UNIVERSIDAD DE GRANADA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE COMUNICACIONES

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

PROGRAMA OFICIAL DE POSTGRADO

"DOCTORADO IBEROAMERICANO DE SOFT COMPUTING"

Tesis Doctoral

Predicción de Mapas de Contactos de Proteínas Mediante Multiclasificadores

Cosme Ernesto Santiesteban Toca

Granada 2014

Editorial: Universidad de Granada. Tesis Doctorales
Autor: Cosme Ernesto Santiesteban Toca
ISBN: 978-84-9125-231-3
URI: http://hdl.handle.net/10481/40874

UNIVERSIDAD DE GRANADA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E
INTELIGENCIA ARTIFICIAL
PROGRAMA OFICIAL DE POSTGRADO
"DOCTORADO IBEROAMERICANO DE SOFT COMPUTING"

Tesis Doctoral

Predicción de Mapas de Contactos de Proteínas Mediante Multiclasificadores

MEMORIA DE TESIS QUE PRESENTA

Cosme Ernesto Santiesteban Toca

COMO REQUISITO PARA OPTAR POR EL GRADO DE
DOCTOR EN INFORMÁTICA

DIRECTORES DE TESIS
Dr. Jesús S. Aguilar-Ruiz

Granada 2015

D. Jesús Salvador Aguilar Ruiz, Catedrático de Universidad adscrito al área de Lenguajes y Sistemas Informáticos de la Universidad Pablo de Olavide, de Sevilla.

CERTIFICA QUE:

D. Cosme Ernesto Santiesteban Toca, Máster en Informática en el Instituto Superior Politécnico José Antonio Echeverría en Cuba, ha realizado bajo su supervisión el trabajo de investigación titulado:

PREDICCIÓN DE MAPAS DE CONTACTOS DE PROTEÍNAS MEDIANTE MULTICLASIFICADORES

Una vez revisado, autoriza la presentación del mismo como tesis doctoral en la Universidad de Granada y estima oportuna su presentación al tribunal que habrá de valorarlo. Dicha tesis ha sido realizada dentro del programa de Doctorado Iberoamericano en Soft Computing, de la Universidad de Granada.

Sevilla, noviembre de 2015

El doctorando Cosme Ernesto Santiesteban Toca y el director de la tesis Jesús Salvador Aguilar Ruiz garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección del director de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Abril 2015

Fdo. Cosme Ernesto Santiesteban Toca	Fdo. Jesús Salvador Aguilar Ruiz
Doctorando	Director

A mis padres, hermano, esposa e hijos

Agradecimientos

Es cierto que "*es mejor viajar lleno de esperanza que llegar*". Y es mi primer agradecimiento a la oportunidad que he tenido de realizar este viaje lleno de tantos los momentos inolvidables y de batallas interminables, donde la mayor recompensa la obtuve de comprobar que siempre podré contar con el apoyo incondicional de mis familiares y amigos.

Al primer amigo que he de agradecer será al Dr. Jesús Salvador Aguilar Ruiz. Por su gran capacidad como investigador, su exigencia, pero sobre todo por confiar en mí y brindarme total autonomía y abrir siempre un espacio al debate. Gracias por mostrarme qué y a la vez permitirme decidir cómo.

A mis entrañables amigos y compañeros del Laboratorio de Informática del Centro de Bioplantas. Y a todos los que han hecho suya la ardua tarea que implicó la realización de la experimentación.

Y, de último, los primeros: mi familia. En primer lugar a mis padres por todo su apoyo, su bondad, el ánimo que siempre me dieron y por tanto y tanto cariño. A mi hermano Jorge por ayudarme experimentar e interpretar. A mis hijos, por serme tan pacientes, y a mi esposa por tantos y tantos sacrificios.

A todos muchas gracias!!!

Resumen

La predicción de estructuras de proteínas involucra varios y muy complejos pasos, entre los que se incluye la predicción de mapas de contactos. En los últimos años se han explorado métodos basados en algoritmos de aprendizaje automático, técnicas estadísticas, bioinspiradas, combinación de clasificadores, entre otras, sin lograr aún los niveles de efectividad deseados. El objetivo de la presente investigación es diseñar un algoritmo, a partir de la información brindada por la secuencia de aminoácidos, que posea capacidad explicativa y permita predecir mapas de contactos de proteínas con una efectividad similar o superior a los algoritmos del estado del arte. Éste se sustenta sobre la base del supuesto de que el análisis de la correlación entre la estructura de residuos covalentes de una proteína y su secuencia de aminoácidos permitiría obtener un algoritmo capaz de predecir los mapas de contactos de una proteína, con una precisión aceptable. Como resultado, se obtuvo el multiclasificador FoDT, el cual es capaz de asignar contactos con una efectividad del 55% reduciendo significativamente el costo computacional. FoDT propone una codificación donde se analizan por separado cada una de las 400 parejas de aminoácidos que pueden formarse. Además, implementa un nuevo algoritmo de re-muestreo basado en una estrategia genética, capaz de reducir el nivel de desbalance que existe en la predicción de mapas de contactos. La comparación con algoritmos del estado del arte, empleando proteínas del CASP9 y el CASP10, muestra que no existen diferencias significativas con éstos, sin embargo, es capaz de brindar un mecanismo de interpretación de su base de conocimiento.

Índice General

Índice de figuras

Índice de Tablas

Parte I

Introducción

Capítulo 1

Introducción

1.1.- Motivación

La Biología Teórica actual centra su atención en la investigación de las estructuras básicas de la vida. Una de estas estructuras básicas es el sistema bioquímico que hace posible el flujo de la información genética en los organismos vivos, el código genético.

El código genético es la piedra angular del sistema de información genética. Consecuentemente, es de esperar que toda construcción teórica que intente explicar las relaciones cuantitativas y cualitativas existentes en el sistema de información genética tome como punto de partida el código genético. El desarrollo alcanzado por las Ciencias Biológicas ha permitido la acumulación de mucha información experimental disponible en grandes bases de datos, lo cual ha dado lugar al surgimiento de la Bioinformática.

Bioinformática es una rama interdisciplinaria de las Ciencias de la Computación que estudia sistemas de cómputo y tratamiento de la información para el análisis de datos experimentales (de nivel molecular, principalmente) de sistemas biológicos, así como la simulación de los mismos. Algunas de las principales aplicaciones de la bioinformática son la simulación, la minería de datos (*data mining*), y el análisis de los datos obtenidos en el estudio de moléculas relevantes para la vida, principalmente del ADN/ARN/genoma (Proyecto Genoma Humano) o de las proteínas (cuyo conjunto en un determinado organismo biológico forma su proteoma), así como el diseño y desarrollo de herramientas tales como bases de datos, directorios web, etc.

Éste, es un campo que ha emergido vertiginosamente, influenciado principalmente por los avances en la secuenciación del ADN y las técnicas de mapeo. Uno de los grandes desafíos de la bioinformática es la predicción de estructura, donde se desea

determinar la estructura tridimensional (3D) de una proteína, a partir de su secuencia de aminoácidos [1]–[3].

La organización de una proteína viene definida por cuatro niveles estructurales denominados: estructura primaria, estructura secundaria, estructura terciaria y estructura cuaternaria. Cada una de estas estructuras informa de la disposición de la anterior en el espacio.

Es conocido que las proteínas se pliegan de manera espontánea y reproducen una estructura 3D única, en solución acuosa [4]. Por otra parte, se ha demostrado que los esquemas de predicción de plegamiento basados en el empleo de proteínas solubles parecen ser inapropiados para la predicción de proteínas de membrana, las cuales tienden a plegarse de forma diferente. Esto se debe a que en la secuencia no se encuentra toda la información de la estructura de la proteína, sino que requieren de la presencia de membranas cotemporalmente con el proceso de biosíntesis, para el plegamiento y la inserción [5].

La base de datos de proteínas (PDB)[6], almacena las coordenadas tridimensionales de los átomos de miles de estructuras de proteínas. La mayoría de estas proteínas pueden agruparse en alrededor de 700 familias de plegamientos, basadas en sus similitudes. Y se supone que existen unas 1000 familias[7]. Estas bases de datos ofrecen un nuevo paradigma para la predicción de estructuras de proteínas, mediante el empleo de métodos de minería de datos como la clusterización, clasificación, las reglas de asociación, los modelos ocultos de Markov, etc.

La predicción y el análisis de estructuras de proteínas involucra varios y muy complejos pasos, incluyendo la búsqueda de secuencias homólogas, el alineamiento de múltiples secuencias, modelos comparativos de alineamiento de estructuras, evaluación y medición, visualización tridimensional, entre otros [8]–[17].

La habilidad de hacer predicciones exitosas implica el entendimiento de la relación entre la secuencia y la estructura de la proteína [18], [19]. Los contactos entre los residuos, condicionan el plegamiento de las proteínas y caracterizan las diferentes estructuras de proteínas. La predicción de los contactos entre residuos requiere del

estudio de las distancias entre los residuos de la proteína, relacionado con el par de aminoácidos específico.

1.2.- Planteamiento

En los últimos años, se han desarrollado múltiples métodos destinados a la predicción de mapas de contacto. Los primeros esfuerzos partieron de procesar la distribución de las distancias de los residuos de un par en proteínas de estructura 3D conocida con el objetivo de abordar los problemas de plegamiento. Recientemente, la clasificación de los contactos residuales ha sido relacionada con patrones estructurales. Otros ejemplos más específicamente apuntan al tema de la predicción de los contactos residuales usando la información derivada de la ocurrencia de mutaciones correlacionadas en proteínas similares.

Han sido desarrollado métodos que combinan las mutaciones correlacionadas con la información estadística derivada de las bases de datos de estructuras de proteínas conocidas y otras propiedades. Algunas de estas propiedades son: la conservación de secuencias a partir del cálculo de múltiples alineaciones de secuencias, la separación de secuencias a lo largo de la cadena, la estabilidad de la alineación y la ocupación del contacto de un residuo específico como evaluación de la estructura 3D de la proteína.

Más recientemente se han explorado métodos basados en algoritmos de aprendizaje automático y redes neuronales para predecir las distancias entre los pares acoplados de residuos, así como los mapas de contacto de las proteínas y su aproximación basada en la clasificación de residuos. Estos sistemas han logrado predecir las estructuras tridimensionales con una efectividad del 35% para todas las proteínas, lo cual demuestra claramente que los predictores basados en aprendizaje automático mejoran la eficiencia de los predictores estadísticos, pero todavía no presentan una precisión suficiente para la predicción de la estructura de las proteínas.

Entre las principales causas que dieron origen a esta investigación se encuentran:

- Bajo nivel de efectividad de los algoritmos propuestos hasta la actualidad.
- La mayoría de los algoritmos de predicción de estructuras de proteínas presentan un alto grado de complejidad computacional.

- La mayoría de los algoritmos emplean técnicas cuyas base de conocimiento no son interpretables, convirtiéndolos en cajas negras.

Es por ello que, el **problema** que aborda esta investigación es que en la actualidad no se cuenta con técnicas de predicción de estructuras de proteínas, a partir de la secuencia de aminoácidos, que logren una precisión aceptable y sean capaces de brindar un modelo fácilmente interpretable por los especialistas biólogos.

Esta investigación se **enmarca** en el proceso de predicción de mapas de contactos de proteínas. Específicamente en el empleo de técnicas de aprendizaje automático que permitan dilucidar dicho proceso.

1.3.- Objetivos

El **objetivo principal** de esta investigación es diseñar un algoritmo, a partir de la información brindada por la secuencia de aminoácidos, que posea capacidad explicativa y permita predecir mapas de contactos de proteínas con una efectividad similar o superior a los algoritmos del estado del arte.

Este objetivo se sustenta sobre la **hipótesis** de que si se diseña una heurística basada en la correlación entre la estructura de residuos covalentes de una proteína y su secuencia de aminoácidos, se lograría un algoritmo capaz de predecir los mapas de contactos de una proteína, con una precisión aceptable.

Para dar cumplimiento a este objetivo, se trazaron las siguientes tareas de investigación:

1. Realizar el análisis del marco teórico sobre métodos de predicción de estructura de proteínas, profundizando en la etapa de predicción de los mapas de contactos.
2. Diseñar un método de extracción y pre-procesamiento de la información contenida en la secuencia de las proteínas que incluya la codificación y el tratamiento del desbalance entre contactos y no contactos.
3. Diseñar un método de predicción de mapas de contactos de proteínas, a partir del empleo de árboles, que sólo tome en cuenta la información contenida en la secuencia y que permita obtener un modelo descriptivo del proceso de plegamiento de las proteínas.

4. Validar el método propuesto mediante el empleo de conjuntos de proteínas del PDB y la comparación con otros predictores del estado del arte.

1.4.- Fundamentación de la investigación

Los contactos entre los residuales de las proteínas, condicionan su plegamiento y caracterizan su estructura. Por tanto, la predicción de los contactos residuales es un problema interesante cuya solución puede ser útil para el reconocimiento de plegamientos. A partir del conocimiento de los contactos residuales puede ser deducida la estructura tridimensional de una proteína.

La **importancia** de esta investigación se debe a que la estructura es quien determina la función que realiza la proteína, las cuales pueden ser muy variadas. Entre estas funciones se destacan la enzimática, hormonal, transportadora (hemoglobina), defensiva (anticuerpos), estructural (colágeno), etc. De ahí, el interés que tiene para la industria farmacéutica y para la medicina en general, el poder predecir la estructura de proteínas no conocidas con un nivel efectividad superior al existente en la actualidad. Esto, combinado con los datos proteómicos de expresión de arreglos (*microarrays*), brindaría un modelo flexible para la célula completa, potencialmente capaz de predecir las propiedades emergentes del sistema molecular, tales como las señales de las sendas de transducción, la diferenciación y la respuesta inmune. En Cuba y el resto del mundo, muchos son los esfuerzos que se dedican a la producción de nuevos fármacos y medicamentos, donde se presentan problemas de predicción de estructuras de proteínas que aún estar por resolver. La aplicación de técnicas de *Soft Computing* en este campo resulta relevante desde el punto de vista científico, económico y social.

En esta investigación se planteará un algoritmo original para la predicción de mapas de contactos de proteínas, a partir del empleo de árboles de decisión. Este algoritmo sigue los supuestos *ab initio* para las técnicas de predicción de estructuras. Donde toda la información para el proceso de predicción se extrae únicamente de la secuencia de proteínas.

Esta propuesta es supervisada y se formalizará en el algoritmo FoDT. Para su evaluación se seguirán los principios OECD establecidos por la Unión Europea para la validación de modelos químicos y biológicos [20] , incluyendo la medición de medirá la robustez del algoritmo, su bondad de ajuste y su capacidad de generalización. Además,

se definirá su dominio de aplicación y se explicará su mecanismo de interpretación de la predicción. En el proceso de experimentación serán empleadas bases de datos conocidas, otras extraídas de la base de datos de proteínas (PDB, *Protein Data Bank*) [6] y, para la comparación con algoritmos del estado del arte, la base de datos del CASP9 [21]–[23].

La **novedad** de la investigación está en que se propone un predictor basado en un multiclasificador. El cual presenta como ventajas fundamentales el bajo costo computacional y su capacidad explicativa, a lo cual se le agrega el empleo de la información que puede brindar la separación entre los residuos.

Como **aporte científico-metodológico**, el algoritmo presentado introduce la idea de tratar el problema de la predicción de contactos inter-residuales de forma independiente para cada uno de las parejas de aminoácidos posibles a formarse. Con este fin, se presenta un nuevo modelo de combinación de clasificadores con un nivel de efectividad que compite con los predictores del estado del arte y que brinda un modelo de conocimientos de fácil interpretación.

1.5.- Principales contribuciones

Las principales contribuciones de este trabajo son:

Publicaciones (JCR):

- Santiesteban-Toca Cosme E., Casañola-Martín Gerardo M, Aguilar Ruiz Jesús S. A divide-and-conquer strategy for the prediction of protein contact map. **Letters in Drug Design & Discovery**. Vol. 12, No.2, 2015. Impacto: **0.845**.
- Santiesteban-Toca Cosme E., Aguilar Ruiz Jesús S. Las técnicas de aprendizaje automático en la predicción de estructura de proteínas: Un enfoque desde la bioinformática. **AFINIDAD** IQS. 2014. ISSN: 0001-9704. Impacto: **0.145**.
- Santiesteban-Toca Cosme E., Aguilar Ruiz Jesús S. A new multiple classifier system for the prediction of protein's contacts map. **Information Processing Letters**. [Factor de Impacto: **0.488**]

Publicaciones referenciadas:

- Cosme E. Santiesteban-Toca, Julio C. Quintana-Saez, Jesús S. Aguilar-Ruiz. "Predicting protein contact's map employing a divide-and-rule multiple classifier". UNICA 2014.
- Alfonso E. Márquez-Chamorro, Federico Divina, Jesús S. Aguilar-Ruiz, Cosme E. Santiesteban-Toca: Improving the efficiency of MECoMaP: a protein residue-residue contact predictor. **Lecture Notes in Computer Science**. Volume 8259, 2013, p. 166 ff.
- Cosme E. Santiesteban-Toca, Milton García-Borroto, Jesús S. Aguilar-Ruiz: Using Short-Range Interactions and Simulated Genetic Strategy to Improve the Protein Contact Map Prediction. MCPR 2012: **Lecture Notes in Computer Science**, pages 166-175. Springer, 2012. ISBN: 0302-9743.
- Santiesteban-Toca Cosme E., Ascencio-Cortes Gualberto, Márquez-Chamorro Alfonso, Aguilar-Ruiz Jesús S. "Short-range interactions and decision tree-based protein contact map predictor". **Lecture Notes in Computer Science** series, volume 7246. Springer, 2012. ISBN: 0302-9743.
- Márquez-Chamorro Alfonso E., Divina-Federico, Aguilar-Ruiz Jesús S., Bacardit Jaume, Asencio-Cortés Gualberto, Santiesteban-Toca Cosme E. "A NSGA-II Algorithm for the Residue-residue Contact Prediction". **Lecture Notes in Computer Science** series, volume 7246. Springer, 2012. ISBN: 0302-9743.
- Asencio-Cortes Gualberto, Aguilar-Ruiz Jesus S., Marquez-Chamorro Alfonso E., Santiesteban-Toca Cosme E., Ruiz-Sanchez Roberto. "Prediction of mitochondrial matrix protein structures based on feature selection and fragment assembly". **Lecture Notes in Computer Science** series, volume 7246. Springer, 2012. ISBN: 0302-9743.
- Santiesteban-Toca Cosme E., and Aguilar-Ruiz Jesús S. DTP: Decision tree-based predictor of protein contact map. IEA/AIE (2), volume 6704 of **Lecture Notes in Computer Science**, pages 367-375. Springer, 2011.

- Santiesteban-Toca Cosme E., Márquez-Chamorro Alfonso E., Asencio-Cortés Gualberto, and Aguilar-Ruiz Jesús S. A decision tree-based method for protein contact map prediction. EvoBio, volume 6623 of **Lecture Notes in Computer Science**, pages 153-158. Springer, 2011.
- Santiesteban-Toca Cosme E., Aguilar-Ruiz Jesús S. Entorno de experimentación para predicción de estructuras de proteínas. Springer. 2010. ISBN: 959-25-0525-4.

Otras publicaciones:

- Santiesteban-Toca Cosme E., Aguilar Ruiz Jesús S. Multiple Trees classifier system for protein's contact map prediction. 9th International Congress of Biotechnology and Plan Culture. **Bioveg 2013**, pages 15-16. ISBN: 978-959-16-2045-3.
- Antón-Vargas Jarvin A., Santiesteban-Toca Cosme E. Selección de la mejor estrategia para la predicción de contactos interresiduales de proteínas. 9th International Congress of Biotechnology and Plan Culture. **Bioveg 2013**, pages 16-17. ISBN: 978-959-16-2045-3.
- López-Aparicio Liuben, Santiesteban-Toca Cosme E., Marrero-Ponce Yovany. Aid selection of protein and peptide sequences. 9th International Congress of Biotechnology and Plan Culture. **Bioveg 2013**, page 17. ISBN: 978-959-16-2045-3.
- Santiesteban-Toca Cosme E., Aguilar Ruiz Jesús S. Protein contact map prediction based on short range-interactions. XII Congreso de la Sociedad Cubana de Matemática y Computación. **COMPUMAT 2011**. ISBN: 978-959-250-658-9.
- Santiesteban-Toca Cosme E., Aguilar Ruiz Jesús S. Short-range interaction-based protein contact map predictor. Cuba-Flanders workshop on machine learning and knowledge discovery**. CF-WML-KD2011**. ISBN: 978-959-250-658-9.
- Santiesteban-Toca Cosme E. and Aguilar-Ruiz Jesús S. Predicción de mapas de contacto de proteínas basado en árboles de regresión. VIII Congreso Internacional de Biotecnología Vegetal. **BioVeg 2011**, page 96. ISBN: 959-16-0300-2.
- Mena-Torres Dayrelis, and Santiesteban-Toca Cosme E. Algoritmo de Edición de Datos Biológicos para la Predicción de Estructuras de Proteínas. VIII Congreso Internacional de Biotecnología Vegetal. **BioVeg 2011**, page 97 ISBN: 959-16-0300-2

- Santiesteban-Toca Cosme E., and Aguilar-Ruiz Jesús S. DTP: algoritmo de predicción de mapas de contacto de proteínas basado en árboles de decisión. XIV Convención y Feria internacional. **Informática 2011**. ISBN: 978-959-7213-01-7.
- Santiesteban-Toca Cosme E., Aguilar-Ruiz Jesús S. Software platform for the research in protein structure prediction methods. VII International Congress of Biotechnology and Plan Culture. **Bioveg 2009**. ISBN: 959-16-0300-2.
- Santiesteban-Toca Cosme E., Aguilar-Ruiz Jesús S. Entorno de experimentación para predicción de estructuras de proteínas. **Taller Internacional de SoftComputing. UCLV. Abril/09**.

Premios recibidos:

- **Premio Anual Provincial de la Academia de Ciencias de Cuba por su Impacto Científico**. Resolución No. 18/2014. Trabajo: "Predicción de Mapas de Contacto de Proteínas basados en Multiclasificadores". Autores: Cosme E. Santiesteban Toca, Julio César Quintana Zaez y Jesús S. Aguilar Ruiz.
- **Premio Anual Provincial de la Academia de Ciencias de Cuba por su Impacto Científico**. Resolución No. 20/2012. Trabajo: "Predicción de Mapas de Contacto de Proteínas basados en árboles". Autores: Cosme E. Santiesteban Toca, Jesús S. Aguilar Ruiz y Julio César Quintana Zaez.

Tutoría de tesis de grado:

- Julio César Quintana Zaez. "Algoritmo de predicción de mapas de contactos de proteínas basado en las interacciones de estructuras secundarias y árboles de decisión". **Tesis de Máster en Informática Aplicada**. Facultad de Informática. Universidad de Ciego de Ávila. (2014).
- Boris Luis Fajardo Estevez. "Predicción de mapas de contacto basado en árboles de regresión". **Tesis de Ingeniería en Informática**. Facultad de Informática. Universidad de Ciego de Ávila. (2012).
- Julio César Quintana Zaez. "Algoritmos de predicción de estructura proteínas basados en mapas multiclases". **Tesis de Ingeniería en Informática**. Facultad de Informática. Universidad de Ciego de Ávila. (2011).

Registro de Software:

- Cosme Ernesto Santiesteban Toca. "Contact maps predictor" Plataforma para la investigación de métodos de predicción de estructuras de proteínas. Certificación de depósito legal facultativo de obras protegidas. **Registro: 150-2011**.

1.6.- Organización

El trabajo que se presenta, está estructurado en cuatro partes y varios capítulos organizados de la siguiente manera:

Capítulos 2 y 3: en estos capítulos se expone la fundamentación de teórica que permite el entendimiento de la investigación. En el primero se abordan conceptos básicos sobre la estructura, las funciones, la síntesis y el plegamiento de las proteínas. Se realiza un análisis del proceso de predicción de estructura de las proteínas, haciendo especial énfasis en los mapas de contactos inter-residuales. En el segundo, elementos sobre multiclasificadores, construcción y esquemas básicos.

Capítulo 4: en este capítulo se hace una valoración crítica de las diferentes técnicas abordadas para la predicción de mapas de contactos y estructuras de proteínas. Se hace un estudio taxonómico de dichas técnicas y un análisis de la distribución de las mismas.

Capítulo 5: éste es el capítulo más importante del documento, pues aquí se presenta la metodología propuesta para realizar la predicción de los mapas de contactos de las proteínas. Se explica cómo se realiza la codificación, el pre-procesamiento, el multiclasificador y el algoritmo de filtrado diseñados para este propósito.

Capítulo 6: se realiza la validación de la propuesta presentada en la presente investigación. Se muestran los resultados experimentales para el análisis del dominio de aplicación del algoritmo, así como de la validación interna y externa del mismo. Adicionalmente, se expone el mecanismo de interpretación de la propuesta.

Capítulo 7: se presentan las conclusiones y recomendaciones, enfatizándose en los resultados alcanzados y proponiendo líneas de continuación del trabajo aquí presentado.

Por último se muestran las referencias bibliográficas y los materiales anexos a la presente investigación.

Capítulo 2

Fundamentos sobre la predicción de estructuras de proteínas

Las proteínas son cadenas de aminoácidos unidos a través de enlaces covalentes. Éstas están consideradas las macromoléculas biológicas más abundantes y se encuentran presentes en todas las células y partes de las mismas. Se presentan en una gran variedad que pueden variar en su tamaño desde simples péptidos relativamente pequeños hasta polímeros con gran masa molecular. De igual forma, éstas exhiben gran diversidad en cuanto a su función biológica.

En el presente capítulo se abordan conceptos básicos sobre la estructura, las funciones, la síntesis y el plegamiento de las proteínas. Se realiza un análisis del proceso de predicción de estructura de las proteínas, haciendo especial énfasis en los mapas de contactos inter-residuales.

2.1.- Introducción

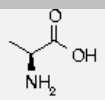
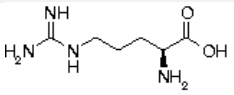
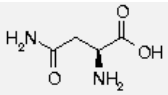
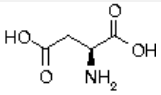
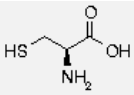
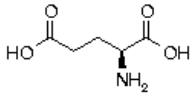
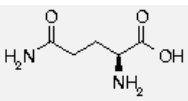
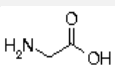
Las proteínas son los instrumentos moleculares mediante los que se expresa la información genética. Están formadas por cadenas lineales de aminoácidos, de los cuales existen veinte especies fundamentales y que se unen entre sí mediante enlaces peptídicos (enlaces covalentes).

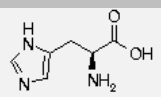
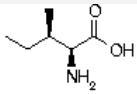
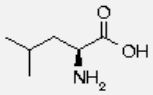
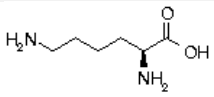
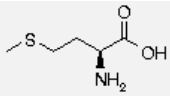
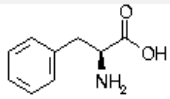
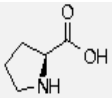
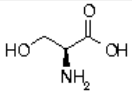
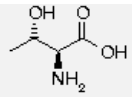
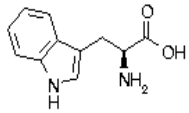
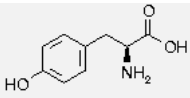
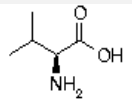
2.2.- Conceptos básicos de proteínas

Las proteínas son largas cadenas de aminoácidos unidas por enlaces peptídicos entre el grupo carboxilo (COOH) y los grupos amino (NH_2) de residuos de aminoácidos adyacentes.

Estos aminoácidos, aunque presentan rasgos estructurales comunes, difieren unos de otros en sus cadenas laterales. Entre sus principales características se puede mencionar la presencia de un átomo de carbono central, el C_α portador de un grupo amino, un grupo carboxilo, un átomo de hidrógeno y una cadena lateral o grupo R (residuo). Los residuos varían en su composición y sus características físico-químicas como la carga y la solubilidad en agua, lo cual brinda al aminoácido sus características definitorias en función de las cuales se clasifican. Existen 20 aminoácidos estándar, a los cuales se les han asignado nombres y abreviaturas de una y tres letras (Tabla 1) [24].

Tabla 1. Aminoácidos estándar. Denominación, códigos de tres y una letra y algunas de sus propiedades físico-químicas.

Nombre	Abrevia-turas		Abundancia relativa (%) E.C.	MW	pK	*VdW* volumen ($Å_3$)	Cargado, Polar, Hidrofóbico, Neutro	Moléculas
Alanina	Ala	A	13.0	71		67	H	
Arginina	Arg	R	5.3	157	12.5	148	C+	
Asparagina	Asn	N	9.9	114		96	P	
Aspartato	Asp	D	9.9	114	3.9	91	C	
Cisteína	Cys	C	1.8	103		86	P	
Glutamato	Glu	E	10.8	128	4.3	109	C	
Glutamina	Gln	Q	10.8	128		114	P	
Glicina	Gly	G	7.8	57		48	N	

Nombre	Abreviaturas		Abundancia relativa (%) E.C.	MW	pK	*VdW* volumen ($Å_3$)	Cargado, Polar, Hidrofóbico, Neutro	Moléculas
Histidina	His	H	0.7	137	6.0	118	P,C+	
Isoleucina	Ile	I	4.4	113		124	H	
Leucina	Leu	L	7.8	113		124	H	
Lisina	Lys	K	7.0	129	10.5	135	C+	
Metionina	Met	M	3.8	131		124	H	
Fenilalanina	Phe	F	3.3	147		135	H	
Prolina	Pro	P	4.6	97		90	H	
Serina	Ser	S	6.0	87		73	P	
Treonina	Thr	T	4.6	101		93	P	
Triptófano	Trp	W	1.0	186		163	P	
Tirosina	Tyr	Y	2.2	163	10.1	141	P	
Valina	Val	V	6.0	99		105	H	

La unión de un conjunto de aminoácidos forma **péptidos**, los cuales se pueden encontrar en la naturaleza en tamaños que contienen desde dos hasta miles de

aminoácidos. Los péptidos formados por pocos aminoácidos son denominados **oligopéptidos**, mientras que cuando se unen muchos aminoácidos se les denomina **polipéptidos**. Razón por la cual, en muchos casos los términos **polipéptidos** y **proteínas** suelen intercambiarse. Sin embargo, suele aceptarse que las moléculas con masas molares inferiores a 10.000 son consideradas **polipétidos** y con masa molar superior, **proteínas**.

2.2.1.- Estructura

La organización de una proteína se encuentra definida por cuatro niveles estructurales comúnmente denominados como estructuras: primaria, secundaria, terciaria y cuaternaria (Figura 1). Cada una de estas estructuras informa de la disposición de la anterior en el espacio [25].

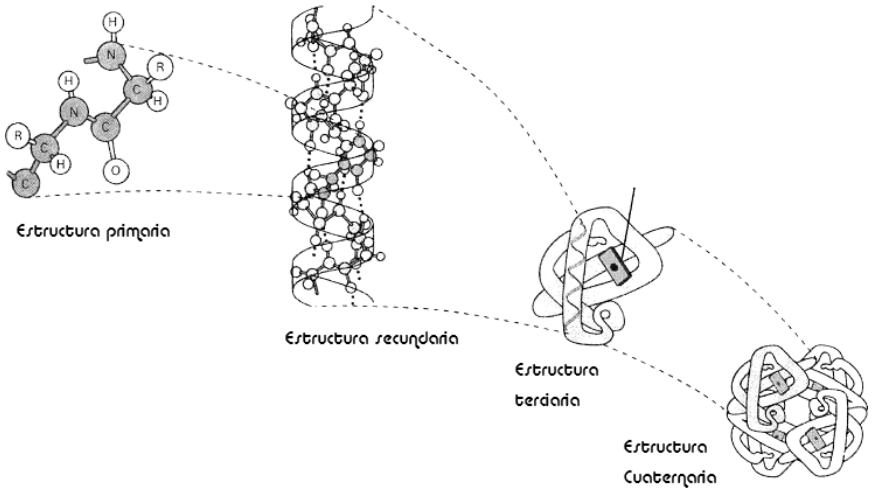


Figura 1. Estructura de las proteínas: primaria, secundaria, terciaria y cuaternaria.

El primer nivel estructural describe todos los enlaces covalentes (principalmente los enlaces peptídicos y puentes disulfuros) que se pueden encontrar en una proteína. Está constituido tanto por el número y la variedad de aminoácidos que entran en su composición como por el orden en que se disponen éstos a lo largo de la cadena polipeptídica, también llamado secuencia (Figura 2). A este primer nivel se le llama estructura primaria.

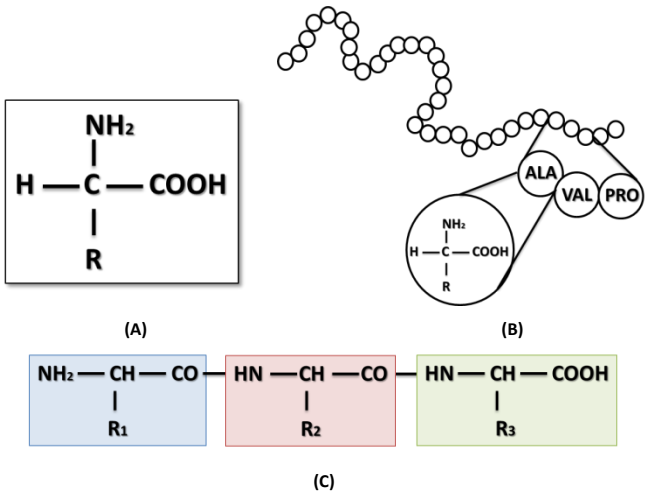


Figura 2. Estructura primaria de la proteína. (A) Aminoácido. (B) secuencia de aminoácidos. (C) cadena peptídica.

Cada polipéptido tiene una composición característica, debido a que los 20 aminoácidos casi nunca se presentan en cantidades iguales en una proteína. Pudiendo aparecer múltiple, una o ninguna vez por molécula.

El segundo nivel estructural se refiere a la relación espacial que guarda un aminoácido con respecto al que le sigue y al que le antecede en la cadena polipeptídica. Estas disposiciones particularmente estables dan lugar a motivos estructurales repetitivos que se muestran, en algunos casos, en el polipéptido entero o algunas zonas de éste. A este segundo nivel se le llama estructura secundaria. Existen dos tipos de estructura secundaria: la alfa-hélice (α), donde los aminoácidos se enrollan en forma helicoidal como si formaran un resorte; y, la conformación beta (β), donde los aminoácidos se mantienen extendidos (Figura 3). Son muy frecuentes asociaciones entre estructuras secundarias que suelen ser muy estables. Una hélice puede enrollarse con otra y formar una superhélice. También hay combinaciones de hélices y hojas plegadas.

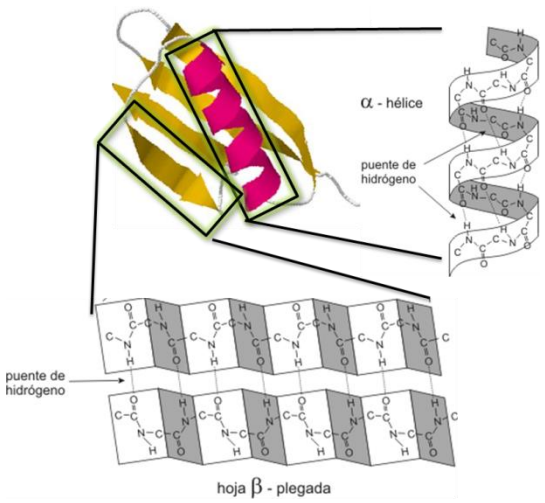


Figura 3. Estructura secundaria de la proteína, láminas beta (β) y hélices alfa (α).

El tercer nivel estructural se refiere a la relación espacial que guardan entre sí las diferentes zonas o áreas de cada cadena polipeptídica que forman a una proteína y describe todos los aspectos del plegamiento tridimensional de un polipétido. A este nivel se le llama estructura terciaria (Figura 4).

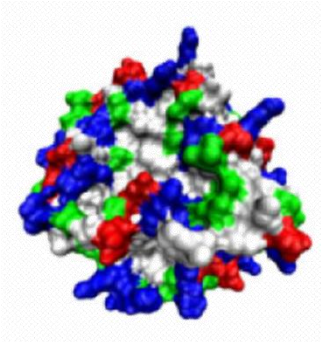


Figura 4. Estructura terciaria de la proteína.

En una proteína compuesta de una sola cadena polipeptídica, el nivel máximo de estructuración corresponde precisamente a su estructura terciaria. Cuando se trata de una proteína oligomérica, que es aquel tipo de proteína que está compuesta de más de una cadena polipeptídica, se puede considerar un siguiente nivel de organización. Este otro nivel se refiere a la manera en que cada cadena polipeptídica en la proteína se arregla en el espacio en relación con las otras cadenas polipeptídicas que la constituyen. A este nivel estructural se le llama estructura cuaternaria.

La estructura terciaria informa sobre la disposición de la estructura secundaria de un polipéptido al plegarse sobre sí misma originando una conformación globular. Esta conformación globular facilita la solubilidad en agua y así realizar funciones de transporte, enzimáticas, hormonales. Esta disposición se mantiene estable gracias a la existencia de enlaces entre los radicales R de los aminoácidos. Además, aparecen varios tipos de enlaces: los puentes disulfuro entre los radicales de aminoácidos que tiene azufre; los puentes de hidrógeno; los puentes eléctricos; y, las interacciones hidrófobas.

La estructura cuaternaria deriva de la conjunción de varias cadenas peptídicas que, asociadas, conforman un ente que posee propiedades distintas a la de sus monómeros componentes. Esta asociación ocurre mediante interacciones no covalentes, como pueden ser los puentes de hidrógeno, las interacciones hidrofóbicas o los puentes salinos.

2.2.2.- Funciones

Las proteínas ocupan un lugar de máxima importancia entre las moléculas constituyentes de los seres vivos (biomoléculas). Prácticamente todos los procesos biológicos dependen de la presencia y/o actividad de este tipo de sustancias.

Su estructura terciaria determina la función que realiza la proteína, las cuales pueden ser muy variadas. Entre estas funciones se destacan la enzimática (catalizadores de reacciones químicas en organismos vivientes), hormonal (reguladores de actividades celulares), transportadora (hemoglobina y otras moléculas con funciones de transporte en la sangre), defensiva o anticuerpos (encargados de acciones de defensa natural contra infecciones o agentes extraños), estructural (colágeno), los receptores de las células (a los cuales se fijan moléculas capaces de desencadenar una respuesta determinada), la actina y la miosina (responsables finales del acortamiento del músculo durante la contracción), el colágeno (integrante de fibras altamente resistentes en tejidos de sostén), etc.

Las proteínas dirigen la totalidad de los procesos celulares, incluso su propia síntesis. Las funciones de mayor importancia de las proteínas en los seres vivos son:

Función estructural, como el colágeno, la tubulina de los microtúbulos, las de las cápsides virales, etc. (Tabla 2).

Las moléculas de colágeno son ejemplos típicos de las proteínas simples fibrosas. Son la clase de proteínas más abundantes de nuestro cuerpo, son componentes de la matriz extracelular del tejido conectivo, de modo que las podemos encontrar en tendones, ligamentos, membrana basal, etc.

Aunque existen distintos tipos de colágeno que se diferencian en las secuencias de aminoácidos y en las proporciones con que se encuentran los mismos, podemos hacer una generalización acerca de su estructura. El colágeno es una proteína fibrosa que posee una estructura de orden superior. Está formado por unidades compuestas por tres cadenas polipeptídicas de aproximadamente 1000 aminoácidos cada una. Un tercio de esos aminoácidos está constituido por la glicina, prolina y lisina hidroxiladas, constituyendo una estructura rígida. El procolágeno, su unidad precursora, es secretado por el fibroblasto a la matriz extracelular junto a dos enzimas. Estas enzimas catalizan la separación de los extremos de la molécula de procolágeno para producir la triple hélice de tropocolágeno. Las moléculas de tropocolágeno se asocian espontáneamente formando microfibrillas. Las microfibrillas se empaquetan unas junto a otras para formar fibras de colágeno maduro.

Otro ejemplo de proteínas simples fibrosas lo constituyen las queratinas, que dan protección externa (piel, uñas, cabello, cuernos, etc.). Son producidas por las células epidérmicas. Su estructura secundaria es en gran parte α-hélice. En el caso particular de las queratinas del cabello encontramos en su estructura primaria un gran número de cisteínas (en el R contienen grupos SH), lo que permite la formación de puentes disulfuro, que son uniones covalentes que se dan entre dos grupos SH y que estabilizan la estructura proteica. El calor o el tratamiento con determinados productos químicos pueden reducir los puentes disulfuro, o bien formar puentes nuevos, estirando u ondulando el cabello.

Tabla 2. Algunas Proteínas Fibrosas y sus Funciones.

Proteína	Origen	Función
F-actina	Intracelular, todas las células.	Formación de microfilamentos en el citoesqueleto, movimiento contráctil.
Colágeno	Matriz extracelular, huesos, piel, vasos sanguíneos.	Resistencia a la tensión.
Desmina	Células musculares.	Estructuras que sirven de armazón dentro de la célula.
Elastina	Vasos sanguíneos, ligamentos.	Elasticidad.
Fibroína	Seda	Fuerza sin flexibilidad.
Queratina	Piel, cabello, etc. Intracelular.	Estructuras protectoras, resistencia a la tensión de los epitelios.
Lamina (Laminina nuclear)	Lamina nuclear.	Estructural.
Esclerotina	Exoesqueleto de los artrópodos.	Rigidez
Espectrina	Membrana de los eritrocitos.	Se enlaza con la F-actina, lo que permite que la membrana sea flexible.

Función Reguladora: como las ciclinas que controlan el ciclo celular y los factores de transcripción que regulan la expresión de los genes.

Función Motora: actina y miosina del músculo.

Función de Transporte: Globulinas en general, hemoglobima, mioglobina y las lipoproteínas son algunos ejemplos.

La hemoglobina y la mioglobina son proteínas globulares conjugadas, es decir que en su estructura encontramos a parte del polipéptido un grupo no proteico que en este caso corresponde al grupo Hemo.

La mioglobina consta de una sola cadena polipeptídica asociada a un grupo hemo que es el responsable de la unión del oxígeno, en tanto que la hemoglobina está formada por cuatro cadenas polipeptídicas cada una con su correspondiente grupo hemo. Por lo tanto la hemoglobina presenta estructura cuaternaria lo que le permite variar su afinidad por el oxígeno, la cual se ve afectada por el pH sanguíneo, la temperatura y la concentración de 2,3 DGP (2,3- difosfoglicerato).

Función de Reserva: La ovoalbúmina, componente principal de la clara de huevo o la gliadina del trigo.

Función de Receptores: como las proteínas receptoras de membrana.

Función Enzimática: La enzimas catalizan todas las reacciones metabólicas. Dada su importancia biológica, este tema será tratado con más detalle en el próximo capítulo.

Función de Defensa: Los anticuerpos son proteínas simples globulares y son sintetizadas por las células plasmáticas (linfocitos B activados), son también conocidas como inmunoglobulinas o gammaglobulinas. Estas proteínas presentan gran diversidad ya que cada anticuerpo es específico para un determinado antígeno. Sin embargo, podemos mencionar que en general están compuestas por cuatro cadenas polipeptídicas dos contienen 220 aminoácidos (cadenas livianas) y las otras más largas con 440 aminoácidos cada una (cadenas pesadas).

Función de mensajeros químicos: La mayor parte de las hormonas son proteínas o glucoproteínas. También ciertos aminoácidos, derivados de aminoácidos y oligopéptidos son neurotransmisores en el sistema nervioso.

Cada proteína lleva a cabo una determinada función y lo realiza porque posee una determinada estructura primaria y una conformación espacial propia; por lo que un cambio en la estructura de la proteína puede significar una pérdida de la función. A este fenómeno se le denomina **especificidad**. No todas las proteínas son iguales en todos los organismos, cada individuo posee proteínas específicas suyas. Esto se pone de manifiesto en los procesos de rechazo de órganos trasplantados o en la semejanza entre proteínas en individuos con un grado de parentesco y su uso en la construcción de "árboles filogenéticos".

De ahí, la importancia que tiene para la industria farmacéutica y para la medicina en general, el poder predecir la estructura de proteínas no conocidas con un nivel efectividad superior al existente en la actualidad. Esto, combinado con los datos proteómicos (expresión de arreglos), brindaría un modelo flexible para la célula completa, potencialmente capaz de predecir las propiedades emergentes del sistema molecular, tales como las señales de las sendas de transducción, la diferenciación y la respuesta inmune.

2.2.3.- Plegamiento

El plegamiento de proteínas es el proceso por el que una proteína alcanza su estructura tridimensional, de lo cual depende su función. Si una proteína no se pliega correctamente será no funcional y, por lo tanto, no será capaz de cumplir su función biológica.

Existen muchos indicios que señalan que la información necesaria para un correcto plegamiento de una proteína globular está contenida en la estructura primaria, es decir en su secuencia de aminoácidos. Una proteína debe ser capaz de plegarse correctamente sin más información que la contenida en su secuencia de aminoácidos y las interacciones que se establecen entre ellos.

La estructura tridimensional de una proteína en condiciones fisiológicas se conoce como estructura nativa, y se considera la estructura más estable de todas las estructuras posibles. Además la estructura nativa es la funcionalmente activa. Si se cambian las condiciones ambientales, la estructura nativa se pierde, este proceso se denomina desnaturalización.

La estabilidad de la proteína es la tendencia a mantener la conformación nativa y se debe en lo fundamental a las interacciones débiles. Los enlaces covalentes son claramente más fuertes que las interacciones débiles individuales. Pero, al ser tantas las interacciones débiles, éstas son las que logran la fuerza estabilizadora predominante. Si se tiene en cuenta que la conformación nativa de una proteína es el estado en que presenta la conformación de más baja energía libre (la más estable), entonces ésta se corresponde con la que posee el mayor número de interacciones débiles.

Sin embargo, la estabilidad de una proteína no es únicamente el resultado de la suma de las energías libres de conformación de las muchas interacciones débiles de su interior. El plegado de proteínas globulares a partir de sus conformaciones desnaturalizadas es un proceso notablemente rápido, que se completa en menos de un segundo.

Anfinsen demostró a finales de los sesenta que al desplegar la enzima ribonucleasa A con urea y mercaptoetanol aumentaba su volumen aparente y desaparecían sus

propiedades catalíticas. Al dializar la proteína volvía a plegarse. El plegamiento de las proteínas no está inducido por la célula sino que es el resultado de la interacción de la secuencia polipeptídica con el agua. Toda la información necesaria para adquirir su estructura tridimensional está presente en la secuencia de aminoácidos por lo que, algún día, se podrá predecir.

Dada la flexibilidad de los polipéptidos el número de conformaciones posible de una proteína es enorme. Esto puede resultar paradójico tal y como puso de manifiesto Levinthal en 1968 [26]: si una proteína se pliega explorando al azar todas las conformaciones posibles tardará mucho más que la edad que tiene el Universo. Por ejemplo: una proteína pequeña como la RNAsa A, que tiene 124 aminoácidos tiene 10^{50} conformaciones posibles. Si la molécula pudiese probar una configuración cada 10^{-13} segundos, serían necesarios 10^{30} años para probarlas todas. Sin embargo, se ha comprobado in vitro que la RNAsa de pliega en aproximadamente 1 minuto. Como las proteínas se pliegan muy deprisa (típicamente en milisegundos o segundos) está claro que no exploran todas las conformaciones al azar.

