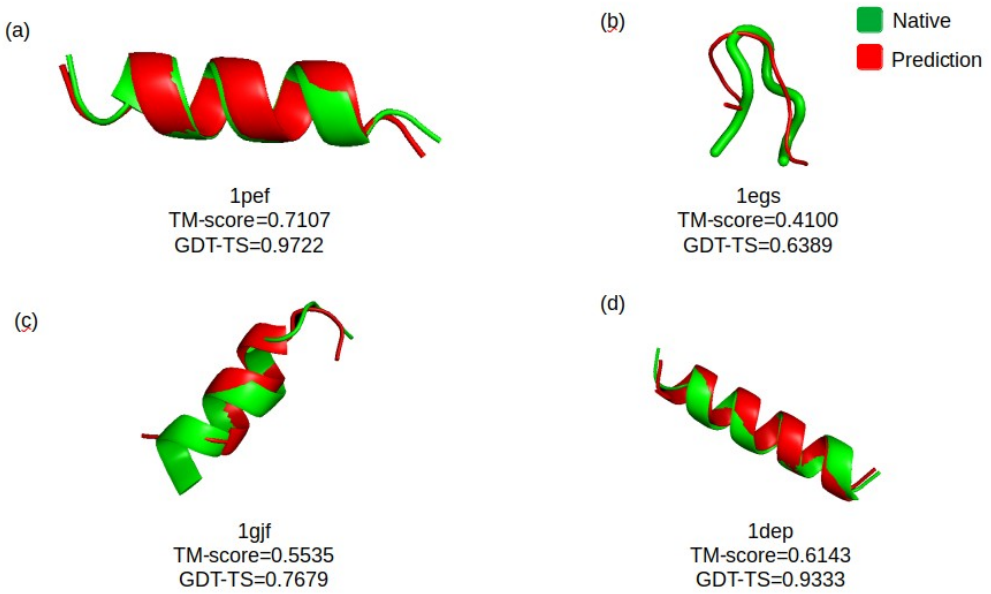


Figura 3.8 Modelos tridimensionales de péptidos refinados por GRSA2 (rojo) y la estructura nativa (verde)

3.6 Algoritmos del estado del arte.

Existen algoritmos que realizan la predicción de estructuras tridimensionales de proteínas a partir de la secuencia de aminoácidos algunos de ellos lo realizan por medio de los siguientes métodos homología, threading, y utilizando fragmentos de proteínas conocidas, se describen a continuación los siguientes Pep-Fold3, I-Tasser, Quark y Rosetta.

PEP-FOLD3 es un algoritmo que realiza predicciones de la estructura terciaria de péptidos entre 5 y 50 aminoácidos a partir de su estructura primaria. El proceso de predicción inicia con la secuencia de aminoácidos del péptido, con esa secuencia se prediccen fragmentos usando un alfabeto estructural; posteriormente con un algoritmo Monte Carlo se realiza el ensamble de fragmentos y se realiza el refinamiento de la estructura formada por los fragmentos, por último se elige la estructura con la mejor conformación [Lamiable, 2016].

I-TASSER (Iterative Threading ASSEmbly Refinement) es un algoritmo de Zhan-Lab que predice estructuras tridimensionales de proteínas y anotaciones de funciones basadas en estructuras. El proceso de predicción de I-TASSER inicia con la secuencia de aminoácidos y con ella identifica plantillas de estructuras de la de PDB mediante el enfoque de threading múltiple LOMETS [Zheng, 2019], construye modelos completos mediante el ensamble de fragmentos basado en plantillas, estos modelos son refinados con el algoritmo de Monte Carlo. Al final del proceso se generan las anotaciones de funciones mediante modelos tridimensionales a través de una base de datos de funciones de proteínas de BioLip [Yang, 2013]. I-TASSER genera estructuras tridimensionales de proteínas de 10 a 1500 aminoácidos [Yang, 2015].

Quark es un algoritmo que utiliza la estrategia de *ab initio* para predecir estructuras de proteínas que tienen una secuencia entre 20 y 200 aminoácidos. Este algoritmo ensambla fragmentos de longitudes continuas en 1–20 residuos usando un algoritmo Monte Carlo para la construcción de la estructura tridimensional [Xu, 2013].

Rosetta es un algoritmo que predice estructuras tridimensionales de la estructura primaria a partir de 27 aminoácidos, empleando la predicción de estructuras secundarias. Rosetta genera fragmentos a partir de su estructura primaria. Los fragmentos son ensamblados mediante un algoritmo de recocido simulado Monte Carlo. Posteriormente realiza un ajuste mediante estructuras de proteínas conocidas. Finalmente genera la estructuras tridimensional [Ovchinnikov, 2016].

AlphaFold [Senior, 2015] denominado por sus autores como sistema de inteligencia artificial utiliza métodos basados en el aprendizaje profundo y combina tres redes neuronales (NN): la primera, predice la distancia entre pares de residuos dentro de la proteína; la segunda NN, se aplica para estimar la precisión de las estructuras candidatas; finalmente, la tercera NN se utiliza para generar la estructura de la proteína NS. Las combinaciones de estas NN hacen uso de dos SA con memoria y generación de fragmentos neuronales [Simons, 1997] con potencial GDT-net y potencial de distancia [Senior, 2020];

además, se aplica un descenso de gradiente repetido de potencial de distancia [Conway, 2022]. En el evento CASP14, AlphaFold2 [Mirdita, 2022] obtuvo un rendimiento excelente. AlphaFold2 utiliza métodos de detección de homología muy sensibles como MMseqs2 [Mirdita, 2019] para encontrar plantillas homólogas.

Estos algoritmos, métodos o sistema (programa) mediante su respectivo servidor fueron utilizados para predecir la estructura tridimensional del conjunto de 45 instancias en sus respectivas comparativas, conjunto de la Tabla 3.1, en el siguiente capitulo se realiza la comparación de los resultados obtenidos por los algoritmos del estado del arte, los métodos con algoritmos HSA con la secuencia de aminoácidos, método con los algoritmos HSA y la predicción de la estructura secundaria, método con el algoritmo GRSA2 y la estrategia de ruleta, y método con el algoritmo GRSA2 con predicción de fragmentos mediante una red neuronal convolucional para analizar el comportamiento y desempeño de cada método.

Capítulo 4

4 Análisis y Resultados

En este capítulo se presentan los resultados de las predicciones utilizando los algoritmos HSA (SA, GRSA, EGRSA & GRSA2) para el conjunto de proteínas de la Tabla 3.1, aplicando los algoritmos HSA a cada instancia para evaluar cada método. Primero se presentan los resultados de los algoritmos HSA con la secuencia de aminoácidos y se realiza una comparación del mejor algoritmo de ellos con los algoritmos del estado del arte. Posteriormente se muestran resultados de HSA con la secuencia de aminoácidos y la predicción de estructuras secundarias, y el mejor algoritmo de HSA se compara con los algoritmos del estado del arte. Se presentan resultados de GRSA2-SSPR estrategia de ruleta para el refinamiento por cadenas laterales. Posteriormente se presentan resultados del método con ensamble de fragmentos y refinamiento de GRSA2 para un conjunto de 60 péptidos.

Los algoritmos HSA fueron ejecutados con instancias de la Tabla 3.1 para la predicción de las estructuras tridimensionales, a continuación se muestran los resultados con los algoritmos HSA con sus diferentes variaciones basadas en la metodología general.

4.1 Métodos HSA.

En las siguientes Figuras (Figura 4.1 a Figura 4.3) se muestra el comportamiento de HSA con la secuencia de aminoácidos y los algoritmos SA, GRSA, EGRSA y GRSA2.

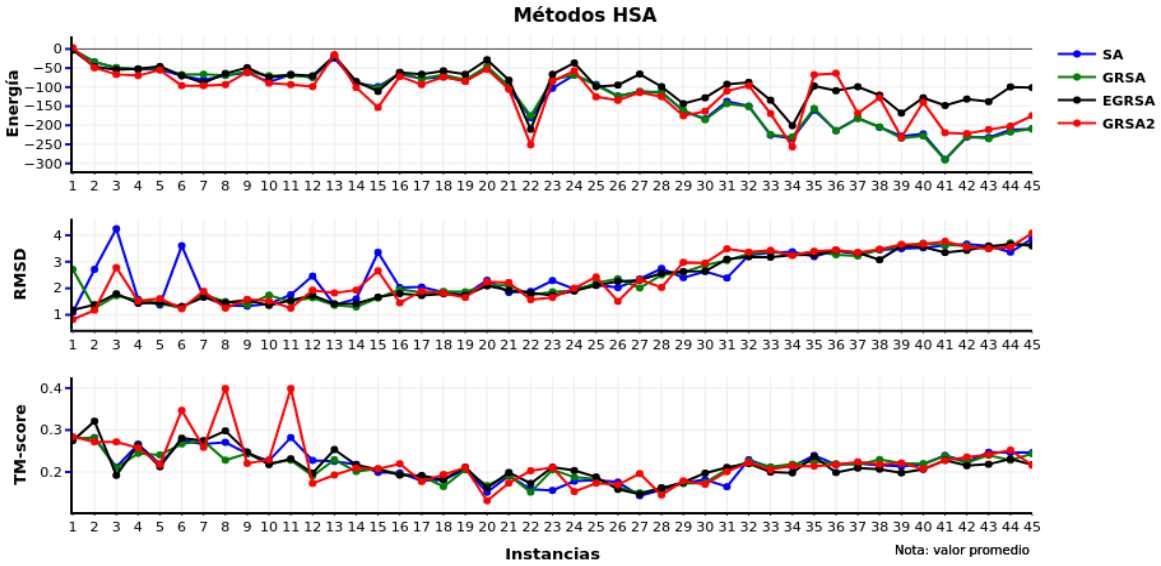


Figura 4.1 Resultados de HSA con secuencia de aminoácidos.

La Figura 4.1 muestra los resultados de Energía, RMSD y TM-score [Zhang, 2004]. Si el valor de TM-score es más cercano a uno, significa que tiene una mejor estructura tridimensional, para el RMSD entre más cercano a cero tiene una mejor estructura y la energía debe ser mínima; se tomaron los resultados promedio de 30 ejecuciones por cada instancia (45), para cada instancia se realiza una sintonización de parámetros previa y se comparan los métodos SA, GRSA, EGRSA y GRSA2. Los valores que tienen un valor cercano a uno poseen una predicción de estructura cercana a la estructura nativa lo que representa una buena predicción y un buen resultado. La numeración de la instancia es correspondiente a la Tabla 3.1 la cual va del rango del 1 al 45 de menor a mayor número de aminoácidos.

Como se observa en la Figura 4.1 el GRSA2 tiene resultados buenos en algunas instancias de TM-score superando el valor 0.3 en 3 instancias lo que nos muestra que

obtiene predicciones de estructuras similares a la nativa comparado con los otros 3 métodos (SA, GRSA, EGRSA), el GRSA2 mejora algunos resultados de predicción, los métodos SA, GRSA y EGRSA tienen resultados similares entre ellos por lo que el GRSA2 tiene un comportamiento en las primeras instancias donde es mejor.

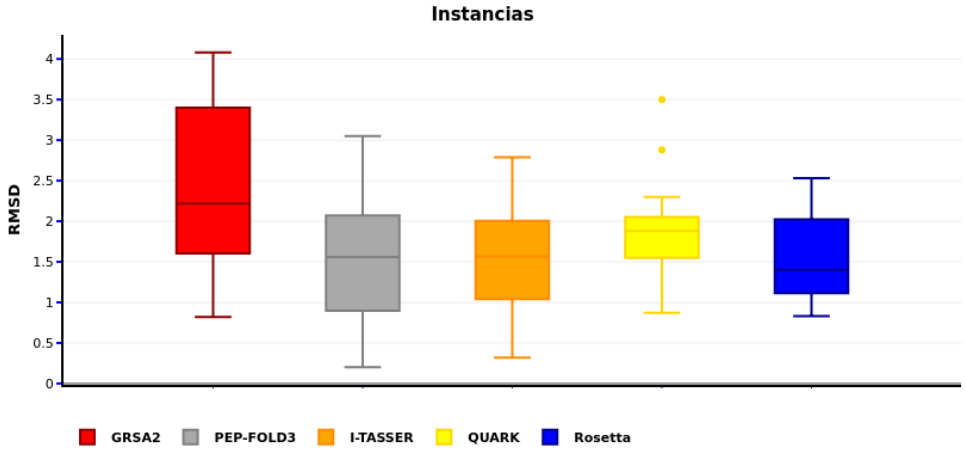


Figura 4.2 Resultados de GRSA2 y algoritmos del estado del arte; en RMSD.

En la Figura 4.2 se muestra el RMSD de las predicciones de las estructuras tridimensionales de las 45 instancias realizadas por GRSA2, PEP-FOLD3, I-TASSER, QUARK y Rosetta. Los mejores valores de RMSD son obtenidos por I-TASSER y posteriormente QUARK, el GRSA2 incluso tiene valores altos de RMSD.