Las proteínas, al parecer forman su estructura como una secuencia de eventos ordenados y secuenciales, llamados "**rutas de plegamiento**" (*pathway*). Las rutas de plegamiento podrían interpretarse como la selección acumulativa, según la cual los intermediarios parcialmente correctos se retienen. Los sitios de iniciación favorecen la estabilización de estructuras más complejas. La naturaleza de estos eventos, o bien están restringidos a contactos nativos (definidos como contactos que se mantienen en la estructura final) o bien pueden incluir interacciones no específicas.

Visión clásica del plegamiento

Esta visión de las proteínas supone que todas las moléculas siguen el mismo camino pasando en su caso por los mismos intermediarios y por el estado de transición. Durante mucho tiempo las dos teorías principales sobre cómo se pliegan las proteínas son el modelo "molten globule" o "hydrophobic collapse" (en función de las interacciones no específicas) y el modelo "framework" o "nucleation/condensation" (basado en las sendas restrictivas sólo para los contactos nativos). El modelo "folding funnel" [27] combinó los modelos "hydrophobic collapse" y "nucleation/ condensation".

Los modelos de tipo armazón (framework) asumen que los procesos iniciales consisten en la formación temprana de algunos de los elementos de estructura secundaria que aparecen en el estado final. Estas regiones chocan y dan lugar a intermediarios que terminan plegándose. El modelo del colapso hidrófobo, sin embargo, supone que el primer suceso relevante es el colapso al azar del polipéptido para ocultar los residuos hidrófobos. A partir del estado colapsado se va organizando el polipéptido y aparece la estructura secundaria de los intermediarios y, finalmente, del estado nativo. También se postuló un modelo (de puzle) que asumía que cada molécula seguía un camino distinto.

Nueva visión del plegamiento

Los modelos clásicos secuenciales han sido cuestionados en los últimos años y se ha propuesto una "nueva visión" del plegamiento en la que cada molécula se pliega por una ruta distinta (como en el antiguo modelo de puzle). La razón de que el plegamiento sea rápido es que las moléculas se mueven por un paisaje de energía en forma de embudo de modo que las interacciones nativas que aparecen tienden a conservarse y la proteína sigue plegándose "cuesta abajo". En general las barreras de energía entre conformaciones son bajas pero donde el embudo presenta rugosidades aparecen intermediarios.

2.2.4.- Elementos conformacionales.

En muchas proteínas existen zonas con entidad estructural independiente, y a menudo funciones bioquímicas específicas. La estructura cuaternaria deriva de la conjunción de varias cadenas peptídicas que conforman un nuevo ente con propiedades distintas.

Las proteínas se organizan en múltiples unidades. Los dominios estructurales son elementos de la estructura de las proteínas que se autoestabilizan y, frecuentemente, estabilizan a los motivos conformacionales con independencia del resto de la secuencia de la proteína. Los dominios suelen aparecer en una variedad de proteínas, aunque muchos son únicos y proceden de una única secuencia de un gen o una familia génica. A menudo, son seleccionados evolutivamente por poseer una función prominente en la biología de la proteína a que pertenecen.

Un motivo puede referirse a una combinación específica de elementos estructurales secundarios (ejemplo: hélice-beta-hélice o beta-giro-beta). Estos elementos son llamados a menudo superestructuras secundarias. Los motivos conformacionales son, entonces, un tipo de motivo de forma global, como los barriles-beta. Los motivos a menudo incluyen giros de longitud variable en estructuras indeterminadas, y esto crea la plasticidad necesaria para unir dos elementos en el espacio sin que estén codificados por una secuencia de ADN inmediatamente adyacente en un gen.

Aunque en el *Protein Data Bank* existen más de 105.000 estructuras, hay muchos menos dominios, motivos estructurales y pliegues. Esto se debe, en gran medida, a la evolución. O sea, un dominio de una proteína puede ser trasladado de una a otra, dando así una nueva función a las proteínas. Esto hace que los dominios o motivos estructurales puedan ser comunes a varias familias de proteínas.

2.3.- Predicción de estructura de proteínas.

A pesar del avance en las técnicas experimentales para proporcionar modelos aproximados de la estructura y la dinámica de las proteínas (cristalografía de rayos X o resonancia magnética nuclear tridimensional), cada día aumenta la diferencia entre el número de secuencias y el de estructuras conocidas. La Bioinformática, la Biología Computacional, así como otras disciplinas, han incursionado en la predicción de la estructura de las proteínas. Los métodos de predicción de estructuras tienen por objetivo disminuir esta diferencia entre el número de secuencias conocidas y el de estructuras [28].

Sin embargo, la predicción de la estructura terciaria de las proteínas se ha convertido en uno de los grandes retos de la Bioinformática. Este proceso implica predecir las estructuras secundaria y terciaria de la proteína desde su estructura primaria. En la actualidad, se ha convertido en uno de los principales objetivos de la bioinformática y de la química teórica debido a su importancia en la medicina, para el diseño de fármacos, y en la biotecnología, para el diseño de nuevas enzimas.

La predicción de la estructura a partir de la secuencia de aminoácidos ha resultado difícil, fundamentalmente debido a las interacciones de largo alcance que estabilizan las estructuras secundarias y terciarias. A esto se le une que una proteína típica se

compone de centenares de enlaces individuales y si se toma en cuenta la libertad de rotación de dichos enlaces, entonces una misma proteína podría adoptar un número infinito de formas en el espacio.

El éxito de una predicción se determina por su comparación con los resultados de aplicar el algoritmo DSSP (método estándar para asignar una estructura secundaria a los aminoácidos de una proteína dadas sus coordenadas atómicas de resolución) a la estructura cristalina de la proteína. Para ácidos nucleicos, podría determinarse por el patrón de puentes de hidrógeno.

Características 1D, 2D y 3D de las secuencias.

Las características 1D de una secuencia son aquellas que pueden ser representadas por un solo valor asociado a cada aminoácido [29], [30]. Entre ellas se incluyen los valores de características físico-químicas de amino-ácidos, o las predicciones de estructura secundaria.

Se han desarrollado algoritmos para la detección de patrones específicos bien definidos tales como hélices transmembrana y hélices superenrolladas en las proteínas, o estructuras de microARN en el ARN.

Las características 2D de las proteínas, por otro lado, corresponden a la descripción de los contactos entre los residuos de la proteína, ya sea a corta distancia o a larga distancia. Donde, por contactos se entiende cualquier tipo de enlace entre residuos (puente de hidrogeno, puente disulfuro). Los contactos a corta distancia están relacionados con el tipo de estructura secundaria. Los contactos a larga distancia, sin embargo, dan información de la organización de los elementos de estructura secundaria. La predicción de contactos, como herramienta para predecir la estructura terciaria es un campo poco desarrollado aún.

Las características 3D se refieren a la estructura terciaria de las proteínas. Actualmente no existen métodos capaces de predecir la estructura 3D de una proteína a partir de su secuencia.

Se supone que la estructura 3D de una proteína está determinada únicamente por la especificidad de la secuencia. También es conocida la influencia que ejercen las proteínas chaperonas en el plegamiento, y se sigue asumiendo que, generalmente, la

estructura final de la proteína es la que representa el mínimo de energía libre. Por estas razones se afirma que toda la información sobre la estructura nativa de una proteína esta codificada en su secuencia, aunque es específica del medio en solución en que se encuentre.

Una simplificación del problema de predicción de estructura 3D es su proyección en cadenas de asignaciones estructurales. Por ejemplo, se puede asignar estados de estructura secundaria o solvatación para cada residuo identificándolos con un símbolo. De hecho, los mayores avances en bioinformática de la última década se han alcanzado en el campo de la predicción de estructura secundaria. Estos avances se han logrado al combinar algoritmos matemáticos complejos con la información evolutiva disponible en las bases de datos.

Para intentar resolver el problema de la predicción de estructura de proteínas, se han adoptado dos aproximaciones fundamentales (Figura 5):

1. La predicción basada en la secuencia: parten de la asunción de que la información necesaria para conocer la estructura tridimensional de una proteína está en su secuencia de aminoácidos.
    - *Ab initio*: intenta resolver el problema analizando las propiedades físico–químicas de los aminoácidos en la secuencia sin tener en cuenta ninguna de las estructuras conocidas.
    - *De novo*: incorporan el análisis de la información evolutiva de las proteínas, tomando en cuenta las secuencias conservadas y las mutaciones correlacionadas.
2. El modelado por comparación: intenta resolver el problema a partir de estructuras conocidas. Esta aproximación, a su vez, puede ser dividida en otras dos vertientes: el modelado por homología y el modelado por hilvanado.
    - Homología: cuando existe una proteína de secuencia parecida y estructura conocida se puede construir un modelo realista basado en la estructura conocida. Este método asume que la estructura está más conservada que la secuencia, de modo que si la proteína que se quiere modelar presenta más de un 30% de identidad con una proteína de estructura conocida, ambas

proteínas serán estructuralmente semejantes. La mayoría de los métodos para la predicción de la estructura secundaria de las proteínas comienzan con el alineamiento de la secuencia que se pretende modelar.

- Hilvanado (*threading*): Cuando no existe ninguna proteína de secuencia parecida en que inspirarse se recurre al modelado por hilvanado o reconocimiento del plegamiento que consiste en plegar la proteína de todas las maneras empleadas por proteínas conocidas y calcular en cuál de ellas la energía es menor.

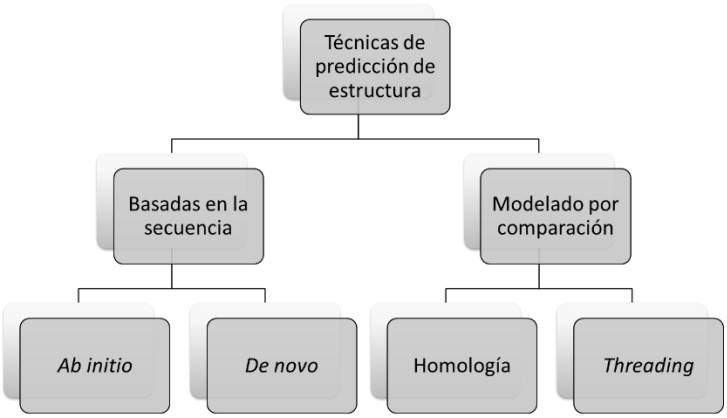


Figura 5. Taxonomía de técnicas de predicción de estructura de proteínas.

2.3.1.- Ab initio.

Estos métodos parten de la asunción de que la información necesaria para conocer la estructura tridimensional de una proteína está en su secuencia de aminoácidos. Se toma la estructura nativa de la proteína como la conformación que se corresponde al mínimo global de una función potencial, que "representa" a la proteína. Para optimizar esta función se emplean distintos métodos de búsqueda en el espacio conformacional. Entre ellos se encuentran simulaciones Monte Carlo, algoritmos de mecánica molecular combinados con dinámica molecular o bases de datos de elementos de estructura secundaria estándar [31].

Un sistema de predicción de estructura de proteínas *ab initio*, usualmente está compuesto por dos elementos: un algoritmo para buscar en el espacio de posibles configuraciones de la proteína para minimizar alguna función de costo; y la función de costo. Esta función está compuesta por varias restricciones derivadas de leyes físicas,

características estructurales (por ejemplo: estructura secundaria, accesibilidad, mapas de contacto entre residuos). La predicción puede realizarse mediante máquinas de aprendizaje o cualquier otro tipo de sistema estadístico, o empleando posibles restricciones obtenidas experimentalmente.

Estos métodos son costosos computacionalmente y su eficiencia disminuye con el tamaño de la proteína (no confiables para péptidos de más de 150 aminoácidos). La principal ventaja que tienen es que es posible modelar proteínas que corresponden a plegamientos no conocidos debido a que sólo se necesita la secuencia como información de partida [11], [13], [32]–[35].

Por lo general, las técnicas *ab initio*, no son más efectivas que las basadas en modelos, pero su diseño es mucho más simple. Múltiples investigaciones están enfocadas a los mapas de contacto binarios. Lo cual se debe a que los mapas de contacto binarios proveen información suficiente para la posterior reconstrucción. La razón fundamental es que si los mapas de contacto son equivalentes a la estructura de la proteína, entonces la predicción de mapas de contacto es equivalente a la predicción de estructura. No obstante, la calidad de la predicción basada en mapas de contacto no logra los niveles de efectividad deseados [33].

2.3.2.- De novo.

El empleo de la información contenida en cada residuo que conforma la cadena, para realizar la predicción de la estructura secundaria de la proteína, es una de las técnicas propuestas [36]. Donde una buena predicción de un residuo sería la asignación correcta de su estructura, pero no en dependencia de uno o varios residuos sino de todos los residuos que conforman la proteína. Éste, es un método cuya implementación presupone un bajo costo computacional que tiene como mérito fundamental el tomar en cuenta las interacciones intermedias en la cadena. Sin embargo, la efectividad del método depende de la optimización de sus variables, las cuales pueden tomar múltiples valores en dependencia del residuo al que se predice su estructura.

Una técnica que se ha generalizado es la alineación de secuencias homólogas. Para realizar la predicción de estructura secundaria y sitios activos se emplea la información

disponible en la familia de secuencias homólogas [37]. Esta aproximación está basada en la información contenida en cada residuo [36], para residuos alineados, y en la observación de que las inserciones y la gran variabilidad de las secuencias tienden a ocurrir en regiones cerradas entre estructuras secundarias.

Esta técnica, primeramente, alinea todas las secuencias. Un método estándar para alinear dos proteínas es la programación dinámica, que consiste en una matriz de similitudes (identidad, propiedades químicas, etc.) [14], [38], entre pares de aminoácidos donde el algoritmo establece una alineación incluyendo inserciones que tiendan a un mejor puntaje. Posteriormente, se evalúa la extensión de las secuencias conservadas para cada posición obtenida. Este valor modifica el resultado del método de predicción de Garnier [39] intentando lograr mejoras.

Además de la conservación de secuencias y de la predicción de la estructura secundaria, pueden localizarse varias regiones activas de las enzimas o residuos funcionalmente importantes (FIR, por sus siglas en inglés). Esto incluye residuos directamente involucrados en la catálisis. Los FIRs pueden estar formados por uno o más residuos secuenciales y que frecuentemente se mantienen invariables para familias de enzimas. La identificación de estos FIRs puede ser empleada para localizar determinados residuos para mutagénesis específicas.

El principal mérito de esta técnica es la capacidad de extracción de información de la alineación de las secuencias homólogas. Lo cual, a su vez, representa una desventaja porque su efectividad estaría en dependencia directa del método de alineación de secuencias empleado.

En la práctica, las técnicas basadas en la secuencia no se emplean para deducir la estructura de una proteína completa, sino como apoyo a otras técnicas más potentes y que consiguen más éxitos. Este conjunto de técnicas constituyen el segundo grupo de métodos de predicción, el modelado por homología.

2.3.3.- Homología.

Parte de la idea de que todas las parejas de proteínas que presentan una identidad de secuencia mayor al 30% tienen estructura tridimensional similar. Tomando en cuenta esto, se puede construir el modelo tridimensional de una proteína de estructura

desconocida, si se parte de la semejanza de la secuencia con proteínas de estructura ya conocidas [13], [37], [40]–[42].

Las etapas del proceso para el modelado por homología (Figura 6), son esencialmente:

- Identificación de plantillas. Se identifican estructuras conocidas que estén relacionadas con la secuencia diana. Para ello se emplean métodos de comparación de secuencias como FASTA, BLAST o PSI-BLAST.
- Alineamiento. Se alinea la proteína objetivo con las plantillas encontradas. Es la etapa más importante y sensible, debido a que la construcción del modelo depende del alineamiento realizado. En este paso se emplean programas típicos de alineamiento de secuencias como CLUSTAL.
- Construcción del modelo. Existen varias aproximaciones para construir las coordenadas espaciales de la secuencia diana desde el alineamiento realizado.

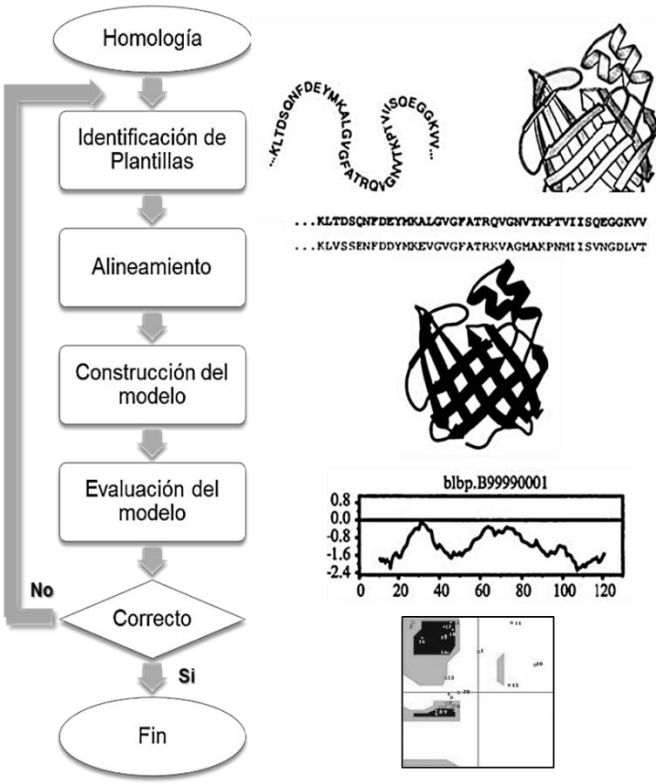


Figura 6. Algoritmo de la predicción de estructuras por homología.

- Evaluación del modelo. La información que se puede obtener del modelo depende de su calidad, de modo que es importante poder evaluarla. Entre las pruebas existentes que se pueden realizar sobre un modelo se incluyen comprobaciones estéricas, químicas, representaciones de Ramachandran[1], etc.

Se estima que el modelado por homología sólo es aplicable a un tercio de todas las secuencias proteicas.

2.3.4.- Threading (Hilvanado).

Cuando la similitud entre la secuencia objetivo y la plantilla es demasiado baja no es posible realizar un buen alineamiento y no se puede aplicar con éxito el modelado por homología. En estos casos aún podemos obtener información estructural de la proteína empleando técnicas de *threading* [28], [40], [43], [44].

Este método consiste en colocar la secuencia objetivo en diferentes plegamientos conocidos y evaluar cómo se "encuentra de bien" o cómo encaja en cada uno de ellos. Para este fin, por "encajar" se entienden cosas diferentes según el tipo de threading: coincidencia de estructura secundaria, residuos en ambientes parecidos a como se encuentran en la base de datos, etc.

2.5.- Los mapas de contactos interresiduales de proteínas.

Los mapas de contacto interresiduales son una importante representación bidimensional de la estructura espacial de las proteínas y tienen potencial aplicación en el área del entendimiento de los mecanismos de pliegue de las proteínas [45]–[47].

2.5.1.- Principios teóricos.

Los mapas de contacto son un paso crítico en la predicción de estructura de proteínas (problema no resuelto), razón por la cual se presta mucha atención en la predicción de mapas de contacto. Los mapas de contactos constituyen una "huella dactilar" de las proteínas, debido a que a partir de estos pueden identificarse algunos atributos de su conformación como estructura secundaria, topología de pliegues, entre otros [48] (Figura 7).

[1] Gráfico donde se pueden visualizar todas las combinaciones posibles de ángulos diédricos Ψ (psi) contra Φ (phi) en los aminoácidos de un polipéptido, y que contribuyen a la conformación de la estructura de las proteínas.

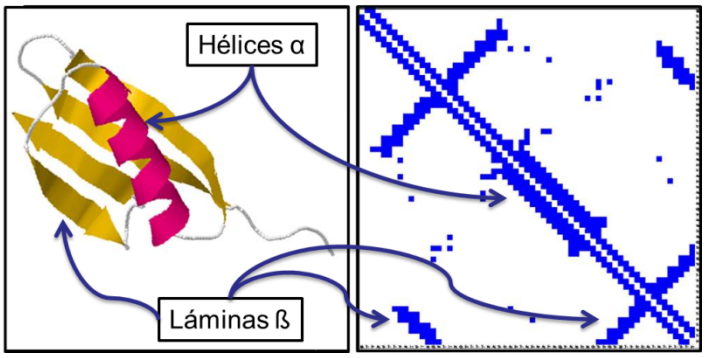


Figura 7. Representación del mapa de contacto de la proteína.

Esta información contribuye a entender cómo se organizan los residuos en el espacio y a descifrar los procedimientos de pliegue de las proteínas [4], [5]. De igual manera, la información del mapa de contacto se emplea en la predicción de estructuras desconocidas y funciones de las proteínas [51].

En los últimos 30 años, se han desarrollado múltiples métodos destinados a la predicción de mapas de contacto, tales como aproximaciones estadísticas basadas en las mutaciones correlacionadas [29], métodos basados en algoritmos de aprendizaje [9, 10] y aproximación basada en la clasificación de residuos [54]

Sin embargo, los resultados de las predicciones aún son insatisfactorios, debido a la naturaleza no balanceada de los mapas de contacto entre residuos, así como la limitada formulación de reglas de los diferentes métodos.

2.5.2.- Representación.

Existen tres tipos de representaciones diferentes de los mapas de contacto: los mapas de contactos basados en distancias o mapas de distancias, los mapas de contacto binarios y los mapas de contactos difusos. Cada uno de ellos aporta informaciones diferentes acerca de la estructura de la proteína.

Mapas de distancia.

En este tipo de representación de la estructura tridimensional de una proteína, se emplea una matriz simétrica, cuadrada de valores reales (Figura 8), que muestran las distancias entre todos los residuos que componen la proteína [33], [55].

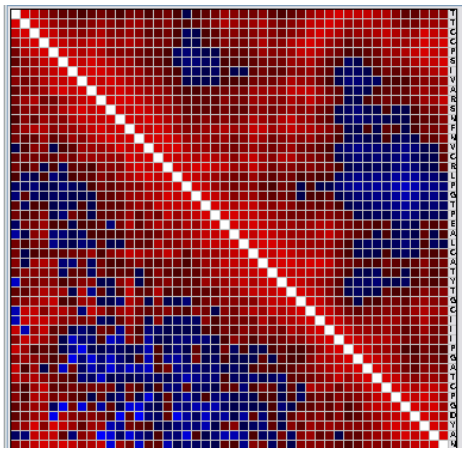


Figura 8. Representación de los mapas de contacto basados en distancias.
Mapa de distancias deseado (triangular superior) y predicho (triangular inferior). La representación de las distancias se realizó empleando 20 tonos de discretización de los colores rojo (distancias pequeñas), al azul (distancias grandes).

La distancia puede medirse entre los átomos Cα-Cα [12, 13], entre átomos Cβ-Cβ o tomarse la mínima distancia entre átomos pertenecientes a la cadena, o núcleo del par de residuos [3, 4].

Como medida de distancia espacial entre los centros geométricos de los residuos, se emplea la distancia Euclídea (1):

$$D_{i,j} = \left| \overrightarrow{r_i} - \overrightarrow{r_j} \right| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (1)$$

El empleo de este tipo de representación, combinado con un adecuado esquema de discretización del color, permite crear una idea más clara de la interacción entre los residuos de la proteína (cuestión importante, debido que puede ayudar en la reconstrucción de la estructura 3D de la proteína).

Mapas de contacto binarios.

Los mapas de contacto binarios son una representación compacta de la estructura tridimensional de una proteína en una matriz simétrica, cuadrada y booleana que muestra los contactos entre residuos (Figura 9) [57], [58]. Dos aminoácidos de una proteína se contactan entre sí formando una interacción no covalente (uniones de hidrógeno, efecto hidrofóbico, etc.). Normalmente, se dice que dos aminoácidos entraron en contacto, cuando se encuentran a al menos algún valor de umbral. El valor de umbral más común es 8 Angstrom (1 Å = 1 x 10^{-10} m = 0,1 nm).

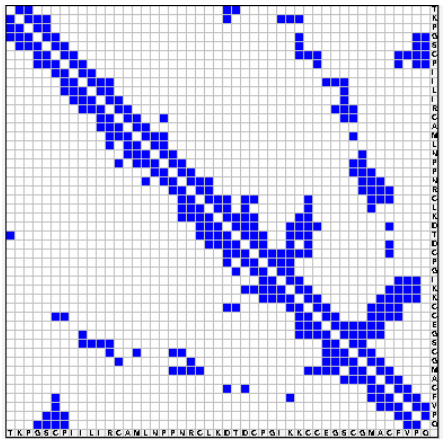


Figura 9. Representación de los mapas de contacto binarios.
Mapa de contacto deseado (triangular superior) y predicho (triangular inferior) de la proteína 1flel, a un umbral de 8 angstrom.

Dada una secuencia de proteína con N aminoácidos, S = $\{r_1, r_2, r_3,\ldots, r_N\}$, dos residuos r_i y r_j donde $(0 < i, j \leq N)$ se consideran contactos si la distancia espacial entre ellos es menor que un umbral C_t.

El mapa de contacto de una proteína de longitud N puede representarse mediante una matriz cuadrada, de N x N donde sus elementos se definen como (2 ó 3):

$$C_{i,j} = \begin{cases} 1 \text{ contacto entre los residuos } i \text{ y } j \\ 0 \text{ en otro caso} \end{cases} \quad (2)$$

$$C_{i,j} = \begin{cases} 1 \text{ si } D_{i,j} < C_t \\ 0 \text{ en otro caso} \end{cases} \quad (3)$$

Donde C_t es el umbral y $D(r_i, r_j)$ la distancia entre los pares de residuos (r_i, r_j).

Ésta es una representación evidentemente redundante, debido a que sólo se requieren $N(N-1)/2$ grados de libertad en lugar de N^2. Sin embargo, este tipo de representación sigue siendo importante, debido a que la redundancia puede ayudar en la reconstrucción de la estructura tridimensional (3D) de la proteína [48].

Selección del umbral adecuado.

La selección del valor de umbral, en Angstrom (Å), es un factor muy importante sobre el cual existen diferencias de criterios y que definen el número de contactos que se tendrían en cuenta. El menor umbral propuesto es 4.5 Å, con el objetivo de garantizar que no sea ni el tercer residuo ni una molécula de agua entre dos aminoácidos. A esta

propuesta se le agrega el no tener en cuenta los contactos entre residuos cuya separación en la secuencia sea menor de 4 residuos, lo cual evita la aparición de pequeños contactos falsos dados por la cercanía entre los residuos [59].

Se han realizado múltiples estudios relacionados con la selección del umbral adecuado, en la mayoría de los casos coinciden en que se encuentra entre 5, 6, 7, 8 y 12 Å [23], [60]–[62]. Además, se propone el empleo de una separación ≤ 7 residuos, con el objetivo de prever el efecto de aprendizaje de contactos locales y, particularmente, para polarizar las predicciones de los contactos en las inversiones y las hélices [48].

Mapas de contacto difusos.

Los mapas de contacto difusos fueron introducidos con la intención de tomar en cuenta el error potencial de medición en las coordenadas de los átomos y resaltar las características que aparecen a diferentes valores de umbral [63].

La definición formal de un contacto difuso está dada por la ecuación (4):

$$F_{i,j} = \mu\left(\left[\overline{i,\ j}\right], Ct\ \right) \tag{4}$$

Donde, $\mu()$ es una definición particular del contacto difuso, $\left[\overline{i,\ j}\right]$ es la distancia Euclidiana entre los residuos i, j y C_t es el umbral. La Figura 10 muestra definiciones alternativas para los contactos. Cada panel en la figura es un mapa de contacto difuso donde los puntos aparecen para cada par de residuos donde $F_{i,j} > 0$.

Los mapas de contacto se pueden generalizar eliminando las restricciones (en el modelo original) de tener un solo umbral R, como distancia de referencia. La definición formal de Contacto Difuso General (GFC) estaría dada por (5):

$$F_{i,j} = \max\ \left\{\mu_1\left(\left[\overline{i,\ j}\right], \Re_1\right),\ldots,\ \mu_n\left(\left[\overline{i,\ j}\right], \Re_n\right)\right\} \tag{5}$$

Con el mapa de contacto C definido como (6):

$$C^{r\,x\,r} = (F_{i,j})\ :\ 0 \le i, j \le r \tag{6}$$

Esto significa que pueden existir hasta n umbrales diferentes y n interpretaciones semánticas diferentes de "contacto" que pueden emplearse para definir el mapa de

contacto *r* x *r* siendo r el número de residuos en la proteína. Donde las funciones de pertenencia μ_1 y μ_2 se definen para patrones cortos y largos.

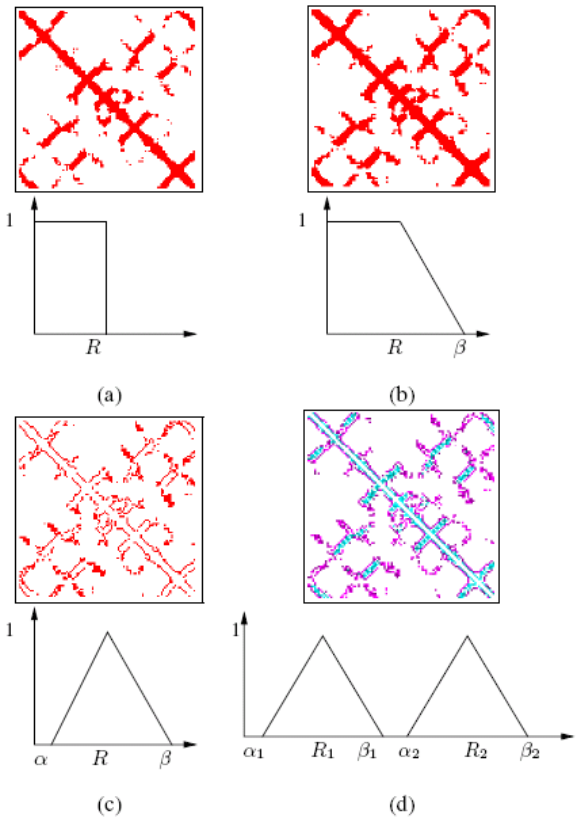


Figura 10. Ejemplos de mapas de contacto difusos.
(a) Modelo estándar. (b) Generalización difusa simple. (c) Otra generalización. (d) Mapa de contacto difuso de 2 umbrales y 2 pertenencias.

La Figura 10 muestra varios tipos de mapas de contactos difusos. Desde el punto de vista de la implementación, la parte superior del triángulo del mapa de contacto contiene los valores $F_{i,j\in}$ [0, 1], mientras que la parte inferior contiene el índice de la función de pertenencia donde se encuentra el máximo de la función (6). Para este caso los valores que tomaría el mapa serían 0 para los no contactos, 1 para los contactos cortos y 2 para los contactos largos.

2.5.3.- Codificación.

Para realizar el proceso de entrenamiento del predictor, es necesario determinar el método adecuado de codificación de la información contenida en el mapa de contacto. Esta codificación está en función propiamente de dicha información y del predictor que se implemente.

Basada en la frecuencia de ocurrencia de los contactos.

Uno de los métodos más empleados para la codificación de los mapas de contactos es el empleo de la frecuencia de ocurrencia de los contactos en la base de datos de proteínas como función de la separación de los residuos en la secuencia (Figura 11) [4, 15].

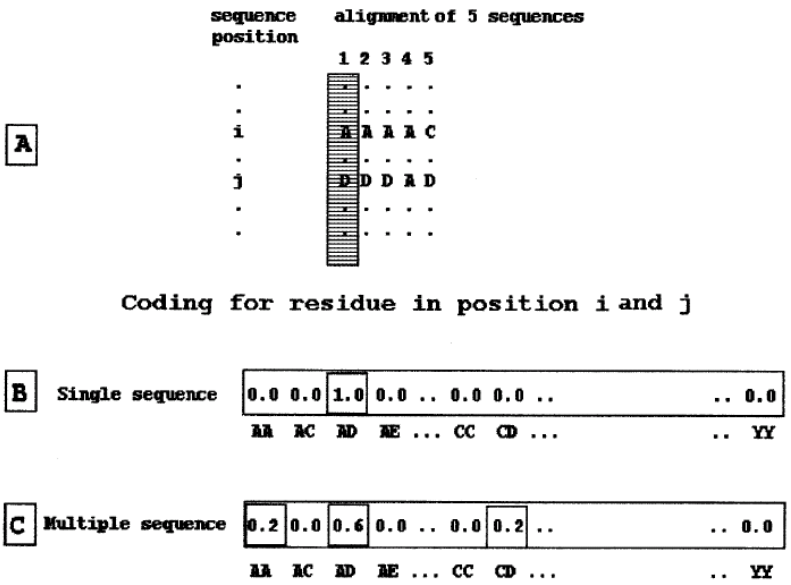


Figura 11. Representación de los códigos basados en las parejas ordenadas
(A) representación de secuencias previamente alineadas. (B) codificación simple, se asigna el valor 1 al par correspondiente a los residuos i y j, al resto de los pares se le asigna el valor 0. (C) codificación de múltiples secuencias, toma en cuenta la frecuencia de aparición de cada par de aminoácidos en las posiciones i y j de las secuencias alineadas, al resto de los pares se le asigna el valor 0.

Este método consiste en representar los códigos basados en las parejas ordenadas. En la Figura 11.A, se representa una alineación de cinco secuencias (hidrofóbicas) en un fichero HSSP [50]. Donde i y j son índices de la posición de los dos residuos hagan o no contacto (A y D en la secuencia inicial o secuencia 1).

La codificación más común es la "simple" (Figura 11.B). Donde la posición que representa las parejas (AD) en el vector está a 1, mientras el resto de las posiciones están en 0. Otro modo de realizar la codificación es mediante el empleo de múltiples secuencias (Figura 11.C). En este caso, para cada secuencia en la alineación (1 a 5 en la Figura 11.A) se cuenta una pareja de residuos en posición i y j. El código final de entrada representa la frecuencia de cada pareja en la alineación, el cual se normaliza para el número de secuencias.

Basada en conservación y mutaciones correlacionadas.

Ésta, es un tipo de codificación más compleja, en la cual se tienen en cuenta las características químicas que tienen algunas posiciones, con el objetivo de mantener su función y ajustes necesarios para mantener la proteína estable ante la tendencia a mutar.

Para incluir la conservación de las secuencias en la codificación, se toma en cuenta la variabilidad de la secuencia. Haciendo cero la variabilidad cuando las posiciones en múltiples secuencias alineadas se mantienen conservadas completamente (invariables) e incrementándose proporcionalmente con el número de cambios en los aminoácidos en dicha posición.

Mientras que las mutaciones correlacionadas son calculadas como un vector de distancias empleado para codificar cada posición en el alineamiento (Figura 12). El vector de la posición específica, contiene todas las distancias residuo – residuo entre todos los pares posibles de las secuencias en dicha posición. El valor de correlación entre cada par de posiciones en la alineación es calculado como la correlación de dos arreglos para cada par de residuos posibles. Los elementos correspondientes en los arreglos contienen las distancias entre las mismas dos secuencias en las dos posiciones bajo comparación.

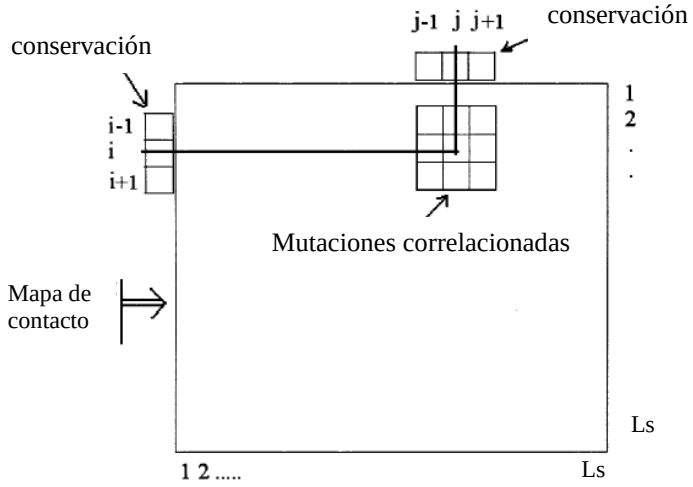


Figura 12. Conservación y Mutaciones correlacionadas.

La matriz de puntuación define las distancias entre los residuos. Las posiciones con un porcentaje de diferencia ≥ 10% son puestas a un valor de correlación de –1 y las posiciones conservadas son puestas al valor de correlación 0.

Codificación binaria.

Estrategia de codificación binaria, empleando 19 bits, que integra las características de los pares formados: clasificación de los residuos, estructura secundaria, longitud de la secuencia e información sobre la separación en la secuencia [61]. Diseñada con vista a su empleo en algoritmos genéticos (Figura 13), con esta codificación se busca un esquema capaz de capturar las reglas de mapeo de los pares de residuos, derivadas de la secuencia primaria, para la conformación de los contactos [16, 17, 21].

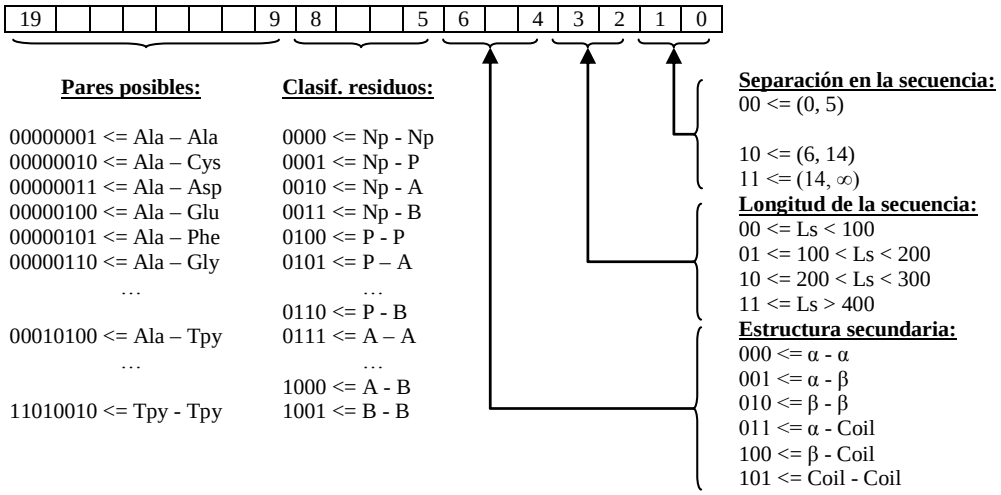


Figura 13. Codificación binaria, empleando 19 bits.
En la columna de clasificación: Np (no polar); P (polar); A (ácido); B (base). En la columna de separación en la secuencia se adopta la definición de Park y colaboradores en el 2001 [51], donde las interacciones entre residuos son agrupadas en largas (mayores o iguales a 14), las medias entre (6, 13) y las pequeñas (1, 5).

Para cada par de residuos (r_i, r_j) de la secuencia, son adoptados cinco grupos de atributos para formular las relaciones entre la secuencia primaria y el mapa de contacto, con el propósito de capturar diferentes aspectos de los aminoácidos y las posiciones en la secuencia. Estos atributos incluyen posibles parejas de residuos, clasificación química, estructura secundaria, longitud de los residuos y la separación en la secuencia.

Capítulo 3

Fundamentación sobre multi-clasificadores

El reconocimiento de patrones es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos o abstractos, con el propósito de extraer información que permita establecer propiedades entre conjuntos de objetos. Uno de los principales objetivos del reconocimiento de patrones es la clasificación, donde se quiere clasificar un objeto dependiendo de sus características. Una forma de abordar el problema de la clasificación es la utilización de multiclasificadores [65].

3.1.- Introducción

Un multiclasificador es un conjunto de clasificadores, conocidos como clasificadores base. Estos combinan sus predicciones siguiendo un determinado esquema, con el fin de obtener una predicción más fiable que la que normalmente serían capaces de obtener en solitario.

La combinación de clasificadores ha sido abordada en la literatura a través de distintos términos, entre ellos: ensamblados (*ensembles*) [66], [67]; modelos múltiples (*multiple models*) [68], [69]; sistemas de múltiples clasificadores (*multiple classifier systems*) [70]; combinación de clasificadores (*combining classifiers*) [71]; integración de clasificadores (*integration of classifiers*) [72]; mezcla de expertos (*mixture of experts*) [73], [74]; comité de decisión (*decision committee*) [75]; comité de expertos (*committee of experts*); fusión de clasificadores (*classifier fusion*) [76], [77] y aprendizaje multimodelo (*multimodel learning*).

Existen tres razones fundamentales que justifican el uso de un esquema de combinación de clasificadores, en lugar un único clasificador:

1. Estadística: elegir y usar un único clasificador entre varios es arriesgado, aun cuando el error de entrenamiento de ese clasificador sea cero, debido a que no se conoce la respuesta que va a tener frente a datos desconocidos. En ese sentido, combinar varios clasificadores no es mejor que quedarse con el mejor clasificador posible, pero reduce el riesgo de tomar uno que esté lejos de serlo (Figura 14).

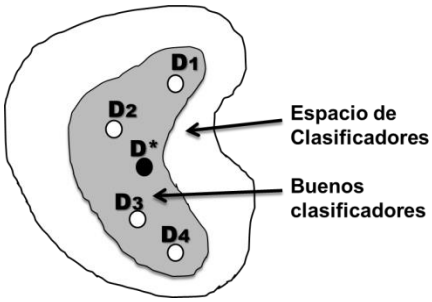


Figura 14. Razón estadística. El espacio de búsqueda de los clasificadores D1, D2, D3 y D4 permiten encontrar la solución que devolvería el clasificador ideal D*.

2. Computacional: los algoritmos de entrenamiento de muchos clasificadores dependen de algún elemento que los hace llegar a un mínimo local del espacio de posibles clasificadores para un conjunto de datos dado. La combinación de varios clasificadores puede atenuar este efecto (Figura 15).

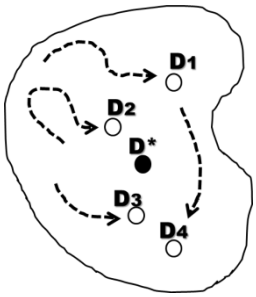


Figura 15. Razón computacional. D* es el clasificador ideal y las líneas discontinuas son las trayectorias.

3. Representación del problema: se basa en las limitaciones de los clasificadores para lograr un modelo que represente adecuadamente un determinado problema. Por ejemplo, en problemas no separables linealmente, no debe

emplearse un modelo lineal, sin embargo, es posible que la combinación de varios clasificadores lineales puedan lograr una aproximación que se adapte mejor al problema (Figura 16).

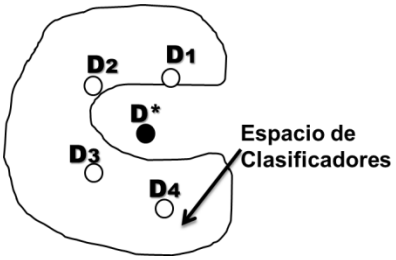


Figura 16. Representación del problema. D* es el clasificador ideal y la zona circulada es el espacio de búsqueda de los clasificadores seleccionados.

3.2.- Construcción de multiclasificadores

La arquitectura de un multiclasificador depende del esquema que se emplee al combinar los clasificadores base para garantizar una toma de decisión. La combinación puede ser por eliminación de hipótesis (decisiones dependientes), con independencia entre ellos o a través de la cooperación de clasificadores (cada uno soluciona un problema) [78], [79]. Teniendo en cuenta esto, los multiclasificadores pueden adoptar distintas arquitecturas: en serie (secuencial o vertical), paralela (horizontal), e híbrida (mezcla de la arquitectura serie con la paralela, con interacción, etc.).

Arquitectura serie

En el modelo secuencial o vertical la información resultante de un clasificador se convierte en información de entrada para otro. La Figura 17 muestra los esquemas de este modelo. Los niveles de decisión sucesivos permiten reducir progresivamente el número de clases posibles. Un único clasificador por nivel tiene en cuenta la respuesta proporcionada por el clasificador colocado anteriormente y debe tratar los rechazos y confirmar la decisión obtenida en el eslabón anterior. Con este objetivo existen los enfoques de reducción del conjunto de clases y el de re-evaluación.

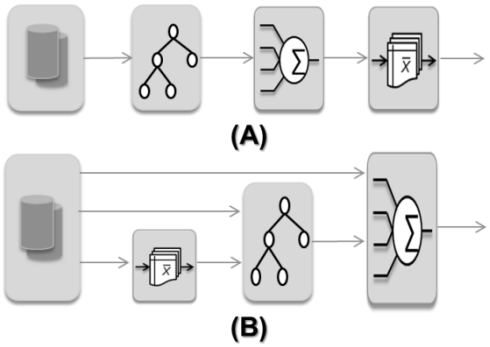


Figura 17. Arquitectura serie. (A) enfoque de reducción del conjunto de clases (B) enfoque de reevaluación.

En el enfoque de reducción del conjunto de clases (Figura 17.A), el clasificador primario en la configuración genera una lista de posibilidades que constituye un subconjunto del número total de clases. El resto de los clasificadores se limitan a analizar el subconjunto de clases generado por el clasificador anterior. Esta arquitectura está acompañada de un filtrado progresivo de las decisiones destinada a la reducción de la ambigüedad.

En el enfoque donde se emplea la re-evaluación (Figura 17.B), cada clasificador realiza la búsqueda de una solución en todo el dominio de clases. En lugar de indicar la clase a que los datos pertenecen, genera un valor de confianza que se corresponde al indicador de decisión. Este proceso se repite sucesivamente hasta que uno de los clasificadores encuentre un valor de confianza suficientemente alto o el clasificador final emita su decisión.

En sentido general, la arquitectura serie es altamente sensible al orden en el cual se colocan los clasificadores, por lo que se debe tener un conocimiento a priori del comportamiento de cada uno de los clasificadores. De manera general, es difícil de optimizar el conjunto ya que existe dependencia.

Arquitectura paralela

La arquitectura paralela es un esquema muy fácil de aplicar (todo lo contrario a la arquitectura serie). Los clasificadores operan independientemente unos de otros las respuestas de cada uno se fusionan en busca de un consenso entre los clasificadores para llegar a una única decisión (Figura 18). Éstos, no requieren de una reparametrización en caso de que existan modificaciones en el conjunto. La desventaja

de esta arquitectura radica en que la activación de todos los clasificadores conlleva a un elevado costo computacional.

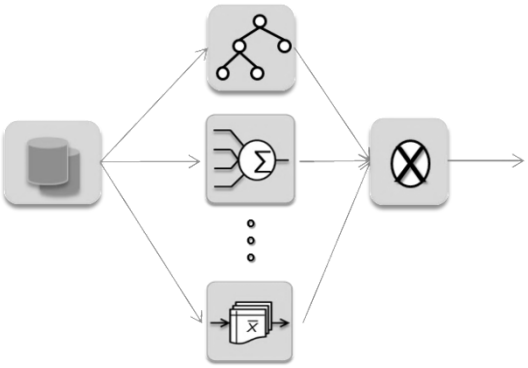


Figura 18. Arquitectura horizontal o paralela.

Arquitectura híbrida

La arquitectura híbrida busca combinar las ventajas de las arquitecturas anteriores (serie y paralela). Ésta, reduce el conjunto de las clases posibles y lograr un consenso entre los clasificadores (Figura 19).