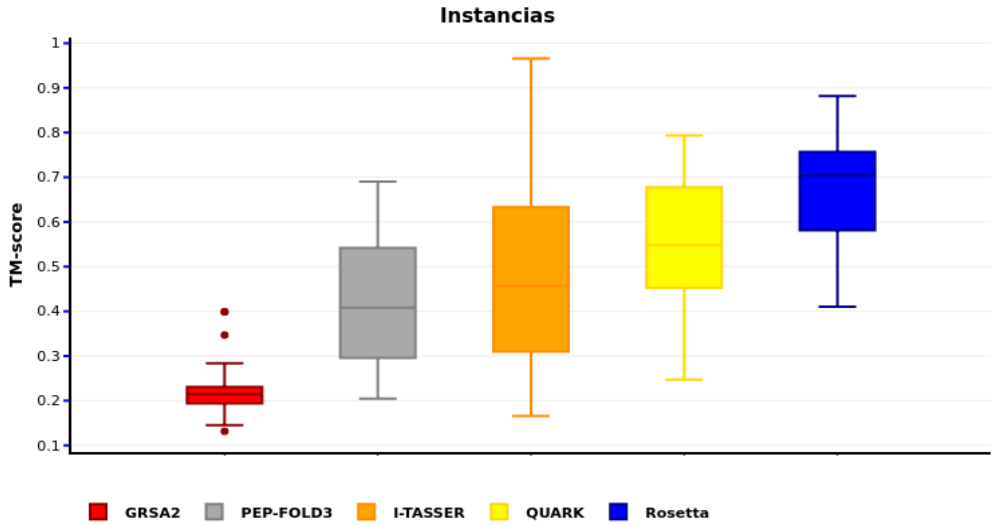


Figura 4.3 Resultados de GRSA2 y algoritmos del estado del arte; en TM-score.

Se observa en la Figura 4.3 se presenta el TM-score de las predicciones de las estructuras tridimensionales de las 45 instancias realizadas por GRSA2, PEP-FOLD3, I-TASSER, QUARK y Rosetta. Los mejores valores de TM-score son obtenidos por Rosetta

aunque su algoritmo puede predecir a partir de 27 aminoácidos por lo que tiene menos instancias evaluadas y como segundo mejor está QUARK aunque también predice a partir de 20 aminoácidos, el GRSA2 aún tiene valores bajos de TM-score por lo que la calidad de sus resultados aún está baja con los algoritmos del estado del arte.

A continuación se presenta el comportamiento de los algoritmos HSA con la secuencia de aminoácidos y la predicción de estructura secundaria con los algoritmos SA, GRSA, EGRSA y GRSA2 para identificar cada algoritmo se ha etiquetado GRSA0-SSP para SA, GRSA1-SSP para GRSA, GRSAE-SSP para GRSA Evolutivo y GRSA2-SSP para el GRSA2. Se comparan cada algoritmo de GRSA-SSP con los algoritmos GRSA sin predicción de la estructura secundaria los que solo emplean la estructura primaria con las etiquetas GRSA0, GRSA1, GRSAE y GRSA2 en las Figuras 4.4, 4.5, 4.6 y 4.7 :

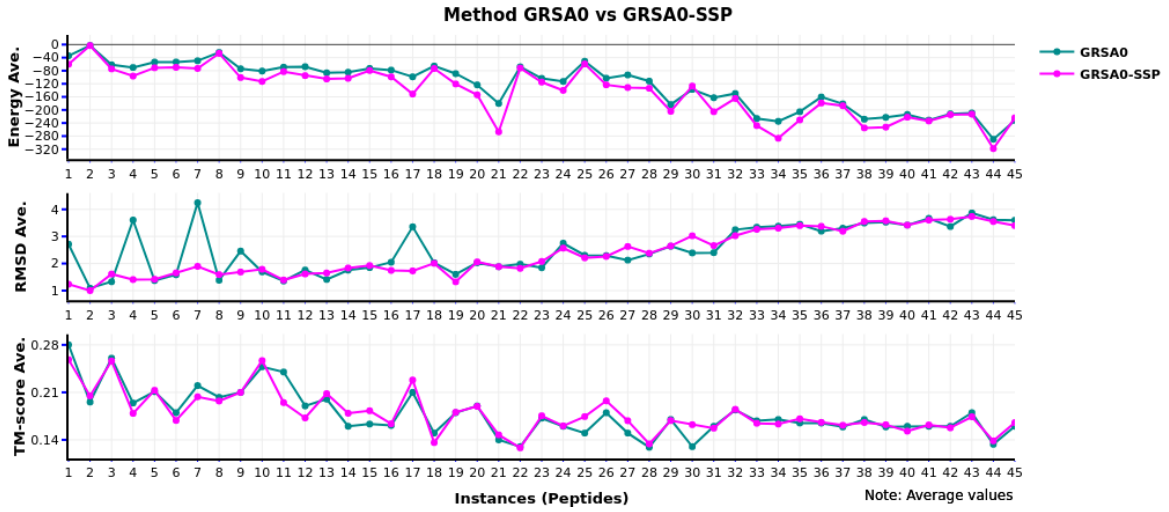


Figura 4.4 Comparativa de GRSA0 y GRSA0-SSP.

La Figura 4.4, presenta un comportamiento para GRSA0-SSP mejor que el comportamiento de GRSA0. En la mayoría de los péptidos GRSA0-SSP obtiene la menor energía. En el caso del RMSD el GRSA0-SSP no presenta tanta variación como el GRSA0 en las instancias de la 1 a la 17. Cuando se compara el TM-score el comportamiento en general es muy similar. En está comparación GRSA-SSP obtiene mejora de resultados en energía y RMSD.

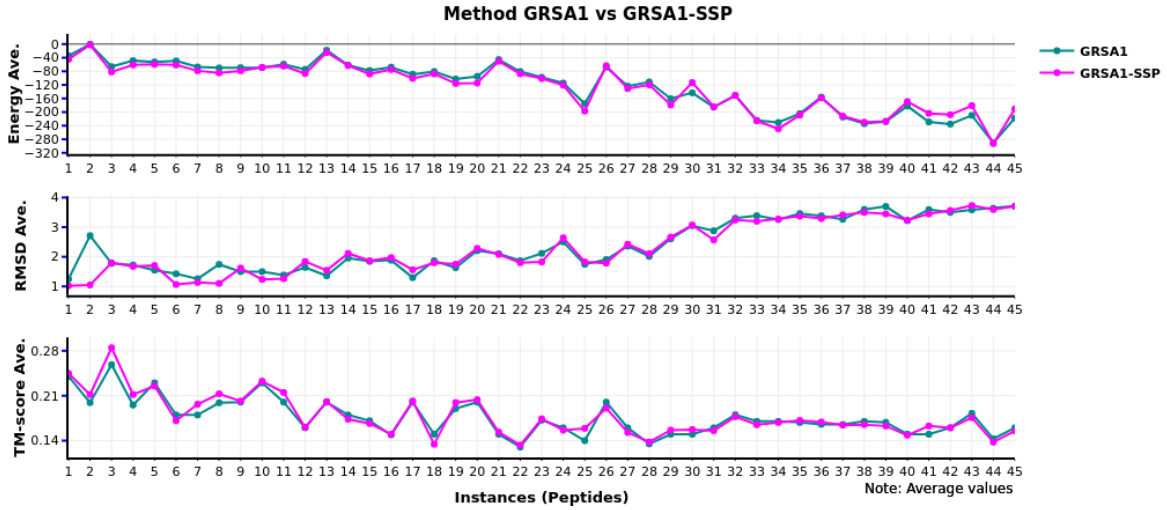


Figura 4.5 Comparativa de GRSA1 y GRSA1-SSP.

La Figura 4.5, presenta el el comportamiento de GRSA1-SSP contra GRSA1, con las métricas de energía, RMSD y TM-score; obteniendo un comportamiento similar entre ambos algoritmos.

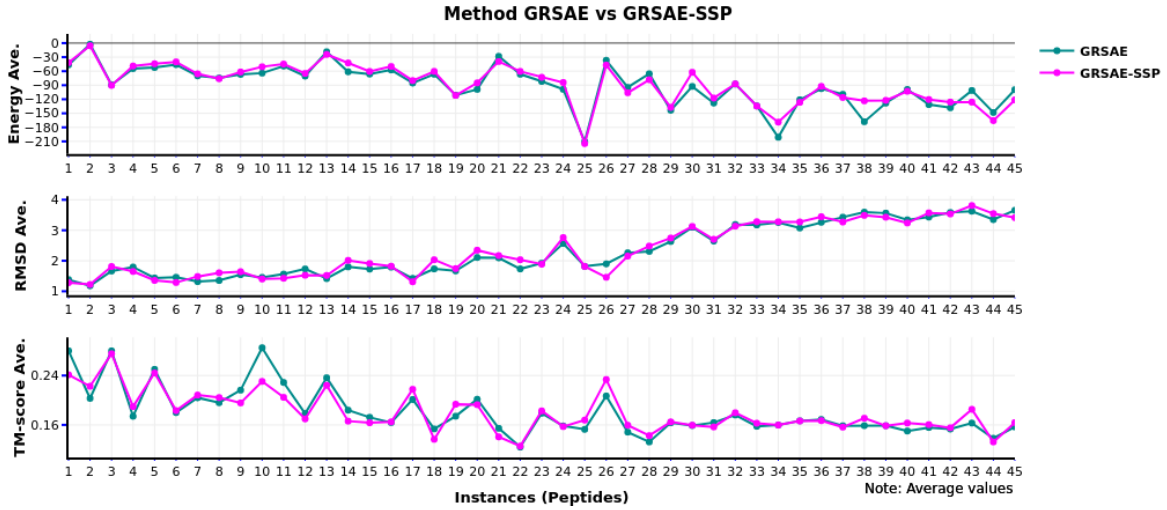


Figura 4.6 Comparativa de GRSAE y GRSAE-SSP.

La Figura 4.6, muestra el comportamiento de GRSAE-SSP comparándose este contra GRSAE, se puede observar que los resultados son similares en energía, RMSD y TM-score.

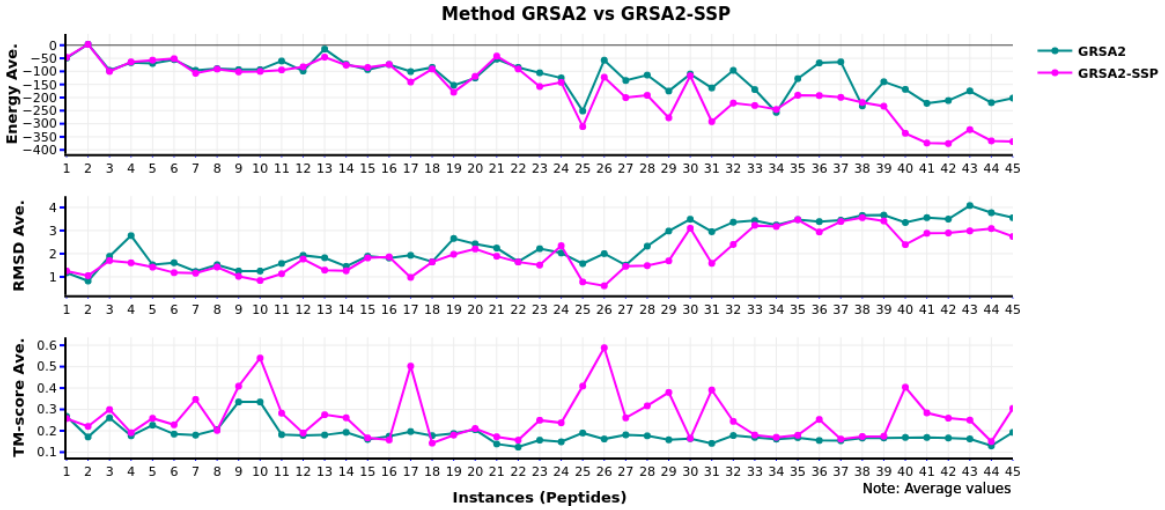


Figura 4.7 Comparativa de GRSA2 y GRSA2-SSP.

En la Figura 4.7, se presenta la comparación entre GRSA2 y GRSA2-SSP. Los resultados obtenidos para cada instancia es muy notable y se aprecia una superioridad de GRSA2-SSP usando las métricas de energía, RMSD y TM-score. Al aplicar la metodología GRSA-SSP se mejora considerablemente el comportamiento del algoritmo GRSA2.

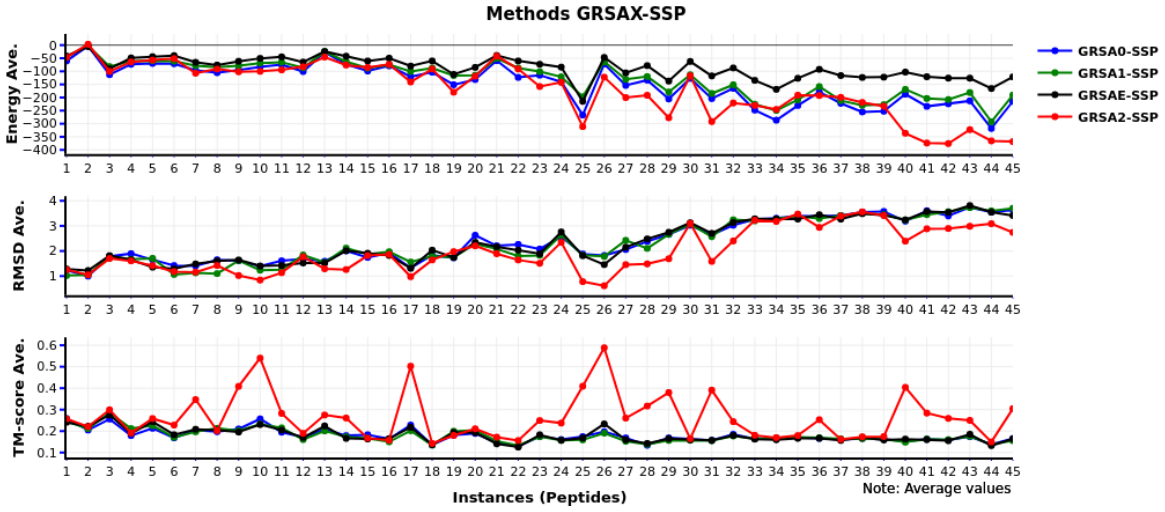


Figura 4.8 Comparativa de GRSA's con SSP.

En la Figura 4.8, se muestra una comparación de los algoritmos GRSAX-SSP donde X representa cada algoritmo de la familia GRSA. Se puede observar que GRSA2-SSP obtiene los mejores valores en varias instancias contra los demás algoritmos. Por lo tanto el mejor comportamiento de los algoritmos con la predicción de la estructura secundaria es GRSA-SSP.

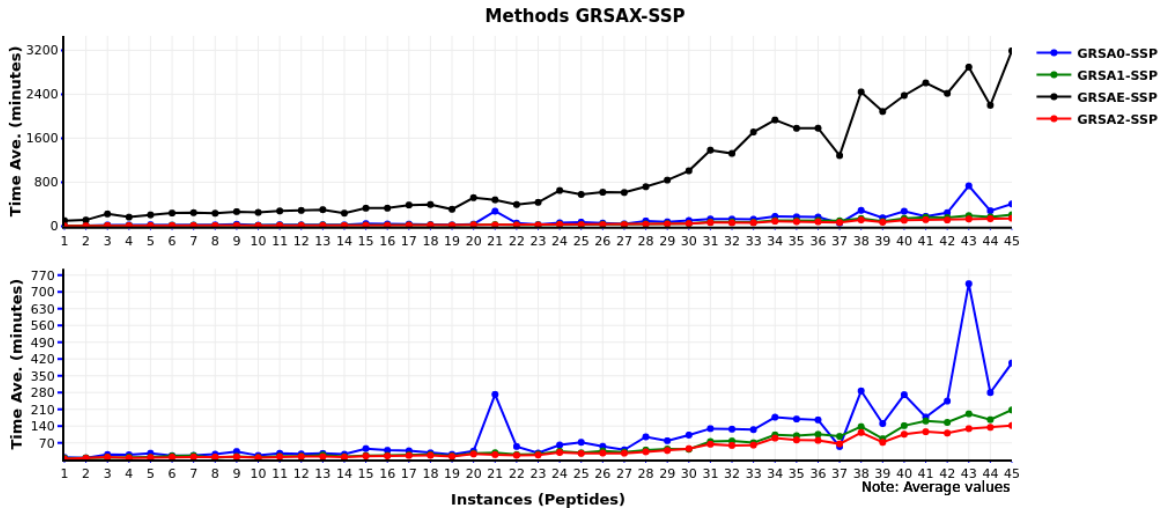


Figura 4.9 Comparativa de tiempo de los algoritmos GRSAX-SSP.

En la Figura 4.9, se presenta una comparación del tiempo computacional de los algoritmos GRSAX-SSP donde el GRSA2-SSP tiene el mejor comportamiento con valores más bajos en la mayoría de instancias comparado con los otros algoritmos.

La Figura 4.10, nos muestra el comportamiento del algoritmo GRSA2-SSP con las instancias de prueba por tipo de estructura secundaria. Las instancias o péptidos son agrupados de acuerdo al tipo de estructura secundaria predominante en beta strand (B), alpha-helix (A) y none (N) que no es ni mayormente beta o alpha. Se observa que el mejor comportamiento es en las estructuras tipo alpha (A) tanto para el mejor valor como en el promedio en RMSD y Tm-score.

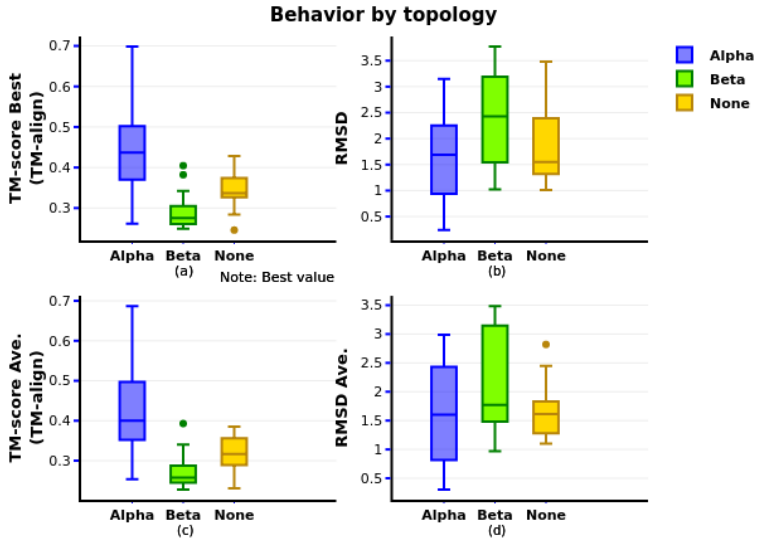


Figura 4.10 GRSA2-SSP de acuerdo al tipo de estructura secundaria.

En las Figuras 4.11 a 4.13, se presentan el comportamiento del algoritmo GRSA2-SSP y se compara con los resultados de los algoritmos de PEP-FOLD3, I-TASSER, QUARK y Rosetta. Para realizar está comparación se han dividido en 3 grupos de 15 instancias el conjunto de la Tabla 3.1 formando grupo 1 de la instancia 1 a 15, grupo 2 de la 16 a 30 y grupo 3 de la 31 a 45. Las comparaciones de esto 3 grupos son usando las métricas de RMSD, TM-score, GDT-TS [Zemla, 2001], presentando el mejor valor para TM-score y RMSD de las 5 mejores predicciones. Anexando también el GDT-TS y TM-score promedio de las 5 mejores predicciones.

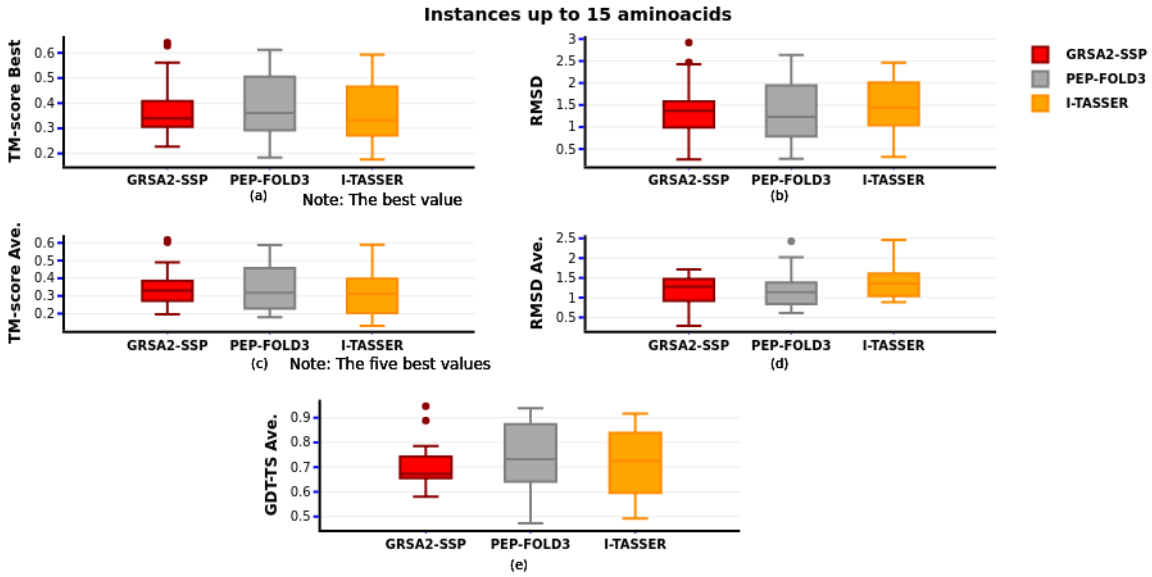


Figura 4.11 Comparativa GRSA2-SSP con servidores para el grupo 1.

La Figura 4.11, compara el grupo 1 que consta de 15 instancias que tienen en su secuencia de aminoácidos (aa's) hasta 15 aa's. Se puede observar que el GRSA2-SSP tiene un comportamiento similar a I-TASSER y PEP-FOLD3, sin embargo PEP-FOLD3 es ligeramente mejor que GRSA2-SSP e I-TASSER también cuando se compara el GDT-TS. Los algoritmos Rosetta y QUARK no se añadieron debido a que predicen instancias a partir de 27 y 20 aa's respectivamente.

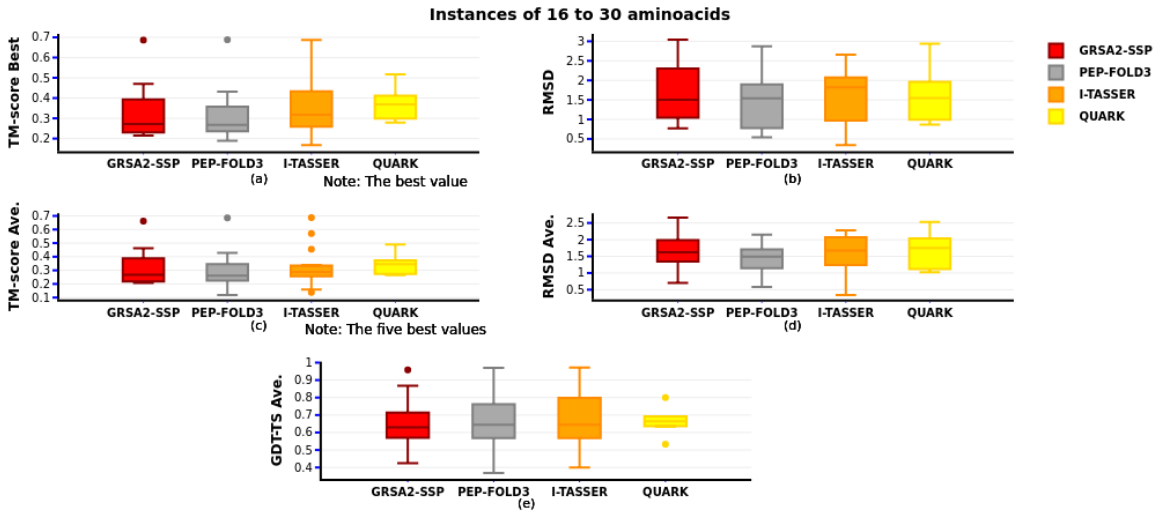


Figura 4.12 Comparativa GRSA2-SSP con servidores para el grupo 2.

En la Figura 4.12 se compara el grupo 2 de 16 a 30 aa's, con la mejor y el promedio de las 5 mejores predicciones obtenidas usando las métricas de TM-score, y su respectivo RMSD y GDT-TS. En este grupo se anexa QUARK y Rosetta es omitido porque solo tiene 3 instancias en este grupo. GRSA2-SSP tiene un comportamiento muy similar a PEP-FOLD3, I-TASSER y QUARK en Figura 4.12 (a). En Figura 4.12 (c) I-TASSER obtiene los mejores resultados seguido por PEP-FOLD3 y GRSA2-SSP. Para el GDT-TS de la Figura 4.12 (e), GRSA2-SSP tiene un comportamiento similar a PEP-FOLD3, I-TASSER y QUARK. Se puede decir que GRSA2-SSP e I-TASSER tienen un comportamiento similar en promedio.

En el grupo 3 de 31 a 49 aa's podemos observar en la Figura 4.13 que I-TASSER es el mejor algoritmo de este grupo, ya que en TM-score mejor y promedio obtiene mejores resultados, seguido por Rosetta, QUARK, PEP-FOLD3 y GRSA2-SSP.

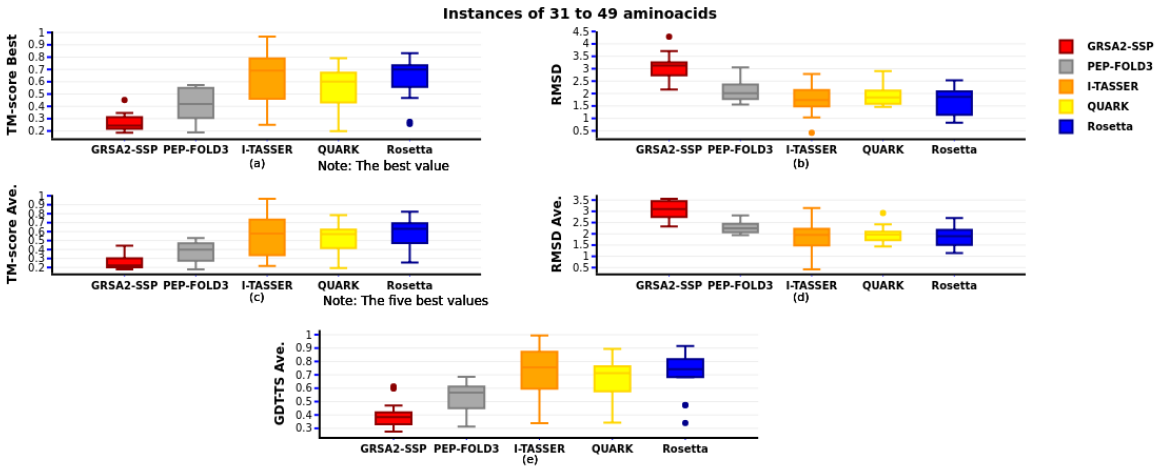


Figura 4.13 Comparativa GRSA2-SSP con servidores para el grupo 3.