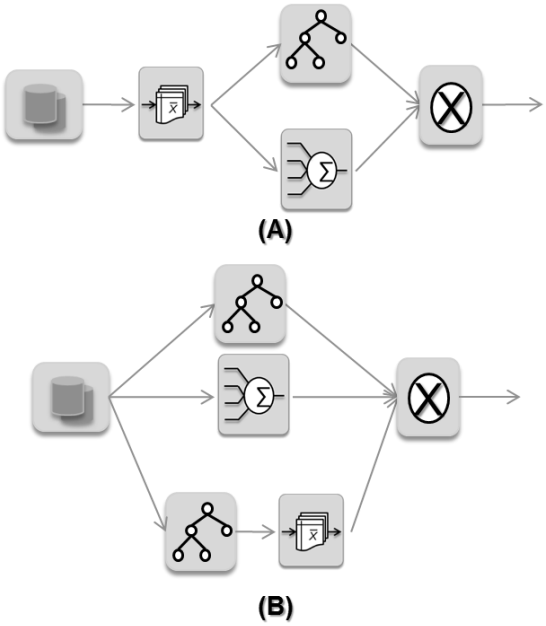


Figura 19. Esquemas de arquitecturas híbridas. (A) serie – paralela (B) paralela con serie incluida.

Teóricamente, esta arquitectura debe lograr mejor provecho de cada uno de los clasificadores utilizados. En la actualidad existen múltiples esquemas de combinación para extraer lo mejor de los datos. En el esquema que se propone en la Figura 19.A, los

datos deben tratarse de forma dependiente y resulta aún más complejo de optimizar que el esquema de la Figura 19.B o en la arquitectura serie.

Niveles de actuación para crear multiclasificadores

Según Kuncheva [65], existen cuatro niveles de actuación en la construcción de multiclasificadores (Figura 20):

1. Nivel de combinación: a este nivel existen distintos modos de combinación de las predicciones individuales de los clasificadores base.

   Existen dos estrategias fundamentales para la combinación de clasificadores: la selección y la fusión. En la fusión de clasificadores, se supone que cada miembro del ensamblado tiene conocimiento de todo el espacio de características. En la selección de clasificadores, se supone que cada miembro del ensamblado conoce sólo una parte del espacio de características y es responsable de los objetos en ese espacio.

   Selección: ésta es, probablemente, la mejor de las dos estrategias, sin embargo, no es la estrategia más empleada en la actualidad.

   Fusión: se establecen estrategias que permitan que las salidas de cada clasificador se puedan fusionar en una sola para el conjunto. Aunque las salidas de los clasificadores individuales así como la del conjunto pueden tomar diferentes formas, la filosofía de la mayoría de las técnicas se puede aplicar a cualquiera de ellos, siempre atendiendo a las ventajas y desventajas del empleo de cada una (Tabla 3). Algunos de estos métodos son:

   - Métodos de nivel abstracto (*Abstract-level methods*)
     - Voto mayoritario simple.
     - Voto mayoritario por peso.
     - Reglas basadas en el enfoque de Bayes.
   - Métodos de nivel de rango (*Rank level methods*)
     - Método de la cuenta de Borda.
     - Método de la cuenta de Borda por peso.

- Métodos de nivel de medidas
    - Reglas de promedio simple, el producto y otros operadores estadísticos.
    - Operadores pesados.

Tabla 3. Comparación entre los diferentes grupos de métodos de fusión.

Métodos	Ventajas	Desventajas
Nivel abstracto	• Puede ser aplicado siempre.	• Demanda alta calidad y cantidad de datos.
Nivel de rango	• Es conveniente en problemas con muchas clases, donde la clase correcta puede aparecer a menudo cerca de la lista superior. • Puede ser preferido a las salidas para evitar la carencia de la consistencia al usar diversos clasificadores. • Puede ser preferido a las salidas para simplificar el diseño de combinación.	• No están soportados sobre una base teórica. • Los resultados dependen en la escala de los números asignados a las diferentes opciones.
Nivel de medidas	• Puede explotar una cantidad de información más alta con respecto a los otros métodos de fusión. • Los combinadores complejos pueden ser diseñados y requeridos con frecuencia por clasificadores que exhiben diferentes funcionamientos y correlaciones complejas.	• Se requiere normalización de las salidas de los clasificadores utilizan si se diferentes clasificadores. • Es necesario con frecuencia conjuntos de datos de gran tamaño y buena calidad

2. Nivel de clasificadores base: selección del tipo de clasificadores base que se van a utilizar.
3. Nivel de las características: es posible obtener distintos clasificadores base quitando, añadiendo y/o modificando las características del conjunto de datos. Empleando distintos subconjuntos de características para cada clasificador base, aun cuando éstos sean del mismo tipo.

4. Nivel del conjunto de datos: lograr, mediante algún criterio, que los conjuntos de datos de cada clasificador base sean diferentes, aun cuando los clasificadores base sean del mismo tipo.

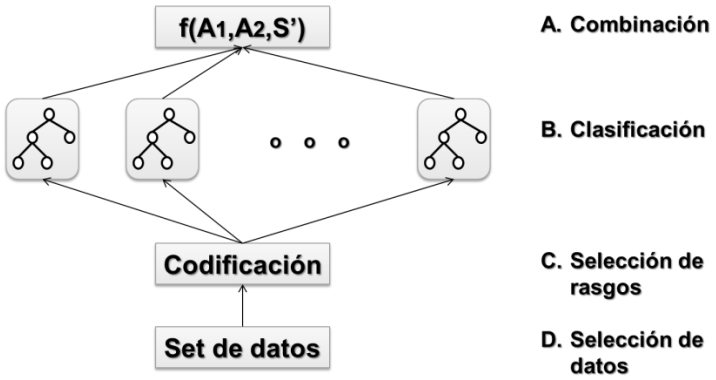


Figura 20. Modelo general de actuación para la creación de multiclasificadores.

Una cualidad que, idealmente, deberían presentar los clasificadores base de un multiclasificador es la diversidad. Y es en virtud de la cual, las predicciones de los mismos tienden a ser distintas. Construir un multiclasificador exitoso requiere que sus clasificadores base acierten en la mayoría de sus predicciones, pero que cuando se equivoquen cada uno lo haga en distintas instancias; de lo contrario la ventaja de tener varios clasificadores colaborando, frente a uno solo, no existiría.

Los multiclasificadores pueden ser catalogados atendiendo a estas tres perspectivas:

1. Fusión contra selección: se refiere al método de combinación de los resultados de los clasificadores base.
2. Optimización de la decisión contra optimización de la cobertura. Los métodos con optimización de la decisión se centran en cómo combinar la decisión de los clasificadores base. Mientras que los de optimización de la cobertura, asumiendo cualquier técnica de combinación, se centran en cómo entrenar a los clasificadores base para lograr la diversidad.
3. Entrenables contra no entrenables: se refiere a si el multiclasificador sigue un esquema fijo de combinación o si es necesario un entrenamiento extra en la etapa de combinación.

Los multiclasificadores pueden ser catalogados desde estas tres perspectivas. Sin embargo, ninguna de las tres categorías por separadas establece una frontera clara que permita incluir a todo multiclasificador a un lado u otro de la misma.

3.3.- Algoritmos clásicos

Existen varios métodos propuestos para la construcción de multiclasificadores. Éstos, podrían agruparse en aquellos que utilizan un único algoritmo de construcción de clasificadores base y en aquellos que combinan varios algoritmos de construcción de clasificadores para crear un único clasificador final.

En el primer grupo se encuentran: *Bagging, Random Forests, Random Subspaces* y *Boostinq*. Estos métodos logran la diversidad variando los datos de entrenamiento que procesan, ya sean instancias o atributos. En el segundo grupo se encuentran: *Cascading, Stacking y Grading*. Estos métodos, los datos que se procesan incluyen la información de salida del nivel anterior.

Bagging [80]: construye N clasificadores base, empleando el mismo algoritmo, pero con distintos conjuntos de entrenamiento. Estos conjuntos se obtienen mediante remuestreo con reemplazo (*bootstrap*) de un determinado porcentaje de instancias del conjunto de entrenamiento original (por lo general se emplea el 100%).

Al elegir correctamente un buen clasificador base de *Bagging* es preferible que sea sensible a las pequeñas variaciones que introduce el remuestreo en el conjunto de entrenamiento (clasificadores inestables). El resultado final del multiclasificador es la clase de más votos a partir de las predicciones de sus clasificadores base.

Random Forests [81], [82]: engloba cualquier multiclasificador basado en árboles, en el que se varía algún parámetro de cada árbol de forma aleatoria. Breiman describe dos implementaciones de este multiclasificador que son variantes de *Bagging* que emplean árboles sin podar: *Forest-RI* (se ramifica a partir de subconjunto de atributos del conjunto inicial) y *Forest-RC* (construye nuevas características en cada uno de los árboles, como combinación lineal de los atributos del conjunto inicial). En ambos casos se emplea el remuestreo aleatorio y la predicción por medio de voto mayoritario.

Como característica distintiva, cada clasificador base mantiene la diversidad que le viene dada por el remuestreo con reemplazo, la cual aumenta teniendo en cuenta que

en cada nodo varía de forma aleatoria los atributos a considerar. Y, al no estar podados, mantiene la inestabilidad necesaria para los algoritmos. Este clasificador es robusto al ruido y altamente paralelizable.

Random Subspace [83]: construye cada clasificador base empleando un subconjunto aleatorio del total de todos los rasgos del conjunto de entrenamiento inicial. El número de atributos para cada clasificador base siempre es el mismo y debe ser el 50% aproximadamente. La clasificación se obtiene promediando las respuestas de los clasificadores base.

Boosting [84]: construye iterativamente los clasificadores base, de manera que en cada iteración se proporcione más importancia a las instancias que han sido mal clasificadas hasta el momento. El resultado final de la clasificación se realiza a través de la ponderación del voto de los clasificadores base, asignando mayor peso a los que mayor tasa de acierto tienen.

Dentro de los algoritmos adaptativos de la familia *Boosting*, dos de los más empleados son el *AdaBoost* (*AdaBoostM1* y *AdaBoostM2*) [85] y el *MultiBoosting* [86].

Cascading [87]: tiene una arquitectura generalmente de dos niveles. El primer nivel se entrena con el conjunto de datos original. El segundo, con un conjunto de datos aumentado, que contiene al conjunto original más la salida del primer nivel de clasificación. La salida de este primer nivel es un vector con la estimación de probabilidad condicional de cada clase. En cada nivel se emplea un clasificador de naturaleza distinta al otro.

Stacking [88]: clasificador de tipo jerárquico. Puede tener varios niveles aunque lo usual son dos. En el primer nivel, por lo general, hay varios clasificadores entrenados con algoritmos diferentes pero con conjuntos de entrenamiento que tienen el mismo espacio de características. En el segundo nivel existe un único clasificador que toma como entrada las salidas de los niveles inferiores. Al igual que en *cascading*, los clasificadores de cada nivel son de naturalezas distintas. Este algoritmo suele degradarse con datos multiclases.

Grading [89]: es muy similar a *stacking*, solo que en el primer nivel realiza una clasificación previa que sirve como entrada al siguiente nivel, el nivel superior decide

qué clasificadores del nivel anterior estaban en lo cierto. En caso de conflicto, se realiza una votación haciendo uso de la confianza en la predicción de los clasificadores base. Si la votación fuera empate, entonces se toma la clase mayoritaria en el conjunto de entrenamiento como decisión final.

Además de estos algoritmos de multiclasificación clásicos, existen otros muchos esquemas de ensamblado de clasificadores, entre los que se encuentran:

Modelo de selección de clases (MCS) [90]: que divide el conjunto de clases en subespacios a través de reglas obtenidas empíricamente. Asigna a cada espacio un clasificador de tres posibles (árboles de decisión, función discriminante o clasificador basado en instancias).

Mezcla de expertos (ME) [91]: igual que MCS pero los subespacios pueden estar solapados entre sí. Otro clasificador combina las salidas de los expertos. Existe la variante jerárquica (HME) en que los espacios se descomponen recursivamente en nuevos subespacios.

Árbitros de árboles (AT) [92]: se parte de un número de particiones disjuntas del conjunto de entrenamiento. Con cada una se entrena un clasificador base del primer nivel. Estos clasificadores se emparejan y por cada pareja se entrena otro de nivel superior que actúa como árbitro, que también es un árbol. Este esquema se extiende recursivamente.

Combinación de árboles (CA) [92]: es similar a AT, solo que los clasificadores que no son hojas se entrenan en las salidas del nivel anterior. Emplean las mismas reglas que *cascading* o *satcking*.

NBTree [93]: este multiclasificador es un árbol que en sus hojas hay un clasificador *Naïve Bayes*.

3.4.- Tipos de errores en los multiclasificadores

Además de la arquitectura que se emplee para crear un multiclasificador, una diferencia que resalta entre ellos es el tipo de error que generan en la clasificación. Si se analizan las predicciones para cada instancia de un conjunto de entrenamiento dado, se podría calcular con qué probabilidad se predice cada clase por los clasificadores obtenidos a partir de cada algoritmo.

Tendencia central de una instancia [86], es la denominación que se emplea para la clase mayoritaria predicha por los predictores de un determinado algoritmo y para un conjunto de datos determinado. O sea, no es más que la predicción realizada por el multiclasificador al combinar las predicciones por separado de cada clasificador base entrenados en dicho conjunto de entrenamiento. Tomando como base la tendencia central, el error puede descomponerse en dos componentes que se conocen como **bias** y **varianza** [94]–[96].

- **Bias**: es el error que se produce cuando el clasificador se equivoca al predecir la tendencia central. Se debe a las propias limitaciones del clasificador. Por ejemplo, un clasificador lineal requiere regiones linealmente separables, si un grupo reducido de instancias está en el interior de una región en la que hay abundantes instancias de la clase contraria, las instancias de dicho grupo estarán normalmente en el lado incorrecto de cualquiera de los hiperplanos generados a partir de diferentes muestras del mismo conjunto de datos para ese mismo clasificador lineal. Por tanto, la tendencia central de ese clasificador será predecirla como perteneciente a la clase equivocada, y por tanto se genera un error debido a la componente **bias**.
- **Varianza**: es el error que se produce cuando el clasificador se equivoca al predecir cualquier otra clase que no sea la tendencia central. Este error aumenta por tanto, si para una misma entrada, el mismo algoritmo de clasificación (entrenado cada vez con un conjunto de entrenamiento distinto), es capaz de generar distintos clasificadores, habiendo muchos capaces de hacer predicciones distintas a la tendencia central. Debido a que el único cambio que se ha introducido para que tengan lugar estas diferencias es el conjunto de entrenamiento, es claro que este error se manifiesta cuanto mayor es la sensibilidad del algoritmo a dicho cambio. Por tanto, es una componente del error que aumenta cuando el algoritmo tiene en cuenta en exceso a las instancias que representan casos que se dan con poca frecuencia (*outliers*). Este error, es el habitual cuando el clasificador sufre un sobre entrenamiento.

En el caso de los árboles de decisión, no podarlos aumentaría la componente **varianza** del error respecto al caso podado. Por el contrario, una poda muy agresiva dará lugar a una alta componente de **bias** en el error.

Parte II

Estado del arte

Capítulo 4

Estado actual de las técnicas de predicción de contactos interresiduales

Como se planteó con anterioridad, la predicción de estructura de proteínas, se ha abordado empleando dos enfoques fundamentales: los métodos *ab initio* y los métodos de modelado por comparación.

Los métodos *ab initio* tratan de predecir el plegamiento sin tener en cuenta la información evolutiva de las proteínas. Estos métodos exploran la energía hipersuperficial para establecer una conformación de energía mínima, la cual se cree se corresponda con el estado nativo de la proteína [11], [13], [56], [97], [98]. Por otra parte, los métodos de homología molecular tratan el problema de la predicción a partir de la información que implica la caracterización de múltiples secuencias alineadas basadas en la conservación de las secuencias y las mutaciones correlacionadas [29], [40], [48], [59], [99].

4.1.- Valoración crítica del estado del arte.

La estadística ha sido el soporte de muchos análisis de datos biológicos por años, sin embargo, los datos biológicos han cambiado en tamaño y estructura. Usualmente los conjuntos de datos contienen miles de instancias y de atributos. Los algoritmos han evolucionado teniendo en cuenta estas características y han sido validados por medio de pruebas estadísticas con datos sintéticos y reales. Técnicas de minería de datos, inteligencia artificial, heurísticas complejas, entre otros, han ocupado este espacio con

el objetivo de manejar grandes volúmenes de datos, lograr optimización y eficiencia [100]–[103].

A pesar de la clasificación usual de las técnicas de predicción en *ab initio* o de modelado, en este trabajo se propone una nueva agrupación de éstas, teniendo en cuenta el paradigma a que pertenecen. La Figura 21 muestra una taxonomía que agrupa estas técnicas en métodos estadísticos, redes neuronales, máquinas de vectores soporte, bio-inspirados y combinación de clasificadores.

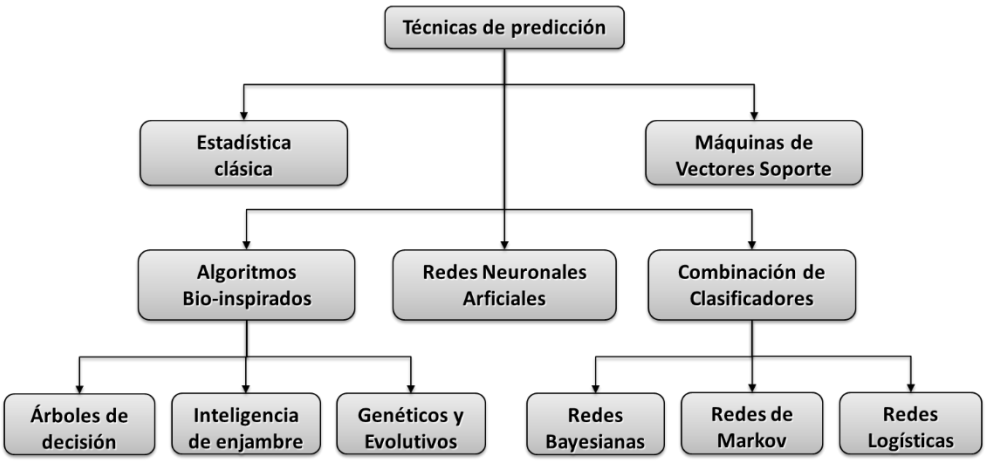


Figura 21. Taxonomía de métodos de predicción de estructuras de proteínas.

4.1.1.- Estadística clásica contra aprendizaje automático.

Los primeros modelos de predicción clasificaban los aminos ácidos de forma cualitativa como: destructores de hélices, formadores de hélices, helicoidal y antihelicoidal. También, realizaban el ajuste de fragmentos helicoidales de estructuras conocidas y ruedas helicoidales para localizar los arcos hidrofóbicos en la región helicoidal seleccionada [5].

En estos métodos se tuvieron en cuenta los parámetros de iniciación de las hélices (σ) y el ancho de las hélices (s), así como los ángulos del centro del aminoácido para varias secuencias tripeptídicas en proteínas de estructura conocida (ϕ, ψ). Lo cual permitió construir una tabla de 20x20 para todos los tripéptidos, mostrandose las frecuencias de ocurrencia en las regiones helicoidales y no helicoidales. Además, se pudo observar la no correlación debido a interacciones entre pares de residuos adyacentes en regiones helicoidales y no helicoidales. Teniendo esto en consideración, los autores llegaron a la conclusión de que la estructura secundaria de una cadena polipeptídica

depende principalmente de las interacciones en la cara de la cadena con el núcleo (*backbone*) más que las interacciones entre las caras de la cadena [104].

Para la identificación de las regiones de α hélices y las estructuras β en las proteínas globulares nativas, se propone un método basado en la teoría de la estructura secundaria de proteínas globulares solubles en agua, determinada mediante la técnica de difracción cristalográfica de rayos X. Se presenta un sistema de reglas, por medio de las cuales se puede localizar las regiones de α hélices y otras, 8 reglas específicas, para localizar las regiones en las estructuras β [105]. El uso de las propiedades hidrofóbicas de los aminoácidos de las proteínas es el aspecto más destacable de este método.

Más recientemente, se ha seguido incursionando en métodos como la predicción por regresión [106], desarrollado para los casos en que no sea encontrada una cadena homóloga a la cadena objetivo. La función de regresión se obtiene a partir del cálculo del orden de contactos absoluto. Esta ecuación surge de la observación de la correlación existente entre el orden de contactos (CO) y la combinación lineal del porcentaje de residuos en alfa hélices ($p_{(\alpha)}$), el porcentaje de residuos en cintas beta ($p_{(\beta)}$) y la longitud de la proteína (L). Esta técnica tiene el mérito de generar una ecuación lineal simple que suple la necesidad de búsqueda de cadenas homólogas. Sin embargo, tienen la desventaja de que asume la supuesta linealidad en la predicción de los contactos interresiduales lo que puede en determinadas ocasiones afectar la fiabilidad de las predicciones.

La implementación de los métodos estadísticos implica un bajo costo computacional, sobre todo debido a su simplicidad. Sin embargo, éstos presentan una serie de desventajas debido a que se basaban en pocas proteínas pequeñas (20) de estructura conocida, determinadas por cristalografía de rayos X [104], [105], en el análisis de secuencias homólogas [104]–[106], o asumen linealidad en la predicción de los contactos interrsiduales [106]. Además, la aparición de un conjunto de reglas cualitativas, implica la necesidad del empleo de sentido común para realizar las predicciones [104]–[106]. Todo ello sugiere la necesidad de emplear técnicas más poderosas como las de aprendizaje automático.

4.1.2.- Redes Neuronales Artificiales.

Con el objetivo de lograr predictores con mayor nivel de generalización y robustez, se comenzaron a emplear ampliamente las redes neuronales artificiales [107]. En este campo, ha aparecido una amplia gama de técnicas basadas en modelos Feed-Forward (ANN) [30], [42], [48], [108]–[112], modelos recurrentes (RNN) [33], [53], [62], [113], modelos que emplean funciones de base radial (RBFNN) [60], [61], [64], modelos de transciente caótico [114], entre otros [99].

Más recientemente, han sido empleadas las redes neuronales de campos condicionales (CNF) y de campos aleatorio (CRF). En estos casos las CNF han demostrado mayor efectividad que las CRF, mostrando un mejor desempeño en la predicción de la mayor parte de las α–proteínas y pequeñas β–proteínas, no siendo así para las proteínas β más grandes [99].

Topologías

En la mayoría de los modelos se emplea una capa de entrada (variable según la codificación realizada). Una capa oculta con un número de neuronas que varía en dependencia del modelo y que oscila entre dos y 300 neuronas [42], [48], [59], [108]–[110], [113], [114]. Y, por último, una capa de salida, cuyo número de neuronas también varía en dependencia de la predicción que se realice: una neurona si lo que se predice es si hay o no contacto [48], [53], [59], [113]; de dos a tres neuronas en dependencia de la codificación, si lo que se predicen son los motivos estructurales [42], [108]–[110]; otros casos emplean muchas más neuronas de salida para representar las estructuras 2D y 3D [50], [60], [61], [64].

Estos modelos varían tanto en su arquitectura (figura 7), en la codificación que emplean en sus entradas, como en los métodos de entrenamiento.

También han sido propuestas arquitecturas de esquemas de combinación de redes neuronales por niveles. Donde, en un primer nivel, una RNA se encarga de la predicción secuencia – estructura, para predecir la estructura secundaria; en un segundo nivel otra RNA realiza la predicción estructura – estructura, para predecir la estructura terciaria a partir de la secundaria; y, un tercer nivel de RNA que se encarga del promediado de la predicción[30]. Estos esquemas suelen ser verdaderamente

complicados en cuanto a su implementación y alto costo computacional. Sin embargo, emplean información evolutiva a partir del alineamiento de múltiples secuencias.

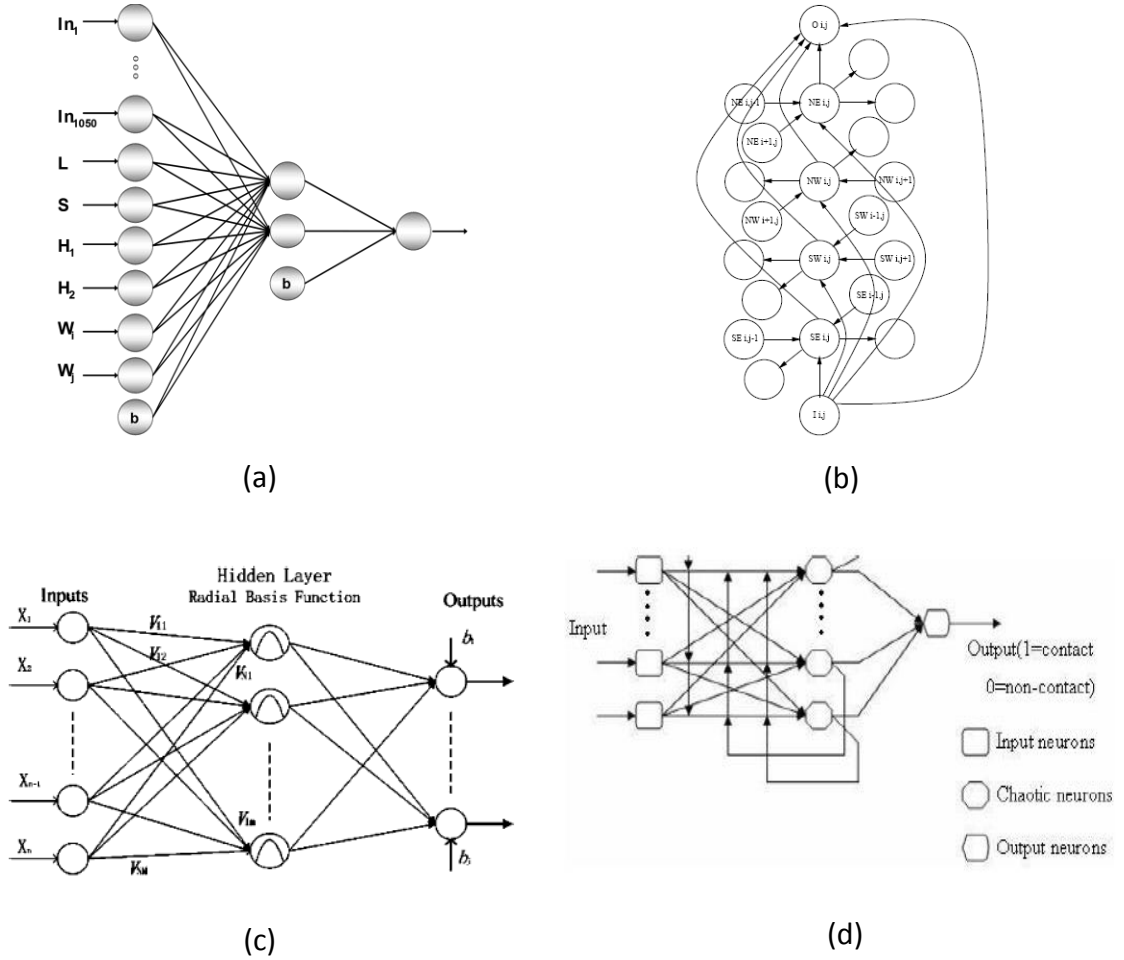


Figura 22. Topologías de redes neuronales, en dependencia del tipo de red.
(a) Ejemplo de modelo Feed-Forward (ANN), la arquitectura que se muestra es la propuesta por Fariselli y Casadio 2001 [48], la cual está compuesta por 1086 neuronas, distribuidas en: 1050 neuronas de entrada, 2 neuronas más para incluir la información de la longitud de la proteína y la separación en la secuencia (L y S), otras 2 neuronas para agregar la información de hidrofobicidad (H1 y H2), 2 neuronas más para incluir la conservación de los pesos para las posiciones i y j (Wi y Wj), 2 neuronas que codifican para la conservación de cada residuo en las posiciones i y j (Ci y Cj), 1 neurona para las mutaciones correlacionadas (Cm(i,j)) y 18 neuronas más para incluir la estructura secundaria predicha (PS1..PS18); 8 neuronas en la capa oculta y una neurona de salida. (b) Ejemplo de modelo recurrente (RNN) propuesta por Pollastri 2002 [53], la unidad de entrada está conectada a los cuatro planos ocultos. La unidad de entrada y las unidades ocultas están conectadas por la unidad de salida. Se muestran las conexiones de cada unidad oculta con su rejilla vecina con el mismo plano. (c) Arquitectura de una red neuronal basada en funciones de base radial (RBFNN), propuesta por Zhand 2004, 2005 y 2007 [60], [61], [64]. (d) Modelo basado en RNA de transiente caótico propuesto por Liu 2006 [114], la arquitectura consta de tres capas de neuronas: una neurona de salida que representa la propensión a contacto, una capa oculta de 10 neuronas y una capa de entrada con 1050 neuronas para cinco parejas de residuos, 10 neuronas para la clasificación de los residuos acorde a la hidrofobicidad, polaridad, ácida y básica, seis neuronas para la información de la estructura secundaria.

Codificación

La codificación de la información para el entrenamiento de las redes neuronales es un factor decisivo tanto por su influencia en el éxito final de la predicción como en la determinación de la estructura de la misma.

Una vertiente muy generalizada es la de tener en cuenta la vecindad del aminoácido para el cual se está realizando la predicción. En este caso la capa de entrada codifica a una ventana móvil en la secuencia de aminoácidos y la predicción es realizada para el residuo ubicado en el centro de la misma. El ancho de la ventana es obtenida experimentalmente y varía según los autores [30], [42], [50], [108], [110]. En algunos casos se plantea que ésta debe ser de 17 aminoácidos, basada en la evidencia de la correlación estadística entre la estructura secundaria de un residuo y los 8 residuos vecinos al punto de predicción hacia ambos lados [108].

Otro tipo de codificación, igualmente aceptado, implica la representación de la frecuencia de ocurrencia de contactos incluyendo el contexto de la secuencia para cada residuo, tomando una ventana centrada en el residuo objetivo, con una longitud de tres aminoácidos. Son considerados los pares paralelo y antiparalelo de los dos segmentos entrados en i y j. {i – 1, j – 1}, {i , j}, {i + 1, j + 1} (pares paralelos), {i – 1, j + 1}, {i , j}, {i + 1, j – 1} (pares antiparalelos). Se incluye, además, la longitud de la proteína y la separación de la secuencia y otro par asociado a la información de hidrofobicidad [48], [59]. La mayor dificultad de esta codificación está en su complejidad.

Otras codificaciones tienen en cuenta la información evolutiva, el contexto de la secuencia (una ventana de 3 residuos de longitud para los pares paralelos y antiparalelos) y la información relacionada con las mutaciones correlacionadas y la predicción de la estructura secundaria [48].

Como técnicas de pre-procesamiento de los datos, se ha tratado la reducción de la dimensionalidad de los rasgos con técnicas como el análisis de componentes principales (PCA) y los mapas auto-organizados (SOM). Por otra parte han sido empleadas algunas técnicas de procesamiento de imágenes y de estadística [115].

Algoritmos de Entrenamiento

Las técnicas de entrenamiento empleadas, varían en dependencia del nivel de complejidad de la RNA propuesta. Una de las más empleadas es algoritmo *back-propagation* [42], [48], [59], [110]. Otra técnica de ajustes en los pesos es el empleo del gradiente descendente del total del error de salida [30], [108], definido por (8).

$$E = \sum_{c} \sum_{j} (O_{j,c} - D_{j,c})^2 \qquad (8)$$

Donde, $O_{j,c}$ es la salida observada en la neurona j para el caso de entrenamiento y $D_{j,c}$ es la salida deseada. El entrenamiento se detiene cuando la reducción en E se vuelve asintótico (en la práctica sería cuando el cambio por ciclo en E es menor que 2×10^{-4}).

También se han empleado algoritmos genéticos para optimizar los parámetros iniciales, los cuales se ajustan posteriormente con un algoritmo de aprendizaje híbrido para la predicción de los mapas de contacto [13], [64], [116]. El algoritmo de aprendizaje híbrido, combina el paradigma del gradiente y el paradigma del mínimo lineal cuadrático (LLS), y puede ser utilizado para ajustar los centros y los anchos. Este algoritmo incluye dos partes: 1) El paso de avance (*forward*), donde los datos de entrada y las señales de la función son suministrados para calcular la función de salida oculta, y posteriormente el método LLS modifica los anchos de la función encontrada. 2) En el paso de realimentación (*backward*) el error se propaga de la salida a las entradas. En este paso, se mantienen los pesos fijos, y se modifican los centros y los anchos de las neuronas RBF.

Evaluación de la red

Con el objetivo de evaluar el desempeño (efectividad) de los métodos, se han propuesto numerosas medidas. Una medida comúnmente aplicada es (9):

$$Q_3 = \frac{P_\alpha + P_\beta + P_{coil}}{N} \qquad (9)$$

Donde N es el número total de residuos predichos y P_α, P_β y P_{coil} son el número de estructuras correctamente predichas para cada tipo [110].

Otro método para medir la efectividad del predictor es (10):

$$Acc = N_{CP}^* / N_{CP} \qquad (10)$$

Donde N_{CP}* es el número de contactos asignados correctamente y N_{CP} es el total de contactos predichos [48], [59], [113], [114].

El coeficiente de correlación también es una medida comúnmente empleada (11).

$$C_{\alpha} = \frac{(p_{\alpha} n_{\alpha}) - (u_{\alpha} o_{\alpha})}{\sqrt{(n_{\alpha} + u)_{\alpha} (n_{\alpha} + o_{\alpha})(p_{\alpha} + u_{\alpha})(p_{\alpha} + o_{\alpha})}} \quad (11)$$

Donde p_{α} es el número de casos de verdaderos positivos, n_{α} el número de verdaderos negativos, o_{α} el número de falsos positivos y u_{α} el número de errores. Esta misma expresión es aplicable para C_{β} y C_{coil} [109], [110], [113], [114].

El desempeño de la red neuronal es estimado mediante el error cuadrático medio (12).

$$MSE_{v} = \frac{1}{N_T} \sum_{p=1}^{N_T} (y_p - f(x_p))^2 \quad (12)$$

Donde y_p y $f(x_p)$ son las salidas deseada y predichas de la red y N_T es el número de objetos para el entrenamiento [116].

Como técnicas de pre-procesamiento de los datos, se ha tratado la reducción de la dimensionalidad de los rasgos con técnicas como el análisis de componentes principales (PCA) y los mapas auto-organizados (SOM). Por otra parte han sido empleadas algunas técnicas de procesamiento de imágenes y de estadística [115].

En sentido general, las RNA han aumentado el por ciento de efectividad en la predicción de estructuras de proteínas. Esto se debe a su elevada capacidad de generalización y a que, junto con ellas, se ha combinado una gran cantidad de información sobre las proteínas como son: la información evolutiva, la conservación y las mutaciones correlacionadas de las secuencias alineadas; la combinación de la información de la secuencia de aminoácidos y su estructura secundaria; los pares paralelos y antiparalelos; entre otros aspectos. Estas técnicas presentan algunas desventajas como: la complejidad de implementación, el alto costo computacional, la complejidad en la codificación de los datos de entrada, la base de conocimientos no está formada por reglas o casos que permitan una explicación del funcionamiento del sistema, entre otras.

4.1.3.- Máquina de Vectores Soporte (SVM).

Dentro de las técnicas de reconocimiento de patrones empleadas para la predicción de estructura, se encuentran las máquinas de vectores soporte [117], [118]. Éste es un método general para la solución de problemas de regresión, clasificación y estimación. Esta técnica ha sido utilizada para la predicción de estructuras secundarias basadas en máquinas vectores soporte de doble capa y matrices PSSMs (por sus siglas en inglés "*position-specific scoring matrices*") [118]–[120].

Otro método de predicción basado en SVM, soportado en la teoría del aprendizaje estadístico y en su habilidad de generalización, emplea para el entrenamiento de la SVM los rasgos de cada trenzas β de la secuencia de aminoácidos. Para cada trenza en el conjunto de datos, se extrajeron seis atributos con el objetivo de ser disimilares en los dos tipos de trenzas (centrales y de borde) [121]. En este caso, los parámetros de la SVM fueron optimizados empleando una validación cruzada con 18 particiones (*18-fold cross-validation*). La medición del rendimiento del método propuesto se realizó mediante el coeficiente de correlación (11).

También han sido empleadas las SVM, como herramienta de clasificación a partir de varios rasgos basados en la secuencia primaria, la alineación de múltiples secuencias, la predicción de la estructura secundaria y el análisis de mutaciones correlacionadas [119], [122]–[125], donde para cada par de posiciones en la secuencia de la proteína, se identifican cinco grupos rasgos que capturan diferentes aspectos de los aminoácidos en las posiciones seleccionadas: conservación de la secuencia (Con), separación en la secuencia (Sep), análisis de mutaciones correlacionadas (CMA), estructura secundaria predicha (PSS) y perfiles de la secuencia (SP). El predictor propuesto devuelve una puntuación para cada instancia de entrada, donde se asume como contacto cuando la puntuación es alta y no contacto cuando es baja [117], [118], [120], [126]–[132].

4.1.4.- Algoritmos bio-inspirados

Árboles de decisión

Los árboles de decisión es otra técnica ampliamente explorada. Éstos son capaces de producir reglas entendibles para los especialistas, las que pueden ser empleadas para dar explicaciones sobre cómo se ha realizado la predicción [40], [121], [133]–[135].

Los árboles de decisión han sido utilizados en esquemas de clasificación para la predicción de mapas de contactos interresiduales de proteínas [133], [134]. En otros casos los árboles son empleados para clasificar láminas de borde en láminas centrales en un conjunto de proteínas. Este enfoque es muy útil debido a que permite detectar las zonas de interacción proteína – proteína [121].

Inteligencia de Enjambre

Las técnicas de optimización basadas en lógica de enjambre como colonia de hormigas [136], colonia de abejas [137], [138] y enjambre de partículas [139]–[141], han sido ampliamente empleadas.

Dentro del enjambre de partículas la optimización basada en quantums (QPSO) es uno de los modelos bio-inspirados más utilizados [141]. En este algoritmo los aminoácidos se separan en hidrofóbicos y no hidrofóbicos. Una de sus principales ventajas y que constituye su aporte fundamental es el empleo de una variante basada en múltiples capas que incluye mejoras en la localización de las partículas usando estrategias de ajuste de precisión y de exploración. Divide las partículas en tres subpoblaciones separadas, la de élite, la de explotación y la de exploración. Con la formación de estas subpoblaciones, son actualizadas las posiciones de las partículas en la estrategia de ajuste de precisión, QPSO y la estrategia de exploración, respectivamente.

Algoritmos evolutivos y genéticos

Los algoritmos evolutivos y genéticos (AG) también han sido explorados en la predicción de estructuras de proteínas [97], [142]–[144]. Mansour y colaboradores [143] proponen un algoritmo genético basado en el modelo cúbico polar e hidrofóbico. Este tipo de algoritmo propone un nuevo método de cruzamiento y de mutación que asegura soluciones candidatas factibles para el tratamiento del problema.

Los AGs también han sido empleados en la fase de preprocesamiento como método de remuestreo [133], [144], así como en la predicción *ab initio*, combinándolo con técnicas como el modelo de sustitución de los k-vecinos más cercanos [34]. En otros casos se ha empleado el AG para extraer un conjunto de reglas de decisión para la predicción de los mapas de contactos de las proteínas [142].

Algoritmos de Estimación de Distribuciones

Las técnicas de estimación de distribuciones (EDA), son algoritmos evolutivos que trabajan con una población de soluciones candidatas. Los EDAs han sido empleados para analizar la influencia de los factores que intervienen en la estabilidad de las proteínas, entre los que podemos mencionar: las interacciones electrostáticas, los enlaces de hidrógeno, las interacciones de Van der Waals, las propiedades intrínsecas de los aminoácidos que los hacen asumir ciertas estructuras, las interacciones hidrofóbicas y la contribución de la entropía conformacional [145].

Dependiendo del nivel de detalle del modelo, los EDAs han sido empleados en: modelos simplificados de predicción de estructuras de proteínas [146]–[148], reducción del alfabeto de aminoácidos para la predicción de estructuras de proteínas [146], diseño de proteínas mediante minimización de los potenciales de los contactos [147], diseño de ligandos de péptidos de proteínas [149], y posicionamiento de la cara de la proteína [145].

La principal desventaja de los EDAs consiste en la imposibilidad de tomar en cuenta las dependencias complejas entre los rasgos, incluso para modelos multivariados (con un orden estadístico mayor de dos). Otra desventaja que podemos mencionar es que un EDA multivariado implica altos requerimientos computacionales, en cuanto a capacidad de cómputo y de memoria.

4.1.5.- Combinación de clasificadores.

La principal motivación para el empleo de múltiples clasificadores es aumentar la efectividad en la clasificación, pues diferentes algoritmos emplean diferentes formas de generalización y de representación del conocimiento. Es sabido que éstos tienden a equivocarse en diferentes zonas del espacio de búsqueda, pero su combinación adecuada suele corregir los errores individuales no correlacionados. La combinación de la clasificación puede realizarse mediante los paradigmas de selección o de fusión. En el primer caso, se selecciona un único algoritmo para la clasificación de una nueva instancia, mientras que en el segundo caso, se fusiona la decisión de todos los algoritmos.

Diplaris y colaboradores [150], proponen el empleo de ambos paradigmas de combinación de clasificadores, empleando nueve algoritmos de aprendizaje diferentes. Entre los que se encuentran: árboles de decisión, reglas de asociación, vecinos más cercanos, redes Bayesianas, máquinas de soporte vectorial y las redes neuronales artificiales. La principales desventajas del empleo de múltiples clasificadores radica en la complejidad de implementación del esquema de clasificación.

Otros autores también han empleado múltiples esquemas de clasificación, entre el que se destaca el paralelo [23], [150]–[152]. Abu-doleh y colaboradores [23] proponen un modelo de combinación de los resultados basado en un sistema neurodifuso (ANFIS) y un clasificador basado en el vecino más cercano (kNN). Recientemente se propuso un esquema de combinación que descompone el problema de predicción de contactos interresiduales en 400 subproblemas, uno por cada pareja de amino ácidos [133], [134]. Esta solución tiene como ventaja fundamental que brinda un modelo interpretable con el objetivo de explicar el proceso de plegamiento de las proteínas en función de las parejas de aminoácidos y la subsecuencia entre ellos.

Redes Bayesianas

La teoría de las probabilidades de Bayes ha sido empleada en modelos de análisis de la información del plegamiento de las proteínas globulares. Este proceso se trata como una transferencia de mensajes entre la secuencia primaria y la secundaria, donde el segundo se deriva del primero. El método está dirigido a buscar una aproximación más robusta para proteínas pequeñas y grandes. [18].

Las redes Bayesianas tienen como ventaja fundamental que permiten la medición de la contribución de cada información por separado. Al estar basadas en el método de Bayes, emplean poca información y resultan un método más robusto que otras pruebas estadísticas. La principal desventaja que presenta es que el método es válido para su empleo en proteínas que mantengan una relación similar entre la secuencia y su estructura, o sea, familias o conglomerados de proteínas [153]–[159].

Redes de Markov

Los modelos ocultos de Markov (HMM) también han sido ampliamente empleados en la predicción de estructuras de proteínas [160]. Frecuentemente, los HMM son empleados en combinación con otras técnicas de aprendizaje como: algoritmos genéticos, difusión de partículas, redes neuronales artificiales, entre otras, con el fin de explorar su espacio de topologías [161]–[163], o además, con funciones de partición, considerando la similitud de los aminoácidos, la estructura secundaria y la accesibilidad soluble relativa para el cálculo de las matrices de alineación [49]. Usualmente se toman en cuenta las dependencias no sólo entre aminoácidos (emisiones) adyacentes, sino entre los segundos anteriores y siguientes [160]–[162], [164]–[168].

Redes de Regresión Logística

La regresión logística ha sido otra técnica estadística empleada en la predicción de estructuras de proteínas. Estas aproximaciones parten de la base de contar con las predicciones de un conjunto de predictores, donde a diferencia de los esquemas de combinación como el promedio (las predicciones son promediadas) o la votación (se seleccionan las mejores λL predicciones y se emplean para votar por la solución), las predicciones se combinan empleando un análisis de regresión logística (LR) [22], [169]–[171].

Uno de los principales inconvenientes del empleo de esta técnica es la calidad de las predicciones previas, debido a que depende directamente de la selección de los clasificadores a emplear y a que, en algunos casos, pudiera contarse con predicciones parciales falseando la información.

4.2.- Resumen de las técnicas de predicción.

En la tabla 1 se muestra una comparación entre las técnicas revisadas, en base al nivel de efectividad alcanzada, principales ventajas y desventajas. Los valores de efectividad incluidos en esta tabla no han sido obtenidos por vía experimental, sino que son los reportados por los autores de los algoritmos. En muchos de los casos, como ocurre con las primeras aproximaciones de predicción (Chou, Lim, Robson), los valores de efectividad están en dependencia del método empleado para calcularla [36].

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Chou, 1974* [104]	50** – 80% (19 p)	Predicción de regiones de cadenas α y β mediante reglas basadas en los potenciales.	• Bajo costo computacional. • Gran simplicidad.	• Empleo de sentido común (reglas cualitativas).
Lim, 1974* [105]	80 – 85% (25 p)	Predicción de regiones de cadenas α y β mediante reglas basadas en las propiedades hidrofóbicas de las proteínas.	• Tiene en cuenta las propiedades hidrofóbicas de los aminoácidos.	• Empleo de sentido común (reglas cualitativas).
Robson, 1974* [18]	60% "propone el empleo de 20^{21} proteínas".	Teoría de análisis de la información del plegamiento de las proteínas globulares. Método de predicción basado en la teoría de probabilidades de Bayes.	• La teoría de información permite la medición de la contribución de las informaciones, por separado. • El método de Bayes emplea muy poca información. • Modelo mucho más robusto que otras pruebas estadísticas como chi-cuadrado.	• El método de Bayes es más eficiente para información compleja, o sea, mutuamente consistente, que represente modelos alternativos de transferencia de información entre la secuencia y la estructura. • El método es válido para proteínas que mantengan una relación similar entre la secuencia y la estructura.
Garnier, 1978* [36]	60% (30 p)	Emplea la información que contiene cada residuo que conforma la cadena, sobre la estructura de otro residuo.	• Bajo costo computacional. • Tiene mayor efectividad de que el reportado por Robson en 1974. • Toma en cuenta interacciones intermedias en la cadena.	• La efectividad depende de la optimización de variables que pueden tener múltiples valores en dependencia del residuo al que se predice su estructura secundaria.
Holley, 1987 [108]	63 – 79% (48 p)	Basado en el empleo de una RNA Feed-Forward.	• Codificación de los datos de entrada muy sencilla. • Tiene en cuenta la evidencia de la correlación estadística entre la estructura secundaria de un residuo y los 8 residuos vecinos al punto de predicción, hacia ambos lados. • Simple de implementar y entrenar.	• Base de conocimientos no permite crear un mecanismo de interpretación simple.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Zuelebil, 1987 [172]	66% (11 p)	Método para realizar la predicción de estructura secundaria y sitios activos mediante el empleo de la información disponible secuencias homólogas.	•Extrae información de la alineación de secuencias homólogas. •Permite identificar residuos funcionalmente importantes en las enzimas.	•Dependiente del método de alineación de secuencias empleado.
Bohr, 1988 [42]	73% (56 p)	Basado en el empleo de RNA MLP.	•Codificación de los datos de entrada muy sencilla. •Simple de implementar y entrenar.	•Sólo predice las hélices α. •Base de conocimientos no permite crear un mecanismo de interpretación simple.
Qian, 1988 [110]	63 – 65% (106 p)	Basado en el empleo de RNA MLP.	•Entrenamiento con proteínas homólogas y no homólogas (conjuntos de entrenamiento independientes). •Los conjuntos de entrenamiento están balanceados en número de hélices α y láminas β.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Bohr, 1990 [50]	99.9% (13 p)	Basado en el empleo de RNA.	•Entrenada con la información de la secuencia de aminoácidos y su estructura, y la distancia (binara). •Predice las estructuras 2D y 3D. •Simple de implementar y entrenar.	•Diseñado para proteínas homólogas. •Base de conocimientos no permite crear un mecanismo de interpretación simple.
Kneller, 1990 [109]	79% (66 p)	Basado en el empleo de RNA.	•Adiciona la información periódica de la secuencia y la subdivide las proteínas en 8 clases estructurales. •Bajo costo computacional.	•La preparación inicial de los datos es compleja (clasificación de las secuencias).