El conjunto de 45 instancias evaluado en la experimentaciones nos muestra que la aplicación de estructuras secundarias y el refinamiento mediante los algoritmos de la familia GRSA, mejoran el comportamiento en energía, RMSD y TM-score. En el caso de GRSA2-SSP el comportamiento es mejorado considerablemente con respecto a los demás algoritmos de la familia GRSA. Finalmente, el GRSA2-SSP es comparado con PEP-FOLD3, I-TASSER, QUARK y Rosetta en los que GRSA2-SSP muestra un comportamiento bueno en pequeñas instancias (Grupo 1 y 2). Aunque, en grandes instancias (Grupo 3) GRSA2-SSP no es el mejor, en dos grupos de los tres es competitivo.

Los resultados de los algoritmos GRSA-SSP de está investigación fueron publicados en una revista internacional de alto impacto Mathematical and Computational Applications en [Sánchez, 2021].

4.2 Método GRSA2 con estrategias de selección de ruleta.

En este algoritmo nombrado GRSA2-SSPR se presentan los resultados de 5 instancias de péptidos tomadas del conjunto de instancias de 45 péptidos. Se comparan los resultados promedio de GRSA-SSP y GRSA-SSPR con la estrategia de ruleta como se muestra en la Tabla 4.1.

Tabla 4.1 Comparativa de GRSA-SSP vs GRSA2-SSPR (Resultados promedio).

Instances			GRSA2											
			Energy		Time		RMSD		TM-align		Tm-score		GDT-TS	
PDB code	residues	variables	SSP	SSPR	SSP	SSPR	SSP	SSPR	SSP	SSPR	SSP	SSPR	SSP	SSPR
1uao	10	47	-46.4121	**-48.1433**	**3.1360**	5.4949	1.2547	**1.1420**	0.3057	**0.3394**	0.2574	**0.2873**	0.6817	**0.6858**
1dep	15	100	-107.2671	**-140.8803**	**11.0912**	23.1831	1.1577	**1.1427**	0.3944	**0.5719**	0.3469	**0.5433**	0.6389	**0.9006**
1pef	18	124	-122.2355	**-122.4443**	**27.2618**	41.3718	0.6140	**0.4377**	0.6272	**0.6445**	0.5884	**0.5972**	0.9245	**0.9477**
1by0	27	193	-291.9536	**-293.9406**	**64.2842**	83.4278	**1.5843**	1.6147	0.4023	**0.4763**	0.3909	**0.4469**	0.6991	**0.7750**
1yiu	37	206	-245.3790	**-251.8692**	**89.8416**	112.9834	**3.1770**	3.2847	0.2175	**0.2270**	0.1695	**0.1772**	0.3097	**0.3288**

En la Tabla 4.1 se pueden observar 5 instancias con su código de PDB, sus correspondientes residuos y variables (ángulos de torsión). Se compara la energía, el tiempo de ejecución y las métricas estructurales RMSD, TM-align, TM-score y GDT-TS. Estos resultados son el promedio de 30 ejecuciones para cada instancia por algoritmo. Se obtiene una mejora de energía y un mayor consumo de tiempo debido al implemento de la ruleta. En métricas estructurales se obtiene una mejora en TM-align, TM-score y GDT-TS para el método GRSA2-SSPR.

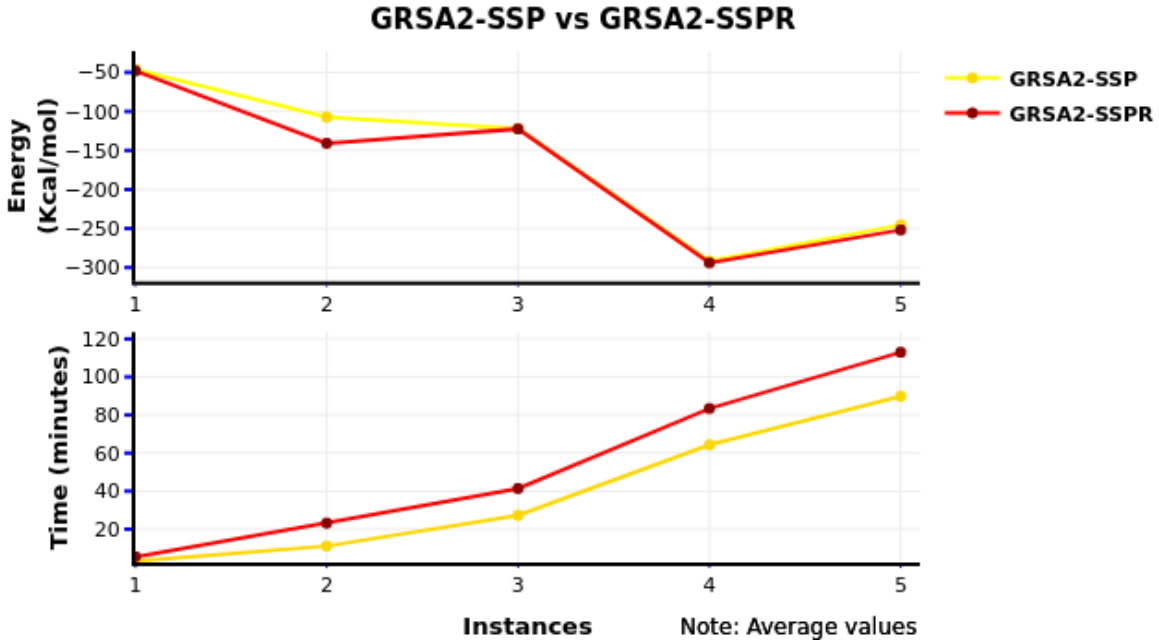


Figura 4.14 Comparativa de GRSA2-SSP y GRSA2-SSPR en energía y tiempo.

En la Figura 4.14 se muestra el comportamiento de la energía y el tiempo para GRSA2-SSP y GRSA2-SSPR para las 5 instancias donde se tiene un coste computacional mayor pero un mejoramiento en energía.

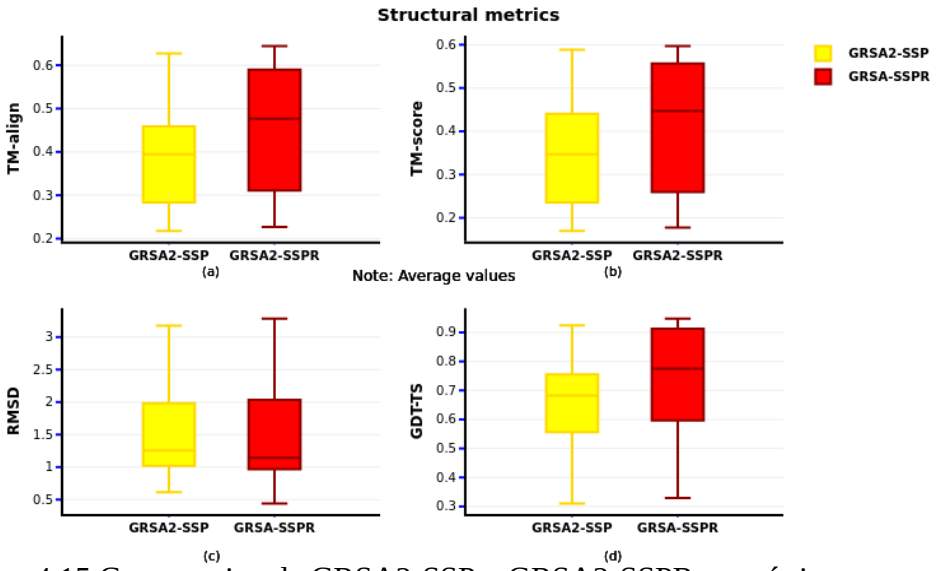


Figura 4.15 Comparativa de GRSA2-SSP y GRSA2-SSPR en métricas estructurales.

En la Figura 4.15 se muestra el desempeño de GRSA2-SSP y GRSA2-SSPR con sus métricas estructurales (RMSD, TM-align, TM-score y GDT-TS) . El GRSA2-SSPR obtiene un mejor comportamiento en las métricas de TM-align, TM-score y GDT-TS como se puede apreciar en los diagramas de cajas.

En la Figura 4.16 se aprecia el comportamiento de GRSA2-SSPR con los algoritmos del estado del arte, en el cual se compara con las métricas estructurales RMSD, TM-align, TM-score, GDT-TS donde se observa un comportamiento similar a los servidores PEP-FOLD3 e I-TASSER. La comparativa con los servidores se realizó con los promedios de los 5 mejores valores para cada algoritmo del estado del arte y el GRSA2-SSPR.

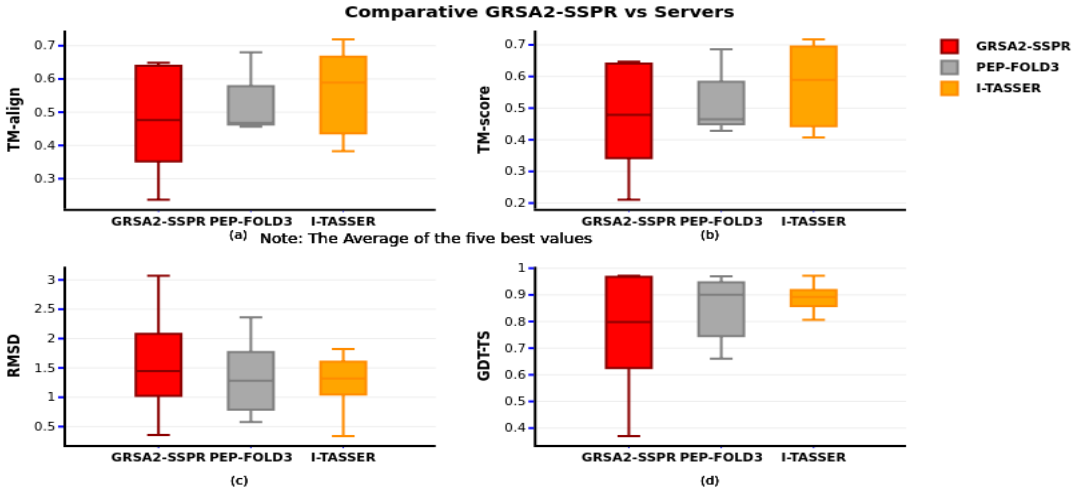


Figura 4.16 Comparativa de GRSA2-SSPR y servidores en métricas estructurales.

4.3 Método GRSA2 con FCNN.

En este método llevamos a cabo la experimentación con la metodología propuesta nombrada GRSA2-FCNN y la comparamos con I-Tasser, Quark, Rosetta, PEP-FOLD3, AlphaFold2 y GRSA2-SSP. Las instancias (péptidos) que utilizamos en este experimento tienen una longitud que varía de 9 a 49 aa's en su estructura primaria. En consecuencia, al variar el número de aa's, también varía el número de ángulos de torsión. Concretamente, el número de ángulos de torsión está dentro del rango [47, 304] para cada instancia de péptido. La Tabla 4.2 muestra el conjunto de datos de péptidos que utilizamos para GRSA2-FCNN y algoritmos del estado del arte. Este conjunto contiene 60 instancias que están representadas con el código PDB y ordenadas por el número de aa's; estas instancias según su SS se clasifican en alfa (estructuras mayoritariamente alfa), beta (estructuras mayoritariamente beta), y ninguna (estructuras sin mayoría alfa o beta).

Tabla 4.2 Conjunto de 60 péptidos.