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Rost, 1993 [30]	70.8% (130 p)	Basado en el empleo de RNA. Este modelo combina tres niveles de redes: secuencia-estructura, estructura-estructura y promediado.	•Emplea información evolutiva a partir del alineamiento de múltiples secuencias.	•Difícil implementación. •Alto coste computacional.
Fariselli y Casadio, 1999 [59]	16% (608 p)	Basado en el empleo de RNA MLP, incluye información evolutiva y conservación de la secuencia.	•Entrenamiento con proteínas no homólogas. •Incluye información evolutiva, conservación de la secuencia, hidrofobicidad, longitud de la secuencia y separación de residuos. •Simple de implementar y entrenar.	•Difícil selección del conjunto de entrenamiento.
Fariselli y Casadio, 2001 [48]	25% (173 p)	Basado en el empleo de RNA MLP, empleando las mutaciones correlacionadas y en la predicción de la estructura secundaria.	•Entrenamiento con proteínas no homólogas. •Incluye información de las mutaciones correlacionadas y en la predicción de la estructura secundaria.	•Sólo toma en cuenta las hélices β para realizar la predicción. •La codificación de los datos de entrada se hace compleja.
Pollastri, 2002 [53]	45 – 60.5% (1484 p)	Basado en RNA Recurrente y Red Bayesiana.	•Muy flexible.	•Extremadamente complejo. •Alto coste computacional.
Zhao, 2002 [120]	22.38% (177 p)	Basado en SVM	•Toma en consideración la secuencia primaria, la alineación de múltiples secuencias, la predicción 2D y las de mutaciones correlacionadas.	•Es necesario obtener previamente la estructura 2D predicha de la proteína.
Guo, 2003 [116]	-	Basado en el empleo de RBFNN optimizado por medio de un Algoritmo Genético.	•Simple de implementar. •Algoritmo genético e hibridaciones para optimizar los parámetros. •Rápida convergencia.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Siepen, 2003 [121]	78% (564 p)	Basado en árboles de decisión y trenzas β centrales y de bordes.	•Entrenamiento con proteínas no homólogas. •Arquitectura C4.5. •Produce reglas entendibles por los especialistas.	•Compleja extracción de rasgos.
Guo, 2004 [119]	(1035 p)	Basado en SVM y PSSM.	•Combina la SVM con PSSM para lograr mayor optimización.	•Requiere de procesamiento intermedio (alinear secuencias) mediante el empleo del PSI-BLAST.
Zhang, 2004[64] 2005[60] 2007[61]	33% (173 p) (18 p) (61 p)	Basado en el empleo de RBFNN optimizado por medio de un Algoritmo Genético.	•Evaluado en proteínas no homólogas (2004), globulina (2005) y Hepatitis C (2007). •Propone un novedoso método de encriptación de las entradas en cromosomas para el AG.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Diplaris, 2005 [150]	(662 p)	Basado en la combinación de clasificadores	•Implementan los métodos de selección y fusión de clasificadores. •Abarca 9 clasificadores de propósito general.	•Alta complejidad de implementación.
Liu, 2005 [113]	8% (105 p)	Basado en RNA recurrente.	•Tiene en cuenta la estructura 2D predicha y la hidrofobicidad. •Tiene en cuenta los pares paralelos y antiparalelos.	•Sólo toma en cuenta las hélices α para realizar la predicción. •Es necesario obtener previamente, la estructura 2D predicha de la proteína.
Vullo, 2006 [62]	72,6% (2171 p)	Modelo de RNA Recurrente Bidireccional.	•Empleo de mapas PE (*principal eigenvector*), lo que reduce la dimensión del problema.	•Poca capacidad de generalización, asociada a la aparición de falta de información en la base de aprendizaje.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Liu, 2006 [114]	8,2% (125 p)	Modelo de RNA de Transiente Caótico.	•Tiene en cuenta la estructura 2D predicha y la hidrofobicidad. •Incluye paralelos y antiparalelos.	•Sólo toma en cuenta las hélices α para realizar la predicción. •Necesita conocer la estructura 2D.
Shi, 2008 [106]	97% (933 p)	Basado en la regresión estadística.	•Se basa en los porcentajes de residuos α, β y en la longitud de la secuencia. •Genera una ecuación lineal simple.	•Asume linealidad en la predicción de los contactos.
Walsh, 2009 [33]	–	Basado en RNA recursiva.	•Empleo de mapas de contacto multiclase. •Empleo de algoritmo de recocido simulado para la reconstrucción.	•Alta complejidad de implementación.
Cheng-yuan, 2010 [141]	(2 p)	Basado en QPSO multicapas.	•Empleo de la heurística PSO. •Basado en el modelo Toy.	•Difícil alcanzar un óptimo global para grandes volúmenes de datos.
Mansour 2010 [143]	(20 p, Ls ≤64)	Basado en Algoritmo Genético.	•Propone nuevo método de cruzamiento y mutación.	•Sólo probado en secuencias pequeñas de 10 y hasta 64 aminoácidos.
Zhao 2010 [99]	(12 p, CASP8, FM)	Basado en RNA Condicionales (CNF).	•Más potente que CRF (*Conditional random fields*). •Buen rendimiento en proteínas α y pequeñas proteínas β	•Mal rendimiento para proteínas β grandes. •Base de conocimientos no permite crear un mecanismo de interpretación simple.
Pezeshk 2010 [160]	(1342 p)	Basado en HMM doble cara.	•Toma en cuenta las dependencias entre emisiones (aminoácidos). •Considera las dependencias entre ambas caras de la proteína.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Higgs 2010 [144]	(200 p, Rosetta)	Basado en AG.	•Método de remuestreo empleando estructuras refinadas con anterioridad. •Emplea la función de calidad de Rosetta [31].	•Depende de lo buena que sea la extracción de la población inicial.
Custódio 2010 [34]	(6 p)	Basado en AG.	•Emplea modelo sustitución basado en similaridad.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Yang 2011 [22]	(CASP9)	Basado en Regresión Logística.	•Emplea la unión de los pares de residuos de las λLs mejores predicciones para cada uno de los p predictores que la componen.	•No puede aprovechar la información directamente de los vectores de contactos.
Howe 2011 [117]	(472 p)	Basado en SVM.	•Emplea una ventana local de rasgos.	•Emplea predicción previa de la estructura secundaria y de la accesibilidad solvente. •Base de conocimientos no permite crear un mecanismo de interpretación simple.
Ascencio-Cortes 2011 [55]	51% (5659 p)	Basado en el vecino más cercano.	•Emplea 30 de 544 propiedades físico-químicas de los aminoácidos. •Predice mapas de distancias.	-
Abu-Doleh 2012 [23]	45.5% (500 p)	Combinación de clasificadores	•Combinación paralela de clasificadores mediante un sistema neurodifuso (ANFIS) y vecinos más cercanos (kNN).	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Marquez-Chamorro 2012 [142]	–	Basado en AG.	•Genera un conjunto de reglas de decisión que son empleadas en la predicción de mapas de contactos.	•No presentan un mecanismo de interpretación de la base de conocimiento.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Deka 2012 [112]	(8 p)	Basado en RNA.	•Devuelve tres predicciones de la estructura, realizando la clasificación por voto mayoritario. •Incluye algoritmo de gradiente descendiente y con aprendizaje adaptivo.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Mandle 2012 [118]	(4 p)	Basado en SVM.	•Emplea función de base radial como núcleo. •Emplea un espacio de rasgos de alta dimensión.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Chetia 2012 [115]	(3 p)	Basado en RNA.	•Emplean algunas técnicas de procesamiento de imágenes y de estadística. •Emplea análisis de componentes principales (PCA) y mapas auto-organizados (SOM) para la reducción de la dimensión de los datos.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Ding 2013 [173]	58.86% (5300p)	CNNcon. Basado en RNA	•Emplea cascadas de redes neuronales. •Efectiva para proteínas grandes de hasta 450 amino ácidos.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.
Wang 2013 [174]	(1500p)	Basado en restricciones evolutivas y físicas. Emplea programación integrativa lineal y *Random Forest*.	•Brinda más información que los métodos basados en mutaciones correlacionadas.	•No presentan un mecanismo de interpretación de la base de conocimiento.
Habibi 2013 [175]	15%	Basado en multiclasificador de tipo comité de expertos.	•Los clasificadores base son RNAs.	•Alto costo computacional. •Base de conocimientos poco entendible.

Métodos	Efectividad	Descripción	Ventajas	Desventajas
Feinauer 2014 [176]	31.7% (384p)	Basado en una máquina de inferencias que analiza la máxima vecindad de los contactos.	•Reduce el número de enlaces espurios. •No se afecta la calidad de la predicción con presencia de gaps.	•Altamente dependiente de la calidad del alineamiento realizado.
Kukic 2014 [177]	(3645p)	Basado en RNA recursivas bidimensionales.	•Plantea un algoritmo de reconstrucción de la estructura terciaria de la proteína.	•Base de conocimientos poco entendible.
Schneider 2014 [178]	53.2% (CASP10)	Basado en SVM	•Combina la información evolutiva con las características físico-químicas de los amino ácidos.	•Base de conocimientos no permite crear un mecanismo de interpretación simple.

* Debido a la gran diversidad de métodos para el cálculo de la efectividad empleado por los autores, a la reducida cantidad de proteínas empleadas en la experimentación y a la importancia atribuida a estos métodos, otros autores han recalculado el valor de efectividad el cual resulta entre 50 y 53% [110].

** Este valor de efectividad ha sido proporcionado por otros autores, vía experimental [42].

Como se puede apreciar, el uso de las técnicas de aprendizaje automático en la predicción de estructuras de proteínas ha experimentado un crecimiento acelerado en los últimos tiempos, alcanzando el 87% del total de las técnicas usadas hasta el momento. Esto se debe, fundamentalmente, al alto nivel de adaptabilidad de estos algoritmos. Dentro de las técnicas de aprendizaje automático las más empleadas son los sistemas de combinación de clasificadores con un 27% y las redes neuronales artificiales con un 25%. Mientras que las máquinas de vectores soporte y los algoritmos bio-inspirados han logrado el interés de los investigadores con un 17% y un 16%, respectivamente. Es por ello que este estudio está enfocado a estas técnicas. Se muestran además las nuevas perspectivas y tendencias en el campo de la predicción de estructuras de proteínas.

4.3.- Conclusiones parciales.

El análisis del estado del arte permitió realizar un estudio extensivo de los métodos y técnicas de aprendizaje automático aplicado a la predicción de estructuras de proteínas, proponiéndose una nueva taxonomía. Se evidenció, además, que desafortunadamente los clasificadores más efectivos suelen ser no comprensibles o no presentan un mecanismo de interpretación adecuado. Lo cual representa una ventaja para las tareas de clasificación, pero un impedimento para el entendimiento del proceso de plegamiento de las proteínas.

Se evidencia que los datos derivados de la secuencia (estructura primaria), sólo contribuyen en un 60 – 65% a la información necesaria para la predicción de la estructura secundaria. Además, para lograr un incremento considerable en la efectividad de las predicciones, se hace necesaria la incorporación de información relacionada con la estructura terciaria [108], [110].

A modo de conclusión de este estudio, se visualizan tres líneas de investigación fundamentales: 1) crear una modelación adecuada del proceso de plegamiento de las proteínas, a partir de la información brindada por la secuencia de aminoácidos y su relación con la estructura terciaria; 2) abordar el problema de predicción mediante el empleo de combinación de clasificadores; 3) trabajar en mecanismos de interpretación adecuados para el modelo obtenido.

Parte III

Propuesta

Capítulo 5

Multiclasificador propuesto (FoDT)

En este capítulo se presenta la metodología propuesta para realizar la predicción de los mapas de contactos de las proteínas. La metodología tiene como objetivo conocer hasta qué punto un sistema basado en árboles es capaz de aprender la correlación entre la estructura de residuos covalentes de una proteína y su mapa de contacto, ya que éste se calcula a partir de su estructura 3D conocida.

5.1.- Introducción

Recientemente se han explorado métodos basados en algoritmos de aprendizaje automático y redes neuronales para predecir las distancias entre los pares acoplados de residuos, así como los mapas de contacto de las proteínas y aproximación basada en la clasificación de residuos. En el algoritmo que se presenta, de igual forma que en otros propuestos por la literatura, la predicción se aborda como un problema de clasificación, aunque desde la perspectiva de la multiclasificación.

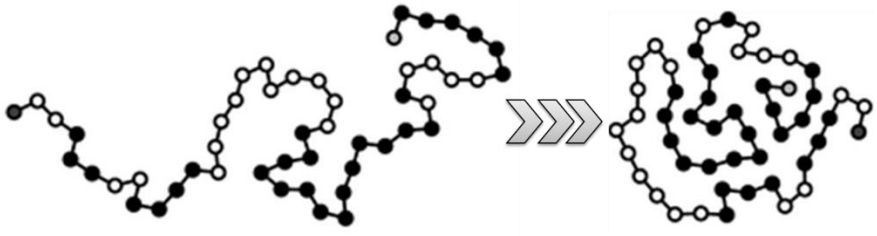
Las principales motivaciones por las cuales se aborda la predicción de mapas de contacto con multiclasificadores radican en: que en más de 30 años y el empleo de una gran variedad de algoritmos no se ha logrado encontrar uno que dé solución al problema (razón estadística); la mayoría de los algoritmos propuestos parecen tener dificultades para modelar este problema, el cual está magnificado por su nivel de desbalance, la alta cantidad de datos y la dificultad para diferenciar claramente instancias o rasgos que permitan establecer fronteras entre las clases (razón de representación).

5.2.- Hipótesis biológica

La caracterización de los estados conformacionales, así como la elucidación del rol de intermediarios, como las chaperonas y otras macromoléculas, conllevan al análisis de nuevas hipótesis sobre el proceso de plegamiento [179], [180].

El colapso hidrofóbico es un evento que, supuestamente, ocurre durante el plegamiento de las proteínas globulares [181]. Esta hipótesis está basada en la observación de que a menudo las proteínas en su estado nativo contienen en su interior un núcleo hidrofóbico de aminoácidos no polares de la cadena lateral, quedando la mayoría de los aminoácidos polares o cargados en la cara expuesta de la proteína (Figura 23). La estabilización energética se confiere a la proteína por el medio acuoso circundante [182], [183].

La hipótesis postula que el colapso hidrofóbico es un evento que ocurre antes de la formación de muchas de las estructuras secundarias y contactos nativos presentes en la estructura terciaria de la proteína, aunque todavía no se han establecido todas las interacciones entre aminoácidos.



(A) (B)

Figura 23. Colapso hidrofóbico de proteínas globulares.
A) representa los aminoácidos sin plegar. B) representa la forma compacta de la proteína. Los aminoácidos hidrofóbicos están representados con esferas negras y tienden a formar núcleos hidrofóbicos de aminoácidos no polares en el centro de la proteína.

El análisis de la hipótesis del colapso hidrofóbico como evento importante en el proceso de plegamiento de las proteínas globulares sugiere que:

1) La estructura nativa de una proteína viene determinada por su secuencia de aminoácidos, es única, estable y se corresponde con un mínimo de energía libre;
2) El plegamiento de proteínas está guiado por la formación de interacciones locales entre aminoácidos que actúan como núcleos de plegamiento;

3) La fuerza motora que determina el plegamiento de una proteína es la estabilización energética que se logra secuestrando los residuos hidrofóbicos en el interior de la proteína;

4) Y, que las proteínas chaperonas no contienen información sobre modelos particulares de plegamiento, sino que ayudan a las proteínas mal plegadas a encontrar su conformación nativa.

Lograr una correcta modelación del plegamiento de las proteínas es una tarea complicada, sobre todo por el gran número de actores que intervienen en dicho proceso. Con el objetivo de lograr la representación del proceso de plegamiento, en esta investigación se realizó una simplificación del mismo que implica:

1) Tomar en cuenta las características físico – químicas de los aminoácidos objetivo (que podrían o no entrar en contacto), lo cual podría ayudar a entender las interacciones locales;

2) Descomponer el análisis del plegamiento a cada una de las 400 posibles parejas de aminoácidos que pueden formarse de forma independiente;

3) Tomar en cuenta la subsecuencia implícita entre los aminoácidos objetivos no adyacentes, con la intención de analizar la influencia de dicha secuencia en el plegado de la proteína.

4) Ignorar la presencia de las chaperonas y otros actores externos que intervengan el proceso del plegado correcto de la proteína.

La hipótesis aquí formulada, así como la necesidad de empleo de un esquema adecuado de clasificación brindan el sustento teórico del algoritmo propuesto en la presente investigación.

5.3.- Codificación del vector de entrada

En la bibliografía revisada son empleados diversos tipos de codificaciones, como se muestra en el epígrafe 2.5.3 de este documento. Sin embargo, es de consenso general el empleo de una visión holística del problema, donde los clasificadores son entrenados con todos los vectores que se forman a partir de las proteínas del conjunto de entrenamiento.

En esta investigación se propone emplear un nuevo esquema para la representación del problema que minimiza esa visión. Sobre la base de que la predicción de contactos interresiduales requiere del estudio de las distancias entre los residuos y su relación con el tipo específico de residuo. Se propone analizar por separado cada una de las parejas de aminoácidos que pueden formarse, lo cual representa la descomposición del problema en 400 sub-problemas (20 x 20 aminoácidos). Esta aproximación no simplemente cambia de la visión holística a una visión reduccionista, sino que propone una simplificación al proceso de entrenamiento.

Como codificación de las entradas, se emplean vectores de longitud 35. Éstos incluyen la información de la subsecuencia implícita entre los aminoácidos no adyacentes, propiedades físico-químicas, así como la distancia y la separación a que se encuentran en la secuencia (Figura 24).

A_0	A_1	—— Aminoácidos ——	A_{19}	Prop. Físicas			Prop. Químicas		S_S	S_L	Clase
0	1		19	20		26	27		31	32	33	34

Figura 24. Esquema de codificación de las entradas del multiclasificador.

Leyenda:

- Aminoácidos (**Aa**) [20 elementos] – Frecuencia de aparición de los aminoácidos presentes en la sub secuencia que se forma entre los aminoácidos analizados.
- Propiedades físicas (**Pf**) [7 elementos] – Frecuencia de las propiedades físicas de los aminoácidos presentes en la sub secuencia que se forma entre los aminoácidos analizados.
  Pf = {Hidrófobos, No Hidrófobos, Hidrofobicidad desconocida, Polares, No Polares, Básicos, Ácidos}.

- Átomos (**At**) [5 elementos] – Frecuencia de los componentes químicos (átomos) presentes en la sub secuencia que se forma entre los aminoácidos analizados.
  At = {Hidrógeno (H), Carbono (C), Oxígeno (O), Azufre (S), Nitrógeno (N)}.

- Separación en la secuencia (Ss) [1 elemento] – Separación en la secuencia (Ss) de los aminoácidos analizados.
- Longitud de la secuencia (SL) [1 elemento] – Tamaño de la proteína.
- Clase (**CL**) [1 elemento] – Si se produce o no un contacto.

$$\mathbf{C_L} = \begin{cases} \{Contacto, No - Contacto\}, & \text{Para mapas de contacto binarios} \\ \text{Distancia euclideana } |A_1 - A_2|, & \text{Para mapas de distancias} \end{cases}$$

Para un par de aminoácidos (**A1**, **A2**), los primeros 20 elementos del vector (**Aa**) representan cada uno de los aminoácidos principales que conforman las proteínas. Aquí se registran las frecuencias de aparición de los mismos en la subsecuencia que se forma entre **A1** y **A2** cuando no son adyacentes. Para representar las propiedades físico – químicas de los aminoácidos (**Pf**), son necesarios siete elementos en el vector. En

este segmento se busca encontrar la influencia de las propiedades de hidrofobicidad y carga de los aminoácidos en el plegamiento de la secuencia. También resulta interesante conocer la relación que puedan tener los átomos (**At**) de los residuos de cada aminoácido en la formación de los enlaces no covalentes o débiles, que dieron lugar al plegamiento que presenta la subsecuencia analizada. Para ello se emplean cinco elementos en el vector, en los cuales están registrados la cantidad de moléculas de hidrógeno, carbono, oxígeno, azufre y nitrógeno que existe en la subsecuencia. Por último, en el vector se incluyen datos más generales como la separación a que se encuentran los aminoácidos objetivo en la secuencia (**Ss**), la longitud de la proteína (**Ls**) y la clase que se le asigna a dicho vector.

Para asignar el valor que toma la clase (**CL**), es necesario calcular primeramente la distancia Euclídea (**D**) entre las posiciones espaciales de ambos aminoácidos (Ecuación 1). Luego, en dependencia de si se está empleando un mapa de distancias, entonces se asigna el valor continuo **D**. Pero, si se está empleando mapa de contactos, entonces es necesario discretizar **D** antes de asignar el valor a **CL**. El criterio de discretización empleado se describe en las ecuaciones 2 y 3.

Caso de estudio

Para comprender el funcionamiento de la codificación se diseñó el siguiente caso de estudio (Figura 25): dada una proteína con secuencia "GVIANVKCAKISRQVALEPCKKG-MRFGKCMNGKCHCTPQ", se desean extraer los vectores correspondientes a la pareja de aminoácidos [A, A].

Figura 25. Codificación propuesta, caso de estudio.

Para extraer los vectores que se forman para la pareja [A, A] son analizadas, en primer lugar, todas las combinaciones de subsecuencias que pueden formarse donde el primer y el último aminoácido sea A. Una vez que se extraen las subsecuencias, se

computan cuántas veces aparecieron los distintos aminoácidos que la componen. Al número de veces que aparece un aminoácido se le denomina frecuencia de aparición en la subsecuencia. El valor de dicha frecuencia es reflejado en el vector, en la posición correspondiente a cada aminoácido. En caso de los aminoácidos que no aparecen en la subsecuencia, se le asigna valor de frecuencia cero. Asimismo se procede con las propiedades físico-químicas, donde se computa cuántos aminoácidos hay que sean o no hidrofóbicos, polares o no polares, ácidos o básicos.

El mismo patrón se sigue para calcular la cantidad de átomos que están implicados. En ese caso sólo se toman en cuenta los átomos pertenecientes a los residuos, debido a que los del esqueleto principal de cada aminoácido (*backbone*) no varían en su composición. El objeto de este segmento de la codificación es intentar descubrir la relación entre los átomos y los enlaces débiles que se puedan formar.

De igual manera que como se procedió con la pareja de aminoácidos [A, A], se forman los vectores para cada pareja de aminoácidos posible a formarse en la secuencia de la proteína dada.

5.4.- Preprocesamiento de los datos

En la clasificación supervisada se conoce un universo de objetos pertenecientes a un conjunto de clases y el problema consiste en, llegado un nuevo objeto poder establecer su relación con cada una de las clases [184]. La utilidad del conocimiento extraído a partir de los datos mediante los métodos de aprendizaje automatizado depende en gran medida de la calidad de esos datos. De manera general, independientemente del clasificador supervisado que se utilice, la calidad de los resultados de clasificación estará dada en gran medida por la calidad del conjunto de entrenamiento. Si éste no es representativo del problema que se investiga, posee ruido, o tiene objetos innecesarios o superfluos, el clasificador supervisado se verá afectado irremediablemente.

Dentro de las características deseadas de un algoritmo clasificador se encuentran una alta precisión y eficiencia computacional. Lo cual se traduce en que el algoritmo clasifique bien, y que además lo haga con poco coste computacional, tanto de almacenamiento como de tiempo. Sin embargo, en problemas como la predicción de

mapas de contactos de proteínas, no existe claridad respecto a qué rasgos son relevantes para la tarea de clasificación, y se incluyen varios de ellos para tratar de no perder ningún posible parámetro. Muchos de estos rasgos son completamente irrelevantes con respecto al proceso de clasificación, haciéndolo más complejo e introduciendo ambigüedades en los datos.

Una vía para mejorar los conjuntos de entrenamiento de los clasificadores supervisados es la selección de objetos o edición del conjunto de entrenamiento. En este proceso se trata de obtener sólo los ejemplos más representativos del conjunto de datos, ya sea tratando de reducir el número de objetos sin afectar la calidad del clasificador o eliminando los objetos ruidosos.

La predicción de mapas de contactos de proteínas encara los tres desafíos fundamentales dentro del aprendizaje automático:

- Volumen: se genera un gran número de vectores por cada una de las proteínas. La cantidad de vectores está determinada por la longitud de las secuencias de las proteínas (Ls), estando definida por la ecuación Ls * (Ls -1) / 2. Lo cual significa que para proteínas de Ls = 100, se generan 4950 vectores. En el conjunto de entrenamiento de 12830 proteínas las longitudes de las secuencias oscilan entre 100 y 400 aminoácidos.
- Coste: el coste de las predicciones de los contactos no es el mismo que el de los no contactos, debido al desbalance que existe entre las clases (la proporción entre contacto y no contacto sigue aproximadamente la ratio 1:13).
- Desbalance: el entrenamiento a partir de datos desbalanceados, generalmente, provoca mal funcionamiento de los algoritmos de aprendizaje automático. Los clasificadores logran muy buenas precisiones con la clase mayoritaria, no siendo así con la minoritaria. En estos casos muchos clasificadores pueden considerarlos como rarezas o ruido y no tener en cuenta la distribución de los datos, centrándose únicamente en los resultados de las medidas globales. Es por ello que muchas de las medidas que tradicionalmente son usadas para medir la calidad de un clasificador son consideradas inadecuadas en el contexto de los conjuntos de entrenamiento no balanceados.

Teniendo en cuenta las características de este problema, se hace necesario realizar el análisis de la naturaleza de los datos y diseñar una estrategia de preprocesamiento adecuada.

5.4.1.- Análisis de la naturaleza de los datos

Con el objetivo de determinar tanto la técnica de preprocesamiento que más se ajusta a la naturaleza de los datos, como la codificación más apropiada para el algoritmo que se propone, fueron realizados los análisis de la distribución de los contactos interresiduales, el nivel de balance/desbalance entre clases (contactos y no contactos) y la distribución de los motivos estructurales.

Distribución de contactos

Con el objetivo de realizar el estudio de la distribución de los contactos interresiduales, fueron analizadas las frecuencias de aparición de los mismos en función de la separación de los residuos en la secuencia para diferentes umbrales. Se empleó como rango desde 5Å hasta 15Å. Estos valores se tomaron teniendo en cuenta los más frecuentemente citados en la literatura (5 – 12Å). Además, fueron analizados umbrales en el rango de 13 y 15Å tal que permitiera analizar el comportamiento de la distribución de contactos para estos valores a medida que aumentaban.

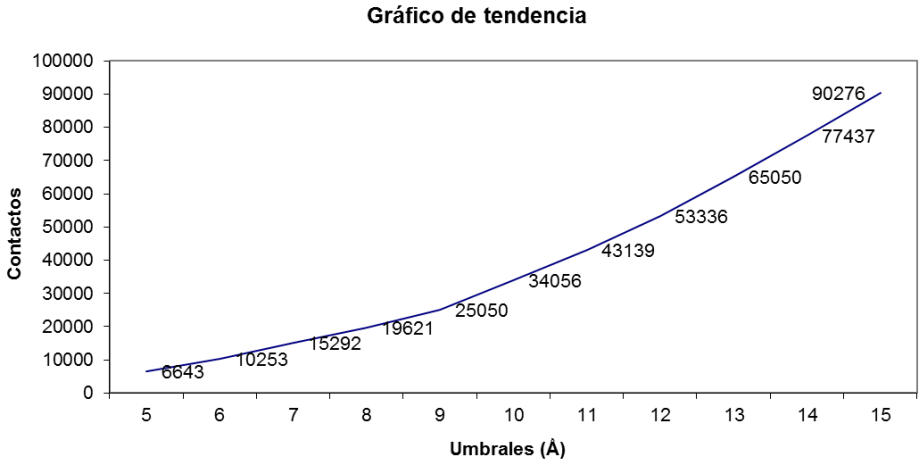


Figura 26. Comportamiento de la ocurrencia de los contactos en función del umbral. Análisis realizado con 53 proteínas globulinas. Longitud de secuencia máxima de 300 aminoácidos. Los contactos fueron calculados teniendo en cuenta la secuencia completa. Los umbrales empleados se establecieron en el rango de 5 a 15Å. El incremento de los contactos describe una función exponencial.

En la Figura 26 se muestra la gráfica del comportamiento de la ocurrencia de los contactos en función del umbral. Este análisis permite observar el incremento que se produce a medida que establecen valores de umbrales mayores para la ocurrencia de contactos. Este incremento parece describir una función exponencial con muchos grados de apertura.

Mediante la representación del histograma de la frecuencia de los contactos en la base de datos de proteínas en función de la separación de la secuencia de residuos, se demuestra que el patrón de distribución de los contactos está lejos de ser uniforme (Figura 27), y que los contactos se concentran, predominantemente, en los residuos que se encuentran cercanos en la secuencia.

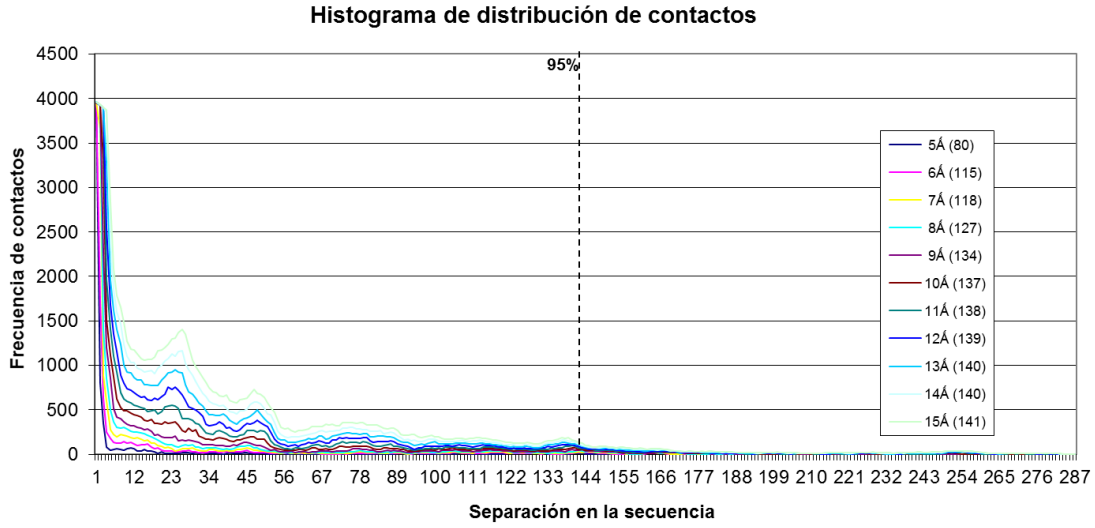


Figura 27. Histograma de distribución de los contactos.
Representación gráfica de las frecuencias de los contactos en función de la separación en la secuencia de los residuos para umbrales de 5 a 15Å. Asumiendo una pérdida del 5% de los contactos, el 95% se concentra en las separaciones en la secuencia, menores e iguales que 140 aminoácidos.

No obstante, a partir de una separación en la secuencia superior a los 100 aminoácidos, se produce un comportamiento casi lineal en la aparición de contactos, prácticamente con monotonía cero, lo cual evidencia que el número de contactos (N_C) se incrementa, aproximadamente, linealmente con la longitud de la secuencia de la proteína, mientras que el número de no contactos (N_{NC}) se incrementa con el cuadrado de la longitud de la proteína.

Teniendo en cuenta que el modelo propuesto podría hacerse impráctico debido a la gran cantidad de recursos que podría consumir, se analizó a qué separaciones en la secuencia se concentró el 95% de los contactos, para cada uno de los umbrales (ver leyenda, Figura 27). Este estudio demostró que para una separación de 25 residuos se puede encontrar el 80% de los contactos, mientras que para una separación máxima de 140 residuos se puede garantizar retener la información referente a la aparición de los contactos en las proteínas con apenas un 5% de error. Lo cual permite crear un modelo más factible desde el punto de vista de coste computacional, tanto en almacenamiento como en procesamiento de la información.

Este estudio también evidenció que tanto para el umbral de 8Å (el más empleado en la literatura) como para el resto de los umbrales, se obtiene prácticamente la misma distribución de contactos, notándose que el 95% de los contactos se encuentra en separaciones inferiores a 140 residuos. Para aportar más información al proceso de aprendizaje, se decide entrenar el algoritmo propuesto con los contactos que se encuentren a una longitud máxima de 140 aminoácidos.

Balanceo de clases

Con el fin de determinar el nivel de desbalance entre contactos y no contactos, fueron analizados los mapas de contactos de una muestra de 12830 proteínas heterogéneas, tomadas del *Protein Data Bank* (PDB) [6], las cuales cumplen con el criterio de identidad menor del 30%. Se calculó cuántas veces aparece cada par posible de aminoácidos, y de ellas, cuántas veces estuvieron o no en contacto (Figura 28).

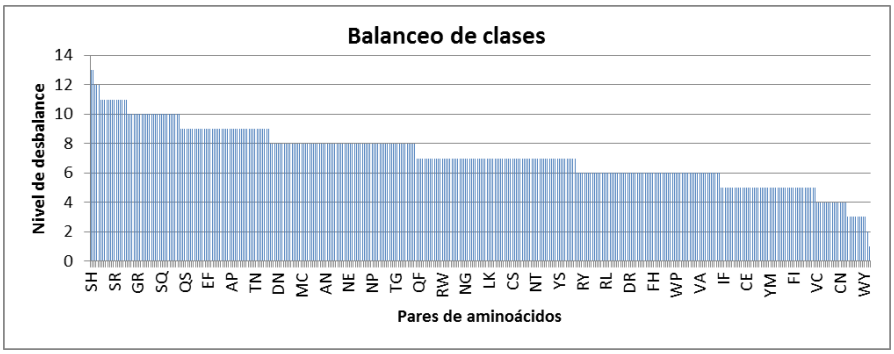


Figura 28. Histograma de distribución de clases realizado a12830 proteínas de identidad 30%.

Acorde a lo reportado por la literatura especializada, este análisis evidenció que los mapas contienen en promedio un número de contactos (N_C) mucho menor que el

número de no contactos (N_{NC}) [48], [59], [185], donde la razón de desbalance (N_C/N_{NC}) es de alrededor 1 contacto por cada 13 no contactos.

Distribución de los motivos estructurales

En este estudio se tuvo en cuenta la distribución de los motivos estructurales (hélices alfa y láminas beta), atendiendo al número de residuos que conforman el motivo (Figura 29).

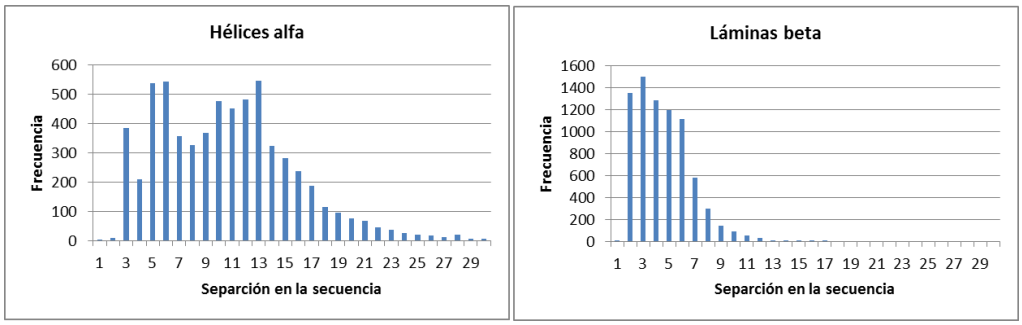


Figura 29. Histogramas de distribución de motivos estructurales.
El 80% de los motivos estructurales se encuentra en separaciones entre 2 y 10 residuos.

Como se puede apreciar en la Figura 29, la longitud promedio de las láminas beta son menores que las hélices alfa. Las hélices aparecen con tamaños de 3 a 20 aminoácidos, mientras que las láminas beta ocupan regiones de 2 a 10 aminoácidos. Como promedio, el 80% de los motivos estructurales se concentran entre los 2 a 10 aminoácidos. Esta conclusión confirma el análisis hecho con la distribución de contactos.

A pesar de que existen evidencias de que la interacción a largo alcance sea quién estabilice el plegamiento de la proteína, el análisis de la naturaleza de los datos demuestra que los contactos no se encuentran distribuidos aleatoriamente. Sino que se agrupan mayormente en interacciones cercanas, concentrándose el 80% a menos de 25 residuos, un 15% a no más de 140 residuos, y sólo un 5% a más residuos de separación.

5.4.2.- Estrategia genética simulada (EGS)

El coste y el desbalance se encuentran estrechamente relacionados en el problema de predicción de mapas de contactos. Sin embargo, debido a la dificultad que implica

asignar costes a la predicción de contactos, en este problema se atacó el problema del desbalance.

Existen numerosas técnicas para el tratamiento del desbalance de datos [186]. Éstas pueden clasificarse empleando la taxonomía que se muestra en la Figura 30, donde las principales aproximaciones son:

1. A nivel de los algoritmos de aprendizaje: no modifican la distribución de los datos, se centran en el ajuste de coste por clase, ajuste de la estimación de probabilidad en las hojas de árboles de decisión, aprender de una sola clase, entre otras [187]–[189].
2. A nivel de distribución de los datos: modifican la distribución de los datos haciendo remuestreo, ya sea reduciendo la clase mayoritaria eliminando ejemplos (submuestreo), o aumentando la clase minoritaria creando nuevos ejemplos (sobremuestreo) [190]–[192].
3. Aproximaciones *Boosting*: generan clasificadores con pesos de forma secuencial y construyen un ensamblado por votación. Ésta es una solución mucho más compleja pero tiene como ventaja que es aplicable a cualquier tipo de error de clasificación [186].

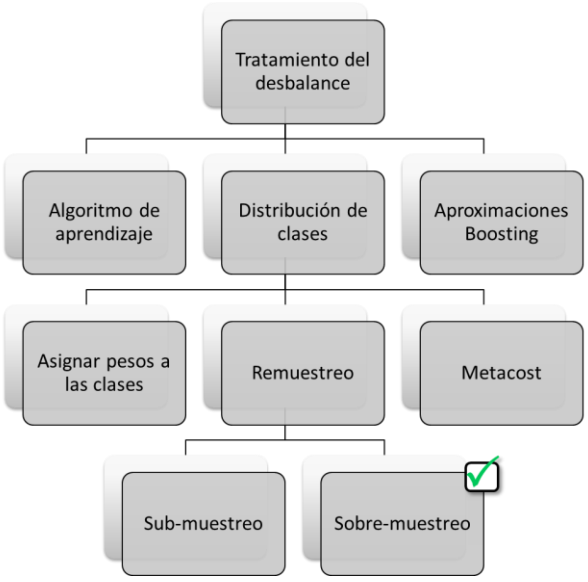


Figura 30. Taxonomía de técnicas de tratamiento del desbalance.

En esta investigación se diseñó un nuevo algoritmo de remuestreo, la Estrategia Genética Simulada (EGS). Este algoritmo realiza un sobremuestreo basado en la probabilidad de aparición de los aminoácidos en las secuencias y emplea una estrategia genética para la generación de nuevas instancias.

Para generar las nuevas instancias, se emplea la estrategia genética que se muestra a continuación (Algoritmo 1):

Algoritmo 1. Estrategia genética simulada (EGS).

Entrada:

- *P*: Población inicial, contiene todas las instancias de la clase "Contacto".

Salida:

- *P'*: Población final.

Algoritmo:

```
1:  P' = P
2:  while (funciónParada(P'))
3:  |   H = Cruzar(SelecciónPadres(P'))
4:  |   H' = Mutar(H)
5:  |   H'' = Evaluar(H')
6:  L   P' = Seleccionar(H'')
7:  return P'
```

Como se muestra en el algoritmo, la **población inicial** es el total de instancias de la clase "Contacto". Esto se debe a que ya el número de vectores es significativamente menor que los de la clase "No Contacto". Ahora, al seleccionar una población menor se corre el riesgo de no cubrir adecuadamente el espacio de búsqueda.

El esquema que se siguió para la selección de los padres es la selección por torneo. Este esquema constituye un procedimiento de selección de padres muy extendido y en el cual la idea consiste en escoger al azar un número de individuos de la población, tamaño del torneo (en este caso con reemplazo), seleccionar el mejor individuo de este grupo (se le asigna una probabilidad de cruce que favorece a aquellos que no se han cruzado antes), y repetir el proceso hasta que el número de individuos seleccionados coincida con el tamaño de la población. El algoritmo propuesto permite la selección de individuos sin que necesariamente sean los mejores, asignándoles una probabilidad de selección distinta a cero, por lo que puede ser considerada una selección preservativa.

La estrategia genética simulada aplica el supuesto que siguen los algoritmos de predicción de estructura basados en plantillas, la cual plantea que secuencias similares, con más de un 30% de identidad, deben tener conformaciones espaciales similares.

Teniendo en cuenta esto, se empleó el **cruce** basado en un punto, en el cual los dos individuos seleccionados para jugar el papel de padres son recombinados por medio de la selección de un punto de corte, para posteriormente intercambiar las secciones que se encuentran a los lados de dicho punto. El punto de cruce se estableció en el centro del cromosoma que permite cruzamiento en el vector (Figura 31), lo cual garantiza que los hijos se parezcan en, al menos, un 50% a sus padres. De esta forma se garantiza cumplir con el supuesto del modelado por homología.

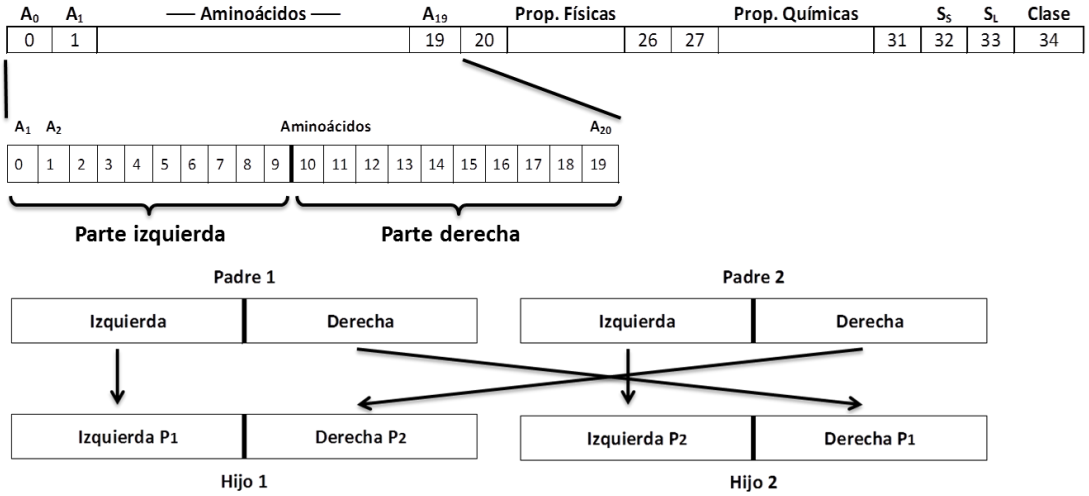


Figura 31. Operador de cruce empleado en la EGS.
En la figura se muestran dos de los cuatro nuevos hijos que se obtienen combinando la parte izquierda y derecha del cromosoma a la hora de realizar el cruce entre padres. El resto del vector (propiedades físico-químicas, separación en la secuencia y longitud de la secuencia), tiene que volver a ser calculado, la clase se asigna "Contacto" para mapas binarios o el promedio de ambos padres para mapas de distancias.

Las propiedades físico-químicas, las frecuencias de los átomos, la longitud de la sub-cadena, son parámetros que tienen que ser recalculados para cada uno de los cuatro hijos resultantes del cruce. En el caso de la longitud de la secuencia, se asume la mayor de ambos padres (valor obtenido empíricamente). Para vectores de mapas de contactos, la clase se mantiene siendo "Contacto", sin embargo, cuando se esté empleando mapas de distancias, a la clase se le asigna el promedio de las distancias de ambos padres.

La **mutación** es un operador básico, que proporciona un pequeño elemento de aleatoriedad en la vecindad (entorno) de los individuos de la población. Aunque el operador de cruce es el responsable de efectuar la exploración a lo largo del espacio de posibles soluciones, el operador de mutación introduce un nuevo elemento en la creación de individuos en la nueva población.

En la estrategia propuesta, el operador de mutación reemplaza un aminoácido en el cromosoma del nuevo hijo. La selección de qué aminoácido será reemplazado se realiza aleatoriamente, aunque no se reemplaza por otro aminoácido cualquiera, sino que se recurre al empleo de una matriz de sustitución. Debido a que cada nuevo hijo tiene identidad del 50% con sus padres, la matriz seleccionada es BLOSUM64 (Anexo 1). De esta forma se garantiza que hijos mutados y padres mantengan similar identidad.

En la mayoría de las implementaciones de algoritmos genéticos se asume que tanto la probabilidad de cruce como la de mutación permanecen constantes, sin embargo, en este caso la probabilidad de mutación se incrementa en 0.025 a medida que aumenta el número de iteraciones. Esto se hace con la intención de disminuir la probabilidad de que existan hijos repetidos en la población.

Para la **evaluación** de los nuevos individuos, se realiza un proceso de alineamiento con sus padres, empleando alineamiento local con el algoritmo Smith-Waterman [176], [193], [194], el cual está basado en el uso de algoritmos de programación dinámica, de tal forma que garantiza que el alineamiento local encontrado es óptimo con respecto al sistema de puntajes usado (Anexo 1). Como medida de calidad del individuo, se le asigna el valor promedio de la identidad entre éste y sus padres.

De los hijos evaluados sólo se podrán **seleccionar** en la población inicial aquellos cuya evaluación sea superior al 50%, una vez más, para garantizar el supuesto del modelado por homología.

La **función de parada** de la EGS incluye dos criterios: el algoritmo propuesto realiza tantas iteraciones como sean necesarias, hasta lograr que el nivel de desbalance entre las instancias de “Contacto” y “No Contacto” desaparezca; realizar un máximo de iteraciones equivalente al doble de la razón de desbalance, valor que se obtuvo

empíricamente y busca garantizar suficientes hijos, tal que los desestimados por el operador de inserción no afecten el tamaño de la generación resultante.

El **costo computacional** de este algoritmo puede calcularse en función del tamaño de la población, el número de generaciones y de costo de la función de evaluación. Donde, si se toma en cuenta que el tamaño de la población inicial se corresponde con las instancias de la clase "Contacto" y que el número de hijos producto del cruzamiento equivale a 4 veces el número de padres, entonces el costo asociado al tamaño de la población equivale a ***4/13·n*** (ver epígrafe 5.4.1, análisis de la naturaleza de los datos). El número de generaciones máxima equivale al doble de la razón de desbalance, por lo que se considera un costo constante (26 para el caso más crítico). Mientras que la función de evaluación compara el nivel de balance entre clases (tamaño de los arreglos), considerándose constante. De lo cual se deduce que la estrategia genética simulada (EGS) es un algoritmo con complejidad lineal, ***O(n)***.

5.5.- Diseño del multiclasificador FoDT

La solución que se presenta en este trabajo, denominada FODT (Forest of Decision Trees), combina los resultados de 400 árboles, uno por cada pareja de aminoácido posible (20 x 20). Para obtener como resultado final una única predicción conjunta, FoDT combina las respuestas de los clasificadores base, tomando en cuenta la predicción realizada por el árbol que fue entrenado para un par de aminoácidos específico (Figura 32). De esta manera, FoDT va construyendo el mapa de contacto correspondiente a la proteína que se está prediciendo.