N°	PDB-CODE	N° AA'S	VAR.	TYPE SS	EXP	N°	PDB-CODE	N° AA'S	VAR.	TYPE SS	EXP
1	1egs	9	49	none	NMR	**31**	1t0c	31	163	none	NMR
2	1uao	10	47	beta	NMR	**32**	2gdl	31	201	alpha	NMR
3	1l3q	12	62	none	NMR	**33**	2l0g	32	183	alpha	NMR
4	2evq	12	66	beta	NMR	**34**	2bn6	33	200	alpha	NMR
5	1le1	12	69	beta	NMR	**35**	2kya	34	210	alpha	NMR
6	1in3	12	74	alpha	NMR	**36**	1wr3	36	197	beta	NMR
7	1eg4	13	61	none	X-ray	**37**	1wr4	36	206	beta	NMR
8	1rnu	13	81	alpha	X-ray	**38**	1e0m	37	206	beta	NMR
9	1lcx	13	81	none	NMR	**39**	1yiu	37	212	beta	NMR
10	3bu3	14	74	none	X-ray	**40**	1e0l	37	221	beta	NMR
11	1gjf	14	79	alpha	NMR	**41**	1bhi	38	216	none	NMR
12	1k43	14	84	beta	NMR	**42**	1jrj	39	208	beta	NMR
13	1a13	14	85	none	NMR	**43**	1i6c	39	218	alpha	NMR
14	1dep	15	94	alpha	NMR	**44**	1bwx	39	242	alpha	NMR
15	2bta	15	100	none	NMR	**45**	2ysh	40	213	beta	NMR
16	1nkf	16	86	alpha	NMR	**46**	1wr7	41	222	beta	NMR
17	1le3	16	91	beta	NMR	**47**	1k1v	41	279	alpha	NMR
18	1pgbF	16	93	beta	X-ray	**48**	2dmv	43	229	alpha	NMR

19	1niz	16	97	beta	NMR	**49**	1res	43	268	beta	NMR
20	1e0q	17	109	beta	NMR	**50**	2p81	44	295	alpha	NMR
21	1wbr	17	120	none	NMR	**51**	1ed7	45	247	beta	NMR
22	1rpv	17	124	alpha	NMR	**52**	1f4i	45	276	alpha	NMR
23	1b03	18	109	beta	NMR	**53**	2l4j	46	250	beta	NMR
24	1pef	18	124	alpha	X-ray	**54**	1qhk	47	272	alpha	NMR
25	1l2y	20	100	alpha	NMR	**55**	1dv0	47	279	alpha	NMR
26	1du1	20	134	alpha	NMR	**56**	1pgy	47	304	none	NMR
27	1pei	22	143	alpha	NMR	**57**	1e0g	48	294	none	NMR
28	1wz4	23	123	alpha	NMR	**58**	1ify	49	290	none	NMR
29	1yyb	27	160	alpha	NMR	**59**	1nd9	49	303	alpha	NMR
30	1by0	27	193	alpha	NMR	**60**	2hep	85	535	alpha	NMR

Nota: Las filas de la tabla están ordenadas según el número de aa. Var (variables) y Exp (método experimental).

GRSA2-FCNN se evaluó procesando cada instancia treinta veces. Se utilizó el paquete de software SMMP [Eisenmeger, 2001] para calcular una estructura de proteína con la función de energía (ECEPP/2). Los parámetros de temperatura inicial y final para cada instancia se determinaron mediante un método de ajuste analítico [Frausto, 2007]. Los algoritmos de la metodología propuesta GRSA2-FCNN se ejecutaron en el cluster Ehecatl en el TecNM/IT Ciudad Madero, y sus características son procesador Intel ® Xeon ® a 2.30 GHz, memoria: 64 GB (4 × 16 GB) ddr4-2133, sistema operativo Linux CentOS, lenguajes de programación FORTRAN y Python.

Para evaluar nuestra metodología utilizamos las métricas TM-score [Zhang, 2004] y Global Distance Test-Total Score (GDT-TS) [Zemla, 2001]. El TM-score y el GDT-TS son métricas utilizadas por la comunidad científica para evaluar la calidad estructural. El TM-score tiene un rango de valores [de 0 a 1] para medir la similitud entre dos estructuras proteicas. Los valores superiores a 0,5 y cercanos a 1 en el TM-score indican una alta similitud estructural, mientras que los valores inferiores a 0,5 indican una baja similitud estructural. El GDT-TS también se utiliza para evaluar la similitud entre una estructura proteica predicha y una estructura de referencia. El valor va de 0 (una predicción sin sentido) a 1 (una predicción perfecta). Para el caso del RMSD entre más cercano sea el valor a cero mejor será la similitud de la estructura con respecto a la nativa.

En primer lugar, en la Figura 4.17 mostramos el comportamiento de GRSA2-FCNN por tipo de estructura secundaria principal alfa, beta y ninguna. En la estructura secundaria none, no hay ningún caso en el que alfa o beta tengan una mayoría significativa. GRSA2-FCNN obtuvo mejores resultados para el caso de los péptidos con más estructuras alfa que tienen valores altos en TM-score y GDT-TS. Por el contrario, los péptidos con mayoría de estructuras beta tienen los valores más bajos de TM-score y GDT-TS.

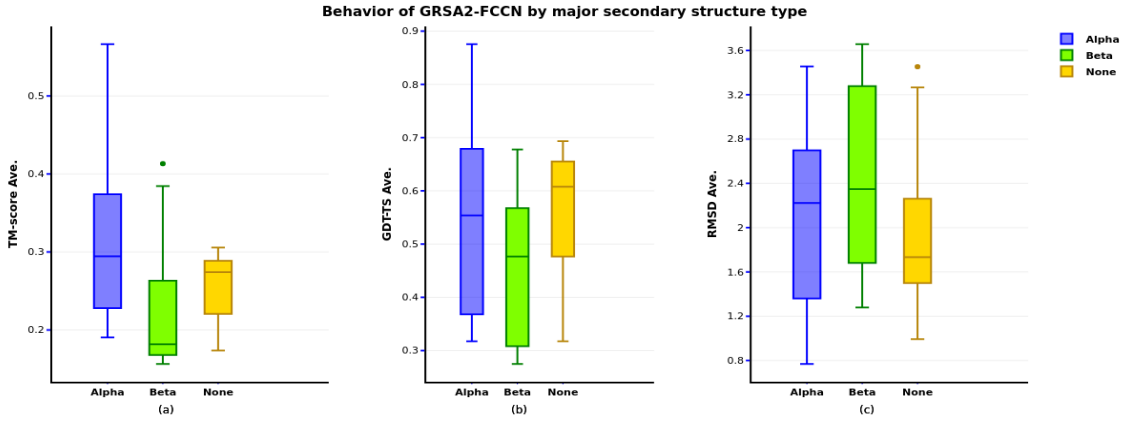


Figura 4.17 Gráficas de comportamiento de GRSA2-FCNN con la mayoría del tipo de estructura secundaria en TM-score, GDT-TS y RMSD, evaluando los cinco mejores resultados para cada instancia : (a) TM-score; (b) GDT-TS (c) RMSD para las instancias tipo Alfa, Beta y None respectivamente.

Las figuras 4.18 a 4.21 muestran los resultados de GRSA2-FCNN en comparación con los algoritmos del estado del arte que se ejecutaron en sus servidores. Las instancias están numeradas de {1} a {60} ordenadas por cantidad de aminoácidos que van de 9 a 49 aa's y divididas en 4 grupos y separadas en 3 subgrupos de 5 instancias para cada figura: hasta 15 (grupo 1), de 16 a 30 (grupo 2), 31 a 40 (grupo 3) y más de 40 (grupo 4) de acuerdo al número de aa's. En cada instancia, cada algoritmo se etiqueta con un color y el que tiene el mejor resultado para cada instancia y su respectiva métrica se etiqueta con una letra W que representa el método ganador para el grupo. Para la métrica TM-score, presentamos la media de las cinco mejores puntuaciones de cada algoritmo y la media de las puntuaciones correspondientes en GDT-TS, así como el RMSD. Para cada algoritmo, realizamos un conteo W para determinar el ganador más frecuente.

En la Figura 4.18, mostramos los resultados obtenidos para los péptidos más pequeños (hasta 15 aa's), donde AlphaFold2 obtuvo diecisiete W's (cuatro en TM-score,

nueve en GDT-TS y cuatro en RMSD), mientras que para I-TASSER hay cuatro W (dos en GDT-TS y dos en RMDS), GRSA2- SSP obtuvo siete W (tres en GDT-TS y cuatro en RMSD), y PEP -FOLD3 obtuvo cinco W (una en TM-score y cuatro en RMSD). GRSA2-FCNN, obtuvo trece W (diez en TM-score, una en GDT-TS y dos en RMSD). Los tres mejores algoritmos para instancias con hasta 15 aa fueron AlphaFold2, GRSA2-FCNN y GRSA2-SSP.

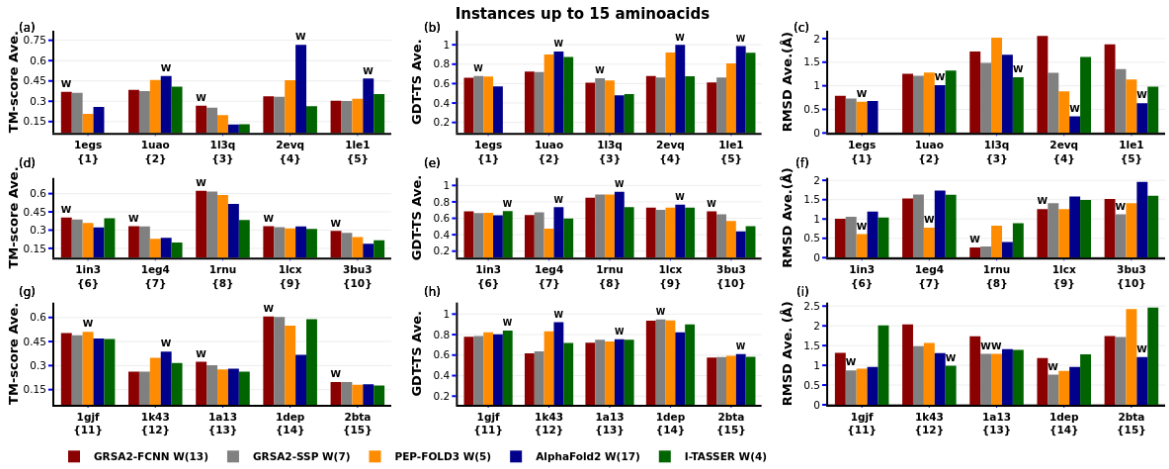


Figura 4.18 Comparación GRSA2-FCNN frente a I-TASSER, AlphaFold2, PEP-FOLD3 y GRSA2-SSP (hasta 15 aa's): Las Figuras (a, d, y g) muestran la media de las cinco mejores predicciones TM-score; las Figuras (b, e, y h) su correspondiente GDT-TS para cada instancia; y las Figuras (c, f, e i) muestran el RMSD.

La figura 4.19 muestra los resultados obtenidos para péptidos de longitudes comprendidas entre dieciséis y treinta aa. AlphaFold2 obtuvo quince W (tres en TM-score, siete en GDT-TS y cinco en RMDS), I-TASSER tuvo doce W (cuatro en TM-score, cinco en GDT-TS y tres en RMSD), GRSA2- SSP obtuvo dos W (una en GDT-TS y otra en RMSD), QUARK obtuvo sólo dos W (una en TM-score y otra en RMSD), y PEP -FOLD3 obtuvo cuatro W (todas ellas en RMSD). Por el contrario, la metodología GRSA2-FCNN obtuvo doce W (siete en TM-score, cuatro en GDT-TS y una en RMSD). Por lo tanto, para este caso, el rendimiento de GRSA2-FCNN es mejor que el de todas las alternativas cuando se utiliza TM-score para la comparación.

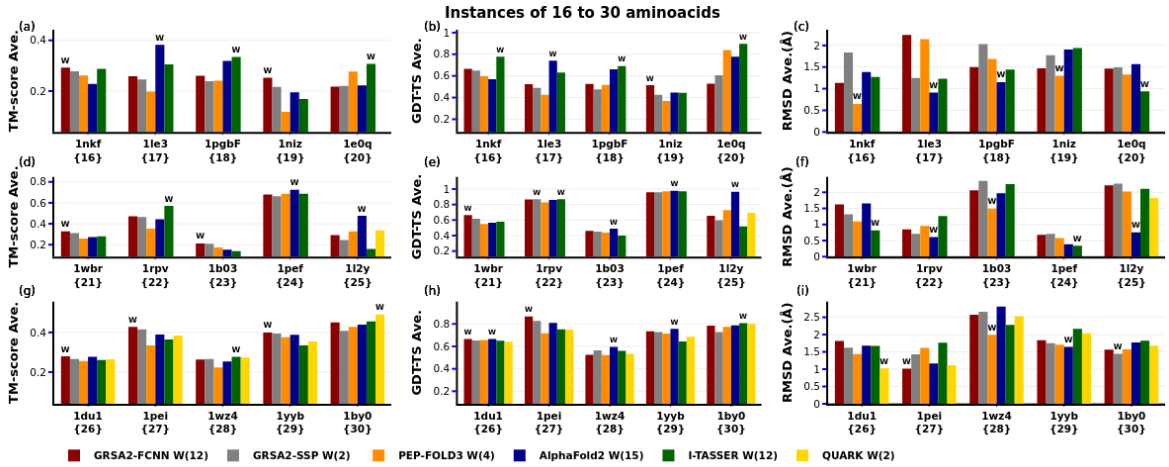


Figura 4.19 Comparación de GRSA2-FCNN frente a I-TASSER, AlphaFold2, QUARK, PEP-FOLD3 y GRSA2-SSP (Instancias de 16 a 30 aa's): Las figuras (a, d y g) muestran la media de las cinco mejores predicciones TM-score. Las Figuras (b, e, y h) muestran las correspondientes GDT-TS para cada instancia, y las Figuras (c, f, e i) muestran los resultados RMSD.