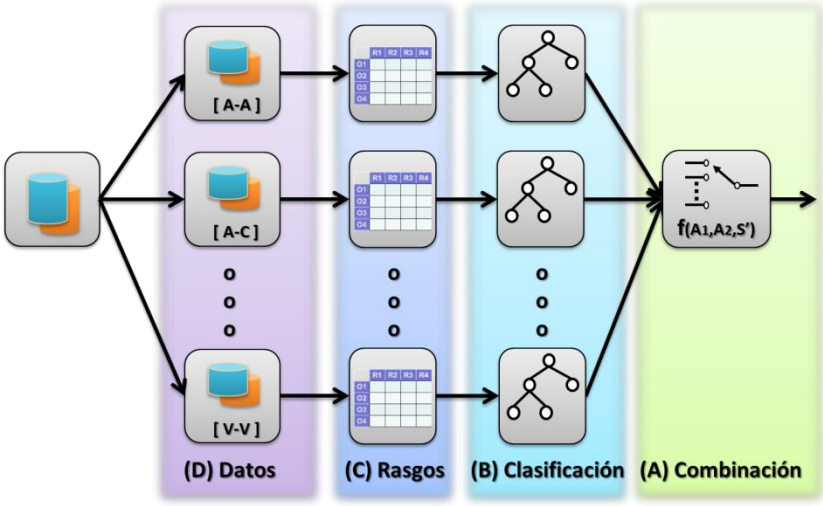


Figura 32. Arquitectura del multiclasificador FoDT.
La diversidad en el multiclasificador se logra en el nivel de datos (D), donde cada clasificador base recibe sólo la información referente al par de aminoácidos para el que será entrenado. En el nivel de rasgos (C), se generan los vectores según la codificación explicada en el epígrafe 5.3. Los clasificadores base (B), emplean el mismo algoritmo de construcción de árboles. La combinación (A), se realiza mediante la selección del clasificador adecuado para cada caso que entra al multiclasificador

En FoDT, cada uno de los 400 clasificadores base se especializa en la solución de un problema específico, relacionado con la predicción de la aparición de posibles contactos para la pareja de aminoácidos que representa. Para lograr este efecto, se adoptó una arquitectura horizontal o paralela.

Esta arquitectura resulta muy sencilla de implementar debido a la independencia entre los clasificadores base y a que la parametrización inicial de cada uno es la misma. A diferencia de la mayoría de los esquemas de este tipo, como *Random Forests*, *Árbitros de Árboles* y *Combinación de Árboles*, entre otros, donde todos los clasificadores base están activados a la vez, en FoDT sólo se activa un clasificador por vez, en dependencia del par de aminoácidos que se esté prediciendo. Esto reduce significativamente el elevado coste computacional que caracteriza a los multiclasificadores de arquitectura paralela.

En FoDT se emplea un único algoritmo de construcción de los clasificador base, la diversidad se logra a partir de la variación de los datos de entrenamiento. Por lo que FoDT podría agruparse en la misma categoría que *Bagging, Random Forests, Random*

Subspaces y *Boostinq*. No obstante, muestra notables diferencias respecto a estos clasificadores.

Los conjuntos de entrenamientos que se emplean en FoDT son disjuntos y no emplean reemplazamiento con *bootstrap*, en lo cual difiere de *Bagging y Random Subspaces*. Además, el vector de entrada en el entrenamiento de cada uno de los clasificadores contiene todos los atributos del conjunto inicial y la predicción ocurre mediante la selección del clasificador adecuado para cada problema, lo cual hace FoDT notablemente diferente de *Random Forests* y cualquiera de sus variantes.

Esta arquitectura de 400 clasificadores base es rígida, debido a que responde a todas las posibles combinaciones de parejas de aminoácidos que puedan formarse (20 x 20). Esta característica hace que FoDT difiera también de *Boosting,* que construye los clasificadores base de forma iterativa en dependencia de las instancias mal clasificadas.

En cuanto a la arquitectura paralela escogida para FoDT y al empleo de árboles como clasificadores base, también podríamos incluirlo en la familia de multiclasificadores como *Árbitros de Árboles y Combinación de Árboles*. Pero, en este caso, difiere de ambos clasificadores en que FoDT sólo emplea un nivel de clasificación, o sea, la salida de clasificadores base no son empleadas como entrada de un nuevo nivel en cascada.

5.5.1.- Análisis de la diversidad

La diversidad de los clasificadores base ha sido obtenida a través de distintas estrategias, pero la mayoría se basa en hacer algún tipo de modificación en el conjunto de entrenamiento de cada uno de los clasificadores base. Ya sea tomar distintos subconjuntos para el entrenamiento (*Bagging y Random Subspaces*), selección de los atributos que se emplean (*Random Forests*), entrenamiento iterativo variando los pesos de las instancias que se van a utilizar (*Boosting*).

La diversidad de FoDT se logra, de igual manera, incidiendo sobre el nivel de base de datos (Figura 32.D). Las instancias que componen el set de entrenamiento son separadas para cada pareja de amino ácidos posible a formarse (400 subconjuntos de datos). Sin embargo, en el nivel de extracción de rasgos (Figura 32.C), se mantiene el mismo vector de características para todos los clasificadores base. Esto fuerza a que

cada clasificador base se construya, únicamente, en dependencia de las instancias que recibe, logrando la deseada diversidad (Figura 32).

Este método presenta varias ventajas:

- Sencillez: es la mayor de todas las ventajas, debido a que no requiere de un esfuerzo computacional extra para lograr la diversidad ni depende de configuraciones de estados anteriores
- Versatilidad: la estrategia para lograr la diversidad es totalmente independiente al clasificador base empleado, lo que la hace fácil de exportar.
- No ambigua: no presenta componente aleatorio alguno, dividiendo el espacio de búsqueda en regiones fijas, por lo que siempre genera las mismas 400 poblaciones iniciales de instancias. Esto permite la reproducción exacta de la experimentación.

5.5.2.- Selección del clasificador base

Algunos algoritmos constructores de clasificadores base que podrían haber sido empleados son: las redes neuronales artificiales (ANN) [195], las máquinas de vectores soporte (SVM) [196], los clasificadores bayesianos (BNs) [197], vecinos más cercanos (kNN) [198], árboles [199], entre otros.

En los casos de las ANN y SVM, se trata de clasificadores que maximizan su desempeño en problemas linealmente separables. En la (Tabla 4) se presenta una comparación rápida de los tipos clasificadores base más populares y el clasificador base propuesto (árboles de decisión, DTs).

Tabla 4. Comparativa de clasificadores base.

Algoritmo	Tipo de datos	Tipo de Problema	Tipo de Clasificador	Base conocimiento
ANN	Numéricos	Se beneficia de que el problema sea linealmente separable	Multiclase	Matriz numérica
SVM	Numéricos	Se beneficia de que el problema sea linealmente separable	Dividen el espacio en dos (clasificadores binarios)	Matriz vectores
BNs	Numéricos y Nominales	Cualquier tipo de problema	Multiclase	Probabilidades
kNN	Numéricos y Nominales	Cualquier tipo de problema	Multiclase	Matriz de entrenamient

				o
Árboles	Numéricos y Nominales	Cualquier tipo de problema	Multiclase (decisión) / Clase numérica (regresión)	Conjunto de reglas

Debido a que se pretende trabajar con datos numéricos (distancias) y nominales (contactos y no contactos), a que no se conoce si el problema es linealmente separable o no y a que se requiere contar con un modelo que sea fácilmente interpretable, la mejor opción podría ser el empleo de árboles, tanto de regresión (M5') como de decisión (C4.5). Sin embargo, se impone realizar un estudio experimental para la selección del algoritmo de construcción del clasificador base más adecuado.

Análisis experimental

Para el desarrollo de la experimentación fueron seleccionados algunos de los algoritmos de aprendizaje supervisado más empleados: red neuronal *feet-forward* (ANN), red bayesiana (BNs), k-vecinos más cercanos (kNN) y árbol de decisión (DTs). Fueron empleadas las implementaciones de Weka [200], manteniendo sus configuraciones por defecto. Como vector de entrada, se utilizó la codificación propuesta en esta investigación.

Esta experimentación busca determinar el domino de aplicación de los algoritmos seleccionados. Con este propósito, fue empleado el conjunto de proteínas propuesto en el Anexo 2. En la Tabla 5, se muestran los resultados de la experimentación para cada uno de los motivos estructurales analizados.

Tabla 5. Estudio comparativo del desempeño de los algoritmos en cuanto a las medidas de Efectividad (Acc), Cobertura (Cov), Media armónica (F-measure) y área bajo la curva ROC (AUC).

Dominios	Métodos	Acc	Cov	Fmeasure	AUC
alpha	ANN	0.874	0.540	0.667	0.907
alpha	BNs	0.182	**0.792**	0.296	0.901
alpha	DTs	**0.884**	0.613	**0.724**	**0.910**
alpha	kNN	0.503	0.583	0.540	0.863
beta	ANN	0.715	0.568	0.633	0.862
beta	BNs	0.184	**0.740**	0.295	0.876
beta	DTs	**0.825**	0.558	**0.666**	**0.887**
beta	kNN	0.525	0.519	0.522	0.834
alpha/beta	ANN	0.809	0.596	0.686	0.881
alpha/beta	BNs	0.162	**0.767**	0.267	0.893
alpha/beta	DTs	**0.849**	0.651	**0.737**	**0.913**
alpha/beta	kNN	0.561	0.583	0.572	0.853
alpha+beta	ANN	**0.918**	0.426	0.582	0.811

alpha+beta	BNs	0.181	**0.695**	0.287	0.850
alpha+beta	DTs	0.874	0.528	**0.659**	**0.903**
alpha+beta	kNN	0.461	0.449	0.455	0.803

Con la intención de establecer si existen diferencias estadísticamente significativas en el comportamiento de dichos algoritmos, fueron aplicados un test de Friedman con rangos y los procedimientos post-hoc de Bonferroni-Dunn, Holm, Hochberg and Hommel con un $\alpha = 0.10$ [201].

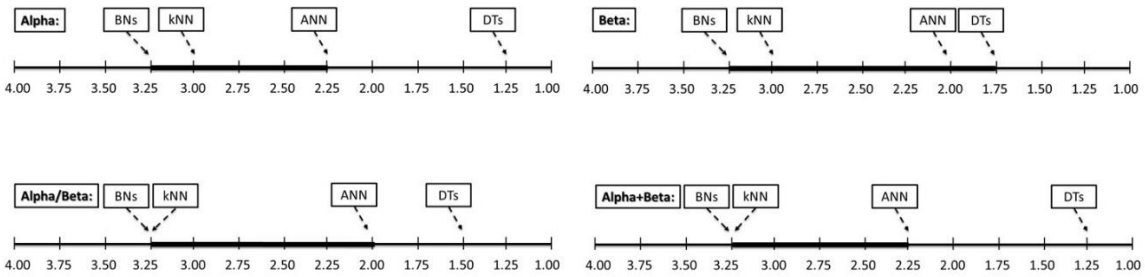


Figura 33. Comparación visual del desempeño de los algoritmos para cada dominio de aplicación. Los algoritmos conectados por la línea gruesa no muestran diferencias estadísticamente significativas en su rendimiento. Luego de aplicar los test post-hoc de Bonferroni-Dunn, Holm, Hochberg and Hommel con un $\alpha = 0{,}10$.

La Figura 33 muestra claramente que, para cada uno de los dominios de aplicación, el árbol de decisión presenta un mejor desempeño que el resto de los algoritmos. A excepción de las proteínas con motivos estructurales beta, donde no existen diferencias significativas en el comportamiento de los algoritmos, en el resto de los subconjuntos, DTs es significativamente mejor.

Estos resultados en conjunción con la capacidad explicativa de los DTs, lo convierten en el algoritmo adecuado para la construcción de los clasificadores base del multiclasificador que se propone en la presente investigación.

Como valor añadido se puede señalar que los árboles son una colección de nodos conectados entre sí, cada uno con un ascendiente (a excepción del nodo raíz que no tiene), y cero o más descendientes. A los nodos sin descendientes se les conoce como hojas y tienen asociada una clase (para el caso de decisión) o una función de regresión (para el caso de regresión). Los nodos que no son hojas contienen una función con la que evaluar la instancia que se esté clasificando[199], [202], [203]. Donde, el proceso de clasificación consiste en ir recorriendo los nodos desde la raíz hasta una hoja. El recorrido viene dado por cómo "responda" la instancia a cada una de las decisiones

que el árbol planteará en cada nodo, de forma que cada posible respuesta puede interpretarse como un conjunto de reglas ***Si, Entonces, Sino***. Dándoles así capacidad explicativa a estos algoritmos.

Por tal motivo, el multiclasificador FoDT fue evaluado empleando tanto árboles de decisión como de regresión. En el caso de decisión, se empleó el J48, implementación de Weka del C4.5, propuesto por Quinlan [199], [204]. En el caso de regresión se empleó el M5' [205], también la implementación de Weka.

5.5.3.- Combinación de los resultados

Para garantizar una toma de decisión del multiclasificador propuesto, se diseñó una arquitectura que emplea una combinación simple de los clasificadores base, mediante la selección, donde cada clasificador base soluciona un problema por separado, por lo que puede considerarse independencia y cooperación entre clasificadores.

El proceso de selección ocurre tanto en la fase de entrenamiento como en la de clasificación. Debido a que a cada uno de los 400 clasificadores base llegan sólo las instancias pertenecientes a un par de aminoácidos específico, luego durante la clasificación, para una instancia dada, se selecciona cuál de los 400 clasificadores debe ser activado y se toma el resultado que devuelve.

5.6.- Filtrado de la clasificación

Uno de los principales inconvenientes de FoDT es la sobrepredicción de contactos. El número de contactos predicho viola el número de contactos reales, observados, que puede establecer un aminoácido en la estructura 3D de la proteína. Con la intención de disminuir este efecto indeseado, se diseñó un procedimiento de filtrado que incluye el análisis del orden de contactos (CO), la propensión de los aminoácidos de entrar en contacto y restricciones basadas en la estructura secundaria.

5.6.1.-Orden de contactos (CO) e Índice de contactos múltiples (MCI)

Como parte del filtrado se adicionó una nueva restricción al algoritmo tal que se tenga en consideración la correlación entre la constante de plegamiento (***k***) y la topología de la proteína, mediante el parámetro orden de contactos (***CO***).

$$\mathrm{CO} = \frac{1}{\mathrm{LN}} \sum_i^N \sum_j^N \Delta \mathrm{Z_{i,j}} \qquad (17)$$

donde ***N*** es la cantidad total de contactos en la proteína, $\Delta \mathbf{Z_{i,j}}$ es la separación en la secuencia a que se encuentran los residuos ***i*** y ***j***, y L es la longitud de la proteína.

En una proteína con bajo ***CO***, por lo general, los residuos interactúan con otros que se encuentran cercanos en la secuencia. Por lo que un alto ***CO*** implicaría la existencia de un gran número de interacciones a largo alcance [32], [56], [206].

Para disminuir la sobrepredicción de contactos, los contactos predichos son filtrados teniendo en cuenta la cantidad de contactos que cada tipo de residuo puede tener [48]. El procedimiento de filtrado se basa en el número de coordinación de residuos (*occupancy data or residue coordination numbers*). Este valor se obtiene estadísticamente del conjunto de proteínas y tiene en cuenta los tipos de estructuras secundarias y la solubilidad de cada residuo (*solvent exposition*).

Teniendo en cuenta esto, el número de contactos predichos estaría en función del entorno estructural. Por tal motivo, el número de coordinación de residuos es considerado como un estimado del máximo número de contactos que cada residuo puede tener. Luego, este valor se emplea para limitar el número de contactos predicho para cada aminoácido.

5.6.2.- Matriz de propensión de contactos

Para garantizar que esta estrategia tenga sentido, desde el punto de vista biológico, es necesaria la definición de qué residuos tienen mayor probabilidad de entrar en contacto, por lo que se calculó una matriz de propensión de contactos.

La matriz de propensión de contactos, se obtiene a partir de un análisis estadístico del set de entrenamiento, calculándose los porcentajes de contactos para cada uno de los 400 posibles pares, empleando la fórmula:

$$Paa = \frac{N_C}{N_C + N_{NC}} \qquad (18)$$

donde, ***Paa*** es la propensión a contacto del par de aminoácidos ***aa***, ***Nc*** es la frecuencia de contactos y N_{NC} la frecuencia de no contactos.

Los valores de la tabla son normalizados en base al par de amino ácidos que mayor propensión tenga (Tabla 6). En este caso, el par CC (Cisteína – Cisteína).

Tabla 6. Matriz de propensión de contactos.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	0,36	0,31	0,21	0,37	0,44	0,32	0,33	0,32	0,25	0,46	0,44	0,35	0,34	0,43	0,25	0,31	0,27	0,31	0,35	0,39
R	0,33	0,3	0,35	0,33	0,37	0,33	0,35	0,28	0,28	0,41	0,31	0,27	0,42	0,36	0,23	0,3	0,25	0,4	0,34	0,48
N	0,42	0,33	0,28	0,26	0,55	0,39	0,25	0,33	0,48	0,28	0,32	0,33	0,3	0,32	0,29	0,27	0,39	0,37	0,34	0,35
D	0,33	0,41	0,35	0,27	0,42	0,29	0,32	0,33	0,19	0,36	0,3	0,25	0,25	0,34	0,33	0,34	0,41	0,21	0,42	0,27
C	0,5	0,4	0,52	0,28	**1**	0,37	0,48	0,54	0,43	0,57	0,48	0,41	0,47	0,74	0,39	0,4	0,4	0,68	0,63	0,4
Q	0,37	0,27	0,25	0,2	0,32	0,33	0,29	0,42	0,22	0,36	0,35	0,31	0,32	0,34	0,23	0,25	0,4	0,36	0,28	0,39
E	0,36	0,28	0,27	0,22	0,42	0,28	0,31	0,28	0,32	0,29	0,34	0,29	0,26	0,35	0,24	0,22	0,27	0,35	0,24	0,31
G	0,3	0,29	0,32	0,24	0,46	0,31	0,24	0,3	0,33	0,31	0,33	0,35	0,3	0,4	0,32	0,32	0,34	0,43	0,38	0,34
H	0,42	0,31	0,57	0,24	0,55	0,52	0,25	0,31	0,29	0,46	0,43	0,22	0,55	0,27	0,26	0,4	0,47	0,47	0,39	0,41
I	0,44	0,29	0,32	0,3	0,61	0,39	0,36	0,3	0,43	0,44	0,41	0,32	0,23	0,46	0,3	0,32	0,5	0,42	0,44	0,63
L	0,42	0,27	0,3	0,29	0,5	0,45	0,3	0,34	0,35	0,42	0,4	0,31	0,52	0,41	0,31	0,21	0,38	0,4	0,37	0,43
K	0,32	0,28	0,24	0,26	0,41	0,26	0,33	0,37	0,27	0,38	0,37	0,27	0,3	0,42	0,27	0,28	0,35	0,33	0,31	0,39
M	0,31	0,32	0,32	0,37	0,44	0,31	0,38	0,31	0,37	0,3	0,36	0,35	0,35	0,38	0,35	0,25	0,28	0,29	0,28	0,39
F	0,4	0,39	0,4	0,3	0,55	0,39	0,29	0,25	0,41	0,37	0,42	0,3	0,24	0,42	0,2	0,34	0,43	0,31	0,43	0,51
P	0,28	0,18	0,43	0,32	0,37	0,38	0,29	0,33	0,24	0,25	0,33	0,31	0,3	0,36	0,23	0,36	0,33	0,32	0,33	0,32
S	0,29	0,28	0,25	0,34	0,46	0,43	0,35	0,36	0,2	0,36	0,32	0,3	0,31	0,36	0,32	0,27	0,38	0,3	0,26	0,36
T	0,32	0,26	0,23	0,33	0,37	0,34	0,37	0,35	0,29	0,39	0,35	0,25	0,29	0,36	0,23	0,26	0,37	0,44	0,32	0,4
W	0,54	0,38	0,43	0,5	0,68	0,51	0,32	0,49	0,34	0,55	0,42	0,29	0,61	0,76	0,29	0,38	0,44	0,46	0,28	0,51
Y	0,42	0,3	0,41	0,29	0,72	0,46	0,29	0,34	0,38	0,43	0,44	0,38	0,3	0,46	0,31	0,28	0,41	0,41	0,6	0,43
V	0,4	0,33	0,26	0,25	0,48	0,38	0,39	0,29	0,38	0,49	0,46	0,38	0,39	0,57	0,37	0,37	0,44	0,35	0,49	0,55

La definición de qué contactos se eligen estaría en dependencia directa de del orden de contactos (si ***CO*** es bajo, se priorizan los contactos a corto alcance, de lo contrario los de largo alcance), de la cantidad de contactos máximo que cada residuo puede tener (número de coordinación de residuos) y de la probabilidad de que un residuo entre en contacto con otro (matriz de propensión de contactos).

5.6.3.- Restricciones basadas en la estructura secundaria

Por último, fueron agregadas cuatro restricciones basadas en la frecuencia de ocurrencia de estos casos [14]:

- Una hebra **β** puede formar láminas **β** con un máximo de dos hebras **β**.
- Los amino ácidos de inicio y fin de un segmento de lanzo no entran en contacto.
- Un residuo ***Ai*** no puede entrar en contacto con los residuos ***Aj*** y ***Aj+2*** cuando ***Aj*** y ***Aj+2*** están en la misma hélice ***α***.
- Una pareja de amino ácidos ***Ai*** y ***Aj*** están en contacto si sus amino ácidos paralelos y antiparalelos están en contacto.

5.7.- Formalización y Algoritmo

Para realizar la formalización del algoritmo, se tendrá en cuenta cuáles son los parámetros de entrada y salida tanto en el proceso de entrenamiento como en el proceso de predicción. Se mostrará el pseudocódigo de ambos algoritmos y se realizará su análisis de complejidad.

5.7.1.- Construcción y entrenamiento de FoDT

La construcción y entrenamiento del multiclasificador FoDT comienza con la extracción de los vectores que se forman por cada una de las proteínas del conjunto de entrenamiento. Posteriormente se subdividen las instancias según al par de aminoácidos a que pertenezcan, quedando formados 400 subconjuntos de entrenamiento. Cada uno de estos subconjuntos es preprocesado, empleando la estrategia genética simulada (EGS) propuesta en esta investigación (Algoritmo 1). Por último cada uno de los 400 árboles es entrenado con el subconjunto que le corresponde.

Para facilitar el entendimiento del proceso de construcción y entrenamiento de FoDT, éste podría formalizarse de la siguiente manera (Algoritmo 2):

Algoritmo 2. FoDT: Entrenamiento y construcción del algoritmo

Entrada:

- *MC*: Conjunto inicial de matrices de mapas de contacto empleados para el entrenamiento.

Salida:

- *modelo*: Matriz de 400 árboles (20 x 20 aminoácidos).

Algoritmo:

```
1:  foreach mc ∈ MC
2:  |   for A1 ∈ Fila : Fila = {1, ..., L-1} ⊆ mc[Fila, Columna]
3:  |   |   for A2 ∈ Columna : Columna = {A1+1, ..., L} ⊆ mc[Fila, Columna]
4:  ⌊   ⌊   ⌊   matrizVectores[A1, A2]  = Codificar(A1, A2)
5:  for A1 ∈ Fila : Fila = {1, ..., 20} ⊆ matrizVectores[Fila, Columna]
6:  |   for A2 ∈ Columna : Columna = {1, ..., 20} ⊆ matrizVectores[Fila, Columna]
7:  |   |   matrizVectores[A1, A2] = EGS(matrizVectores)      // ver algoritmo 1
8:  ⌊   ⌊   modelo[A1, A2]  = ConstruirÁrbol(matrizVectores[A1, A2])
9:  return modelo
```

Para calcular el coste computacional de la construcción del algoritmo FoDT es necesario analizar cada paso del algoritmo independientemente. Pero, para hacer más

simple el entendimiento, la complejidad del algoritmo será analizada a partir del análisis de sus dos ciclos principales: de la instrucción 1 a la 4 y de la 5 a la 8.

El primer ciclo (de la instrucción 1 a la 4) resulta constante, debido a que depende del número de mapas de contactos que se empleen durante la construcción del predictor. Los siclos subsiguientes (instrucciones 2 y 3) recorren una diagonal para cada una de las matrices que representan estos mapas de contactos, por lo cual la complejidad se puede describir como $n \cdot (n-1)/2$, donde n se corresponde con la longitud de la secuencia de las proteínas. La codificación (instrucción 4), es un proceso lineal $O(n)$. Considerándose, entonces este primer conjunto de instrucciones con complejidad $n^2 \cdot (n-1)/2$, o lo que es igual, complejidad cúbica $O(n^3)$.

El segundo ciclo (de la instrucción 5 a la 8) resulta constante, debido a que siempre recorre una matriz de 20 x 20. El preprocesamiento, mediante el empleo de la estrategia genética simulada (EGS) equivale al empleo de 400 algoritmos genéticos (AG), por lo que la complejidad puede definirse como $a \cdot O(n)$, donde a es el número de AG (ver epígrafe 5.4.2, estrategia genética simulada). De igual forma, la construcción del modelo equivale al coste de construir 400 árboles. Esto podría escribirse como $a \cdot O(m \cdot n \cdot log(n))$, donde a es 400 (número de árboles) y m el número de atributos. Considerándose, entonces este segundo conjunto de instrucciones con complejidad superior a la logarítmica pero inferior a la cuadrática.

A partir de este análisis, podría decirse que el coste promedio de la construcción del predictor es $O(n^3) + a \cdot O(m \cdot n \cdot log_2(n))$. Donde, la mayor complejidad es $O(n^3)$, por lo que la complejidad FoDT es **cúbica.**

5.7.2.- Predicción de mapas de contacto empleando FoDT

El mapa de contactos de una proteína desconocida se predice evaluando, en el modelo construido por FoDT, la secuencia implícita entre cada pareja de aminoácidos que se pueden formar en la secuencia. Este proceso podría formalizarse de la siguiente manera (Algoritmo 3):

Algoritmo 3. FoDT: proceso de predicción de mapas de contacto.

Entrada:

- mc: Mapa de contacto a predecir.
- modelo: Matriz de 400 árboles construida por FoDT (ver algoritmo 2).

Salida:

- mcp: Mapa de contacto predicho.

Algoritmo:

```
1:  for A1 ∈ Fila : Fila = {1, ..., L-1} ⊆ mc(Fila, Columna)
2:  |   for A2 ∈ Columna : Columna = {A1+1, ..., L} ⊆ mc(Fila, Columna)
3:  |   |   S' = Codificar(A1, A2)
4:  |   |   DT = modelo[A1, A2]
5:  └   └   mc'[A1, A2] = DT(S')
6:  mcp = Postprocesar(mc')
7:  return mcp
```

Debido a que la predicción se realiza empleando un mecanismo de combinación por selección, durante este proceso sólo se activa uno de los 400 árboles a la vez. Donde, el número de iteraciones del algoritmo (instrucciones 1 y 2) disminuyen a medida que se recorre la secuencia de la proteína $\boldsymbol{n\cdot(n-1)/2}$. La codificación (instrucción 3), es un proceso lineal $\boldsymbol{O(n)}$. La instrucción 4 es una simple asignación. El coste de la predicción del árbol de decisión (instrucción 5), depende exclusivamente del tamaño medio de los árboles ***h.*** Por tanto, el coste promedio de la predicción empleando FoDT es $\boldsymbol{n^2\cdot(n-1)/2}$, o lo que es igual, complejidad cúbica $\boldsymbol{O(n^3)}$.

5.8.- Conclusiones parciales

En este capítulo se formalizó el algoritmo propuesto, FoDT, el que podría catalogarse como un multiclasificador de combinación por selección. El cuál, a diferencia de los multiclasificadores que optimizan la decisión al combinar la predicción de distintos clasificadores base, realiza la optimización por cobertura, o sea, que logra la diversidad a partir de entrenar los clasificadores base con diferentes subconjuntos de entradas. Además, también se clasifica como no entrenable, debido a que no es necesario realizar un entrenamiento extra en el nivel de combinación para realizar la selección.

Se propone un vector de codificación que combina la información de la subsecuencia implícita entre los aminoácidos no adyacentes, propiedades físico-químicas, así como otras propiedades generales de las proteínas.

Se diseñó un nuevo algoritmo de sobremuestreo basado en una estrategia genética que permite afrontar el nivel de desbalance que presenta la base de datos. Y, con el objetivo de prever el problema de sobrepredicción que presenta FoDT, se incluye una propuesta de mejora al procedimiento de post-procesamiento basado en el orden de los contactos [48], que toma en cuenta la matriz de propensión de contactos entre los aminoácidos.

Parte IV

Validación de los resultados

Capítulo 6

Validación experimental de los resultados

En el presente capítulo se realiza la validación de la propuesta realizada en la presente investigación. Se muestran los resultados experimentales para el análisis del dominio de aplicación del algoritmo propuesto, así como de la validación interna y externa del mismo. Adicionalmente, se expone el mecanismo de interpretación de la propuesta.

6.1.- Principios de validación de modelos de la OECD

La validación de modelos químicos y biológicos ha estado sujeta a múltiples debates por parte de la comunidad científica, considerándose de extrema importancia el desarrollo de un conjunto de principios de validación que provea un medio de regulación neutral, sobre bases científicas. A partir de aquí, recientemente, la Unión Europea ha establecido los principios generales para la validación de modelos en el contexto de quimio y bioinformática. Estos principios inicialmente se conocían como SETUBAL, pero en la actualidad se conocen como principios OECD [20].

Estos principios establecen:

- Definición de un objetivo bien definido.
- Algoritmo no ambiguo.
- Definición del dominio de aplicación.
- Medición apropiada de la bondad de ajuste, la robustez y la capacidad de generalización (predictibilidad) del algoritmo propuesto. Este principio incluye un proceso de validación interna y de validación externa.
- Descripción de un mecanismo de interpretación, en caso de que sea posible.

El proceso de validación de la propuesta realizada en esta investigación, el multiclasificador FoDT, estará guiado acorde a los principios de la OECD.

FoDT tiene un **objetivo bien definido**, que es la predicción de mapas de contactos de proteínas. Para lograr este objetivo, el trabajo de FoDT está dividido en tres tareas fundamentales:

- El **acondicionamiento** (preprocesamiento) de los datos: que tiene como objetivo fundamental mitigar el problema de desbalance que presenta el conjunto de datos de entrenamiento sobre la base de los principios biológicos del plegamiento de las proteínas.
- La **clasificación** de contactos y no contactos: que tiene como objetivo predecir cuándo una pareja de aminoácidos entra en contacto, atendiendo a la subsecuencia que se forma entre éstos y a las propiedades físico-químicas de dicha subsecuencia.
- El **postprocesamiento** de los resultados: que tiene como objetivo aplicar un conjunto de restricciones asociadas al proceso de plegamiento de las proteínas, para evitar así el efecto de sobrepredicción que provoca la etapa de clasificación.

FoDT sigue una arquitectura claramente definida, simple, fácil de reproducir y de aplicar. Sin embargo, lo principal es que siempre que se brinden los mismos datos de entrada, genera los mismos resultados, lo que permite que la experimentación propuesta en esta investigación pueda ser reproducida con facilidad y fiabilidad por cualquier investigador. Estas características hacen que FoDT pueda considerarse como un **modelo no ambiguo**, satisfaciendo al segundo principio.

No obstante, un modelo no ambiguo se caracteriza no sólo por el algoritmo propuesto, sino que incluye el vector de codificación diseñado, así como el procedimiento exacto para calcular dicho vector. Incluso, el software empleado en el desarrollo forma parte del modelo.

6.2.- Resultados experimentales

En este epígrafe se presenta la experimentación realizada, se definen los principales estadígrafos, así como las pruebas de significación estadísticas empleadas. Además, se realiza el análisis y discusión de los resultados.

6.3.1.- Medición de la bondad de ajuste, la robustez y la capacidad de generalización

Para dar cumplimiento al cuarto principio de la OECD, es necesaria definir cuáles serán los estadígrafos que serán calculados, así como los test estadísticos que se emplearán para determinar si existen diferencias significativas en el comportamiento de los algoritmos empleados.

Estadígrafos empleados

Muchas de las medidas comúnmente empleadas para evaluar la efectividad de los predictores no funcionan en predicción numérica. Los principios básicos (usando un conjunto de prueba independiente en lugar los métodos *holdout* y *cross-validation* para la evaluación del desempeño con el conjunto de entrenamiento), son igualmente aplicables a la predicción numérica. Pero la calidad de la medición ofrecida por la razón de error no es muy apropiada: los errores no están, simplemente, presentes o ausentes; ellos se presentan en diferentes magnitudes [82].

Pueden emplearse múltiples alternativas para la evaluación del proceso de predicción numérica, como medidas de error:

Error cuadrático medio (*Mean-squared error*): medida más comúnmente usada; en ocasiones tiende a brindar las mismas dimensiones que los valores predichos (19). Tiende a exagerar los valores de los outliers.

$$MSE = \frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n} \qquad (19)$$

Error absoluto medio (*Mean absolute error*): es una alternativa que promedia las magnitudes de los errores individuales sin tener en cuenta sus signos (20). A diferencia del MSE, no exagera los valores de los outliers, todas las dimensiones del error son tratadas por igual, acorde a su magnitud.

$$MAE = \frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{n} \tag{20}$$

Error cuadrático relativo (*Relative squared error*): el error se hace relativo para su uso en predictores simples (21). Un predictor simple es el promedio de los valores actuales de los datos de entrenamiento. De esta manera, el RSE toma el error cuadrático total y lo normaliza dividiendo por el error cuadrático total del predictor por defecto (o predictor simple).

$$RSE = \frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \overline{a})^2 + \ldots + (a_n - \overline{a})^2}, \quad donde \quad \overline{a} = \frac{1}{n}\sum_i a_i \tag{21}$$

Error absoluto relativo (*Relative absolute error*): es el error absoluto total con el mismo tipo de normalización (22). En estas tres medidas de error relativo, los errores son normalizados mediante el error del predictor simple, el cual predice los valores promedios.

$$RAE = \frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|a_1 - \overline{a}| + \ldots + |a_n - \overline{a}|} \tag{22}$$

Coeficiente de Correlación (*Correlation coefficient*): mide la correlación estadística entre los a's y los p's (23). El coeficiente de correlación se expresa en valores en el rango de 1 para los resultados perfectamente correlacionados, 0 para la no correlación a -1 para los perfectamente correlacionados negativamente.

$$CC = \frac{S_{PA}}{\sqrt{S_P S_A}}, \quad donde \quad S_{PA} = \frac{\sum_i (p_i - \overline{p})(a_i - \overline{a})}{n-1}$$
$$S_p = \frac{\sum_i (p_i - \overline{p})^2}{n-1} \quad y \quad S_A = \frac{\sum_i (a_i - \overline{a})^2}{n-1} \tag{23}$$

* donde p son las predicciones y a son los valores actuales

En ocasiones el error relativo es más importante que el absoluto. Esto significa que si el 10% del error es igualmente importante tanto si es un error de 50 en una predicción de 500 o un error de 0.2 en una predicción de 2, entonces el error absoluto sería absurdo y el error relativo sería más apropiado. Este efecto debe tenerse en cuenta para usar los errores relativos en el cálculo del MSE o del MAE.

La correlación negativa no debe ocurrir en métodos de predicción razonables. La correlación es significativamente diferente de otras medidas debido a que es independiente de la escala. Si se tiene un conjunto de predicción específico, el error es inalterable si toda la predicción es multiplicada por un factor constante y los valores actuales se mantienen invariables. A diferencia del resto de las medidas de error donde los valores pequeños indican buen desempeño, el CC debe mostrar valores elevados [82].

Las evaluaciones que se han presentado hasta ahora no tienen en cuenta el coste de tomar las decisiones erróneas (clasificaciones erróneas). La optimización de la tasa de clasificación sin considerar el coste de los errores puede llevar a resultados indeseados.

Una simplificación del problema sería reducirlo a dos clases: contactos y no contactos. Donde los verdaderos positivos (TP) y verdaderos negativos (TN), pertenecen a clasificaciones correctas y los falsos positivos (FP) y falsos negativos (FN) pertenecen a clasificaciones incorrectas[208].

El cálculo de la **efectividad** (*Ac*) [209] estaría definido como la división de las predicciones correctas entre el total de las predicciones (24).

$$Ac = \frac{TP + TN}{TP + FP + TN + FN} \tag{24}$$

Si se tiene en cuenta la naturaleza desbalanceada de las clases presentes en este problema, donde la relación entre el número de contactos (NC), con respecto al número de no contactos (NNC) es aproximadamente NC / NNC = 1 / 13, la efectividad de la predicción (S) podría calcularse como la razón de los verdaderos positivos [209], también conocida como *recall* o **sensibilidad** (25). Esto se debe a que en esta ecuación se penalizan los no contactos y se priorizan los contactos.

$$S = \frac{TP}{TP + FP} \tag{25}$$

También es posible emplear la medida de la precisión del predictor (***Ap***), la cual estaría definida por (26):

$$Ap = \frac{TP}{TP + FN} \quad (26)$$

Para la competición CASP (*Critical Assessment of Structure Prediction*), la medida ***S*** es tratada como la efectividad (*Acc*) del predictor y la medida ***Ap*** es tratada como la cobertura de la clasificación (*Cov*). De acuerdo con CASP, los contactos predichos para cada pareja de aminoácidos de una proteína específica, deberán ser ordenados descendentemente, acorde a los valores de la predicción. Los primeros ***X*** en la lista son los únicos empleados para calcular la efectividad y la cobertura. Los valores de ***X*** más usuales son ***2L***, ***L***, ***L/2*** y ***L/5***, donde L es la longitud de la proteína y, en CASP, ***X=L/5*** es el más empleado [21]–[23], [31]–[33], [35], [40], [52], [106], [210].

Con el objetivo de poder comparar la efectividad del predictor, comúnmente se emplea una medida de efectividad adicional, que es la **efectividad del predictor aleatorio** (Ar). La cual se define como la razón de los contactos reales de la proteína sobre los posibles contactos (27) [48].

$$Ar = \frac{N_C}{N_P} \quad (27)$$

Donde, N_P es el número de todos los posibles contactos. Con el objetivo de descartar los contactos locales, N_P se calcula teniendo en cuenta un umbral mínimo de separación en la secuencia que deben tener los aminoácidos en contacto[48]. Por lo que el cálculo de N_P estaría definido por la ecuación (28):

$$N_P = (Ls - umbral) * (Ls - umbral - 1) \quad (28)$$

A partir de aquí, el cálculo del mejoramiento del desempeño del predictor propuesto (Ap) sobre el predictor aleatorio (Ar), estaría dado por la ecuación (29).

$$R = \frac{Ap}{Ar} \quad (29)$$

Se adicionó un nuevo índice, que mide la diferencia en la distribución de la predicción respecto al valor real[48]. Este índice se define por la ecuación (30):

$$Xd = \sum_{i=1}^{n} \frac{n * (Pic - Pia)}{n * di} \quad (30)$$

donde, *n* es el número de particiones de la distribución de distancias (15 particiones igualmente distribuidas de 4 a 60Å, agrupando a todos los posibles pares a esas distancias, en la estructura observada); *di* es el límite superior (normalizado a 60Å) para cada partición, por ejemplo: 8Å para la partición de 4 – 8Å; *Pic* y *Pia* son los porcentajes de los pares predichos y observados, respectivamente, entre las distancias *di-1* y *di*. Por definición, *Xd*=0 indica que no existe separación entre ambas poblaciones de distancias, otros valores indican ligeros desplazamientos entre las poblaciones. Dado a que el límite superior para contactos es de 8Å, mientras mayor y positivo sea *Xd*, podrá considerarse más efectiva la predicción.

Como paso final en la medición del desempeño del algoritmo propuesto, son promediados los valores de cada índice de tanto los que se emplean sobre los valores reales (*MSE, MAE, RSE, RAE, CC*), como los que se emplean sobre las clases discretas (*Ac, Cov, Acc, R, Xd*).

Procedimiento de estimación de la efectividad del algoritmo

Existen múltiples aproximaciones para estimar la efectividad de los algoritmos, a partir de los estadígrafos antes mencionados, durante el proceso de validación interna. El procedimiento más común es la validación cruzada (CV, *cross validation*). Sin embargo, existen múltiples variantes, bien aceptadas, de este procedimiento como: "*Leave-one-out"* (LOO), "*Leave-many-out"* (LMO) y "*boostraping"* (Boost) [20].

LOO-CV: en este método se entrena al predictor con todos los elementos de la base de datos, menos uno que es empleado como prueba. Este procedimiento se repite para cada uno de los elementos de la base de datos. Esta es una técnica muy útil en conjuntos de datos pequeños, sin embargo este método sobrestima la capacidad de predicción del predictor. Razón por la cual se considera inadecuado para obtener una valoración real del predictor.

LMO-CV: este método se diferencia del LOO-CV en la cantidad de elementos empleados como prueba. En otras palabras, el conjunto de datos se divide en fragmentos iguales, se usa uno para validación y el resto de los fragmentos para el entrenamiento. Este procedimiento se repite para cada una de las particiones y los resultados que se muestran son los obtenidos de la media de las iteraciones corridas.

Esta es una técnica muy útil para grandes conjuntos de datos, brindando una valoración más realista de la capacidad de predicción del predictor.

Boost-CV: este método es similar a LMO-CV, divide aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba, los resultados que se muestran son los obtenidos de la media de las iteraciones corridas. La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar. A diferencia de ***LOO*** y ***LMO***, ***Boost*** es más eficiente y estable. Boost-CV puede verse como una versión suavizada de la validación cruzada. Este método, por lo general, brinda un estimado mucho más preciso del desempeño del predictor.

Sin embargo, para el desarrollo de esta investigación, y con la intención de poder compara el desempeño del predictor en las mismas condiciones que los predictores del estado del arte, el procedimiento de experimentación que se empleará será ***LMO-CV***. Para ello, el conjunto de datos fue dividido en 10 particiones las cuales serán empleadas en el proceso de entrenamiento y validación (***10-fold cross-validation***).

Test estadísticos

Para realizar las comparaciones entre los resultados de los diferentes algoritmos, resulta aconsejable la consideración de algún test estadístico [211], [212]. La selección de estos test está en dependencia de la naturaleza de los datos con que se cuenta. Existen contrastes que no necesitan establecer supuestos exigentes sobre las poblaciones de donde se extraen las muestras y ni que los datos sean obtenidos con una escala de medida de intervalo o razón, a los que se les llaman pruebas no paramétricas.

Teniendo en cuenta que en este caso no es posible determinar el cumplimiento de los supuestos de normalidad y homogeneidad de varianzas, serán empleadas entonces las pruebas no paramétricas.

El objetivo de las pruebas estadísticas a aplicar es realizar la comparación de las bases de datos para los diferentes algoritmos (análisis de varianza o *two way* ANOVA). Para este propósito, se empleó el test de Friedman [213]. Si el resultado de este test es significativo (< 0,05), entones existe una gran posibilidad de que al menos dos de las muestras presenten poblaciones con diferencias en los valores de las medias.

Para poder determinar cuál de las los algoritmos presenta diferencias significativas, es necesario analizar si la distribución de los datos coincide en el sentido que no haya predominio de los incrementos ni de reducciones en la diferencia de los mismos. Para verificar esta hipótesis es empleado el test de Wilcoxon [211]–[215]. Además, se emplean los test de Bonferroni-Dunn, Holm, Hochberg y Hommel, los cuales no realizan análisis de varianza [211].

Test de Friedman

Esta prueba puede considerarse como una extensión de la prueba de Wilcoxon para el caso de más de dos muestras. En el caso de que las asunciones de la prueba ANOVA fuesen satisfechas, el análisis se realizaría de acuerdo a un diseño de ANOVA de dos factores sin repetición, en el que los factores serían respectivamente el conjunto de entrenamiento (bloques) y los algoritmos. Para el procesamiento estadístico se exigió una razón de confianza de 0,95.

Test de Wilcoxon

El test de Wilcoxon [211]–[215], el cual propone calcular las diferencias por pares y ordenarlas en conjunto. Este contraste tiene en cuenta, no sólo, el signo de las diferencias entre los valores de la muestra y la mediana que queremos contrastar, sino también, la magnitud de tales diferencias.

Si la hipótesis fundamental es cierta, el número y magnitud de veces en que los resultados de un algoritmo es mayor que el de otro no debe diferir mucho del número y magnitud de veces que ocurre lo contrario y las diferencias ranqueadas deben equilibrarse. Para el procesamiento estadístico se exigió una razón de confianza de 0,01.

Test de Bonferroni-Dunn

En el test de Bonferroni-Dunn se utiliza la t de Student convencional pero con unos niveles de confianza más exigentes en función del número de contrastes que se van a hacer.

Para este test se utiliza la probabilidad (p) que expresa el nivel de confianza dividida por el número de comparaciones previstas, así si el nivel de confianza es de 0,05 y se prevén realizar tres comparaciones, se utilizará como nivel de confianza 0,05/3 = 0,0167; en este caso 0,0167 equivale a un nivel de confianza de 0,05. También, si se conoce la probabilidad exacta (p), esta puede ser multiplicada por el número de contrastes para ver si llega a 0,05.

El problema de este contraste es que es muy conservador. Esto se traduce como que tiene poca potencia para rechazar la hipótesis nula cuando realmente es falsa, o sea, que da muchos falsos negativos. Por tanto, la interpretación de un resultado depende de que el análisis se haga en solitario o junto con otros análisis.

Test de Holm

El test de Holm se emplea para comparaciones de múltiples clasificadores. Éste ajusta el valor de α empleando un método descendente. Sean ***p1, …, pm*** valores ordenados de probabilidad (de mayor a menor) y ***H1, …, Hm*** las hipótesis correspondientes. El procedimiento de Holm rechaza ***H1*** para ***H(i-1)*** si ***i*** es el menor entero tal que ***pi > α/(m–i+1)***.

Otras alternativas fueron desarrolladas por **Hochberg** y **Hommel**. Estos test son fáciles de realizar, la diferencia es que suelen ser más fuertes que Holm pero no es muy notable, por lo que los tres suelen tener resultados similares.

6.3.2.- Evaluación de la eficiencia del predictor: validación interna

En este tópico se realiza el análisis y discusión de la validación interna del predictor FoDT. Primeramente se describe el diseño experimental empleado, en el cual incluye la selección del conjunto de datos y la herramienta de experimentación empleada. A continuación, se realiza una serie de experimentos con proteínas heterogéneas que permiten la comparación de FoDT, empleando dos algoritmos diferentes de

construcción de árboles: C4.5 y M5'. Este análisis permite evaluar la mejor variante del algoritmo, así como realizar la evaluación de la robustez y la eficiencia del predictor.

6.3.2.1.- Diseño experimental

Selección del conjunto de entrenamiento

Para analizar el comportamiento de un predictor, por lo general, la selección de los datos de entrenamiento depende del problema que se pretende resolver pero en términos generales el conjunto de entrenamiento debe combinar una cobertura máxima con un mínimo de redundancia. Sobre este principio, se seleccionó una base de datos de proteínas de estructuras conocidas y no homólogas, del banco de datos de proteínas (*Protein Data Bank*), las cuáles fueron empleadas para realizar el entrenamiento y la validación del método de predicción propuesto.

Para obtener este conjunto se extrajeron sólo aquellas proteínas con identidad menor del 30%. De las 12.830 proteínas fueron escogidas sólo cadenas cuya estructura no contuviera secuencias redundantes. Además, con el fin de eliminar contactos falsos debido a la presencia de hetero-átomos, sólo fueron mantenidas en el entrenamiento aquellas proteínas sin ligandos en el fichero PDB. Fueron excluidas aquellas cadenas cuyo eje principal estaba interrumpido. Se excluyeron, además, las proteínas que contenían aminoácidos no convencionales. Con el objetivo de evitar pequeños rangos de contactos espurios, el procedimiento propuesto no incluye los contactos entre residuos cuya separación de secuencia sea menor que cuatro residuos.

Para eliminar aquellas proteínas que difieren de la media en cuanto a su (*outliers*), se realizó un análisis del histograma en base a las longitudes de las secuencias, donde sólo fueron excluidas un 5% de las proteínas.

Como resultado de este exhaustivo proceso de criba, sólo resultaron 7.447 proteínas. Este subconjunto aún es extremadamente grande y, por consiguiente, costoso computacionalmente. Por ello se decidió estratificar este subconjunto y extraer sólo el 30% de las proteínas. El primer paso fue agrupar las proteínas en cinco grupos, según su longitud de secuencia (***Ls***). Luego, analizado sus histogramas, y con la intención de que cada subconjunto resultante mantuviera la misma distribución que el original, se realizaron 10 clústeres por cada uno. A cada clúster se le aplicó un proceso de

selección aleatoria del 30% de las proteínas que dando como resultado: 0 ≤ ***Ls*** < 100 (778 proteínas), 100 ≤ ***Ls***< 200 (772 proteínas), 200 ≤ ***Ls***< 300 (308 proteínas), 300 ≤ ***Ls***< 400 (174 proteínas) y ***Ls*** ≥ 400[2] (184 proteínas). Estos subconjuntos mantienen la misma distribución que el conjunto de proteínas inicial (Anexo 3). El set resultante (30%) se destinará para la validación interna (***Set_VI***). Mientras que el set de proteínas restantes (70%), se destina a la validación externa (***Set_VE***).

Para esta experimentación sólo serán empleados los subconjuntos: 0 ≤ ***Ls***< 100, 100 ≤ ***Ls***< 200, 200 ≤ ***Ls***< 300 y 300 ≤ ***Ls***< 400. El sub-conjunto de proteínas de longitudes mayores de 400 aminoácidos no es incluido, debido al alto coste computacional que implica la experimentación con éste.

Este experimento se realizó para demostrar, en primer lugar, la validez de la arquitectura de multiclasificación propuesta; y, en segundo lugar, comparar las posibles ventajas y/o desventajas de emplear decisión o regresión.

Herramienta de experimentación

Para el desarrollo del presente trabajo se empleó una plataforma diseñada para la investigación y desarrollo de métodos de predicción de contactos interresiduales y estructuras de proteínas. Esta plataforma está compuesta por: *Integrated Experimental Environment for Contact Map Methods* (IEEcm por sus siglas en inglés), un software desarrollado para la experimentación en métodos de predicción de mapas de contactos de proteínas; y, por *Bioinformatics* una biblioteca de clases que incluye todo el soporte necesario para el desarrollo de la experimentación y de posteriores aplicaciones basadas en los métodos implementados.

6.3.2.2.- Análisis de la robustez del algoritmo

Para realizar el análisis del comportamiento del algoritmo se realizaron dos implementaciones de FoDT:

- **FoDT_DT**: para esta implementación se empleó el algoritmo de construcción de árboles de decisión J48 (implementación del C4.5 de Weka). El algoritmo fue

[2] Este subconjunto exige un alto consumo de recursos computacionales, debido a la gran cantidad de vectores que se generan.

configurado con un valor de confianza de 0,2 y un mínimo de dos objetos por hojas ("-C 0.2 -M 2").

- **FoDT_RT**: esta implementación emplea el algoritmo M5' para la construcción de los árboles de regresión de los clasificadores base. El algoritmo fue configurado para aceptar un mínimo de cuatro objetos por hojas ("-M 4.0")

Ambas implementaciones fueron evaluadas empleando un procedimiento de validación cruzada con 10 particiones. Con la intención de resaltar la relación entre los resultados de los algoritmos y el tamaño de las proteínas, los valores de efectividad fueron calculados y analizados por separado para cada grupo de proteínas (Tabla 7).

Tabla 7. Resultados experimentales de la comparación entre los algoritmos FoDT_DT y FoDT_RT. Se utilizó como base de datos un conjunto de proteínas con identidad de hasta el 30%, el cual se dividió en 4 particiones. Como resultado se obtuvieron las variables (Ap) que es la efectividad de la predicción, (R) que es el mejoramiento del desempeño sobre un predictor aleatorio, (Xd) que mide la distribución de la predicción respecto al valor real, (S) que es la sensibilidad del predictor y (F) que es F-Measure, la media armónica entre la precisión y la sensibilidad.

Algoritmos	**0 ≤ Ls < 100**		**100 ≤ Ls <200**		**200 ≤ Ls <300**		**300 ≤ Ls < 400**	
	FoDT_DT	**FoDT_RT**	**FoDT_DT**	**FoDT_RT**	**FoDT_DT**	**FoDT_RT**	**FoDT_DT**	**FoDT_RT**
Ap	0,69	0,69	0,64	0,90	0,60	0,80	0,57	0,74
R	4,81	4,88	9,89	13,74	14,97	20,16	19,71	25,36
Xd	0,42	0,01	0,46	-0,38	0,44	-0,34	0,42	-0,32
S	0,72	0,68	0,58	0,45	0,55	0,43	0,53	0,43
F	0,70	0,68	0,61	0,60	0,57	0,56	0,55	0,54

En la Tabla 7 se puede observar el comportamiento de ambas implementaciones del algoritmo. En general, ambas implementaciones mantienen una efectividad estable para las diferentes particiones, superior a 0,57. Sin embargo para las proteínas mayores de 100 aminoácidos de longitud, FoDT_RT muestra un comportamiento aparentemente superior a FoDT_DT (Figura 34).

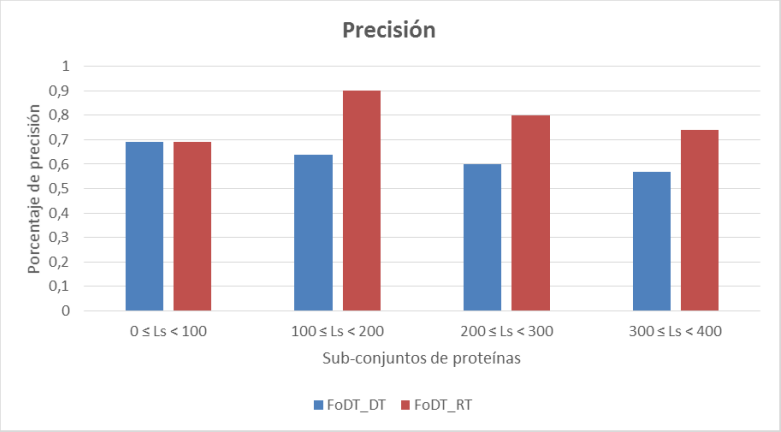


Figura 34. Eficiencia de la predicción de los contactos en función de las longitudes de secuencia de las proteínas para las implementaciones FoDT_DT y FoDT_RT.
En el eje de abscisas se representan los resultados de los predictores, en dependencia de la longitud de las secuencias. El eje de ordenadas representa la efectividad alcanzada.

Sin embargo, el análisis de la sensibilidad o razón de verdaderos positivos (S) muestra un aparente mejor comportamiento del algoritmo FoDT_DT sobre FoDT_RT (Figura 35). Este análisis muestra, además, que la sensibilidad disminuye con el incremento de la longitud de las secuencias.

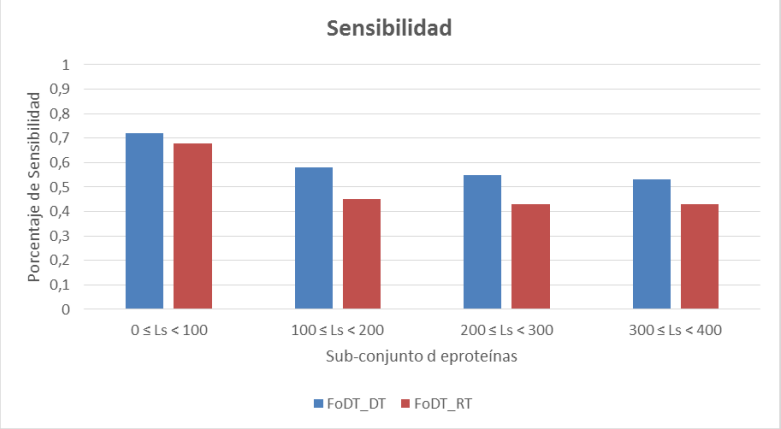


Figura 35. Sensibilidad de las implementaciones FoDT_DT y FoDT_RT, en función de las longitudes de la secuencia delas proteínas.
En el eje de abscisas se representan los resultados de los predictores, en dependencia de la longitud de las secuencias. El eje de ordenadas representa la sensibilidad alcanzada.

Por otra parte el análisis del desempeño del predictor propuesto sobre un predictor aleatorio(R), muestra un comportamiento deseado para ambas implementaciones, debido a que en ambos casos, a medida que aumenta la longitud de las secuencias, FoDT muestra mejor comportamiento que los predictores aleatorios. No obstante, la implementación con árboles de regresión logra un mayor incremento que la

implementación basada en árboles de decisión, alcanzando más de 25 puntos de diferencia en el caso de FoDT_RT (Figura 36).

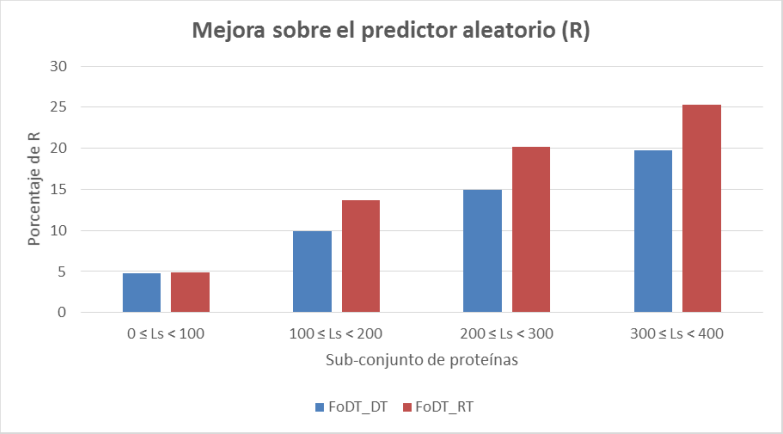


Figura 36. Comportamiento del predictor propuesto sobre un predictor aleatorio (R). Comparación realizada para los algoritmos FoDT_DT y FoDT_RT.

El próximo índice a analizar es Xd, el cual muestra si los predictores son capaces de asignar los contactos con una distribución adecuada. El resultado puede apreciarse en la Figura 37.

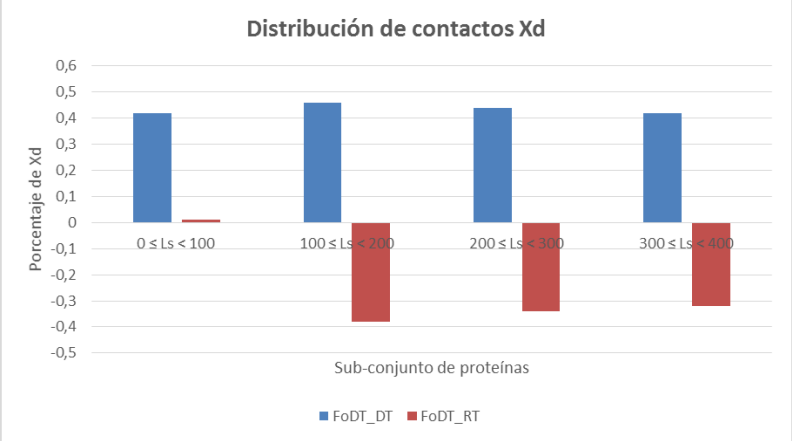


Figura 37. Índice de desempeño Xd, para los algoritmos FoDT_DT y FoDT_RT. Análisis realizado con el objetivo de saber cómo están distribuidos los contactos observados con respecto a los reales.

Resulta interesante analizar el comportamiento de FoDT_RT, donde, para proteínas pequeñas (Ls < 100), muestra una distribución de contactos muy cercana a la aleatoria. Mientras que con el incremento de la longitud de las secuencias, este predictor muestra una correlación negativa en la distribución de los contactos. Teniendo en cuenta que Xd representa la media armónica de las diferencias entre la distribución de las distancias de los contactos predichos y la distribución de las distancias de todos los

contactos, estos valores negativos indican que se predicen mejor las distancias pequeñas que las grandes. O sea, que FoDT_RT asigna mejor los contactos de corto alcance que los de largo alcance, comportamiento que no es el deseado para el predictor. Por otra parte, FoDT_DT sí mantiene una distribución de contactos adecuada (Xd > 0), aún para las proteínas de mayor longitud.

Aparentemente, la implementación FoDT_RT muestra mejor desempeño que FoDT_DT, en cuanto a precisión (Ap). Sin embargo, FoDT_DT parece presentar mayor sensibilidad o capacidad de recuerdo (S) que FoDT_RT. Por lo que resulta más interesante analizar el comportamiento de la medida F-Measure (Figura 38).

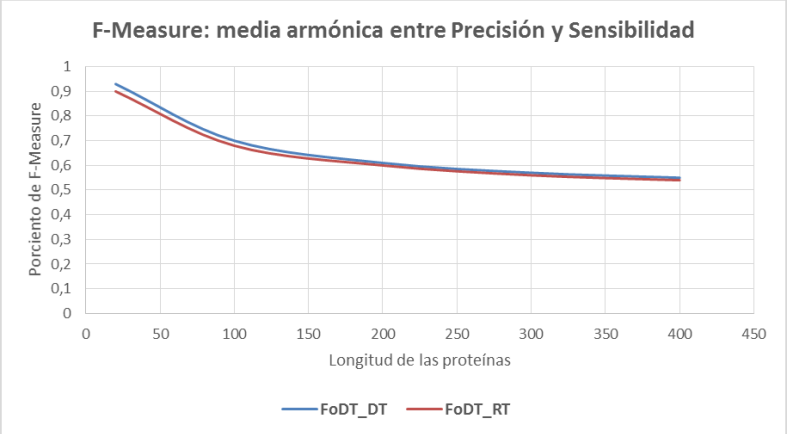


Figura 38. Media armónica entre la precisión y la sensibilidad de los algoritmos FoDT_DT y FoDT_RT.

Como puede apreciarse con claridad, el comportamiento de la media armónica entre la precisión y la sensibilidad (F-Measure) de ambos algoritmos es muy similar. En ambos casos la F-Measure disminuye con el incremento de la longitud de la secuencia de las proteínas. Esta disminución se debe, mayormente, a la aparición de contactos de largo alcance y a la dificultad a la hora de predecirlos.

Basado en el análisis de la sensibilidad y de Xd, aparentemente, la implementación FoDT_DT muestra mejor desempeño que FoDT_RT, sin embargo, FoDT_RT parece presentar mayor precisión que FoDT_DT. Por otra parte, resalta el alto coste computacional de FoDT_RT y que éste asigna contactos con una distribución negativa.

Con la intención de conocer cuál de las implementaciones es la más adecuada, fueron aplicadas pruebas de significación estadística (Anexo 4). Como resultado se demostró

que existen diferencias significativas entre FoDT_DT y FoDT_RT. Teniendo FoDT_DT un comportamiento superior a FoDT_RT. Razón por la cual se decidió emplear árboles de decisión como algoritmo de construcción de los clasificadores base.

6.3.3.- Evaluación de la capacidad de generalización del predictor: validación externa

En este tópico se realiza el análisis y discusión de la validación externa del predictor FoDT (a partir de este momento sólo se empleará la implementación basada en árboles de decisión). Primero se describe el diseño experimental empleado, en el cual incluye la selección de los distintos conjuntos de datos. A continuación, se realiza una serie de experimentos con proteínas heterogéneas, pertenecientes a un virus específico y otras que pertenecen a una misma familia. En todos los casos se analiza el comportamiento del predictor propuesto y se compara con otros del estado del arte.

6.3.3.1.- Diseño experimental

Para realizar esta experimentación se escogieron dos conjuntos de proteínas diferentes. Cada uno de estos conjuntos pretende demostrar cómo es el comportamiento del algoritmo ante diferentes tipos de proteínas. Para este propósito ***FoDT*** es entrenado con el set de validación interna (***Set_VI*** de 2.216 proteínas) y probado con un conjunto de proteínas heterogéneas (5.233) y proteínas 174 del CASP9.

Set de validación externa (Set VE): 5.233 proteínas heterogéneas con identidad máxima del 30%. Con el objetivo de analizar el comportamiento del algoritmo en función de la longitud de la secuencia, este set se divide en los subconjuntos: $0 \leq \boldsymbol{Ls} < 100$ (1.855 proteínas), $100 \leq \boldsymbol{Ls} < 200$ (1.815 proteínas), $200 \leq \boldsymbol{Ls} < 300$ (724 proteínas) y $300 \leq \boldsymbol{Ls} < 400$ (405 proteínas) y $\boldsymbol{Ls} < 400$ (434 proteínas).

Set del CASP9: 174 estructuras de proteínas obtenidas experimentalmente y no incluidas en el PDB (Anexo 5). Este subconjunto fue extraído de la 9^{na} competición de predicción de estructuras CASP9 (*Critical assessment of techniques for proteins structure prediction*) [21]–[23]. El objetivo de este conjunto es comprobar la capacidad de generalización del predictor.

Set del CASP10: 123 estructuras de proteínas obtenidas experimentalmente y no incluidas en el PDB (Anexo 6). Este subconjunto fue extraído de la 10^{ma} competición de

predicción de estructuras CASP10. El objetivo de este conjunto es comparar la capacidad de generalización del predictor con los últimos predictores del estado del arte.

6.3.3.2.- Capacidad de generalización

Para realizar el análisis de la capacidad de generalización de ***FoDT*** se empleó el set de validación externa (5.233 proteínas de identidad máxima del 30%).Con la intención de resaltar la dependencia de los resultados con las dimensiones de las proteínas, los valores de efectividad fueron calculados luego de agrupar las proteínas según la longitud de sus secuencias (Tabla 8).

Tabla 8. Resultados experimentales de la capacidad de generalización de FoDT.
Donde se utilizó como base de datos un conjunto de 5.233 proteínas con identidad de hasta el 30%, el cual se dividió en 4 particiones. Como resultado se obtuvieron las variables (Ap) que es la efectividad de la predicción, (R) que es el mejoramiento del desempeño sobre un predictor aleatorio, (Xd) que mide la distribución de la predicción respecto al valor real, (S) que es la sensibilidad del predictor y (F) que es F-Measure, la media armónica entre la precisión y la sensibilidad.

Algoritmos	**0 ≤ Ls < 100**	**100 ≤ Ls < 200**	**200 ≤ Ls < 300**	**300 ≤ Ls < 400**
Ap	0,69	0,57	0,42	0,32
R	4,83	8,62	10,4	10,89
Xd	0,42	0,44	0,40	0,38
S	0,72	0,59	0,54	0,50
F	0,70	0,58	0,47	0,39

La Tabla 8 muestra un buen comportamiento general del algoritmo propuesto, manteniendo una efectividad superior al 42% para proteínas de hasta 300 aminoácidos de longitud y de 32% para proteínas mayores. Como es de esperar, tanto la efectividad como la sensibilidad del algoritmo disminuyen con el aumento de la longitud de la secuencia (Figura 39). Pero este decrecimiento no es drástico y se justifica por la aparición de interacciones de largo alcance, las cuales son más difíciles de predecir.

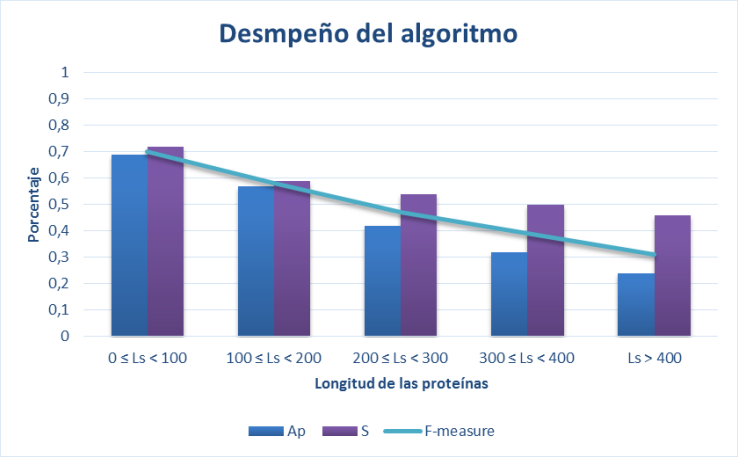


Figura 39. Desempeño del algoritmo ante un conjunto de proteínas heterogéneas. En el gráfico se muestran la efectividad del algoritmo (Ap), la sensibilidad (S) y la media armónica entre ambas medidas o F-Measure (F).

Como se puede observar en la Figura 39, ***FoDT*** la media armónica entre la efectividad y la sensibilidad del algoritmo describe un descenso suave con el incremento de la longitud de la secuencia. Manteniéndose en todo momento sobre el 35% y evidenciando la capacidad de FoDT para predecir correctamente proteínas de hasta 300 aminoácidos de longitud.

En la Figura 40 se muestra cómo se comporta esta asignación de contactos. Es evidente que mientras mayor es la proteína el predictor mejora su capacidad de predicción sobre el predictor aleatorio (R). Característica bastante deseable para cualquier predictor. La distribución de los contactos se mantiene sobre el 36% para las proteínas de cualquier longitud. Aunque, al igual que sucede con la efectividad y la sensibilidad, los contactos son mejor distribuidos para proteínas de longitudes menores que 200 aminoácidos.

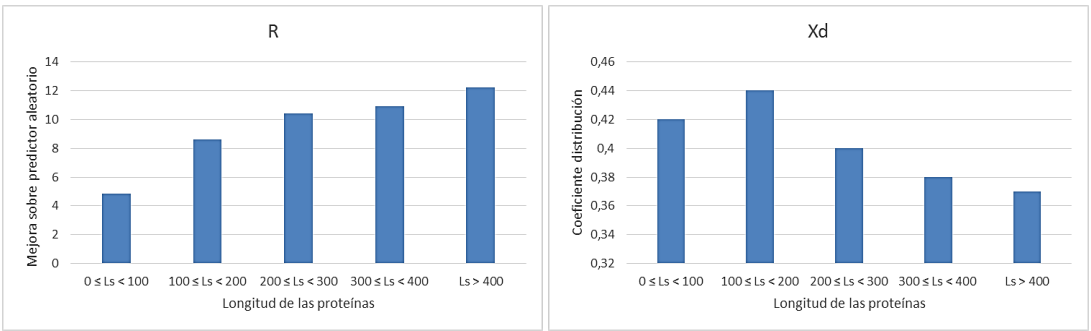


Figura 40. Gráfico de análisis de la capacidad de asignación de contactos de FoDT. (R) Mejora sobre un predictor aleatorio. (Xd) Distribución de los contactos.

6.3.3.3.- Comparación con algoritmos del estado del arte

Para realizar el análisis comparativo del comportamiento de ***FoDT*** con algoritmos del estado del arte, fueron empleados los conjuntos de datos del CASP9 y el CASP10. En ambos casos la experimentación se realizó empleando las reglas del CASP.

Resultados con el set del CASP9

Para la comparación fueron empleados los servidores de predicción más importantes que se encuentran públicamente: SVM-SEQ[216], NNcon [107], FragHMMent [217] y LRcon [22]. Un importante predictor público que fue excluido es SVMcon [33], [218], debido a que emplea el mismo algoritmo que SVM-SEQ[22]. El resultado de la comparación se muestra en la Tabla 9.

Tabla 9. Comparativa de FoDT con algoritmos del estado del arte, con el set del CASP9.

Algoritmos	Mejores L/10 predicciones			Mejores L/5 predicciones		
	Acc	Cov	Fm	Acc	Cov	Fm
FragHMMent	0,37	0,12	0,18	0,34	0,22	0,27
NNcon	0,59	0,19	0,28	0,48	0,31	0,37
SVM-SEQ	0,61	0,19	0,29	0,51	0,33	0,40
LRcon	**0,65**	0,21	0,31	**0,54**	0,35	0,42
FoDT	0,55	**0,61**	**0,57**	0,41	**0,46**	**0,43**

Los algoritmos de SVM-SEQ, NNcon y LRcon presentan un mejor comportamiento que el algoritmo propuesto FoDT, en cuanto a la efectividad (Acc). Mientras que FoDT los supera en cuanto a cobertura (Cov), con valores siempre superior al 46%, y a la media armónica con valores superiores al 43%. Por lo que se realizó una prueba de significación estadística (Anexo 7), cuyos resultados se muestran en la Tabla 10.

Tabla 10. Resultados de las pruebas de significación estadística de la comparación entre algoritmos.

Algoritmos	FragHMMent	NNcon	SVM-SEQ	LRcon	FoDT
FragHMMent		=	=	**+**	**+**
NNcon	=		=	=	=
SVM-SEQ	=	=		=	=
LRcon	**–**	=	=		=
FoDT	**–**	=	=	=	

En la Tabla 10, se muestra claramente que FoDT supera con significación estadística, el comportamiento de FragHMMent e iguala con el resto de los algoritmos. La tabla de rangos de Friedman (Anexo 7), ubica a FoDT en una segunda posición con un índice de 2.0, sólo superado por LRcon, con un índice de 1.6.

Sin embargo, LRcon está basado en regresión logística, NNcon está basado en redes neuronales, SVM-SEQ está basado en máquinas de vectores soporte y FragHMMent en modelos ocultos de Markov. Lo cual significa que, a diferencia de FoDT, los algoritmos con éste que compite no poseen una base de conocimiento fácilmente interpretable, cualidad esta que es sumamente deseable.

Resultados con el set del CASP10

Para realizar esta comparación, los resultados de los métodos de predicción fueron extraídos de: http:/predictioncenter.org/casp10/rr_results.cgi. Donde, son incluidos los 25 mejores algoritmos basados en secuencias según el reporte de evaluación del CASP10 [219].

Los algoritmos están ordenados, empleando la recomendación de los evaluadores del CASP, empleando los resultados para los L/5 contactos de largo alcance, como medida de la robustez de los resultados (Anexo 8). Con la intención de proveer un punto de vista sobre la calidad de los predictores, este ordenamiento se basa en la media entre la efectividad (Acc) y la dispersión de las predicciones (Z-Score).

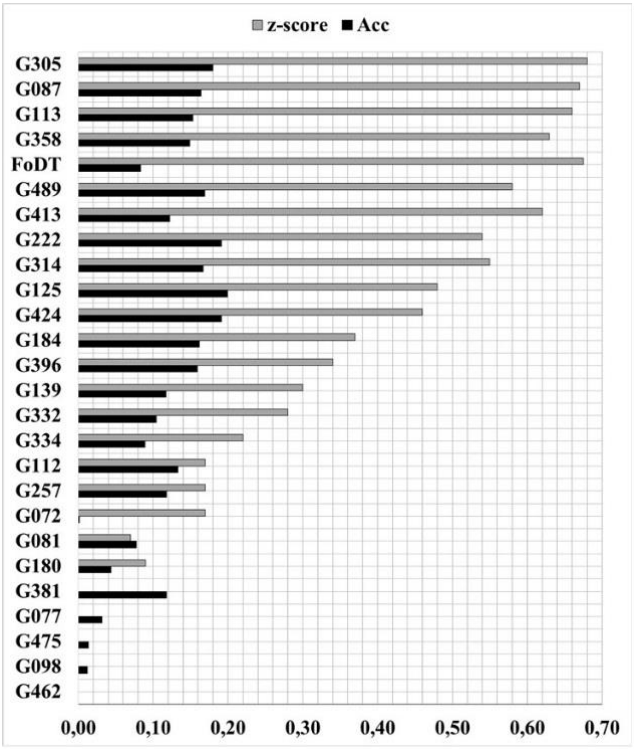


Figura 41. Ordenamiento de los predictores participantes en el CASP10 y FoDT, basado en la media entre el Acc y el Z-Score.

Como puede observarse en la Figura 44, el método propuesto (FoDT) no presenta los mejores valores de efectividad. Sin embargo, muestra el tercer mejor valor de Z-Score. Lo cual significa que FoDT puede asignar contactos con una alta fiabilidad. Lo cual lo ubica entre los mejores algoritmos del ordenamiento.

6.3.- Análisis del dominio de aplicación

La definición del domino de aplicación (*Applicability Domain*) de un algoritmo es un problema de suma importancia. Esto se debe a que no siempre un algoritmo robusto y validado puede predecir adecuadamente todo el universo. Especialmente en el problema de predicción de proteínas, donde existe una gran variabilidad en los datos, la información con que se cuenta no siempre es fiable y, por naturaleza, determinadas proteínas pueden cambiar su estado. Razón por la cual, sólo las predicciones realizadas para el domino de aplicación donde está validado el algoritmo pueden considerarse fiables.

El dominio de aplicación puede ser definido teóricamente como una región o área, donde el modelo propuesto responde adecuadamente. Y está caracterizado por la naturaleza del conjunto de datos de entrenamiento y sus descriptores específicos. Por lo que no debe confundirse el dominio de aplicación del algoritmo con los dominios conformacionales, evolutivos o funcionales de las proteínas [220].

Para este análisis se extrajo un conjunto de 49 proteínas pertenecientes a diferentes dominios estructurales: ***alpha***, ***alpha/beta***, ***alpha+beta*** y ***beta*** (Anexo 2). Este conjunto se seleccionó siguiendo el criterio de Abu-Doleh y colaboradores en [23].

Los resultados experimentales se muestran en la Tabla 11, donde, el funcionamiento del algoritmo fue analizado para cada uno de los dominios estructurales por separado. Tomando en cuenta la efectividad (Ap), la sensibilidad (S), la media armónica (Fm), así como su comportamiento comparado con un predictor aleatorio.

Tabla 11. Comportamiento del algoritmo ante diferentes dominios estructurales.

Experimento	Alpha	Alpha/Beta	Alpha+Beta	Beta
Ap	0,64	0,62	0,52	0,48
S	0,67	0,62	0,54	0,48
Fm	0,65	0,62	0,52	0,47
R	14,7	13,83	10,13	8,04
Xd	0,43	0,38	0,42	0,46

Como se puede observar, el comportamiento del algoritmo para cada uno de los dominios estructurales es similar. Alcanzando una efectividad superior al 60% para las proteínas compuestas mayoritariamente por hélices α, dominios Alpha y Alpha/Beta. Pero manteniendo, en sentido general, la media armónica superior al 47% (Figura 42).

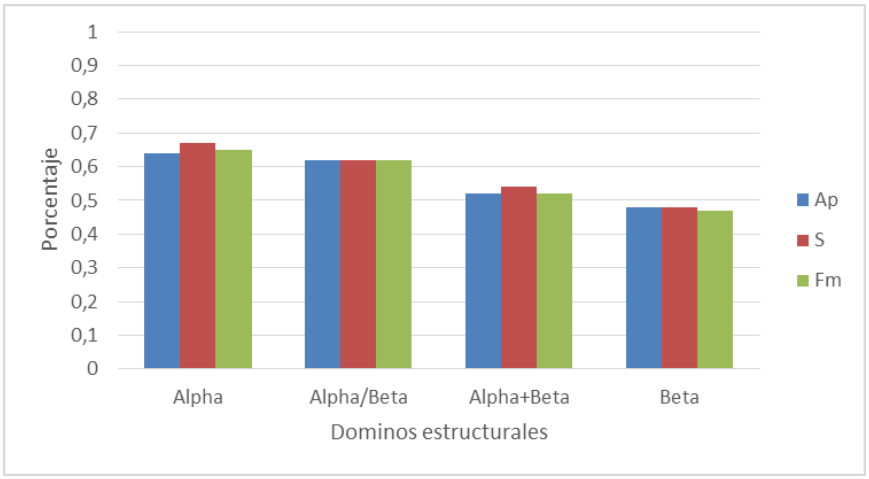


Figura 42. Análisis de la efectividad del algoritmo ante dominios estructurales. El análisis incluyó la efectividad del algoritmo (Ap), la sensibilidad (S) y la media armónica entre ambas medidas o F-Measure (Fm).

FoDT logra asignar contactos con una adecuada distribución (superior al 38%), en todos los casos. Sin embargo, de igual manera que con la efectividad y la sensibilidad, FoDT logra una mejora sobre un predictor aleatorio alrededor de los 14 puntos porcentuales para las proteínas con dominios estructurales Alpha y Alpha/Beta (Figura 43).

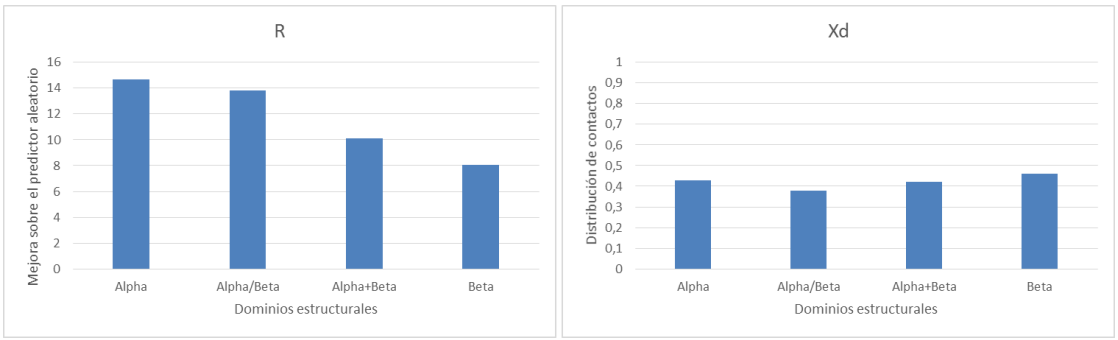


Figura 43. Análisis de la capacidad de asignación de contactos del algoritmo ante dominios estructurales.

Con el objetivo de determinar si existen diferencias estadísticamente significativas en las clasificaciones para los diferentes dominios estructurales, fueron empleados múltiples pruebas estadísticas no paramétricas. Mediante la prueba de Friedman se

comprueba la hipótesis nula de que el algoritmo obtiene resultados similares, como promedio, para todos los dominios. Si la hipótesis nula es rechazada, entonces son aplicadas las pruebas de *post-hoc*: Bonferroni-Dunn, Holm, Hochberg y Hommel (Figura 44).

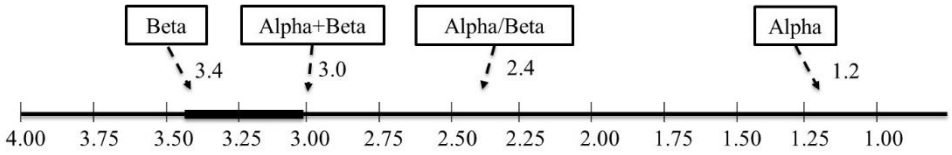


Figura 44. Representación visual de ordenamiento promedio de Friedman con un $\alpha = 0.10$, para Alpha, Beta, Alpha/Beta y Alpha+Beta.
La línea oscura conecta los dominios para los cuales no existen diferencias estadísticamente significativas.

A partir de este análisis, se puede concluir que el algoritmo FoDT puede ser empleado en la predicción de mapas de contactos para proteínas que posean cualquier tipo de dominio estructural. Resultando siempre más efectivos en proteínas formadas mayormente por α-hélices.

6.4.- Mecanismo de interpretación

FoDT, al estar basado en árboles de decisión, tiene la habilidad de poseer un modelo de conocimiento interpretable. Lo cual le proporciona una apreciable ventaja sobre el resto de los algoritmos basados en redes neuronales, modelos probabilísticos, máquinas de vectores soporte, entre otros. Debido a que las bases de conocimientos de las redes neuronales son matrices de pesos o en las máquinas de soporte de vectores son matrices de vectores, las cuales no pueden ser explicadas con facilidad por el biólogo (usuario final).

La concepción misma de un árbol implica un conjunto tangible de reglas. Lo cual es de gran importancia para los especialistas biólogos. Debido a que FoDT está sugiriendo un conjunto de reglas extraídas de regularidades observadas frecuentemente en el aprendizaje de la relación que pueda existir entre la secuencia de aminoácidos y su estructura.

FoDT propone un árbol de decisión por cada una de las 400 parejas de aminoácidos. Donde, cada hoja codifica directamente para contacto (C) o no contacto (NC) con un nivel de confianza que está en dependencia de la cantidad de veces que se observa el

patrón. Cada árbol se construye a partir de diferentes sets de datos, lo cual condiciona la diversidad entre ellos. Esto se puede apreciar en la Figura 45, la cual muestra los árboles de decisiones construidos para los pares de aminoácidos ***K-E*** y ***F-W*** (comunes en las proteínas).

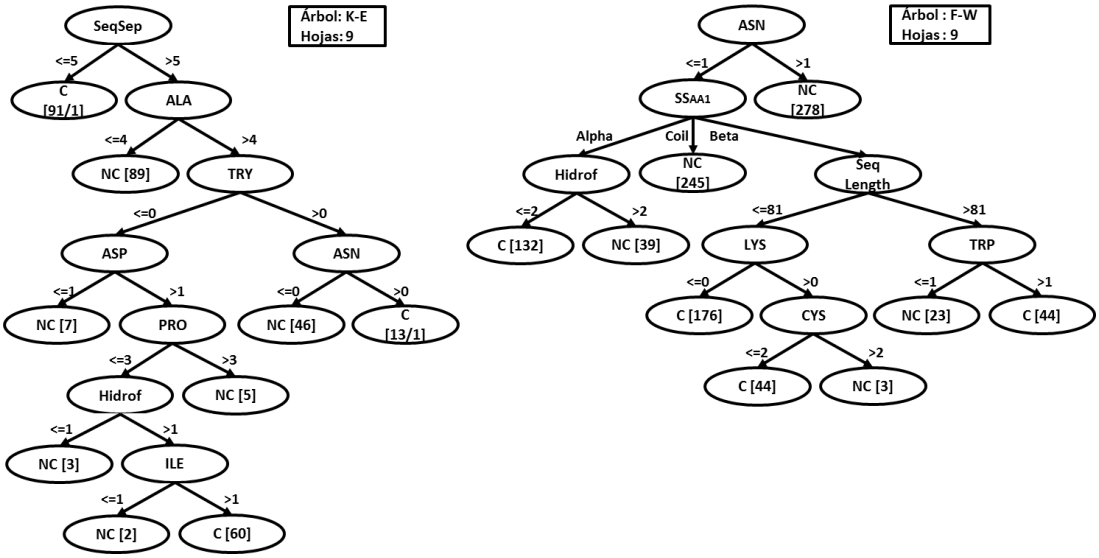


Figura 45. Árbol de decisión construido para los pares de aminoácidos K-E y F-W.

En la Figura 45 puede apreciarse, que el árbol ***K-E*** comienza a cortar tomando en cuenta el atributo "Separación en la Secuencia". Luego toma en cuenta, mayormente, la frecuencia de aparición de los amino ácidos y la propiedad de hidrofobicidad. Mientras que el árbolo ***F-W*** comienza a cortar tomando en cuenta la frecuencia de aparición de un amino ácido y, posteriormente, toma en cuenta otros atributos como la "Longitud de la Secuencia". Este grado de especialización de cada uno de los 400 árboles en el clasificador, contribuye a que la predicción de FoDT supere a otros algoritmos basados en un clasificador simple.

La evaluación de la subcadena, formada entre un par de amino ácidos de una secuencia, en estos árboles devolverá una clase correspondiente a la predicción de la existencia de contacto o no entre los amino ácidos objetivo. La que puede ser representada como un conjunto de reglas con el formato ***si–entonces***, facilitando su comprensión por parte de las personas. Por ejemplo, del árbol ***K-E*** pueden extraerse 9 reglas: 3 para contactos, con una alta confianza (97/1, 60/0 y 13/1 respectivamente) y el resto para no-contactos. Las reglas generadas por el árbol serían las siguientes:

- ***si*** *SeqSep > 5* ***y*** $f_{(ALA)} > 4$ ***y*** $f_{(TRY)} > 0$ ***y*** $f_{(ASN)} > 3$ → ***predecir*** *contacto* (1: 60/0)
- ***si*** *SeqSep ≤ 5* → ***predecir*** *contacto* (0.99: 97/1)
- ***si*** *SeqSep > 5* ***y*** $f_{(ALA)} > 4$ ***y*** $f_{(TRY)} = 0$ ***y*** $f_{(ASP)} > 1$ ***y*** $f_{(PRO)} < 3$ ***y*** $f_{(H?)} > 1$ ***y*** $f_{(ILE)} < 1$ → ***predecir*** *contacto* (0.93: 13/1)
- ***en otros casos*** → ***predecir*** *no-contacto*

donde, ***SeqSep*** es la separación en la secuencia y ***f(aa)*** es la frecuencia de aparición del amino ácido ***aa*** en la subsecuencia. El valor real que aparece a continuación representa el nivel de confianza (entre 0 y 1). Los valores restantes representan la cobertura, o sea, cuántas veces se cumplió la regla del total de veces que se dio la combinación especificada.

FoDT suele generar árboles pequeños que pueden comprender entre 10 hasta 130 reglas. El tamaño de los árboles depende de cuán comunes son las parejas que se forman en las proteínas de entrenamiento. Por lo general, las parejas más comunes generan árboles más grandes, mientras que las parejas poco comunes generan árboles pequeños. Como promedio, FoDT genera reglas con más de un 85% de confianza.

La capacidad de FoDT de generar una pequeña colección de reglas con una alta confianza representa una gran ventaja para su mecanismo de interpretación. FoDT brinda el conjunto de reglas ordenadas en cuanto al nivel de confianza y cobertura. Esto facilita el trabajo de los investigadores, responsables de "descubrir" indicios de cómo se realiza el plegamiento de las proteínas, dando explicación biológica de alguno de dichos supuestos (reglas).

6.5.- Conclusiones parciales

El algoritmo propuesto en esta investigación, fue evaluado según los principios establecido por la OECD. El análisis del dominio de aplicación permitió demostrar que el algoritmo se comporta mejor para proteínas formadas por α-hélices. La validación interna mostró la robustez de FoDT y su bondad de ajuste. Mientras que en el proceso de validación externa, se pudo constatar la capacidad de generalización, demostrando la capacidad de FoDT para asignar contactos con una efectividad similar a los algoritmos del estado del arte, con un coste computacional muy bajo. No obstante, la principal ventaja de FoDT radica en su mecanismo de interpretación de la base de conocimiento.

Parte V

Conclusiones

Capítulo 7

Conclusiones y trabajos futuros

7.1.- Conclusiones

En esta investigación se propuso el algoritmo FoDT, el cuál es capaz de predecir mapas de contactos de proteínas con una efectividad similar o superior a los algoritmos del estado del arte a partir de la información brindada por la secuencia de aminoácidos. Éste, es un multiclasificador de combinación por selección, con optimización de cobertura, basado en árboles de decisión C4.5. La diversidad se logra con la especialización de cada clasificador base en la predicción de contactos para la pareja de aminoácidos que representa. Para lograr este efecto, se adoptó una arquitectura horizontal o paralela, donde sólo se activa un clasificador por vez en dependencia del par de aminoácidos que se esté prediciendo. Este esquema hace que FoDT reduzca significativamente el elevado coste computacional que caracteriza a los multiclasificadores de arquitectura paralela.

El análisis del dominio de aplicación permitió demostrar que FoDT se comporta mejor para proteínas formadas por hélices α. La validación interna mostró la robustez del algoritmo y su bondad de ajuste. Mientras que en el proceso de validación externa, se pudo constatar la capacidad de generalización del mismo ante 5.233 proteínas heterogéneas. Al comparar FoDT con algoritmos del estado del arte, éste fue capaz de asignar contactos con una efectividad del 55%, similar a los algoritmos de los servidores públicos más eficaces de la actualidad, con un coste computacional muy bajo. Sin embargo su principal ventaja radica en la capacidad de brindar un mecanismo de interpretación de la base de conocimiento.

Se propuso, además, un nuevo esquema de codificación que incluye la información de la subsecuencia implícita entre los aminoácidos no adyacentes, las propiedades físico-

químicas, la distancia entre los aminoácidos y la separación a que se encuentran en la secuencia. El principal aporte de esta codificación es el análisis por separado cada una de las parejas de aminoácidos que pueden formarse, lo cual representa la descomposición del problema en 400 subproblemas (20 x 20 aminoácidos). Esta idea no simplemente cambia de la visión holística del problema a una visión reduccionista, sino que propone una simplificación al proceso de entrenamiento de los algoritmos de aprendizaje automático.

Debido al alto nivel de desbalance que existe en la predicción de mapas de contactos, se diseñó un nuevo algoritmo de remuestreo al que se denominó: Estrategia Genética Simulada (EGS). El cual realiza un sobremuestreo basado en la probabilidad de aparición de los aminoácidos en las secuencias y emplea una estrategia genética para la generación de nuevas instancias. El operador de mutación de EGS emplea la matriz de sustitución BLOSUM64 para reemplazar los aminoácidos seleccionados. EGS garantiza que los nuevos individuos mantengan un parecido con sus padres de al menos un 50%.

7.2.- Trabajos futuros

Al concluir esta investigación, se abren nuevas líneas de investigación que podrían darle continuidad:

- Teniendo en cuenta que generalmente las técnicas de predicción *ab initio* no son más efectivas que las basadas en modelos, se impone combinar el análisis de cada una de las parejas de aminoácidos que pueden formarse con la información evolutiva de las proteínas.
- Con el objetivo de lograr incrementar la capacidad de predicción de los algoritmos de predicción de mapas de contactos, se hace necesario investigar en métodos de postprocesamiento más efectivos, que tomen en cuenta la ocurrencia de las interacciones de largo alcance.

Parte VI

Referencias Bibliográficas

Referencias Bibliográficas

[1] C. A. Ouzounis and A. Valencia, “Early bioinformatics : the birth of a discipline — a personal view,” *Bioinformatics*, vol. 19, no. 17, pp. 2176–2190, 2003.

[2] J. Cohen, “Bioinformatics — An Introduction for Computer Scientists,” *ACM Comput. Surv.*, vol. 36, no. 2, pp. 122–158, 2004.

[3] V. Hernán and A. Quiceno, “Bio-informática un Campo por conocer,” vol. VII, no. 11, pp. 1–9, 2006.

[4] J. Hu, X. Shen, Y. Shao, C. Bystroff, and M. J. Zaki, “Mining Protein Contact Maps,” *BIOKDD02 Work. Data Min. Bioinforma.*, pp. 3–10, 2002.

[5] B. A. Wallace, M. Cascio, and D. L. Mielke, “Evaluation of methods for the prediction of membrane protein secondary structures,” *Proc. Natl. Acad. Sci.*, vol. 83, no. Biochemistry, pp. 9423–9427, 1986.

[6] WwPDB, “Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description Version 3.20,” *wwPDB*, 2008. [Online]. Available: http://mmcif.pdb.org/dictionaries/mmcif_pdbx.dic/Index/index.html.

[7] M. I. L. Evitt, “A structural census of the current population,” *Proc. Natl. Acad. Sci.*, vol. 94, no. October, pp. 11911–11916, 1997.

[8] Y. Pan, “Protein Structure Prediction and Understanding Using Machine Learning Methods *,” *IEEE*, no. 2005, pp. 13–13, 2005.