La figura 4.20 presenta los resultados para péptidos de treinta y uno a cuarenta aa. En estos resultados se agregan un algoritmo más Top Model [Mulnaes, 2020]. AlphaFold2 obtuvo doce W (cinco en TM-score, seis en GDT-TS y uno en RMSD), I-TASSER fue el mejor en siete W (tres en TM-score, tres en GDT-TS y uno en RMSD), TopModel obtuvo trece W (cuatro en TM-score, cuatro en GDT-TS y cinco en RMSD), Rosetta obtuvo nueve W (tres en TM-score, dos en GDT-TS y cuatro en RMSD), GRSA2-SSP no tuvo W, QUARK tuvo dos W en RMSD y PEP-FOLD3 una W en RMSD. En este caso, GRSA2-FCNN obtuvo una W en RMSD. En esta prueba, TopModel fue el mejor método en todas las métricas. El rendimiento de GRSA2-FCNN no fue bueno con este conjunto de instancias.

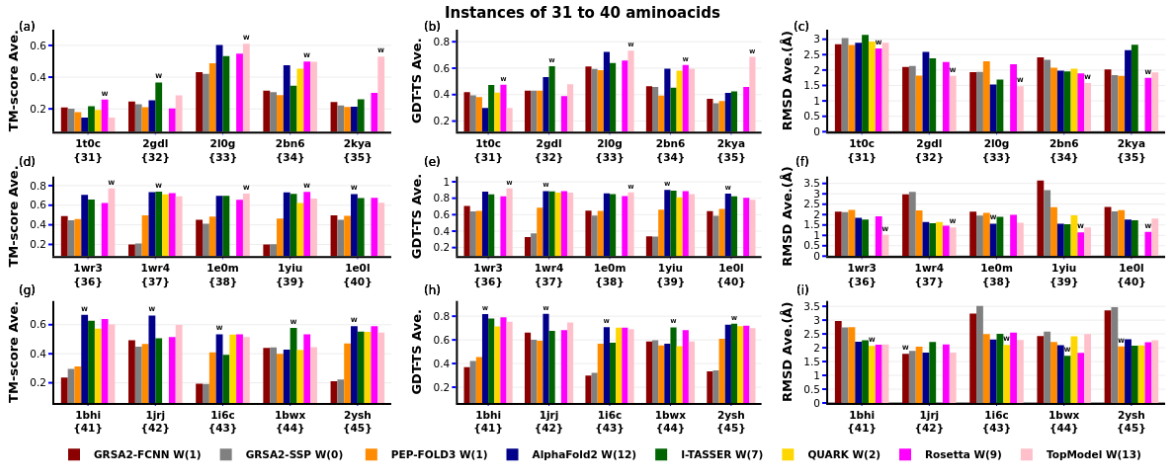


Figura 4.20 Comparación de GRSA2-FCNN frente a I-TASSER, AlphaFold2, Rosetta, QUARK, PEP-FOLD3, TopModel y GRSA2-SSP (de 31 a 40 aa's): Las figuras (a, d y g) muestran la media de las cinco mejores predicciones con TM-score; las figuras (b, e y h) presentan su correspondiente métrica GDT-TS para cada instancia; las figuras (c, f e i) muestran los resultados RMSD.

La figura 4.21 presenta los resultados para péptidos de más de cuarenta aa. AlphaFold2 obtuvo dieciséis W (cinco en TM-score, cinco en GDT-TS y seis en RMSD), I-TASSER fue el mejor con diez W (tres en TM-score, cuatro en GDT-TS y tres en RMSD), TopModel obtuvo tres W (una en GDT-TS y dos en RMSD), Rosetta obtuvo ocho W (tres en TM-score, dos en GDT-TS y tres en RMSD), GRSA2-SSP obtuvo una W en TM-score, QUARK una en RMSD y PEP-FOLD3 ninguna. GRSA2-FCNN obtuvo seis W (tres en TM-score y tres en GDT-TS), y AlphaFold2 fue el mejor método en todas las métricas. El rendimiento de GRSA2-FCNN fue mejor que el de TopModel, QUARK, PEP-FOLD3 y GRSA2-SSP.

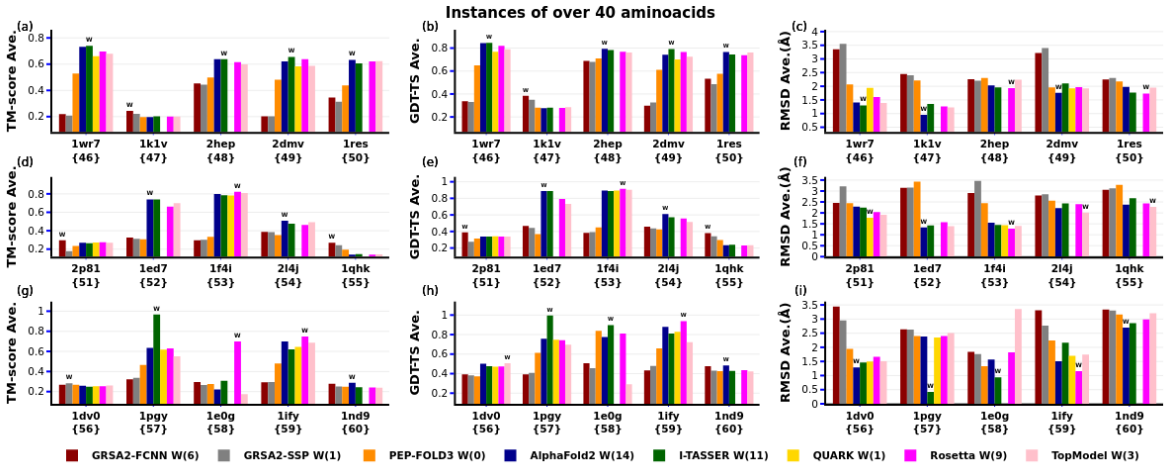


Figura 4.21 Comparación de GRSA2-FCNN con I-TASSER, AlphaFold2, Rosetta, QUARK, PEP-FOLD3, TopModel y GRSA2-SSP (sobre 40 aa's): Las figuras (a, d y g) muestran la media de las cinco mejores predicciones con TM-score; las figuras (b, e y h) presentan su correspondiente métrica GDT-TS; y las figuras (c, f e i) muestran los resultados RMSD.

Además, realizamos una comparación entre AlphaFold2, I-TASSER y nuestro método propuesto GRSA2-FCNN según el tipo de estructura secundaria principal de los péptidos, considerando estructuras mayoritarias alfa, beta y ninguna. Estos resultados se muestran en la Figura 4.22 para las primeras 30 instancias y para las instancias de la 31 a 60 se presentan en la Figura 4.23. En donde se puede ver el comportamiento de GRSA2-FCNN obteniendo buenos resultados en las estructuras alfa y ninguna; sin embargo, está algo limitado en las estructuras beta.

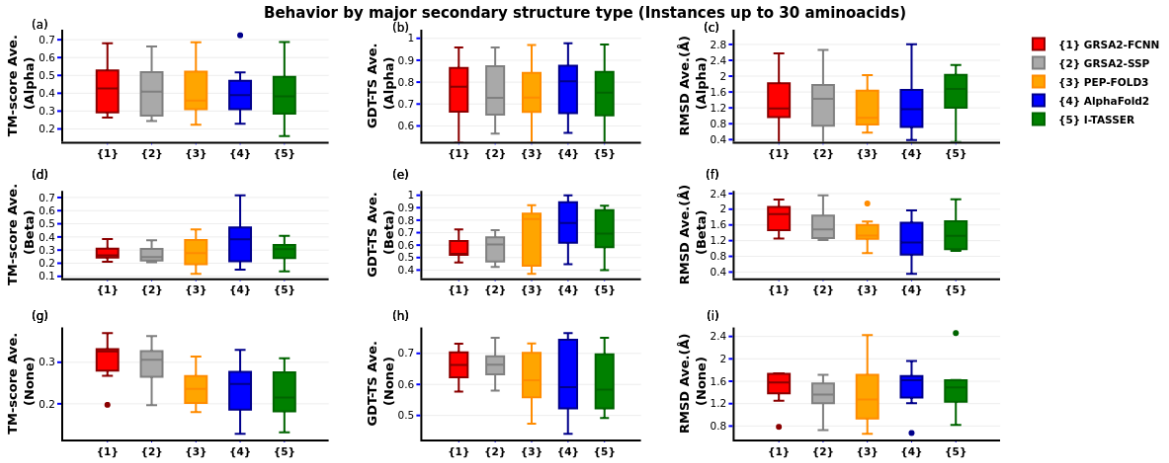


Figura 4.22 Comparación por tipo de estructura secundaria principal de GRSA2-FCNN frente a AlphaFold2, I-TASSER, PEP-FOLD3 y GRSA2-SSP con TM-score y GDT-TS. Las Figuras (a, d, y g) muestran el conjunto de tipo Alpha, Beta, y None evaluado con TM-score (media de las cinco mejores predicciones para cada péptido); las Figuras (b, e, y h) su correspondiente GDT-TS en Alpha, Beta, y None; y las Figuras (c, f, e i) los resultados RMSD en Alpha, Beta, y None

En el segundo grupo sobre 30 aa's (Figura 4.23), realizamos una comparación entre AlphaFold2, I-TASSER, PEP-FOLD3, GRSA2-SSP, Rosetta, QUARK, TopModel, y nuestro método propuesto GRSA2-FCNN. En esta comparación, nuestro método no tuvo el mejor rendimiento en este conjunto de instancias cuando se compararon la estructura secundaria Alpha, Beta y None.

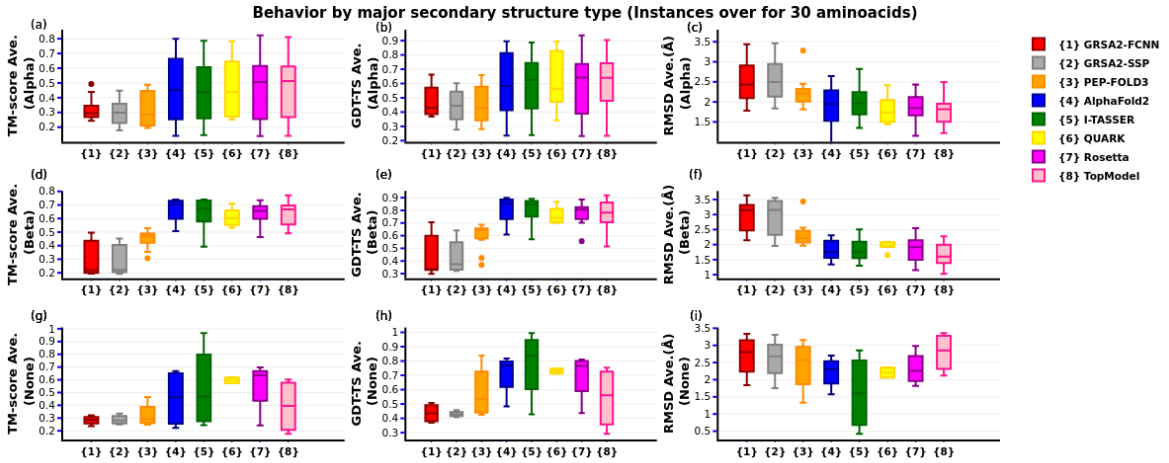


Figura 4.23 Comparación por tipo de estructura secundaria principal de GRSA2-FCNN frente a AlphaFold2, I-TASSER, PEP-FOLD3 y GRSA2-SSP con TM-score y GDT-TS. Las Figuras (a, d, y g) muestran el conjunto de tipo Alpha, Beta, y None evaluado con TM-score (media de las cinco mejores predicciones para cada péptido); las Figuras (b, e, y h) su correspondiente GDT-TS en Alpha, Beta, y None; y las Figuras (c, f, e i) los resultados RMSD en Alpha, Beta, y None.

Para analizar el rendimiento de nuestro algoritmo para cada estructura secundaria, consideramos la longitud de los péptidos, medimos la correlación del conjunto de péptidos en cada estructura y realizamos pruebas de hipótesis tomando la TM-score como métrica principal. En la Figura 4.24, presentamos el rendimiento de nuestro algoritmo GRSA2-FCNN frente a la longitud de cada péptido agrupado por estructura secundaria alfa, beta y ninguna. La figura 4.24a muestra, para la estructura secundaria alfa, que la calidad alcanzada por este algoritmo disminuye con la longitud del péptido para el conjunto de datos; la tendencia mostrada en esta figura es negativa, lo que explica en cierto modo por qué los resultados son más precisos para las estructuras alfa a medida que los péptidos son más pequeños. Las figuras 4.24b y 4.24c muestran que no existe una tendencia clara para las estructuras secundarias beta y nono. La correlación obtenida para las tres estructuras entre la métrica de calidad frente a la longitud de los péptidos fue de -0,5156, 0,0770 y -0,04057; estos valores confirman que los resultados obtenidos por el algoritmo propuesto tienen una tendencia sólo para péptidos pequeños.