[9] M. N. Nguyen, J. M. Zurada, and J. C. Rajapakse, “Towards Better Understanding of Protein Secondary Structure : Extracting Prediction,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, pp. 1–8, 2010.

[10] R. Yamagishi, H. Yagi, M. Furuichi, T. Murase, H. Ishii, H. Mizuno, J. Shimada, H. Minagawa, S. Ohnishi, and H. Kaneko, “Validation of Techniques for Structure Prediction and Thermostabilization of a Protein: A Case Study Using the TIM-barrel Enzyme Lactate Oxidase,” *Chem-Bio Informatics J.*, vol. 9, pp. 62–74, 2009.

[11] C. Hardin, T. V Pogorelov, and Z. Luthey-schulten, “Ab initio protein structure prediction,” *Courrent Opin. Struct. Biol.*, vol. 12, pp. 176–181, 2002.

[12] R. Mao and K. Wu, “A Protein Structure Prediction and Function Identification,” *2010 Int. Conf. Bioinforma. Biomed. Technol. IEEE*, pp. 216–220, 2010.

[13] Z. Zhang, “An Overview of Protein Structure Prediction: From Homology to Ab Initio,” *Bioc218*, pp. 1–10, 2002.

[14] D. Petrey and B. Honig, “Protein structure prediction: inroads to biology.,” *Mol. Cell*, vol. 20, no. 6, pp. 811–819, Dec. 2005.

[15] N. Hamilton and T. Huber, “An introduction to protein contact prediction.,” *Methods Mol. Biol.*, vol. 453, pp. 87–104, Jan. 2008.

[16] D. Baker and A. Sali, “Protein Structure Prediction and Structural Genomics,” *October*, vol. 93, no. 2001, 2008.

[17] R. O. Day, G. B. Lamont, and W. D. Oh, “Protein Structure Prediction by Applying an Evolutionary Algorithm,” *Proc. Int. Parallel Distrib. Process. Symp.*, pp. 2–8, 2003.

[18] B. B. Robson, “Analysis of the Code Relating Sequence to Conformation in Globular Proteins,” *Biochem. J.*, vol. 141, pp. 853–867, 1974.

[19] A. Ramanathan, P. K. Agarwal, and C. J. Langmead, “Using Tensor Analysis to characterize Contact-map Dynamics of Proteins,” 2008.

[20] P. Gramatica, “Principles of QSAR models validation: internal and external,” *QSART Comb. Sci.*, vol. 26, no. 5, pp. 694 – 701, 2007.

[21] “Protein structure prediction (cont) Different approaches Threading – method for structure prediction Target sequence is compared to structures using sequence - structure alignment,” pp. 1–28, 2012.

[22] J. Yang and X. Chen, “A Consensus Approach to Predicting Protein Contact Map via Logistic Regression,” *ISBRA*, vol. 6674, pp. 136–147, 2011.

[23] A. A. Abu-Doleh, O. M. Al-Jarrah, and A. Alkhateeb, “Protein contact map prediction using multi-stage hybrid intelligence inference systems.,” *J. Biomed. Inform.*, vol. 45, no. 1, pp. 173–83, Feb. 2012.

[24] N. V Mahajan, L. G. Malik, and A. Background, “A Need for Development of SDK for Reading PDB File,” pp. 116–120, 2012.

[25] J. Minning, “Correlations in thermodynamics and evolution of proteins,” *Darmstadt*, pp. 1–117, 2012.

[26] C. Levinthal, “Are there pathways for protein folding?,” *J. Chem. Phys.*, vol. 65, pp. 44–45, 1968.

[27] B. Nolting, “The folding pathway of a protein at high resolution from microseconds to seconds,” *Proc. Natl. Acad. Sci. 826–830*, vol. 94, no. 3, pp. 826–830, 1997.

[28] S. Wu and Y. Zhang, “Chapter 11. Protein Structure Prediction,” *Bioinforma. Tools Appl.*, pp. 225–242, 2009.

[29] O. Olmea, B. Rost, and A. Valencia, “Effective Use of Sequence Correlation and Conservation in Fold Recognition,” *J. Mol. Biol.*, vol. 295, pp. 1221–1239, 1999.

[30] B. Rost and C. Sander, “Prediction od Protein Secondary Structure at Better than 70% Accuracy,” *J. Mol. Biol.*, vol. 232, pp. 584–599, 1993.

[31] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. M. Strauss, and D. Baker, “Rosetta in CASP4: Progress in Ab Initio Protein Structure Prediction,” *PROTEINS Struct. Funct. Genet.*, vol. 5, no. January, pp. 119 –126, 2002.

[32] R. Bonneau, I. Ruczinski, J. Tsai, and D. Baker, “Contact order and ab initio protein structure prediction,” *Protein Sci.*, vol. 11, pp. 1937–1944, 2002.

[33] I. Walsh, D. Bau, A. J. M. Martin, C. Mooney, A. Vullo, and G. Pollastri, “Ab initio and template-based prediction of multi-class distance maps by two-dimensional recursive neural networks,” *BMC Struct. Biol.*, vol. 9, no. 5, pp. 1–38, 2009.

[34] J. C. Barbosa and L. E. Dardenne, “Full-Atom Ab Initio Protein Structure Prediction with a Genetic Algorithm using a Similarity-based Surrogate Model,” *IEEE Congr. Evol. Comput.*, pp. 1–8, 2010.

[35] T. W. De Lima, P. H. R. Gabriel, A. C. B. Delbem, R. A. Faccioli, and I. N. Silva, “Evolutionary Algorithm to ab initio Protein Structure Prediction with Hydrophobic Interactions,” *interactions*, pp. 612–619, 2007.

[36] J. Garnier, D. J. Osguthorpe, and B. Robson, “Analysis of the Accuracy and Implications of Simple Methods for Predicting the Secondary Structure of Globular Proteins,” *J. Mol. Biol.*, vol. 120, pp. 97–120, 1978.

[37] M. J. Zvelebil, “Prediction of Protein Secondary Structure and Active Sites using the Alignment of Homologous Sequences,” *J. Mol. Biol.*, vol. 195, pp. 957–961, 1987.

[38] L. Jaroszewski, L. Rychlewski, A. Godzik, L. Jaroszewski, L. Rychlewski, and A. Godzik, “Improving the quality of twilight-zone alignments,” *Protein Sci.*, vol. 9, pp. 1487–1496, 2000.

[39] T. Z. Sen, R. L. Jernigan, J. Garnier, and A. Kloczkowski, “GOR V server for protein secondary structure prediction,” *Bioinforma. Appl. NOTE*, vol. 21, no. 11, pp. 2787–2788, 2005.

[40] J. Peng and J. Xu, "Low-homology protein threading," *Bioinformatics*, vol. 26, pp. i294–i300, 2010.

[41] H. Kim, J. Park, and K. Han, "Predicting Protein Interactions in Human by Homologous Interactions in Yeast §," *Bioinformatics*, pp. 159–165, 2003.

[42] H. Bohr, J. Bohr, S. Brunak, R. M. J., Cotteril, and B. Lautrup, "Protein secondary structure and homology by neural networks," *Biomed. Div.*, vol. 241, no. 1,2, pp. 223–228, 1988.

[43] H. Ashkenazy, R. Unger, Y. Kliger, T. Aviv, T. Mina, E. Goodman, and L. Sciences, "Hidden conformations in protein structures.," *Bioinformatics*, vol. 27, no. 14, pp. 1941–7, Jul. 2011.

[44] T. Huber, A. J. Russell, D. Ayers, and A. E. Torda, "Sausage: protein threading with flexible force fields," *Bioinforma. Appl. NOTE*, vol. 15, no. 12, pp. 1064–1065, 1999.

[45] A. Caprara, R. Carr, S. Istrail, and G. Lancia, "1001 optimal PDB structure alignments: Integer Programming methods for finding the maximum contact map overlap," 2003.

[46] M. Vassura, P. Di Lena, L. Margara, M. Mirto, G. Aloisio, P. Fariselli, and R. Casadio, "Blurring contact maps of thousands of proteins : what we can learn by reconstructing 3D structure," *BioData Min.*, vol. 4, no. 1, p. 1, 2011.

[47] P. Di Lena, K. Nagata, and P. Baldi, "Deep architectures for protein contact map prediction," *Bioinformatics*, vol. 28, no. 19, pp. 2449–2457, 2012.

[48] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio, "Prediction of contact maps with neural networks and correlated mutations," *Protein Eng.*, vol. 14, no. 11, pp. 835–843, 2001.

[49] X. Deng and J. Cheng, "MSACompro : protein multiple sequence alignment using predicted secondary structure , solvent accessibility , and residue-residue contacts," *BMC Bioinformatics*, vol. 12, no. 1, p. 472, 2011.

[50] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, H. Fredholm, B. Lautrupt, and S. B. Petersen, "A novel approach to prediction of the 3-dimensional structures of protein backbones by neural networks," *FEBS Lett.*, vol. 261, no. 1, pp. 43–46, 1990.

[51] J. Cheng, A. N. Tegge, and P. Baldi, "Machine Learning Methods for Protein Structure Prediction," *IEEE Rev. Biomed. Eng.*, vol. 1, pp. 41–49, 2008.

[52] and Y. Z. Bin Xue, Eshel Faraggi, "Predicting residue-residue contact maps by a two-layer, integrated neural-network method," *Proteins*, vol. 76, no. 1, pp. 176–183, 2009.

[53] G. Pollastri and P. Baldi, “Prediction of Contact Maps by Recurrent Neural Propagation From All Four Cardinal Corners,” *Bioinformatics*, vol. 1, no. 1, pp. 1–9, 2002.

[54] H. Kim, “Kim, H., et al., Computational analysis of hydrogen bonds in protein-RNA complexes for interaction patterns. FEBS Lett., 2003. 552: p. 231–239,” *FEBS Lett.*, vol. 552, p. Kim, H., et al., Computational analysis of hydroge, 2003.

[55] G. A. Cortés and J. S. Aguilar-Ruiz, “Predicting protein distance maps according to physicochemical properties.,” *J. Integr. Bioinform.*, vol. 8, no. 3, p. 181, Jan. 2011.

[56] Y. Arkun and B. Erman, “Prediction of Optimal Folding Routes of Proteins That Satisfy the Principle of Lowest Entropy Loss : Dynamic Contact Maps and Optimal Control,” *PLoS One*, vol. 5, no. 10, pp. 1–11, 2010.

[57] J. M. Duarte, R. Sathyapriya, H. Stehr, I. Filippis, and M. Lappe, “Optimal contact definition for reconstruction of Contact Maps,” *BMC Bioinformatics*, vol. 11, no. 283, 2010.

[58] D. Kozma, I. Simon, and G. E. Tusnády, “CMWeb: an interactive on-line tool for analysing residue – residue contacts and contact prediction methods,” *Nucleic Acids Res.*, vol. 40, no. Web Server issue, pp. W329–W333, 2012.

[59] P. Fariselli and R. Casadio, “A neural network based predictor of residue contacts in proteins,” *Protein Eng.*, vol. 12, no. 1, pp. 15–21, 1999.

[60] G. Zhang, D. S. Huang, and Z. H. Quan, “Combining a binary input encoding scheme with RBFNN for globulin protein inter-residue contact map prediction,” *Pattern Recognit. Lett.*, vol. 26, pp. 1543–1553, 2005.

[61] G. Zhang and K. Han, “Hepatitis C virus contact map prediction based on binary encoding strategy,” *Comput. Boiology Chem.*, vol. 31, pp. 233–238, 2007.

[62] A. Vullo, I. Walsh, and G. Pollastri, “A two-stage approach for improved prediction of residue contact maps,” *BMC Bioinformatics*, vol. 7, no. 180, pp. 1–12, 2006.

[63] J. R. González and D. A. Pelta, “On Using Fuzzy Contact Maps for Protein Structure Comparison,” *IEEE*, vol. 1, pp. 1650–1655, 2007.

[64] G.-Z. Zhang and D.-S. Huang, “Prediction of inter-residue contacts map based on genetic algorithm optimized radial basis function neural network and binary input encoding scheme,” *J. Comput. Aided. Mol. Des.*, vol. 18, pp. 797–810, 2004.

[65] L. I. Kuncheva, “Multiple Classifier Systems,” *Comb. Pattern Classif. Methods Algorithms*, pp. 101–110, 2004.

[66] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensembles methods," *Neural Comput.*, vol. 6, pp. 1289–1301, 1994.

[67] W. Yan and K. Goebel, "Designing Classifier Ensembles with Constrained Performance Requirements," in *Proceedings of SPIE Defense & Security Symposium, Multisensor Multisource Information Fusion: Architectures, Algorithms, and Applications*, 2004.

[68] R. Maclin and J. Shavlik, "Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks," *Fourteenth Int. Jt. Conf. Artif. Intell.*, pp. 524–531, 1995.

[69] P. Smyth, "Bounds on the mean classification error rate of multiple expert," *Pattern Recognit. Lett.*, 1995.

[70] G. Giacinto and F. Roli, "Ensembles of Neural Networks for Soft Classification of Remote Sensing Images," *Eur. Symp. Intell. Tech.*, pp. 166–170, 1997.

[71] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining Classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, no. 3, pp. 226–239, 1998.

[72] A. Tsymbal, M. Pecherizkiy, S. Puuronen, and D. W. Patterson, "Dynamic Integration of Classifiers in the Space of Principal Components," *ADBIS*, pp. 278–292, 2003.

[73] P. . Gislason, J. A. Benediktsson, and J. . Sveinsson, "Random Forest classification of multisource remote sensing and geographic data," *IEEE Int. Geosci. Remote Sens. Symp. IGARSS'04*, pp. 1049–1052, 2004.

[74] R. Jacobs, "Methods for combining experts' probability assessments," *Neural Comput.*, vol. 7, pp. 867–888, 1996.

[75] G. I. Webb, "Multiboosting: a technique for combining Boosting and Wagging. Machine Learning," *Kluwer Acad. Publ.*, vol. 40, pp. 159–196, 2000.

[76] R. S. Lynch and P. K. Willet, "Classifier fusion results using various open literature data sets," in *EEE International Conference on Systems, Man and Cybernetics*, pp. 723–728, 2003.

[77] K. A. Toh and W. Y. Yau, "Combination of hyperbolic functions for multimodal biometrics data fusion," *IEEE Trans. Syst. Man Cybern.*, vol. 34, no. 2, pp. 1196–1209, 2004.

[78] A. F. R. Rahman and M. C. Fairhurst, "Multiple classifier decision combination strategies for character recognition: A review.," *Int. J. Doc. Anal. Recognit.*, vol. 5, pp. 166–194, 2003.

[79] M. Last, H. Bunke, and A. Kandel, “A feature-based serial approach to classifier combination,” *Pattern Anal. Appl.*, vol. 5, pp. 385–398, 2002.

[80] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[81] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–3, 2001.

[82] I. H. Witten and E. Frank, *Data Mining: Practical Machie Learning Tools and Techniques*, 2th ed. Morgan Kaufmann, p. 558, 2005.

[83] T. K. Ho, “The random subspace method for constructing decision forests,” *Tin Kam Ho. random Subsp. method IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[84] J. H. Friedman, “Stochastic gradient boosting. Computational Statistics and Data Analysis,” *Comput. Stat. Data Anal.*, vol. 38, pp. 367–378, 1999.

[85] Z. Zhou and Y. Yu, “Chapter 7. AdaBoost,” in *The Top Ten Algorithms in Data Mining*, Taylor & Francis Group, LLC, pp. 127–149, 2009.

[86] G. I.Webb, “Multiboosting: A technique for combining boosting and wagging,” *Mach. Learn.*, vol. 40, no. 2, pp. 980–991, 2000.

[87] J. Gama and P. Brazdil, “Cascade generalization,” *Mach. Learn.*, vol. 41, no. 3, pp. 315–343, 2000.

[88] D. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, pp. 241–260, 1992.

[89] A. K. Seewald and J. Fürnkranz, “An evaluation of grading classifiers,” in *4th International Conference, IDA 2001*, pp. 115–124, 2001.

[90] C. E. Brodley, “Recursive automatic bias selection for classifier construction,” *Mach. Learn.*, vol. 20, no. 1–2, pp. 63–94, 1995.

[91] S. J. Nowlan and G. E. Hinton, “Evaluation of adaptive mixtu res of competing experts,” in *Advances in Neural Information Processing Systems*, pp. 774–780, 1990.

[92] P. Chan and S. J. Stolfo, “A comparative evaluation of voting and meta-learning on partitioned data,” in *wel- fth International Conference on Machine Learning*, pp. 90–98, 1995.

[93] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: A decision- tree hybrid,” *KDD*, pp. 202–207, 1996.

[94] L. Breiman, “Bias, variance, and arcing classifiers,” California, Berkeley, 1996.

[95] R. Kohavi and D. Wolpert, "Bias plus variance decomposition for zero- one loss functions.," in *Thirteenth International Conference on Machine Learning*, pp. 275–283, 1996.

[96] E. B. Kong and T. G. Dietterich., "Error-correcting output coding corrects bias and variance," in *Twelfth Internatio- nal Conference on Machine Learning*, pp. 313–321, 1995.

[97] G. W. Greenwood, J.-M. Shin, B. Lee, and G. B. Fogel, "A survey of Recent Work on evolutionary approaches to the Protein Folding Problem," *IEEE*, vol. 99, pp. 488–495, 1999.

[98] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst, "Multimeme Algorithms for Protein Structure Prediction," in *PPSN VII*, vol. LNCS 2439, pp. 769–778, 2002.

[99] F. Zhao, J. Peng, and J. Xu, "Fragment-free approach to protein folding using conditional neural fields," *Bioinformatics*, vol. 26, no. 2009, pp. 310–317, 2010.

[100] S. Mitra, "Bioinformatics With Soft Computing," *IEEE Trans. Syst. MAN, Cybern. C Appl. Rev.*, vol. 36, no. 5, pp. 616–635, 2006.

[101] J. S. Aguilar-ruiz, J. H. Moore, and M. D. Ritchie, "Filling the gap between biology and computer science," *BioData Min.*, vol. 1, no. 1, pp. 1–3, 2008.

[102] S. Kropp, "Data Mining and Bioinformatics," Monash University Faculty of Information Technology Caulfield, VIC, 2004.

[103] D. S. Marks, T. a Hopf, and C. Sander, "Protein structure prediction from sequence variation.," *Nat. Biotechnol.*, vol. 30, no. 11, pp. 1072–80, Nov. 2012.

[104] P. Y. Chou and G. D. Fasman, "Prediction of protein conformation," *Biochemistry*, vol. 13, no. May, pp. 222–245, 2002.

[105] V. I. Lim, "Algorithms for Prediction of a-Helical and / I-Structural Regions in Globular Proteins," *J. Mol. Biol.*, vol. 88, pp. 873–894, 1974.

[106] Y. Shi, J. Zhou, D. Arndt, D. S. Wishart, and G. Lin, "Protein contact order prediction from primary sequences," *BMC Bioinformatics*, vol. 9, no. 255, pp. 1–21, 2008.

[107] A. N. Tegge, Z. Wang, J. Eickholt, and J. Cheng, "NNcon : improved protein contact map prediction using 2D-recursive neural networks," *Nucleic Acids Res.*, vol. 37, no. Web Server issue, pp. 515–518, 2009.

[108] L. H. Holley and M. Karplus, "Protein secondary structure prediction with a neural network," *Biophysics (Oxf).*, vol. 86, no. January, pp. 152–156, 1989.

[109] D. G. Kneller, F. E. Cohen, and R. Langridge, “Improvements in Protein Secondary Structure Prediction by An Enhanced Neural Network,” *J. Mol. Biol.*, vol. 214, pp. 171–182, 1990.

[110] N. Qian and T. J. Sejnowski, “Predicting the Secondary Structure of Globular Proteins Using Neural Network Models,” *J. Mol. Biol.*, vol. 202, pp. 865–884, 1988.

[111] H. Mathkour and M. Ahmad, “An integrated approach for protein structure prediction using artificial neural network,” in *2010 Second International Conference on Computer Engineering and Applications*, pp. 484–488, 2010.

[112] A. Deka, H. Bordoloi, and K. K. R. Sarma, “TERTIARY PROTEIN STRUCTURE PREDICTION USING A SOFT COMPUTATIONAL FRAMEWORK,” *IRNet Trans. Electr. Electron. Eng.*, vol. 1, no. 2, pp. 58–62, 2012.

[113] G. Liu, C. Zhou, Y. Zhu, and W. Zhou, “Prediction of Contact Maps in Proteins Based on Recurrent Neural Network with Bias Units,” *LNCS*, vol. 3498, pp. 686–690, 2005.

[114] G. Liu, Y. Zhu, and W. Zhou, “Prediction of Contact Maps Using Modified Transiently Chaotic Neural Network,” *LNCS*, vol. 3973, pp. 696 – 701, 2006.

[115] S. Chetia and K. K. Sarma, “PROTEIN STRUCTURE PREDICTION USING CERTAIN DIMENSION REDUCTION TECHNIQUES AND ANN,” *IRNet Trans. Electr. Electron. Eng.*, vol. 1, no. 2, pp. 98–103, 2012.

[116] L. Guo, D. Huang, and W. Zhao, “Combining genetic optimisation with hybrid learning algorithm for radial basis function neural networks,” *Electron. Lett.*, vol. 39, no. 22, pp. 29–30, 2003.

[117] C. W. Howe and M. S. Mohamad, “Protein Residue Contact Prediction using Support Vector Machine,” *World Acad. Sci. Eng. Technol.*, vol. 60, pp. 1985–1990, 2011.

[118] A. K. Mandle, P. Jain, and S. K. Shrivastava, “Protein structure prediction using support vector machine,” *Int. J. Soft Comput.*, vol. 3, no. 1, pp. 67–78, 2012.

[119] J. Guo, H. Chen, Z. Sun, and Y. Lin, “A Novel Method for Protein Secondary Structure Prediction Using Dual-Layer SVM and Profiles,” *Sci. York*, vol. 743, no. September 2003, pp. 738 –743, 2004.

[120] Y. Zhao and G. Karypis, “Prediction of Contact Maps Using Support Vector Machines ∗,” *Dep. Comput. Sci. Univ. Minnesota,minneap.*, pp. 1–8, 2002.

[121] J. A. Siepen, S. E. Radford, and D. R. Westhead, “B Edge strands in protein structure prediction and aggregation,” *Protein Sci.*, vol. 12, pp. 2348–2359, 2003.

[122] Y. Chung, G. Kim, Y. Hwang, and H. Park, "Predicting Protein-Protein Interactions from One Feature Using SVM *," in *IEA/AIE*, vol. LNAI 3029, pp. 50–55, 2004.

[123] C.-W. Hsieh, H.-H. Hsu, and M.-D. Lu, "Protein Disordered Region Prediction by SVM with Post-Processing," in *International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 693–698, 2008.

[124] J. Maudes, J. J. Rodríguez, and C. García-Osorio, "Disturbing neighbors ensembles for linear svm," *Lect. Notes Comput. Sci.*, vol. 5519, no. Multiple Classifier Systems, pp. 191–200, 2010.

[125] D. Zou, Z. He, J. He, and X. Huang, "Influence of Encoding Scheme on Protein Secondary Structure Prediction," in *Worrld Congress on Intelligent Control and Automation*, pp. 1439–1443, 2008.

[126] Y. Pan, "Protein Structure Prediction and Interpretation with Support Vector," in *The Fifth International Conference on Computer and Information Technology (CIT'05)*, pp. 7695–7695, 2005.

[127] H. Kim and H. Park, "Protein secondary structure prediction based on an improved support vector machines approach," *Protein Eng.*, vol. 16, no. 8, pp. 553–560, 2003.

[128] J. Song and K. Burrage, "Predicting residue-wise contact orders in proteins by support vector regression," *BMC Bioinformatics*, vol. 15, pp. 1–15, 2006.

[129] K. Shimizu1, S. Hirose, and T. Noguchi, "POODLE-S: web application for predicting protein disorder by using physicochemical features and reduced amino acid set of a position-specific scoring matrix," *Bioinforma. Appl. NOTE*, vol. 23, no. 17, pp. 2337–2338, 2007.

[130] W. Zhong, C. Science, J. He, and Y. Pan, "Multiclass Fuzzy Clustering Support Vector Machines for Protein Local Structure Prediction," in *7th IEEE International Conference on Bioinformatics and Bioengineering, 2007. BIBE 2007.*, pp. 21–26, 2007.

[131] J. He, H. Hu, R. Harrison, P. C. Tai, Y. Pan, and S. Member, "Rule Generation for Protein Secondary Structure Prediction With Support Vector Machines and Decision Tree," *IEEE Trans. Nanobioscience*, vol. 5, no. 1, pp. 46–53, 2006.

[132] A. Reyaz-ahmed and Y. Zhang, "Protein Secondary Structure Prediction Using Genetic Neural Support Vector Machines," in *7th IEEE International Conference on Bioinformatics and Bioengineering, 2007. BIBE 2007.*, pp. 1355–1359, 2007.

[133] C. E. Santiesteban-Toca, M. García-Borroto, and J. S. Aguilar-Ruiz, "Using Short-Range Interactions and Simulated Genetic Strategy to Improve the Protein Contact Map Prediction," in *MCPR 2012*, vol. 7329, pp. 166–175, 2012.

[134] C. E. Santiesteban-Toca, G. Asencio-Cortes, A. E. Márquez-Chamorro, and J. S. Aguilar-Ruiz, "Short-Range Interactions and Decision Tree-Based Protein Contact Map Predictor," in *EvoBIO*, vol. 7246, pp. 224–233, 2012.

[135] K. D. Kedarisetti, K. Chen, A. Kapoor, and L. Kurgan, "Prediction of the Number of Helices for the Twilight Zone Proteins," in *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB'06*, pp. 1–7, 2006.

[136] H. Guo, "Solving 2D HP Protein Folding Problem by Parallel Ant Colonies," in *2nd International Conference on Biomedical Engineering and Informatics, 2009. BMEI'09*, pp. 1–5, 2009.

[137] H. A. A. Bahamish, R. Abdullah, and R. A. Salam, "Protein Conformational Search Using Bees Algorithm," in *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pp. 911–916, 2008.

[138] H. A. A. Bahamish, R. Abdullah, and R. A. Salam, "Protein Tertiary Structure Prediction Using Artificial Bee Colony Algorithm," in *Third Asia International Conference on Modelling & Simulation*, pp. 258–263, 2009.

[139] H. Zhu, X. Lin, S. Zhang, C. Pu, J. Gu, and M. Su, "Protein Structure Prediction with EPSO in Toy Model," in *Second International Conference on Intelligent Networks and Intelligent Systems*, pp. 673–676, 2009.

[140] H. Firpi, E. Youn, and S. Mooney, "Comparative Study of Particle Swarm Approaches for the Prediction of Functionally Important Residues in Protein Structures," in *22nd International Conference on Advanced Information Networking and Applications - Workshops*, pp. 714–719, 2008.

[141] L. Cheng-yuan, D. Yan-rui, and X. Wen-bo, "Multiple-layer Quantum-behaved Particle Swarm Optimization and Toy Model for Protein Structure Prediction," in *Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 92–96, 2010.

[142] A. E. Marquez-Chamorro, G. Asencio-Cortes, F. Divina, and J. S. Aguilar-Ruiz, "Evolutionary decision rules for predicting protein contact maps," *Pattern Anal. Appl.*, vol. 9, Sep. 2012.

[143] N. Mansour and F. Kanj, "Evolutionary Algorithm for Protein Structure Prediction," in *Sixth International Conference on Natural Computation (ICNC 2010)*, pp. 3974–3977, 2010.

[144] T. Higgs, B. Stantic, and A. Sattar, "Genetic Algorithm Feature-Based Resampling for Protein Structure Prediction," in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2010.

[145] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. Van De Peer, R. Blanco, V. Robles, C. Bielza, and P. Larrañaga, "A review of estimation of distribution algorithms in bioinformatics," *BioData Min.*, vol. 1, no. 6, pp. 1–12, 2008.

[146] R. Santana, P. Larrañaga, and J. Lozano, "Protein folding in 2-dimen- sional lattices with estimation of distribution algorithms," *First Int. Symp. Biol. Med. Data Anal. Lect. Notes Comput. Sci.*, vol. 3337, pp. 388–398, 2004.

[147] R. Santana, "Advances in Probabilistic Graphical Models for Optimization and Learning Applications in Protein Modelling," University of the Basque Country, 2006.

[148] R. Santana, P. Larrañaga, and J. Lozano, "Protein folding in simplified models with estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 418–438, 2008.

[149] I. Belda, S. Madurga, X. Llorà, M. Martinell, T. Tarragó, M. G. Piqueras, E. Nicolás, and E. Giralt, "ENPDA: an evolutionary structure-based de novo peptide design algorithm.," *J. Comput. Aided. Mol. Des.*, vol. 19, no. 8, pp. 585–601, Aug. 2005.

[150] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein Classification with Multiple Algorithms," in *10th Panhellenic Conference on Informatics (PCI 2005)*, vol. 3746, pp. 448 – 456, 2005.

[151] J. Bacardit, P. Widera, A. Márquez-Chamorro, F. Divina, J. Aguilar-Ruiz, and N. Krasnogor, "Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features," *Bioinformatics*, vol. 28, no. 19, pp. 2441–2448, 2012.

[152] J. Eickholt and J. Cheng, "Predicting protein residue–residue contacts using deep networks and boosting," *Bioinformatics*, vol. 28, no. 23, pp. 3066–3072, 2012.

[153] G. Yang, C. Zhou, C. Hu, and Z. Yu, "A Method Based on Improved Bayesian Inference Network Model and Hidden Markov Model for Prediction of Protein Secondary Structure," in *28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, vol. 2, pp. 134–137, 2004.

[154] W. Chu, Z. Ghahramani, A. Podtelezhnikov, and D. L. Wild, "Bayesian Segmental Models with Multiple Sequence Alignment Profiles for Protein Secondary Structure and Contact Map Prediction," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 3, no. 2, pp. 98–113, 2006.

[155] P. Wang and D. Zhang, "Protein Secondary Structure Prediction with Bayesian Learning Method," in *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02)*, pp. 252–257, 2002.

[156] S. Akkaladevi and A. K. Katangur, “Protein Secondary Structure Prediction using Bayesian Inference method on Decision fusion algorithms,” in *IEEE International Parallel and Distributed Procssing Symposium*, pp. 1–8, 2007.

[157] Z. Aydin, Y. Altunbasak, and H. Erdogan, “Bayesian Protein Secondary Structure Prediction With Near-Optimal Segmentations,” *IEEE Trans. SIGNAL Process.*, vol. 55, no. 7, pp. 3512–3525, 2007.

[158] S. C. SCHMIDLER, J. S. LIU, and D. L. BRUTLAG, “Bayesian Segmentation of Protein Secondary Structure,” *J. Comput. Biol.*, vol. 7, no. 1/2, pp. 233–248, 2000.

[159] V. Robles, P. Larrañaga, J. M. Peña, E. Menasalvas, M. S. Pérez, V. Herves, and A. Wasilewska, “Bayesian network multi-classifiers for protein secondary structure prediction,” *Articial Intell. Med.*, vol. 31, pp. 117–136, 2004.

[160] H. Pezeshk, S. Naghizadeh, S. A. Malekpour, C. Eslahchi, and M. Sadeghi, “A Modified Bidirectional Hidden Markov Model and its Application in Protein Secondary Structure Prediction,” in *2nd International Conference on Advanced Computer Control (ICACC)*, pp. 535–538, 2010.

[161] Y. Kang and C. M. Fortmann, “Physical Markov Model for Protein Structure Prediction,” in *IEEE International Conference on Bioinformatics and Biomedicine Workshop*, pp. 2006–2006, 2009.

[162] K. Won, T. Hamelryck, A. Prugel-bennett, and A. Krogh, “Evolving Hidden Markov Models,” in *The 2005 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 33–40, 2005.

[163] T. Aksel, “SuPred: Yapay Sinir Aglari ve Sakli Markov Model kullanarak Protein Ikincil Yapi Tahmin Yontemi,” in *IEEE 14th Signal Processing and Communications Applications*, pp. 1–4, 2006.

[164] Z. Aydin, Y. Altunbasak, and M. Borodovsky, “PROTEIN SECONDARY STRUCTURE PREDICTION WITH SEMI MARKOV HMMS,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’04)*, vol. 5, pp. V–577–80, 2004.

[165] M. Lippi, P. Frasconi, and S. Marta, “Prediction of protein β-residue contacts by Markov logic networks with grounding-specific weights,” *Bioinformatics*, vol. 25, no. 18, pp. 2326–2333, 2009.

[166] M. Madera, “Profile Comparer : a program for scoring and aligning profile hidden Markov models,” *Bioinformatics*, vol. 24, no. 22, pp. 2630–2631, 2008.

[167] A. Kumar and L. Cowen, “Recognition of beta-structural motifs using hidden Markov models trained with simulated evolution,” *Bioinformatics*, vol. 26, pp. 287–293, 2010.

[168] M. Li, X. Wang, L. E. I. Lin, and Y. I. Guan, "PROTEIN SECONDARY STRUCTURE PATTERN DISCOVERY AND ITS APPLICATION IN SECONDARY STRUCTURE PREDICTION *," in *Third International Conference on Machine Learning and Cybernetics*, vol. 3, no. August, pp. 1435–1440, 2004.

[169] K. Sikorska, E. Lesaffre, P. F. J. Groenen, and P. H. C. Eilers, "GWAS on your notebook : fast semi-parallel linear and logistic regression for genome-wide association studies," *BMC Bioinformatics*, vol. 14, no. 166, 2013.

[170] P. J. Munson, V. Di Francesco, and R. Porrelli, "Protein Secondary Structure Prediction using Periodic-Quadratic-Logistic Models: Statistical and Theoretical Issues," in *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pp. 375–384, 1994.

[171] M. M. Gromiha, "Multiple Contact Network Is a Key Determinant to Protein Folding Rates," *J. Chem. Inf. Model.*, vol. 49, no. 4, pp. 1130–1135, 2009.

[172] C. E. Santiesteban-Toca, "Predicción de mapas de contactos basados en distancias," Universidad Central de Las Villas y Universidad Pablo de Olavide, 2010.

[173] W. Ding, J. Xie, D. Dai, H. Zhang, H. Xie, and W. Zhang, "CNNcon: Improved Protein Contact Maps Prediction Using Cascaded Neural Networks," *PLoS One*, vol. 8, no. 4, pp. 1–7, 2013.

[174] Z. Wang and J. Xu, "Predicting protein contact map using evolutionary and physical constraints by integer programming," *Bioinformatics*, vol. 29, no. 13, pp. i266–i273, 2013.

[175] N. Habibi, M. Saraee, and H. Korbekandi, "Protein contact map prediction using committee machine approach," *Int. J. Data Min. Bioinforma.*, vol. 7, no. 4, pp. 397–415, 2013.

[176] C. Feinauer, M. J. Skwark, A. Pagnani, and E. Aurell, "Improving Contact Prediction along Three Dimensions," *PLoS Comput Biol.*, vol. 10, no. 10, pp. 1–13, 2014.

[177] P. Kukic, C. Mirabello, G. Tradigo, I. Walsh, P. Veltri, and G. Pollastri, "Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks," *BMC Bioinformatics*, vol. 15, no. 6, pp. 1–15, 2014.

[178] M. Schneider and O. Brock, "Combining Physicochemical and Evolutionary Information for Protein Contact Prediction," *PLoS One*, vol. 9, no. 10, pp. 1–15, 2014.

[179] V. S. Pande, A. Y. Grosberg, T. Tanaka, and D. S. Rokhsar, "Pathways for protein folding: is a new view needed?," *Curr. Opin. Struct. Biol.*, vol. 8, pp. 68–79, 1998.

[180] L. Mirny and E. Shakhnovich, "PROTEIN FOLDING THEORY: From Lattice to All-Atom Models," *Annu. Rev. Biophys. Biomol. Struct.*, vol. 30, pp. 361–396, 2001.

[181] J. N. Onuchic, Z. Luthey-schulten, and P. G. Wolynes, "THEORY OF PROTEIN FOLDING: The Energy Landscape Perspective," *Annu. Rev. Phys. Chem.*, vol. 48, no. 1, pp. 545–600, 1997.

[182] M. Sadqi, L. J. Lapidus, and V. Muñoz, "How fast is protein hydrophobic collapse?," *Proc. Natl. Acad. Sci.*, vol. 100, no. 21, pp. 12117–12122, 2003.

[183] K. H. Mok, L. T. Kuhn, M. Goez, I. J. Day, J. C. Lin, N. H. Andersen, and P. J. Hore, "A pre-existing hydrophobic collapse in the unfolded state of an ultrafast folding protein," *Nature*, pp. 106–109, 2004.

[184] J. Ruiz-Shulcloper, A. Guzmán-Arenas, and J. F. Martinez-Trinidad, "Logical combinatorial approach to pattern recognition: Feature selection and supervised classification," *Editorial Politécnica*, 2000.

[185] M. Vendruscolo, E. Kussell, and E. Domany, "Recovery of protein structure from contact maps," *Fold. Des.*, vol. 2, no. 5, pp. 295–306, 1997.

[186] Z. Qin, C. Zhang, T. Wang, and S. Zhang, "Cost Sensitive Classification in Data Mining," *NCS*, vol. 6440, pp. 1–11, 2010.

[187] P. Turney, "Types of cost in inductive concept learning. In: Workshop on Cost-Sensitive Learning," in *Seventeenth International Conference on Machine Learning University*, pp. 15–25, 2000.

[188] C. Elkan, "The foundations of cost-sensitive learning," in *Seventeenth International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

[189] T. Wang, Z. Qin, and S. Zhang, "Cost-sensitive Learning - A Survey," *Cost-sensitive Learn. - A Surv.*, 2010.

[190] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Articial Intell. Res.*, vol. 7, pp. 341–378, 2002.

[191] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," *Springer-Verlag*, vol. 8, pp. 878–887, 2005.

[192] G. Cohen, M. Hilario, H. Sax, S. Hogonnet, and A. Geissbuhler, "Learning from imbalanced data in surveillance of nosocomial infection," *Articial Intell. Med.*, vol. 7, 2006.

[193] Z. Aydin, H. Erdogan, and Y. Altunbasak, “Protein fold recognition using residue-based alignments of sequence and secondary structure,” in *ICASSP 2007*, pp. 349–352, 2007.

[194] M. Hauser, C. E. Mayer, and J. Söding, “kClust: fast and sensitive clustering of large protein sequence databases,” *BMC Bioinformatics*, vol. 14, no. 248, pp. 1–12, 2013.

[195] L. I. Kuncheva, “Classifier Selection,” in *Combining Pattern Classifiers. Methods and Algorithms*, 1st ed., Wiley Interscience, pp. 189–202, 2004.

[196] H. Xue, Q. Yang, and S. Chen, “Chapter 3 SVM : Support Vector Machines,” in *The Top Ten Algorithms in Data Mining*, Taylor & Francis Group, LLC, pp. 37–59, 2009.

[197] D. J. Hand, “Chapter 9. Naïve Bayes,” in *The Top Ten Algorithms in Data Mining*, no. 1981, pp. 163–177, 2009.

[198] M. Steinbach and P. Tan, “Chapter 8. kNN: k-Nearest Neghbors,” in *The Top Ten Algorithms in Data Mining*, vol. 1, no. i, Taylor & Francis Group, LLC, pp. 151–161, 2009.

[199] N. Ramakrishnan, “Chapter 1. C4.5,” in *The Top Ten Algorithms in Data Mining*, Taylor & Francis Group, LLC, pp. 1–19, 2009.

[200] E. Frank, M. Hall, L. Trigg, G. Holmes, and I. H. Witten, “Data mining in bioinformatics using Weka,” *Bioinformatics*, vol. 20, no. 15, pp. 2479–2481, 2004.

[201] S. García, A. Fernández, J. Luengo, and F. Herrera, “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power,” *Inf. Sci. (Ny).*, vol. 180, no. 10, pp. 2044–2064, May 2010.

[202] P. Rzepakowski and S. Jaroszewicz, “Decision trees for uplift modeling with single and multiple treatments,” *Knowl. Inf. Syst.*, vol. 32, no. 2, pp. 303–327, 2012.

[203] A. Etemad-Shahidi and J. Mahjoobi, “Comparison between M5 0 model tree and neural networks for prediction of significant wave height in Lake Superior,” *Ocean Eng.*, vol. 36, no. 15–16, pp. 1175–1181, 2009.

[204] M. Learning, K. A. Publishers, A. C. Sciences, and R. August, “Induction of Decision Trees,” pp. 81–106, 2007.

[205] I. A. Nepomuceno Chamorro, “Reconocimiento de Redes de Genes Mediante Regresión,” Universidad Pablo de Olavide de Sevilla, 2010.

[206] A. R. Fersht, "Transition-state structure as a unifying basis in protein-folding mechanisms : Contact order , chain topology , stability , and the extended nucleus mechanism," *PNAS*, vol. 97, no. 4, pp. 1525–1529, 2000.

[207] Z. Wang and J. Xu, "Predicting protein contact map using evolutionary and physical constraints by integer programming.," *Bioinformatics*, vol. 29, no. 13, pp. i266–73, Jul. 2013.

[208] B. Li, L. Hu, L. Chen, K. Feng, Y. Cai, and K. Chou, "Prediction of Protein Domain with mRMR Feature Selection and Analysis," *PLoS One*, vol. 7, no. 6, 2012.

[209] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, pp. 861–874, 2006.

[210] P. Benkert, M. Biasini, and T. Schwede, "Toward the estimation of the absolute quality of individual protein structure models," *Bioinforma. Adv. Access*, vol. 1, pp. 1–8, 2010.

[211] S. García and F. Herrera, "An Extension on 'Statistical Comparisons of Classifiers over Multiple Data Sets' for all Pairwise Comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, 2008.

[212] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[213] D. J. Sheskin, *Handbook of PARAMETRIC and NONPARAMETRIC STATISTICAL PROCEDURES*, Third Edit. Western Connecticut State University: Chapman & Hall/CRC, p. 1184, 2004.

[214] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *23 rd International Con- ference on Machine Learning, Pittsburgh, PA, 2006.*, pp. 2–8, 2006.

[215] T. G. Dietterich, "Statistical Tests for Comparing Supervised Classication Learning Algorithms 1 Introduction," pp. 1–24, 1997.

[216] S. Wu and Y. Zhang, "A comprehensive assessment of sequence-based and template- based methods for protein contact prediction," *Bioinformatics*, vol. 24, pp. 924–931, 2008.

[217] P. Björkholm, P. Daniluk, A. Kryshtafovych, K. Fidelis, R. Andersson, and T. R. Hvid-Sten, "Using multi-data hiddenMarkov models trained on local neighborhoods of protein structure to predict residue-residue contacts," *Bioinformatics*, vol. 25, pp. 1264– 1270, 2009.

[218] S. Wu and Y. Zhang, "Structural bioinformatics A comprehensive assessment of sequence-based and template-based methods for protein contact prediction," *Bioinformatics*, vol. 24, no. 7, pp. 924–931, 2008.

[219] B. Monastyrskyy, D. D'Andrea, K. Fidelis, A. Tramontano, and A. Kryshtafovych, "Evaluation of residue-residue contact prediction in CASP10," *PROTEINS Struct. Funct. Bioinforma.*, vol. 82, no. 2, pp. 138–153, 2014.

[220] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 108, no. 49, pp. E1293–301, Dec. 2011.

Parte VII

Apéndices

Anexo 1.- Matriz de sustitución

Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
	Ala	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**

* BLOSUM (*BLOcks of Amino Acid SUbstitution Matrix,* o matriz de sustitución de bloques de aminoácidos) es una matriz de sustitución utilizada para el alineamiento de secuencias de proteínas. BLOSUM se usa para puntuar alineamientos entre secuencias de proteínas evolutivamente divergentes.

En biología evolutiva una matriz de sustitución, o de puntuación, describe el ritmo al que un carácter en una secuencia cambia a otro carácter con el tiempo.

Anexo 2.- Conjunto de 49 proteínas para el análisis del dominio de aplicación.

Alpha	Ls	Alpha/Beta	Ls	Alpha+Beta	Ls	Beta	Ls
1DT0_A	197	1CVI_A	342	1C9H_A	107	1C3G_A	170
1EFY_A	350	1DXE_A	253	1E8I_A	117	1CUO_A	129
1EW6_A	137	1DXH_A	335	1EOE_A	100	1DN2_A	207
1EYH_A	144	1E6K_A	130	1EUV_A	220	1DQI_A	124
1F06_A	320	1E6L_A	127	1EZV_F	125	1DS0_A	323
1F2U_B	145	1EVI_A	340	1FNT_K	198	1EO2_B	238
1F4O_A	165	1F1M_A	162	1FWK_A	296	1EQD_A	184
1FBV_A	388	1F5O_A	149	1G24_A	211	1FF5_A	219
1FK0_A	93	1FAA_A	120	1G62_A	224	1G43_A	160
1FT5_A	210	1I0Z_A	33	1HQM_E	91	1HWM_B	264
1FYZ_E	168			1IO1_A	395	1I5I_A	174
1HM6_A	345			1QI7_A	253	1I94_Q	104
1HNW_D	208						
1I1R_A	302						
1I94_D	208						

Longitud de la Proteína (*Ls*) es el número de residuos de la estructura covalente.

Anexo 3.- Conjunto de validación interna.