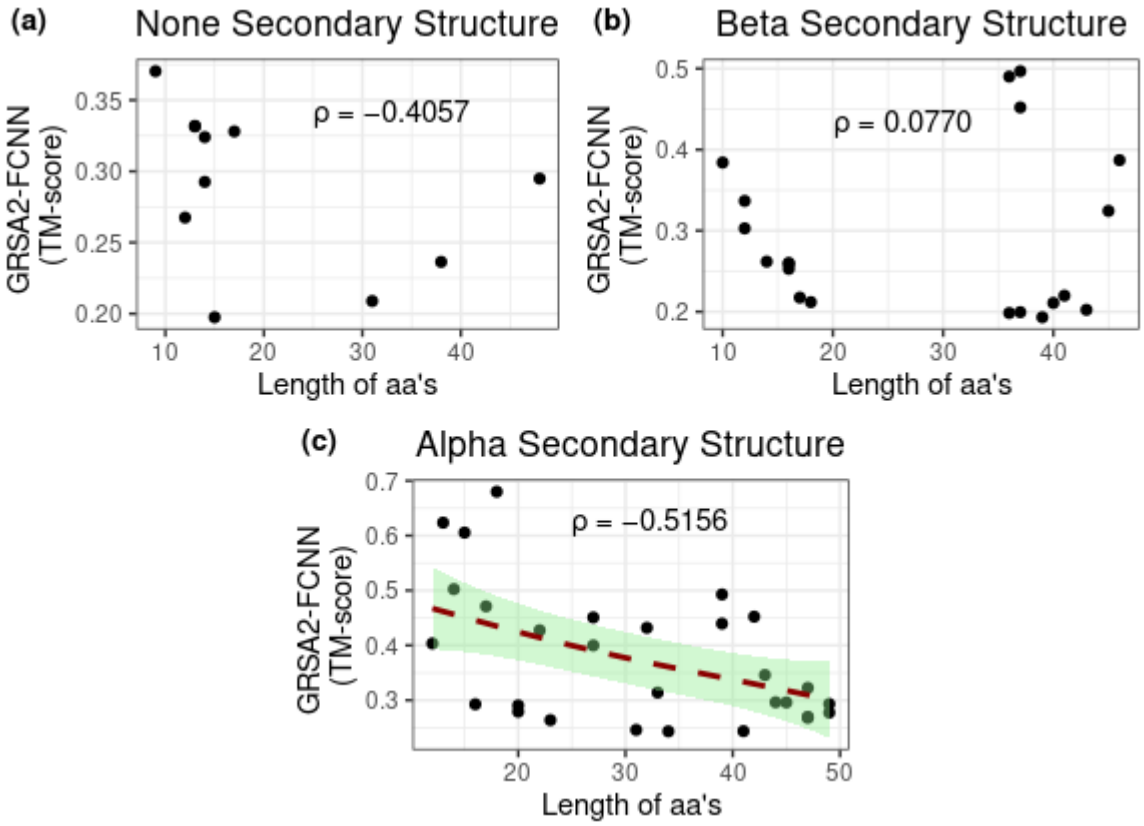


Figure 4.24 Rendimiento de la estructura secundaria frente a la longitud de los péptidos, a) Estructuras alfa, b) Estructuras beta y c) Ninguna estructura.

Para comparar el rendimiento de nuestro algoritmo en cada grupo por estructura secundaria, se realizó una prueba no paramétrica de rangos con signo de Wilcoxon con un valor crítico de 0,05 sobre el p-value. Para las comparaciones, se estableció un ranking de algoritmos según el número de veces que se obtuvo la mejor puntuación de TM-score (Tabla 4.3). En el grupo 1, el algoritmo propuesto tiene en promedio un mejor resultado que AlphaFold2, estos dos algoritmos se compararon estableciendo la siguiente hipótesis: H0: $\mu_1 = \mu_2$ donde μ_1 y μ_2 son las medias para los algoritmos GRSA2-FCNN y AlphaFold3, respectivamente. Del mismo modo, para el grupo dos, el algoritmo propuesto se compara con el segundo clasificado estableciendo la misma hipótesis nula. En los grupos tercero y cuarto, en los que el algoritmo propuesto ocupa los puestos 5 y 4, respectivamente, el algoritmo propuesto se compara con el siguiente mejor clasificado, es decir, I-TASSER y Rosetta.

Tabla 4.3 Posiciones de algoritmos por cada grupo.

Grupo 1	Grupo 2	Grupo 3	Grupo 4
1°GRAS2-FCNN	1°GRAS2-FCNN	1° AlphaFold2	1° AlphaFold2
2° AlphaFold2	2° I-TASSER	2° TopModel	2° I-TASSER
3° PEP-FOLD3	3° AlphaFold2	3° Rosetta	3° Rosetta
4° I-TASSER	4° QUARK	4° I-TASSER	4° GRAS2-FCNN
5° GRSA2-SSP	5° PEP-FOLD3	5° GRAS2-FCNN	5° GRAS2-SSP

En la Figura 4.25, se muestran los gráficos de caja obtenidos con la prueba de hipótesis en las estructuras alfa y beta. En el caso de las estructuras alfa el resultado para las estructuras alfa el algoritmo propuesto es el siguiente: en los grupos, 1, 2, 3, y 4, se obtuvieron los puestos 1, 2, 5, y 4 para el algoritmo propuesto; es decir, fue el mejor en el primer grupo y el peor en el tercero. Además, observamos que a medida que los grupos son más pequeños el algoritmo propuesto tiene mejor rendimiento.

En el caso de las estructuras beta, el tamaño no tiene un impacto importante como se ha comentado anteriormente. En el grupo 1, GRSA2-FCNN compite con AlphaFold2, que obtiene mejores resultados. En el grupo 2, compite con I-TASSER; como se muestra en los gráficos de caja, I-TASSER obtiene mejores resultados. En los grupos 3 y 4, el algoritmo propuesto compite contra I-TASSER y Rosetta, respectivamente; el resultado es que en estos dos grupos el algoritmo tiene un rendimiento pobre. En consecuencia, en estos dos últimos grupos, el algoritmo propuesto debería clasificarse en 5º y 4º lugar. Las estructuras None, no pudieron evaluarse porque el número de muestras era demasiado pequeño; entonces no se obtuvieron los boxplots para las estructuras None.

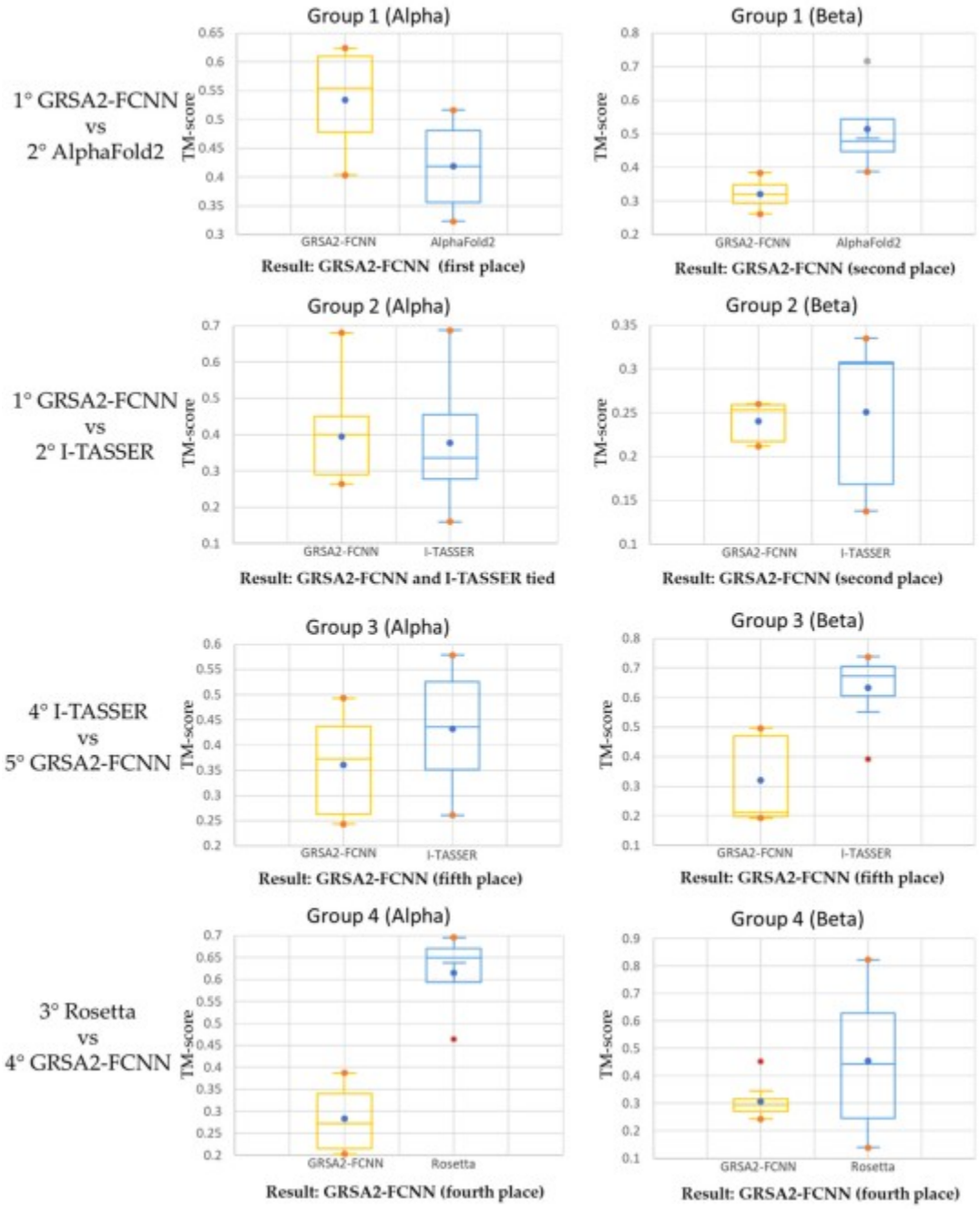


Figure 4.25 Gráficos de caja y p-value para las estructuras alfa y beta.

Los sesenta péptidos del conjunto de datos evaluados para el GRSA2-FCNN muestran un rendimiento similar al I-TASSER, AlphaFold2 y GRSA2-SSP para hasta 30 aa's. Los fragmentos generados por la CNN mejoraron significativamente el modelo inicial. Además, el refinamiento del modelo mejora la predicción final del péptido. En el caso de los péptidos más grandes del conjunto de datos de más de 31 aa's, GRSA2-FCNN no tiene

el mejor rendimiento cuando la comparación es por estructura secundaria Beta y Ninguna. Sin embargo, en el caso de la estructura Alfa, nuestro método es competitivo en la comparación de los resultados obtenidos por I-TASSER, AlphaFold2, Rosetta y TopModel con el conjunto de instancias propuesto en este trabajo. Los resultados que incluyen los de la metodología GRSA2-FCNN fueron publicados en “A Peptides Prediction Methodology with Fragments and CNN for Tertiary Structure Based on GRSA2” en [Sánchez, 2022]

Capítulo 5

5 Conclusiones y Recomendaciones

Los resultados de las predicciones se comparan para observar la calidad de los resultados mediante el RMSD, TM-score y GDT-TS. En la familia de algoritmos de GRSA y la secuencia de aminoácidos se obtuvieron mejores resultados con el método GRSA2 en comparación con el SA, GRSA y EGRSA. Sin embargo el GRSA2 al compararse con los algoritmos del estado del arte PEP-FOLD3, I-TASSER, Rosetta y QUARK, las predicción de las estructuras tridimensionales de estos algoritmos son tanto en RMSD y TM-score aún mejores que el algoritmo GRSA2. En el caso de los algoritmos de GRSA con la secuencia de aminoácidos y predicción de la estructura secundaria, también el GRSA2-SSP obtuvo mejores resultados en comparación con los demás algoritmos de la familia GRSA. El GRSA2-SSP produce resultados muy buenos para péptidos y son tan buenos como PEP-FOLD3, I-TASSER, QUARK y Rosetta para el caso de pequeños y medianos (≤ 30 aminoácidos) péptidos de acuerdo a la experimentación realizada en el conjunto de 45 instancias. Podemos concluir que la metodología GRSA-SSP mejora a los algoritmos GRSA, y con gran relevancia el GRSA2-SSP obtiene resultados en péptidos tan buenos como los mejores algoritmos del estado del arte.

Como se pudo observar conforme se va integrando la metodología general anexar información además de la secuencia de aminoácidos ayuda a mejorar la predicción de una estructura tridimensional de una proteína o péptido. Con la incorporación de los ensambles de fragmentos al método general se han presentado mejoras de los resultados de las predicciones por lo que se está complementando la experimentación con el ensamble de fragmentos para una comparación más robusta con los algoritmos del estado del arte.

El GRSA2-SSPR obtiene un comportamiento mejor que el GRSA2-SSP, por lo que las estrategias de refinamiento en cadenas laterales ayudan a mejorar la estructura tridimensional de una proteína o péptido.

El GRSA2 con fragmentos presenta una mejora en comparación con el GRSA2-SSP, esto debido que al incorporar fragmentos en la construcción del modelo se obtiene un mejor modelo que al ser refinado genera una predicción mejor.

En el caso de la metodología GRSA2-FCNN para la predicción de estructuras peptídicas tridimensionales que incluye el Recocido Simulado de Relación Dorada y las Redes Neuronales Convolucionales. GRSA2-FCNN se compara con los métodos de vanguardia I-TASSER, Rosetta, AlphaFold2, PEP-FOLD3, QUARK y GRSA2-SSP. Probamos el rendimiento de GRSA2-FCNN con un conjunto de 45 péptidos.

La evaluación y comparación de los resultados de GRSA2-FCNN y los algoritmos de última generación se basaron en las métricas TM-score y GDT-TS para 60 péptidos. El conjunto de datos de péptidos se dividió en 4 grupos de acuerdo al tamaño de aminoácidos, y se analizaron los resultados de cada instancia. La evaluación muestra que GRSA2-FCNN se comporta muy bien para hasta 30 aa's en comparación con el estado del arte. Para el grupo 1 (hasta 15 aa’s), encontramos que GRSA2-FCNN se desempeña mejor en TM-score y AlphaFold2 en GDT-TS, mientras que en el grupo 2 (16 a 30 aa’s), GRSA2-FCNN se desempeña mejor en TM-score y AlphaFold2 en GDT-TS. En los últimos grupos 3 y 4 con instancias mayores de 30 aa’s, I-TASSER y Rosetta obtienen los mejores resultados en TM-score e I-TASSER en GDT-TS.

Además, comparamos GRSA2-FCNN según el tipo de estructura secundaria con I-TASSER, y AlphaFold2, ya que fueron los mejores métodos, reportados en los eventos CASP. El rendimiento de GRSA2-FCNN en cuanto al tipo de estructura secundaria muestra buenos resultados para las predicciones de péptidos de tipo mayoritariamente alfa y los que

no son ni alfa ni betas, para el caso de los que son mayoritariamente betas hay resultados limitados en comparación con el AlphaFold2 e I-TASSER.

Analizamos los resultados obtenidos por GRSA2-FCNN en comparación con las otras alternativas y concluimos que GRSA2-FCNN supera a PEP-FOLD3, QUARK y GRSA2-SSP. La metodología propuesta consigue muy buenos resultados y establece un nuevo estado del arte para péptidos de hasta treinta aa's en TM-score y algunos casos en GDT-TS. En conclusión, encontramos que nuestra metodología; además, supera a la mayoría de las mejores metodologías de PFP en el caso de los péptidos.

Al realizar diferentes métodos o metodologías para la predicción de péptidos o proteínas pequeñas hemos observado que el uso de estrategias como ruleta y algoritmos de aprendizaje profundo como CNN ha contribuido en un mejoramiento importante en la predicción de las estructuras tridimensionales logrando obtener resultados similares y en algunos casos mejores que los del estado del arte.

Bibliografía

Anfinsen, C. (1973). Principles that Govern the Folding of Protein Chains. Science, 181(4096), 223-230.

Alzubaidi, L.; Zhang, J.; Humaidi, A. J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Farhan, L. (2021) Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data. 8(1), 1-74.

Bernstein FC, Koetzle TF, Williams GJ, Meyer EE Jr, Brice MD, Rodgers JR et al. (1977) "The Protein Data Bank: A Computer-based Archival File For Macromolecular Structures," J. of. Mol. Biol. 1977, 112: 535.

Cerny V., “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm I,” J. Optim. Theory Appl., vol. 45, no. 41–45, 1985.

Chaudhuri, T., & Paul, S. (2006). Protein-misfolding diseases and chaperone-based theraupetic approaches. FEBS Journal, 273(7), 1331–1349.

Conway, P., Tyka, M. D., DiMaio, F., Konerding, D. E., & Baker, D. (2014) Relaxation of backbone bond geometry improves protein energy landscape modeling. Protein Science, 2014, 23(1), 47-55.

Crick, F. (8 de Agosto de 1970). Central Dogma of Molecular Biology. Nature, 227, 561-563.

De Oliveira, S.H.P. et al. (2015) Building a better fragment library for de novo protein structure prediction. PLoS One. 2015, 10, e0123998.

De Oliveira, S. (2017). Supplementary Information: Sequential search leads to faster, more efficient fragment-based de novo protein structure prediction. Bioinformatics, 34(7), 1132-1140.

Dill, K. (1999). Polymer principles and protein folding. Protein science, 8, 1166-1180.

Dorn, M. (2014). Three-Dimensional Protein Structure Prediction: Methods and Computational Strategies. Computational Biology and Chemistry, 53, 251-276.

Dyrmann, M.; Karstoft, H.; Midtiby, H.S. (2016) Plant species classification using deep convolutional neural network. Biosyst. Eng. 2016,151, 72–80.

Encina, G. (2013). BIOLOGÍA MOLECULAR EN ONCOLOGÍA: LO QUE UN CLINICO DEBIERA SABER. Revista Médica Clínica Las Condes, 24(4), 563-570.

Frausto-Solis J., Román E. F., Romero D., Soberon X., and Liñán-García E. (2007), “Analytically Tuned Simulated Annealing Applied to the Protein Folding Problem,” in Computational Science – ICCS 2007, Springer Berlin Heidelberg, pp. 370–377.

Frausto, J. (2015). Golden Ratio Simulated Annealing for Protein Folding Problem. IJCM, 12(6), 1550037 (20 pages).

Frausto, J. (2019) “GRSA Enhanced for Protein Folding Problem in the Case of Peptides”, Axioms., vol. 8, no. 4, pp. 136.

Frausto-Solís, J.; Hernández-González, L. J.; González-Barbosa, J. J.; Sánchez-Hernández, J. P.; Román-Rangel, E. F. (2021) Convolutional Neural Network–Component Transformation (CNN–CT) for Confirmed COVID-19 Cases. Math. Comput. Appl. 2021, 26, 29. https://doi.org/10.3390/mca26020029.

Goodfellow I., Bengio Y., Courville A., Bengio Y. (2016) Deep learning, vol. 1. Cambridge: MIT press.

Hart, W. E. and Istrail, S. (1997) Robust Proofs of NP-Hardness for Protein Folding: General Laticces and Energy Potentials. Journal of Computational Biology, 4(1):1-22.

Horton, R. (2008). Principios de bioquimica (Cuarta ed.). Mexico: Pearson Educación.

Jiang, P., & Xu, J. (2011). RaptorX: Exploiting structure information for protein alignment by statistical inference. Proteins, 79(10), 161-171.

Jinbo, X., Peng, J., & Zhao, F. (2009). Template-based and free modeling by Raptor++ in CASP8. Proteins, 77(9), 133-137.

Joo, K., Joung, I., & Young, S. (2016). Template based protein structure modeling by global optimization in CASP11. Proteins, 84, 221-232.

Kendrew, J. (1958). A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. Nature, 181, 662-666.

Kinch, L. N., Schaeffer, R. D., Kryshtafovych, A., & Grishin, N. V. (2021). Target classification in the 14th round of the critical assessment of protein structure prediction (CASP14). Proteins: Structure, Function, and Bioinformatics, 89(12), 1618-1632.

Kingma, D.P.; Ba, J. Adam, (2015): A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7–9 May 2015.

Kirkpatrick S., C. D. Gelatt, and M. P. Vecchi (1983), “Optimization by Simulated Annealing,” Sci. New Ser., vol. 220, no. 4598, pp. 671–680.

Lam A. Y. S. and Li V. O. K. (2012), “Chemical Reaction Optimization: A tutorial,” Memetic Comput., vol. 4, no. 1, pp. 3–17.

Lamiable, A.; Thévenet, P.; Rey, J.; Vavrusa, M.; Derreumaux, P.; Tufféry, P. (2016), PEP-FOLD3: Faster de Novo Structure Prediction for Linear Peptides in Solution and in Complex. Nucleic Acids Res, 44, W449–W454.

Levinthal, C. (1968). ARE THERE PATHWAYS FOR PROTEIN FOLDING ? Journal de Chimie Physique, 65(1), 44-45.

Maldonado, F., Frausto, J. (2016), Evolutionary GRSA for Protein Structure Prediction, IJCOPI, vol. 7(3) , 75-86.

Maldonado, F., & Frausto, J. (2018). Comparative Study of Computational Strategies for Protein Structure Prediction. En O. Castillo, P. Melin, & J. Kacprzyk, Fuzzy Logic Augmentation

of Neural and Optimization Algorithms: Theoretical Aspects and real applications (Vol. 749, págs. 449-459). Warsaw: Springer International Publishing.

Mckee, T., & Mckee , J. (2014). Bioquimica Las bases moleculares de la vida (Quinta ed.). Mexico: McGRAW-HILL.

Mirdita, M., Steinegger, M. & Söding, J. (2019) MMseqs2 desktop and local web server app for fast, interactive sequence searches. Bioinformatics, 2019, 35, 2856–2858.

Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. (2022) ColabFold: Making protein folding accessible to all. Nature Methods, 2022.

Morales, L.B.; Garduño-Juárez, R.; Romero, D. Applications of Simulated Annealing to the Multiple-Minima Problem in Small Peptides. J. Biomol. Struct. Dyn. 1991, 8, 721–735.

Muhammad K., Ahmad J., Lv Z., Bellavista P., Yang P., Baik S. W., (2019) "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1419-1434, July 2019, doi: 10.1109/TSMC.2018.2830099.

Mulnaes, D.; Porta, N.; Clemens, R.; Apanasenko, I.; Reiners, J.; Gremer, L.; Gohlke, H., (2020). TopModel: Template-based protein structure prediction at low sequence identity using top-down consensus and deep neural networks. J. Chem. Theory Comput. 2020, 16, 1953–1967.

Ngo, T. (1994). Computational Complexity, Protein Structure Prediction, and Levinthal Paradox. The Protein Folding Problem and Tertiary Structure Prediction, 433-506.

Olivarez Quiroz, L., & García Colín, L. (2004). Plegamiento de las proteínas: Un problema interdisciplinario. Revista de la Sociedad Química de México, 48(1), 95-105.

Ovchinnikov, S.; Kim, D.E.; Wang, R.Y.R.; Liu, Y.; DiMaio, F.; Baker, D. (2016) Improved de novo structure prediction in CASP11 by incorporating coevolution information into Rosetta. Proteins, 84 Suppl 1(Suppl 1), 67-75.

Ovchinnikov, S., Park, H., & Baker, D. (2017). Protein structure prediction using Rosetta in CASP12. Proteins, 86, 113-121.

Raman, S., Vernon, R., & Thompson, J. (2009). Structure prediction for CASP8 with all-atom refinement using Rosetta. Proteins, 77(9), 89-99.

Sánchez-Hernández, J. P., Frausto-Solís, J., González-Barbosa, J. J., Soto-Monterrubio, D. A., Maldonado-Nava, F. G., & Castilla-Valdez, G. (2021). A Peptides Prediction Methodology for Tertiary Structure Based on Simulated Annealing. Mathematical and Computational Applications, 26(2), 39.

Sánchez-Hernández, J. P., Frausto-Solís, J., Soto-Monterrubio, D. A., González-Barbosa, J. J., & Roman-Rangel, E. (2022). A Peptides Prediction Methodology with Fragments and CNN for Tertiary Structure Based on GRSA2. Axioms, 11(12), 729.

Santos, J. (2009). PROTEÍNAS Estructuras fascinantes. Argentina: Industria Argentina.

Senior, A.W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A.W.R.; Bridgland, A.; et al. (2019) Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). Proteins Struct. Funct. Bioinform. 2019, 87, 1141–1148.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., ... & Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. Nature, 577(7792), 706-710.

Soto D. A., Sánchez J. P., Frausto J., Gonzalez J. J. (2021). Roulette selection strategies applied to GRSA2-SSP for refinement of the amino acid side chains regions. 9th International Workshop on Numerical and Evolutionary Optimization (NEO 2021), México. https://neo.cinvestav.mx/NEO2021/images/NEO2021HandBook.pdf

Wang, S., Sun, S., & Xu, J. (2017). Analysis of deep learning methods for blind protein contact prediction in CASP12. Proteins, 86, 67-77.

Yang, J.; Roy, A.; Zhang, Y. (2013) BioLiP: A semi-manually curated database for biologically relevant ligand-protein interactions. Nucleic Acids Research, 41: D1096-D1103.

Yang, J.; Yan, R.; Roy, A. (2015). The I-TASSER Suite: Protein structure and function prediction. Nat Methods, 12, 7–8.

Xu, D.; Zhang, Y. (2013). Toward optimal fragment generations for ab initio protein structure assembly. Proteins, 81, 229–239.

Zemla, A.; Moult, J.; Fidelis, K. Processing and evaluation of predictions in CASP4. Proteins 2001, 45, 13–21.

Zhang, Y., & Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. Proteins: Structure, Function, and Bioinformatics, 57(4), 702–710. doi:10.1002/prot.20264.

Zhang, Y. (2009). I-TASSER: Fully automated protein structure prediction in CASP8. Proteins, 77(9), 100-113.

Zhang, W., Yang, J., & He, B. (2016). Integration of QUARK and I-TASSER for Ab Initio Protein Structure Prediction in CASP11. Proteins, 84(1), 76-86.

ZhangLab. (2016). ZhangLab. Obtenido de https://zhanglab.ccmb.med.umich.edu/

Zhang, C., Mortuza, S. M., He, B., Wang, Y., & Zhang, Y. (2018). Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. Proteins: Structure, Function, and Bioinformatics, 86, 136-151.

Zheng, Y. (2013). Interplay of I-TASSER and QUARK for template-based and ab initio protein structure prediction in CASP10. Proteins, 82(2), 175-187.

Zheng, W.; Zhang, C.; Wuyun, Q.; Pearce, R.; Li, Y.; Zhang, Y., (2019). LOMETS2: Improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. Nucleic Acids Research, 47, W429–W436.

Zhou, G., Gen, M., & Wu, T. (1996, October). A new approach to the degree-constrained minimum spanning tree problem using genetic algorithm. In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929) (Vol. 4, pp. 2683-2688). IEEE.

Zhou, H., Pandit, S., & Skolnick, J. (2009). Performance of the Pro-sp3-TASSER server in CASP8. proteins, 77(9).