LS: 0-100 (778)

1A1P 1A7I 1A7Y 1A8O 1ABA 1ADX 1ADZ 1AFO 1AH9 1AHO 1AJY 1AL1 1ALE 1APJ 1AVY 1AWY 1AZ6 1B22 1B3A 1B4G
1B9U 1BAZ 1BBG 1BCT 1BCV 1BFW 1BOE 1BRV 1BTS 1BW6 1BYO 1BZK 1C01 1C4B 1C4E 1C94 1C9O 1CB3 1CL4 1CQ4
1CQY 1CV9 1CXY 1CYB 1CZQ 1D1N 1D6G 1D7N 1D7Q 1D7T 1D8B 1D8J 1D9L 1DEB 1DFW 1DGN 1DQB 1DU6 1DUM 1DVW
1DXS 1E0B 1E0E 1E74 1E75 1ECI 1EDS 1EI0 1ELO 1EN2 1ENH 1ERY 1ET1 1F0D 1F43 1F4I 1F56 1F62 1F6V 1F9P
1FI0 1FOZ 1FP0 1FRE 1FSB 1FUL 1FW5 1FYJ 1G1Z 1G2B 1G2C 1G2Y 1G6E 1G6U 1GHC 1GHH 1GJF 1GJS 1GMJ 1GO6
1GVD 1GVP 1GW3 1H75 1H7V 1H8B 1H8G 1H98 1HA9 1HDF 1HFE 1HFF 1HGZ 1HJO 1HO7 1HP9 1HQJ 1HR1 1HST 1HTM
1HU5 1HUC 1HY9 1HYK 1HYM 1HZ6 1IO7 1I25 1I2T 1I35 1I71 1I8E 1I93 1IFK 1IFL 1IFP 1IFY 1IGQ 1IHR 1IML
1IQS 1IRZ 1IUR 1IUY 1IV0 1IXU 1IYC 1JOT 1J7M 1J7Q 1J9I 1JAU 1JBF 1JD8 1JEG 1JEI 1JH4 1JI7 1JK4 1JLO
1JMT 1JNI 1JO4 1JXC 1JY4 1JZP 1K1Z 1K36 1K5O 1K7B 1K8V 1KB8 1KBE 1KFT 1KG1 1KJ6 1KMR 1KN6 1KOY 1KQE
1KQI 1KUW 1KVE 1KWA 1KWD 1KZ2 1KZ5 1L4T 1L5P 1L9L 1LBJ 1LG4 1LGH 1LR1 1LU0 1LV4 1LWR 1M7L 1MA6 1MB1
1MEA 1MGQ 1MHN 1MKC 1MLI 1MOF 1MOT 1NO9 1NOD 1NOZ 1N1J 1N5G 1NCS 1NCT 1NEG 1NEI 1NG7 1NHO 1NKF 1NKZ
1NQ4 1NT6 1NTC 1NY4 1NY8 1NZP 1NZS 1O6W 1O8R 1O9Y 1OB6 1OEI 1OM2 1OV2 1OV9 1OW5 1OZZ 1POL 1POO 1P5K
1P82 1P9G 1P9I 1P9K 1PAV 1PB5 1PCE 1PD6 1PDG 1PEH 1PEN 1PFT 1PJV 1PLC 1PLW 1PM3 1PN5 1PYA 1PZW 1QOV
1Q2J 1Q68 1Q8D 1Q8H 1QD6 1QFN 1QJT 1QP6 1QPM 1QS3 1QW2 1QXQ 1QYP 1R05 1R77 1R7J 1R8O 1R94 1R9V 1RBD
1RG6 1RI9 1RIJ 1RMK 1RQ6 1RR7 1RXR 1RYK 1RZS 1S1N 1S3A 1S4H 1S6W 1S7P 1SAN 1SE0 1SG5 1SJ1 1SKK 1SN9
1SSL 1SXD 1T0H 1T0Y 1T1V 1T51 1T52 1T55 1T5M 1T7H 1TER 1TG0 1TH7 1TIF 1TIG 1TOR 1TOS 1TOT 1TP4 1TPG
1TPM 1TTN 1U0I 1U5M 1U9L 1U9P 1UAO 1UAW 1UEO 1UFN 1UGI 1UL5 1UMQ 1UNO 1UTG 1UUC 1UUJ 1UV7 1UVF 1UVG
1V31 1V5O 1V5R 1V66 1V6P 1V7H 1V9O 1V92 1V98 1VBW 1VC3 1VD4 1VDI 1VDL 1VFY 1VH6 1VIG 1VL3 1VM3 1VMG
1VZM 1W1N 1W2I 1W41 1W4J 1WAP 1WBR 1WCO 1WEQ 1WFD 1WHZ 1WII 1WIL 1WJ2 1WJN 1WJV 1WKT 1WM3 1WO3 1WOT
1WQ6 1WQB 1WQE 1WQJ 1WRG 1WRI 1WSO 1WT6 1WU0 1WV9 1WVK 1WXS 1WY3 1WZ4 1X32 1X3U 1X3X 1X4O 1X4R 1X58
1X7V 1X9B 1X9X 1XE1 1XF7 1XGA 1XKM 1XMT 1XOU 1XRZ 1XS3 1XT7 1XU6 1XWR 1XX3 1XY4 1XY5 1YO2 1Y29 1Y43
1Y4E 1Y66 1YCE 1YDO 1YIB 1YJP 1YL9 1YLQ 1YOD 1YSF 1YUK 1YVC 1YWW 1YY1 1Z2Q 1Z4H 1Z6V 1Z9F 1ZAQ 1ZGX
1ZKE 1ZMI 1ZPV 1ZPW 1ZR9 1ZRV 1ZUF 1ZUY 1ZWV 1ZX3 1ZXA 2AO5 2A1C 2A1J 2A26 2A3D 2A6C 2A7Y 2A93 2AB3
2AB9 2AGH 2AIB 2AJE 2AJW 2AKK 2AL3 2AMI 2AQO 2ARI 2AXD 2AYO 2B19 2B5Q 2B7E 2B7T 2B97 2BA2 2BAY 2BC8
2BEQ 2BEY 2BF9 2BFI 2BH8 2BKF 2BP4 2BPS 2BSK 2BV2 2BWF 2BYK 2C0S 2C3G 2C60 2C6A 2C7H 2CJJ 2CK5 2CKC
2CKX 2CP9 2CPG 2CQA 2CQW 2CRL 2CS3 2CS7 2CSA 2CVI 2CX6 2CYU 2CZ4 2D35 2D68 2DCI 2DDI 2DGR 2DIO 2DJR
2DJW 2DK1 2DL6 2DNT 2DO5 2DOG 2DQ5 2E1F 2E2S 2E62 2E6R 2E6W 2E6Z 2E73 2E8D 2EA6 2EAM 2EBB 2EBI 2EBV
2EBW 2ECI 2EE1 2EEM 2EF8 2EGP 2EHE 2EKI 2EKK 2ELN 2ELO 2ELU 2ENV 2EQE 2EQJ 2EQQ 2ERS 2ES6 2EVB 2EVQ
2EW4 2EZK 2EZW 2F3A 2F6M 2FA8 2FB0 2FC6 2FC7 2FDO 2FFM 2FHT 2FK4 2FLG 2FLY 2FMR 2FN2 2FQ8 2FQC 2FQM
2FS1 2FTX 2G2S 2GDL 2GF4 2GFF 2GFR 2GGR 2GIB 2GO8 2GRG 2GTJ 2GX1 2HOD 2H1Z 2H3N 2H4B 2H8S 2HBA 2HDZ
2HF5 2HFR 2HFV 2HG7 2HGC 2HGF 2HGO 2HI3 2HJM 2HL7 2HM2 2HN8 2HNU 2HO2 2HQH 2HTS 2HUG 2IO4 2IOX 2I18
2I1D 2I2H 2I4S 2I5F 2I5U 2I6V 2I9Z 2IC6 2IV5 2IY2 2IZX 2J5H 2J76 2J7J 2JEE 2JMC 2JMD 2JMY 2JN5 2JNH
2JO1 2JOF 2JOR 2JOS 2JOW 2JP6 2JPC 2JPI 2JPW 2JQ1 2JQ3 2JQ4 2JQS 2JQW 2JR3 2JR5 2JRT 2JRW 2JRY 2JS3
2JS9 2JSB 2JSX 2JTB 2JTG 2JTM 2JTU 2JUO 2JV4 2JV5 2JVR 2JWG 2JX5 2JX8 2JXJ 2JXZ 2JY5 2JZ7 2JZ8 2K0B
2K19 2K1P 2K2Q 2K38 2K3C 2K3J 2K47 2K4X 2K52 2K59 2K5J 2K6I 2K6M 2K7G 2K7Y 2K8X 2K9J 2K9L 2K9P 2KA1
2KBC 2KBZ 2KCC 2KCM 2KCN 2KCT 2KCV 2KD3 2KDR 2KE1 2KEG 2KEL 2KEO 2KER 2KFK 2KI9 2KIB 2KIX 2KJ1 2KJ6
2KJF 2KJY 2KK4 2KK7 2KKE 2KL8 2KLU 2KN9 2KNJ 2MLP 2NN4 2NPB 2NQC 2NWT 2NX7 2NZ7 2O0S 2O1K 2O4T 2O6K
2O71 2O8X 2OA5 2OBP 2OCT 2OMP 2OMQ 2OOA 2OPO 2OT2 2OU1 2OVG 2OYV 2OYY 2OYZ 2P06 2P4E 2P5M 2P5T 2P63
2P7R 2P9X 2PJV 2PKA 2PNE 2PNV 2PO8 2PQR 2Q2F 2Q33 2QFF 2QIF 2QKH 2QKQ 2QLX 2QSB 2QSK 2QTX 2QVO 2QYC
2QYW 2QZD 2QZI 2RCZ 2RHF 2RIL 2RK5 2RLL 2RMF 2RMG 2RN9 2RND 2RNL 2RNM 2RNQ 2ROO 2RO3 2RP4 2RPA 2UWQ
2V1R 2V1T 2V2F 2V5F 2V75 2VKN 2VKP 2VLG 2VRD 2VT1 2VXF 2VY5 2WOT 2W1O 2W5O 2W7A 2W84 2W9J 2WG5 2WIE
2WQI 2WQJ 2WT8 2WX3 2YRA 2YRC 2YRK 2YSO 2YSL 2YT8 2YTV 2YUO 2YU4 2YU8 2YUK 2YUM 2YX8 2YYF 2ZOA 2ZOR
2ZDJ 2ZFZ 2ZNF 2ZPM 2ZZT 3A7O 3B4D 3B4S 3BAS 3BD1 3BEY 3BFO 3BHP 3BNO 3BN7 3BPJ 3BQP 3BQS 3BRI 3BRV
3BS3 3BV8 3BW1 3BY7 3BYP 3BZY 3COC 3C6W 3CA7 3CCD 3CEC 3CI9 3CJH 3CMH 3CTV 3CZC 3DOW 3D2Q 3D8L 3DKM
3DM3 3DNL 3DP5 3DWU 3E07 3E0E 3E19 3E4H 3E8V 3ENC 3EUN 3EUS 3EWO 3EWG 3F5H 3F6O 3FB9 3FBL 3FCG 3FOD
3FPO 3FT7 3FVA 3FYB 3G1G 3GGM 3GHD 3GL6 3GZ7 3GZF 3GZM 3H36 3H5G 3HFO 3HGL 3HIL 3HLU 3HTK 3HTY 3HZ7
3HZQ 3I3C 3IF4 3IG9 3JQH 3JTN 3JVO 3K0X 3K3S 3KIK 3KLW 3MRA 3NLA 3SAK 6RLX

LS: 100-200 (772)

1A1X 1A3C 1A6J 1AC0 1AHS 1AQC 1AUY 1AWE 1AYO 1B0B 1B1C 1B24 1B93 1BAK 1BEA 1BGF 1BHD 1BJ8 1BKR 1BM9
1BOU 1BPR 1BTK 1BTN 1BW3 1BXD 1BYF 1BYR 1BZ4 1C05 1C2N 1C7K 1C8N 1CEX 1CID 1CMC 1COZ 1CUK 1CV8 1D2Z
1D4O 1D7M 1D7P 1D9C 1D9S 1DDB 1DGW 1DI0 1DMG 1DOV 1DYO 1E29 1E5K 1E7L 1E88 1EAJ 1EIW 1EJ8 1EJE 1ETE
1EX2 1EX7 1EXG 1EXT 1EZ3 1F1E 1F32 1F3U 1F5M 1F7C 1F7D 1F86 1FGY 1FHG 1FHT 1FJR 1FPZ 1FVG 1FVK 1FYV
1G1T 1G2I 1G31 1G5Q 1G7D 1G84 1G9L 1GCF 1GHE 1GME 1GMX 1GNY 1GPR 1GUI 1GV8 1HO5 1H2I 1H4X 1H97 1HCE
1HD2 1HEK 1HJR 1HKQ 1HML 1HQZ 1HR3 1HRU 1HTN 1HUF 1HUL 1IOR 1I12 1I16 1I17 1I2H 1I58 1I5N 1I6Z 1IBY
1IDP 1IFQ 1IFR 1IG6 1II8 1IJY 1IOO 1IQV 1IRG 1IRS 1IRY 1ITH 1ITV 1IUF 1IVZ 1IWM 1IX5 1J1H 1J3O 1J3W
1JB3 1JBA 1JER 1JFM 1JHC 1JHF 1JHG 1JIG 1JLI 1JMV 1JOC 1JR8 1JT8 1JYA 1K1E 1K2E 1KA6 1KJN 1KKG 1KL9
1KLL 1KLX 1KMV 1KQ6 1KSR 1KU9 1KXG 1L1P 1L3A 1L5I 1LB6 1LF7 1LKI 1LKK 1LKP 1LM5 1LM8 1LSL 1LU4 1LY1
1M1F 1M1H 1M5Q 1MD6 1MG4 1MIL 1MK4 1MKA 1MW5 1MWW 1MXI 1MZK 1N0E 1N0R 1N1A 1N3G 1N3K 1N62 1N6Z 1N9P
1NAO 1NBC 1NCN 1NEP 1NFA 1NG2 1NI7 1NIG 1NL1 1NO5 1NP8 1NRJ 1NWA 1NWW 1NWZ 1NXH 1NXI 1NXJ 1NXM 1NZI
1O22 1O8B 1O9R 1OBO 1OCY 1OD3 1OD6 1OHU 1OI4 1OJ5 1OOO 1OOH 1OP4 1OQA 1OR0 1OR4 1OSC 1OSG 1OTG 1OU8
1OUW 1OV3 1OW4 1OXJ 1OY2 1P0Z 1P32 1P4P 1P5S 1P5T 1P90 1PBU 1PI1 1PL3 1PM4 1PPY 1PQ1 1PQH 1PQW 1PUL
1PWB 1Q0P 1Q77 1Q8C 1Q9C 1QB3 1QCS 1QFO 1QHX 1QJ8 1QUO 1QVC 1QWD 1QYN 1R4V 1R5E 1R5S 1RDU 1RJH 1RLH
1RLJ 1RMD 1ROC 1RTT 1RW6 1RXQ 1S14 1S2X 1S6D 1S79 1S7A 1S7I 1S7K 1S7M 1S8N 1SAU 1SBQ 1SHS 1SJ7 1SJY
1SK7 1SKZ 1SL6 1SMB 1SQL 1SQU 1SQW 1SRA 1SS4 1SS6 1SU0 1SX7 1T1D 1T1J 1T35 1T4Y 1T6U 1T82 1TC1 1TDP
1TFE 1TH8 1THQ 1TOZ 1TP5 1TQ8 1TQG 1TUJ 1TUL 1TUZ 1TVG 1TVM 1U2W 1U3B 1U4F 1U5F 1U79 1U7P 1U9D 1UAP
1UB1 1UB9 1UEB 1UJK 1ULY 1UPQ 1UT7 1UTY 1UUN 1UVQ 1UWO 1UW7 1UWW 1UXO 1UXZ 1V2Y 1V4R 1V5K 1V5M 1V5P
1V87 1V8C 1V8H 1V9V 1V9Y 1VCD 1VHI 1VHU 1VJ2 1VJE 1VKB 1VKC 1VKF 1VKI 1VMB 1VP8 1VR3 1VSR 1VYI 1W1H
1W8I 1W94 1W9A 1W9E 1W9R 1WF6 1WFS 1WFX 1WFY 1WGK 1WGO 1WGR 1WHN 1WI5 1WJ6 1WJ7 1WJJ 1WJQ 1WKQ 1WLM
1WMX 1WNA 1WOL 1WOU 1WPB 1WPV 1WS6 1WUB 1WWI 1WXP 1WY9 1X51 1X94 1X9U 1XB4 1XED 1XFS 1XGW 1XJC 1XJU
1XKE 1XL3 1XM5 1XMA 1XMW 1XPN 1XQB 1XSV 1XT5 1XTE 1XW3 1XWN 1XZO 1YOH 1YOK 1Y63 1Y8M 1Y93 1Y9I 1Y9J
1Y9L 1Y9Q 1YD7 1YFU 1YG2 1YGM 1YGT 1YK9 1YKU 1YM3 1YO7 1YOC 1YPY 1YQ8 1YQH 1YRE 1YRK 1YS5 1YUD 1YWU
1YX4 1YXE 1Z1Y 1Z23 1Z2W 1Z3E 1Z6N 1Z6U 1Z7U 1Z81 1ZAV 1ZBO 1ZCE 1ZDO 1ZHV 1ZLD 1ZSO 1ZTS 1ZUO 1ZVP

1ZZW 2A15 2A4D 2A4H 2A5Y 2A61 2A67 2A7O 2A8N 2A9U 2AAN 2ABY 2AE8 2AF7 2AG4 2AHD 2AJ7 2AKL 2ANX 2AO7
2AO9 2AP3 2AP6 2AR1 2AR5 2ARC 2ARW 2ASF 2ATZ 2AUW 2AVX 2AZE 2AZW 2B0R 2B18 2B1L 2B2A 2B3G 2B3W 2B4A
2B6F 2B7F 2B8M 2BAF 2BBE 2BBZ 2BK9 2BKM 2BMX 2BOU 2BRJ 2BSJ 2BUK 2BUR 2BW2 2BYC 2BZ1 2C34 2CAY 2CCM
2CFX 2CH8 2CLB 2CNT 2COE 2CP5 2CQ8 2CQM 2CSW 2CT9 2CTT 2CUA 2CVE 2CW9 2CWO 2CX7 2CXY 2CYE 2DOO 2D28
2D4G 2D58 2D7E 2D7V 2D86 2D8M 2D9L 2DAO 2DAV 2DAW 2DBA 2DBC 2DC3 2DCP 2DCQ 2DDZ 2DFB 2DGZ 2DHX 2DJF
2DKO 2DMH 2DMW 2DNX 2DOK 2DP9 2DPF 2DSC 2DVK 2E2A 2E2W 2E3V 2E56 2E5E 2E5N 2E7C 2E7M 2E8B 2E8O 2EA9
2EBU 2EBY 2ECU 2ED6 2EDO 2EEE 2EFR 2EHO 2EIS 2EJN 2ELA 2EQS 2EQX 2ERV 2ETD 2EV1 2EWC 2EXU 2FO1 2FOC
2F22 2F4P 2F5J 2F6E 2F76 2F96 2F9F 2FA5 2FB6 2FBH 2FBN 2FCK 2FD5 2FEX 2FG1 2FGC 2FI9 2FIP 2FIW 2FJ8
2FK5 2FKD 2FLH 2FO3 2FQL 2FR2 2FR5 2FSX 2FVV 2FW5 2FXA 2FXM 2FYG 2FZF 2G0B 2G46 2G64 2G7H 2G7S 2G84
2G9T 2GA1 2GAX 2GBB 2GD5 2GD9 2GEC 2GED 2GF6 2GFG 2GFN 2GGZ 2GIA 2GIY 2GKP 2GLZ 2GM3 2GM5 2GPY 2GQB
2GS5 2GUJ 2GUX 2GZB 2H1C 2H1E 2H1T 2H36 2H8E 2HAZ 2HB5 2HCU 2HDO 2HDV 2HEW 2HG6 2HGA 2HH8 2HJQ 2HKU
2HLO 2HLW 2HNG 2HSB 2HTI 2HUH 2HUJ 2HVA 2HW2 2HW4 2HWV 2HWY 2HX6 2HXJ 2HXR 2HY5 2I1O 2I32 2I38 2I52
2I6C 2I6H 2I6J 2I79 2I7A 2I7D 2I7K 2I88 2I8B 2I9Y 2IAF 2IAK 2IBO 2ICG 2IDG 2IEC 2IF7 2IFS 2IFX 2IIH
2III 2IKB 2IM8 2IMS 2IMZ 2INO 2IOJ 2ISB 2ISY 2IT9 2IU1 2IVW 2IWL 2IZ6 2IZW 2JON 2J5A 2J89 2J8A 2J8H
2J8I 2J9C 2JA4 2JAY 2JCB 2JDA 2JE0 2JLJ 2JM1 2JMR 2JOD 2JOK 2JOX 2JOZ 2JP0 2JPE 2JQO 2JQV 2JRK 2JRZ
2JSS 2JTY 2JUA 2JUH 2JUL 2JUN 2JVA 2JWY 2JX3 2JX9 2JXN 2JY9 2JYC 2JZD 2JZX 2JZY 2K0M 2K0Z 2K1D 2K2S
2K2W 2K33 2K3G 2K3Q 2K48 2K49 2K4Q 2K4V 2K5D 2K5H 2K5Q 2K6G 2K7Q 2K7R 2K7W 2K86 2K87 2K8H 2K8Q 2K9K
2K9U 2KAA 2KAT 2KB2 2KBG 2KC5 2KCA 2KCW 2KDX 2KFS 2KGF 2KHD 2KHV 2KIG 2KK0 2KK1 2KKL 2KKP 2KKQ 2KKX
2KLH 2KLL 2KLO 2KMS 2KND 2KNH 2KNR 2KOB 2KQ9 2NML 2NMS 2NO2 2NOG 2NPT 2NRR 2NSC 2NSZ 2NUJ 2NWO 2NX4
2O4D 2O5F 2O66 2O6F 2O6L 2O7T 2O95 2OA2 2OB0 2OB9 2OBB 2ODA 2OER 2OF7 2OFZ 2OGK 2OH1 2OI8 2OIW 2OIZ
2OKQ 2OLM 2OMO 2ONF 2OOC 2OOI 2OOK 2OPC 2OPK 2ORW 2OSE 2OU5 2OWP 2OX1 2OX7 2OXG 2P0B 2P0S 2P12 2P1A
2P1J 2P23 2P25 2P8G 2P8T 2P9R 2PAG 2PCN 2PD1 2PEO 2PHO 2PIM 2PJS 2PKD 2PLN 2PN6 2PPN 2PQ7 2PTH 2PTT
2PWJ 2PYQ 2PZ9 2QOO 2QO3 2Q22 2Q2G 2Q30 2Q3F 2Q3P 2Q3W 2Q3X 2Q47 2Q4M 2Q7M 2Q8O 2QBW 2QCK 2QDL 2QEC
2QEU 2QG8 2QGO 2QGZ 2QH9 2QHK 2QIH 2QJW 2QKO 2QL8 2QND 2QNG 2QNI 2QPV 2QPW 2QQZ 2QS7 2QSQ 2QU8 2QVK
2QWV 2QWW 2QYA 2QZ7 2QZQ 2R18 2R2A 2R32 2R47 2R5S 2R6M 2R6V 2R76 2RA7 2RBB 2RD9 2RE3 2REE 2RER 2RFM
2RGT 2RH3 2RHK 2RIQ 2RJ2 2RJZ 2RK3 2RKK 2RM4 2RNN 2ROP 2RPB 2RQ5 2TNF 2UWN 2UX0 2V0X 2V14 2V1N 2V1O
2V44 2V4X 2V5T 2V6U 2V72 2V7S 2V9H 2VB1 2VBU 2VCL 2VGA 2VN1 2VN6 2VO8 2VO9 2VQ5 2VQ8 2VT3 2VU4 2VX8
2VY6 2VZ5 2W1J 2W1S 2W3X 2W4Y 2W57 2W5E 2W68 2W6K 2W8O 2W82 2WB3 2WEW 2WFO 2WG7 2YS5 2YSK 2YSR 2YTO
2YT2 2YT4 2YTU 2YUD 2YUF 2YUH 2YUI 2YVO 2YVQ 2YW3 2YWI 2YXH 2YYH 2YZJ 2YZK 2ZOB 2Z14 2Z16 2Z51 2Z6F
2ZCA 2ZCM 2ZCW 2ZEX 2ZF9 2ZK9 2ZQO 2ZU1 2ZXJ 3AOY 3A1W 3A35 3A43 3A6S 3A8R 3B33 3B42 3B7C 3B7X 3B8L
3BB9 3BBB 3BBD 3BBY 3BDI 3BED 3BF2 3BI7 3BJA 3BJK 3BJN 3BK6 3BLJ 3BLN 3BLZ 3BM1 3BMZ 3BRO 3BW6 3BY6
3BY8 3BYW 3BZB 3BZH 3C12 3C1D 3C4R 3C6V 3C7M 3C8I 3CCG 3CEX 3CG6 3CGM 3CI3 3CI6 3CJD 3CJE 3CJN 3CK2
3CNU 3CNV 3COO 3COU 3CPM 3CPT 3CQN 3CQV 3CRJ 3CS1 3CT6 3CTD 3CX2 3CXJ 3CXK 3CYP 3CZ6 3D1P 3D3B 3D4R
3D6I 3D6R 3D6W 3D9R 3DB0 3DCZ 3DEE 3DEO 3DKZ 3DMB 3DO8 3DQY 3DRX 3DSB 3DWD 3E0Z 3E1E 3E8M 3E8P 3E99
3EBN 3EDU 3EEF 3EES 3EF4 3EFA 3EGQ 3EHF 3EHH 3ELI 3EMF 3EN8 3EO5 3EPU 3ERB 3ERY 3ES4 3ETN 3EYR 3EYT
3EYX 3F04 3F08 3F0I 3F13 3F14 3F1B 3F1R 3F2I 3F4O 3F4A 3F59 3F5D 3F5T 3F6A 3F8B 3F9S 3F9X 3FAC 3FAJ
3FB2 3FBU 3FCD 3FDX 3FF0 3FF2 3FFL 3FGR 3FH1 3FH2 3FH3 3FHK 3FK8 3FK9 3FLD 3FLH 3FLJ 3FMV 3FMZ 3FN5
3FOG 3FPW 3FRQ 3FRV 3FS3 3FSA 3FSD 3FUY 3FVB 3FXA 3FY6 3FYN 3FZ5 3FZ9 3FZE 3G13 3G1M 3G2E 3G39 3G3Z
3G8W 3GA4 3GG6 3GGN 3GHA 3GM3 3GM5 3GN3 3GPK 3GQH 3GRZ 3GSD 3GWN 3GXH 3GYW 3GZ8 3GZB 3GZI 3HOK 3HON
3H3H 3H4Q 3H51 3H6N 3H8T 3H8U 3HA2 3HA9 3HB7 3HBW 3HEO 3HE1 3HF5 3HF7 3HFI 3HGB 3HHO 3HHC 3HHJ 3HIU
3HKV 3HL4 3HM2 3HM4 3HNR 3HNW 3HOI 3HQX 3HTM 3HU5 3HVU 3HW5 3I08 3IOY 3I38 3I57 3I7D 3I9X 3IBZ 3IC5
3ICU 3IEY 3IJM 3IK8 3IKB 3IM6 3IMI 3IMM 3IOL 3IPR 3IQS 3IR3 3IT7 3IU5 3IV4 3IVP 3IVV 3IWT 3IX3 3IX7
3JRN 3JRV 3JUW 3JX9 3JXS 3JZY 3KOW 3KOZ 3K29 3K44 3K63 3K67 3K69 3K9I 3KDF 3KDH 3KE2 3KG4 3KHN 3KKB
3KKC 3KKG 3KTB 3SXL 3TSS 7FD1

LS: 200-300 (308)

1A04 1A27 1A7J 1AGJ 1AJO 1AOA 1ATG 1B35 1B5E 1B5F 1B8X 1BAM 1BDF 1BOO 1C1G 1C1K 1CBY 1CHK 1CQ3 1D1D
1D4A 1DEK 1DML 1DRW 1DZF 1E1H 1EEJ 1EEM 1EQF 1ES9 1EX4 1EYQ 1F0Y 1F97 1FLK 1FNH 1FVI 1FX8 1FXO 1G2O
1G3Q 1G5B 1G66 1GS5 1H21 1H6G 1HI9 1HK7 1I52 1I6A 1IA9 1IG3 1IK9 1IM4 1IMJ 1IN4 1IX9 1IY2 1IZN 1J1L
1J1T 1JAD 1JG1 1JM1 1JMK 1JR2 1JTV 1K0D 1K0F 1K1B 1K7K 1KCM 1KHX 1KI0 1KKE 1KNV 1KNY 1KQP 1KU2 1KXU
1KY9 1KYF 1KYQ 1KZL 1L6N 1L6W 1LC0 1LVM 1M33 1M4O 1M6Y 1MRZ 1NOW 1N26 1N2Z 1N4K 1NE7 1NF2 1NF9 1NH8
1NHG 1NKV 1NLN 1NLT 1NMO 1NN5 1NNW 1NOX 1NR9 1NST 1NUU 1NV8 1O54 1O65 1O7O 1O7D 1O9G 1O9I 1ODF 1OFC
1OI7 1OK3 1OKC 1OTK 1OTV 1OYZ 1OZN 1POH 1P15 1P4K 1P4X 1P88 1PJH 1PJZ 1PQ7 1PU6 1PUJ 1PV5 1PVT 1PYH
1Q1Q 1Q33 1Q6O 1Q74 1QD5 1QFJ 1QGE 1QL0 1QNX 1QOP 1QRE 1QRF 1QTJ 1QUP 1QUU 1QWJ 1QWT 1ROV 1R44 1R8J
1RC9 1RCW 1RID 1RO7 1RVE 1RWI 1RZ2 1S2E 1S4D 1S68 1SFS 1SLQ 1SQS 1SR8 1SSQ 1T15 1T6L 1T71 1TDH 1THF
1TKE 1TKJ 1TLY 1TMK 1TVC 1TVN 1U00 1UO2 1U0K 1U4G 1U5K 1U6K 1U7B 1U7Z 1U9C 1UA2 1UAL 1UB3 1UR3 1URU
1UYJ 1UYL 1V6Z 1VD6 1VHN 1VHT 1VHY 1VK1 1VKY 1VL1 1VPQ 1VQR 1VQV 1VRB 1W3I 1W8G 1W9Y 1WBE 1WCV 1WDE
1WHS 1WLG 1WOJ 1WOQ 1WOV 1WR8 1WXC 1WYZ 1WZN 1X42 1X8B 1XDN 1XDY 1XDZ 1XGS 1XS5 1XT8 1XVI 1XX1 1YO8
1Y44 1Y7W 1YBI 1YE9 1YI9 1YKI 1YO6 1YUM 1YWH 1YZX 1Z6O 1Z7L 1Z8H 1ZAT 1ZEJ 1ZEL 1ZM8 1ZMB 1ZMT 1ZN6
1ZP2 1ZRN 1ZTC 1ZYB 1ZYM 2A01 2A2J 2A2M 2A6H 2AA4 2AAM 2ABW 2AKO 2ARZ 2AYH 2B0O 2B4H 2BHN 2BHW 2BJO
2BJI 2BKA 2BLN 2BOG 2C3Z 2C46 2C4E 2C4I 2C4X 2CL3 2CWS 2CZQ 2D1L 2D2E 2D3Y 2D4O 2D42 2DDM 2DGD 2DHO
2DPL 2DT8 2DYT 2E0N 2E11 2E1B 2E2O 2E3D 2E74 2E7Y 2E8G 2EFK 2EHZ 2EIY 2EQ5 2ESS 2EWS 2F1K 2F6R 2F7B
2FB1 2FBM 2FEA 2FM9 2FNO 2FQ1 2FZ4 2G5C 2G7C 2G8O 2GAU 2GB4 2GFH 2GGS 2GHP 2GJG 2GKE 2GMN 2GNP 2GOY
2GQT 2GUP 2GVK 2HOA 2HAL 2HBW 2HFK 2HFT 2HHC 2HLS 2HLY 2HNL 2HQB 2HQQ 2HS5 2I2C 2I2O 2I53 2I75 2I9D
2IBA 2ICU 2IEE 2IHY 2II3 2IKS 2ILR 2IMF 2IN3 2IOC 2IPB 2IQT 2IUE 2IUM 2IW0 2IZP 2IZR 2J32 2J3S 2J6G
2J6V 2JBM 2JF2 2JKS 2K7N 2KIN 2NLY 2NO4 2NPN 2NSF 2NTE 2NXO 2NZI 2O0Y 2O1Z 2O2G 2O34 2O57 2O5N 2O8L
2O8N 2O8S 2OCF 2OG4 2OK8 2OOS 2OPJ 2OPW 2OTD 2OUX 2OVJ 2OWL 2OWR 2OYC 2P0K 2P2V 2P4Z 2P7I 2PA8 2PBE
2PBK 2PHC 2PIF 2PQG 2PQX 2PS1 2PSB 2PSQ 2PT6 2PVU 2PX6 2QO2 2QOS 2QOT 2Q3M 2Q46 2Q62 2Q7E 2Q7T 2Q8P
2QDJ 2QDX 2QED 2QF3 2QGQ 2QGS 2QH5 2QJC 2QLT 2QOL 2QQ8 2QZC 2R31 2R6I 2R6Z 2R7A 2R83 2R8B 2R8R 2RBC
2RCK 2RCN 2RGZ 2RJQ 2TBV 2UZJ 2V3K 2V43 2V4B 2V9C 2V9P 2VAN 2VCN 2VDF 2VES 2VFX 2VHK 2VK2 2VKV 2VSH
2VTC 2VXB 2VYO 2W0M 2W1N 2W3Q 2W3Z 2W53 2WAM 2WBM 2WDP 2WFG 2WI8 2WLC 2WM3 2WTA 2YV9 2YWR 2YYV 2ZOJ
2Z1P 2Z2N 2Z2R 2Z6R 2ZBT 2ZD7 2ZE7 2ZG1 2ZIH 2ZOS 2ZVR 2ZW5 2ZZJ 3A8P 3AAG 3B5E 3B5O 3B7K 3B8P 3B95
3BAC 3BBA 3BBJ 3BBL 3BEH 3BGY 3BHD 3BHN 3BO6 3BOS 3BRQ 3BU8 3BUS 3BWH 3BWU 3BXP 3BY9 3BZW 3CO2 3COT
3C37 3C3D 3C3W 3C5Y 3C8E 3C8W 3CAZ 3CC8 3CEG 3CG7 3CG8 3CGX 3CIJ 3CIO 3CIS 3CJJ 3CJP 3CNY 3CSS 3CU2
3CVG 3CVZ 3CWQ 3CYG 3D2Y 3D36 3D8B 3D9H 3D9W 3DA8 3DAO 3DBQ 3DBY 3DCD 3DEM 3DEZ 3DF7 3DHO 3DI4 3DKP
3DKQ 3DKW 3DKX 3DLM 3DOA 3DPT 3DQP 3DR2 3DTE 3DV9 3DX5 3DXI 3E0S 3E2O 3E48 3E4C 3E4R 3E5T 3E8S 3E96
3E9O 3EAY 3EBV 3EC3 3ECS 3EDT 3EEG 3EF2 3EFE 3EGA 3EGO 3EIN 3EJA 3EJX 3EO3 3EPR 3ESH 3EUF 3EVF 3EWB
3EWN 3EXM 3EZO 3F6Y 3F79 3F8M 3FBS 3FCH 3FDJ 3FGN 3FIJ 3FOK 3FTJ 3FY3 3FZX 3G1V 3G3K 3G67 3G87 3G89
3G9H 3G9Q 3G9T 3GBV 3GDH 3GEC 3GFO 3GGG 3GL5 3GO5 3GOH 3GQJ 3GS3 3GUW 3GUY 3GV2 3GWC 3GXA 3GZG 3HO4
3HOU 3H1N 3H1Q 3H2S 3H3L 3H4C 3H7J 3HBN 3HKO 3HNN 3HQO 3HQI 3HR9 3HSL 3HWR 3HXK 3I32 3I3Q 3I5Q 3I7A
3I7J 3I9W 3IB7 3ICV 3ICX 3IEE 3IH5 3IJR 3ILK 3INP 3IRS 3IS6 3ISM 3ISR 3IX6 3JX8 3JY6 3K25 3K66 3K8P
3K93 3KCU 3KDW 3KN3 3KRE 3KST 3PRN 5CSM

Anexo 4.- Pruebas de significación estadísticas entre FoDT_DT y FoDT_RT.

Aplicación de la prueba no paramétrica de Friedman para ***k*** muestras relacionadas, con un nivel de significación de 0,05. Fueron incluidas las variables precisión (*Ap*), mejora sobre el predictor aleatorio (*R*), distribución de contactos (*Xd*), sensibilidad (*S*) y media armónica entre precisión y sensibilidad (*F-Measure*).

Rankings promedio de los algoritmos:

Algoritmo	Ranking
FoDT_DT	1,0
FoDT_RT	2,0

Friedman considerando el desempeño de reducción (distribuido acorde a chi-cuadrado con 1 grado de libertad): 4,0.

P-value calculado por el test de Friedman: 0,04550026389837358.

Iman y Davenport considerando el desempeño de reducción (distribuido acorde a F-distribution con 1 y 3 grados de libertad): Infinito.

P-value calculado por el test de Iman and Daveport: 0,0.

Bonferroni-Dunn rechaza aquellas hipótesis que tenga *p-value* ≤ 0,05.

Hochberg rechaza aquellas hipótesis que tenga *p-value* ≤ 0,05.

Holm / Hochberg paraα = 0,05

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm/Hochberg/Hommel
1	FoDT_RT	2,0	0,0455	0,05

Hommel rechaza todas las hipótesis.

Holm / Hochberg para α = 0,10

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm/Hochberg/Hommel
1	FoDT_RT	2,0	0,0455	0,1

Bonferroni-Dunn rechaza aquellas hipótesis que tenga *p-value* ≤ 0,1.

Hochberg rechaza aquellas hipótesis que tenga *p-value* ≤ 0,1.

Hommel rechaza todas las hipótesis.

p-values Ajustado

t	Algoritmo	P sin ajustar	P Bonf	P Holm	P Hoch	P Homm
1	FoDT_RT	0,0455	0,0455	0,0455	0,0455	0,0455

Holm / Shaffer para = 0,05

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm	Shaffer
1	FoDT_DT vs FoDT_RT	2,0	0,0455	0,05	0,05

Nemenyi rechaza aquellas hipótesis que tenga *p-value* ≤ 0,05.

Shaffer rechaza aquellas hipótesis que tenga *p-value* ≤ 0,05.

Bergmann rechaza estas hipótesis:

- FoDT_DT vs. FoDT_RT

Table 6: Holm / Shaffer Table para = 0,10

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm	Shaffer
1	FoDT_DT vs FoDT_RT	2,0	0,0455	0,1	0,1

Nemenyi rechaza aquellas hipótesis que tenga *p-value* ≤ 0,1.

Shaffer rechaza aquellas hipótesis que tenga *p-value* ≤ 0,1.

Bergmann rechaza esas hipótesis:

- FoDT_DT vs. FoDT_RT

p-values ajustados

t	Algoritmo	P sin ajustar	P Bonf	P Holm	P Hoch	P Homm
1	FoDT_DT vs FoDT_RT	0,0455	0,0455	0,0455	0,0455	0,0455

Anexo 5.- Conjunto de validación externa: 174 proteínas del CASP9.

T0515	T0542_1	T0568	T0596	T0618
T0515_1	T0542_2	T0569	T0597	T0619
T0515_2	T0544	T0570	T0597_1	T0620
T0516	T0544_1	T0571	T0597_2	T0621
T0518	T0544_2	T0571_1	T0598	T0622
T0520	T0545	T0571_2	T0599	T0623
T0521	T0547	T0572	T0600	T0624
T0521_1	T0547_1	T0574	T0600_1	T0625
T0521_2	T0547_2	T0575	T0600_2	T0626
T0522	T0547_3	T0575_1	T0602	T0626_1
T0523	T0547_4	T0575_2	T0603	T0626_2
T0524	T0548	T0576	T0603_1	T0627
T0525	T0549	T0578	T0603_2	T0628
T0526	T0550	T0579	T0604	T0628_1
T0527	T0550_1	T0579_1	T0604_1	T0628_2
T0528	T0550_2	T0579_2	T0604_2	T0628_3
T0528_1	T0551	T0580	T0604_3	T0629
T0528_2	T0552	T0581	T0605	T0629_1
T0530	T0553	T0582	T0606	T0629_2
T0531	T0555	T0582_1	T0607	T0630
T0532	T0556	T0582_2	T0607_1	T0632
T0533	T0557	T0584	T0607_2	T0634
T0533_1	T0558	T0585	T0608	T0635
T0533_2	T0559	T0586	T0608_1	T0636
T0533_3	T0560	T0588	T0608_2	T0636_1
T0534	T0561	T0589	T0609	T0636_2
T0535	T0562	T0589_1	T0610	T0637
T0535_1	T0563	T0589_2	T0611	T0638
T0535_2	T0564	T0590	T0612	T0639
T0536	T0565	T0591	T0613	T0640
T0537	T0565_1	T0591_1	T0613_1	T0641
T0538	T0565_2	T0591_2	T0613_2	T0641_1
T0539	T0565_3	T0592	T0615	T0641_2
T0541	T0566	T0593	T0616	T0643
T0542	T0567	T0594	T0617	

Anexo 6.- Conjunto de validación externa: 123 proteínas del CASP10.

T0644-D1	T0671-D1	T0689-D1	T0712-D1	T0733-D1
T0645-D1	T0671-D2	T0690-D1	T0713-D1	T0734-D1
T0648-D1	T0672-D1	T0690-D2	T0713-D2	T0735-D1
T0649-D1	T0673-D1	T0691-D1	T0714-D1	T0735-D2
T0650-D1	T0674-D1	T0692-D1	T0715-D1	T0736-D1
T0651-D1	T0674-D2	T0693-D1	T0716-D1	T0737-D1
T0651-D2	T0675-D1	T0693-D2	T0717-D1	T0738-D1
T0652-D1	T0675-D2	T0694-D1	T0717-D2	T0740-D1
T0652-D2	T0676-D1	T0696-D1	T0719-D1	T0741-D1
T0653-D1	T0677-D1	T0697-D1	T0719-D2	T0742-D1
T0654-D1	T0677-D2	T0698-D1	T0719-D3	T0743-D1
T0655-D1	T0678-D1	T0699-D1	T0719-D4	T0744-D1
T0657-D1	T0679-D1	T0700-D1	T0719-D5	T0746-D1
T0658-D1	T0680-D1	T0701-D1	T0719-D6	T0747-D9
T0658-D2	T0681-D1	T0702-D1	T0720-D1	T0749-D1
T0659-D1	T0682-D1	T0703-D1	T0721-D1	T0750-D1
T0661-D1	T0683-D1	T0704-D1	T0723-D1	T0752-D1
T0662-D1	T0684-D1	T0705-D1	T0724-D1	T0753-D1
T0663-D1	T0684-D2	T0705-D2	T0724-D2	T0755-D1
T0663-D2	T0685-D1	T0706-D9	T0726-D1	T0756-D1
T0664-D1	T0685-D2	T0707-D1	T0726-D2	T0756-D2
T0666-D1	T0686-D1	T0708-D1	T0726-D3	T0757-D1
T0667-D1	T0686-D2	T0709-D1	T0731-D1	T0758-D1
T0668-D1	T0687-D1	T0710-D1	T0732-D1	
T0669-D1	T0688-D1	T0711-D1	T0732-D2	

Anexo 7.- Pruebas de significación estadísticas: validación externa con CASP9.

Aplicación de la prueba no paramétrica de Friedman para ***k*** muestras relacionadas, con un nivel de significación de 0,05. Fueron incluidas las variables efectividad (*Acc*), cobertura (Cov) y media armónica entre ambas medidas (F-Measure), para los L/5 y L/10 mejores pares predichos.

Rankings promedio de los algoritmos:

Algoritmo	Ranking
LRcon	1,67
FoDT	2,00
SVM-SEQ	2,75
NNcon	3,58
FragHMMent	5,00

Friedman considerando el desempeño de reducción (distribuido acorde a chi-cuadrado con 4 grados de libertad): 17.23333333333332.

P-*value* calculado por el test de Friedman: 0.0017412205342763887.

Iman y Davenport considerando el desempeño de reducción (distribuido acorde a F-distribution con 4 y 20 grados de libertad): 12.733990147783217.

P-value calculado por el test de Iman and Daveport: 2.5966786859065032E-5.

Holm / Hochberg paraα = 0,05

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm/Hochberg/Hommel
4	FragHMMent	3,6514	2,6072	0,0125
3	NNcon	2,0996	0,0358	0,0167
2	SVM-SEG	1,1867	0,2353	0,0250
1	FoDT	0,3651	0,7150	0,0500

Bonferroni-Dunn rechaza aquellas hipótesis que tenga *p-value* ≤ 0.0125.

Holm rechaza aquellas hipótesis que tenga *p-value* ≤ 0.016666666666666666.

Hochberg rechaza aquellas hipótesis que tenga *p-value* ≤ 0.0125.

Hommel rechaza aquellas hipótesis que tenga *p-value* ≤ 0.016666666666666666.

Holm / Hochberg para α = 0,10

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm/Hochberg/Hommel
4	FragHMMent	3,6514	2,6072	0,0250
3	NNcon	2,0996	0,0358	0,0333
2	SVM-SEG	1,1867	0,2353	0,0500
1	FoDT	0,3651	0,7150	0,1000

Bonferroni-Dunn rechaza aquellas hipótesis que tenga *p-value* ≤ 0.025.

Holm rechaza aquellas hipótesis que tenga *p-value* ≤ 0.03333333333333333.

Hochberg rechaza aquellas hipótesis que tenga *p-value* ≤ 0.025.

Hommel rechaza aquellas hipótesis que tenga *p-value* ≤ 0.016666666666666666.

p-values ajustados

t	Algoritmo	P sin ajustar	P Bonf	P Holm	P Hoch	P Homm
1	FragHMMent	2,6072	0,0010	0,0010	0,0010	0,0010
2	NNcon	0,0358	0,1430	0,1072	0,1072	0,1072
3	SVM-SEG	0,2353	0,9433	0,4707	0,4707	0,4707
4	FoDT	0,7150	2,8600	0,7150	0,7150	0,7150

Holm / Shaffer para = 0,05

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm	Shaffer
10	FragHMMent vs LRcon	3,6515	2,6073	0,0050	0,0050
9	FragHMMent vs FoDT	3,2863	0,0010	0,0063	0,0084
8	FragHMMent vs SVM-SEG	2,4648	0,0137	0,0056	0,0084
7	NNcon vs LRcon	2,0996	0,0358	0,0071	0,0084
6	NNcon vs FoDT	1,7345	0,0828	0,0083	0,0084
5	FragHMMent vs NNcon	1,5519	0,1206	0,0100	0,0100
4	SVM-SEG vs LRcon	1,1867	0,2353	0,0125	0,0125
3	NNcon vs SVM-SEG	0,9129	0,3613	0,0167	0,0167
2	SVM-SEG vs FoDT	0,8216	0,4113	0,0250	0,0250
1	LRcon vs FoDT	0,3651	0,7150	0,0500	0,0500

Nemenyi rechaza aquellas hipótesis que tenga *p-value* ≤ 0.005.

Holm rechaza aquellas hipótesis que tenga *p-value* ≤ 0.00625.

Shaffer rechaza aquellas hipótesis que tenga *p-value* ≤ 0.005.

Bergmann rechaza estas hipótesis:

- FragHMMent vs. LRcon
- FragHMMent vs. FoDT

Holm / Shaffer para = 0,10

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm	Shaffer
10	FragHMMent vs LRcon	3,6515	2,6073	0,0100	0,0100
9	FragHMMent vs FoDT	3,2863	0,0010	0,0111	0,0167
8	FragHMMent vs SVM-SEG	2,4648	0,0137	0,0125	0,0167
7	NNcon vs LRcon	2,0996	0,0358	0,0143	0,0167
6	NNcon vs FoDT	1,7345	0,0828	0,0167	0,0167
5	FragHMMent vs NNcon	1,5519	0,1206	0,0200	0,0200
4	SVM-SEG vs LRcon	1,1867	0,2353	0,0250	0,0250
3	NNcon vs SVM-SEG	0,9129	0,3613	0,0333	0,0333
2	SVM-SEG vs FoDT	0,8216	0,4113	0,0500	0,0500
1	LRcon vs FoDT	0,3651	0,7150	0,1000	0,1000

Nemenyi rechaza aquellas hipótesis que tenga *p-value* ≤ 0.01.

Holm rechaza aquellas hipótesis que tenga *p-value* ≤ 0.0125.

Shaffer rechaza aquellas hipótesis que tenga *p-value* ≤ 0.01.

Bergmann rechaza estas hipótesis:

- FragHMMent vs. SVM-SEQ
- FragHMMent vs. LRcon
- FragHMMent vs. FoDT

p-values ajustados

t	Algoritmo	$Z=(R_0 - R_1)/S_E$	p	Holm	Shaffer
1	FragHMMent vs LRcon	2,6073	0,0026	0,0026	0,0026
2	FragHMMent vs FoDT	0,0010	0,0091	0,0061	0,0061
3	FragHMMent vs SVM-SEG	0,0137	0,1097	0,0823	0,0548
4	NNcon vs LRcon	0,0358	0,2503	0,2146	0,2145
5	NNcon vs FoDT	0,0828	0,4970	0,4902	0,2485
6	FragHMMent vs NNcon	0,1206	0,6035	0,4902	0,4828
7	SVM-SEG vs LRcon	0,2353	0,9413	0,9420	0,7060
8	NNcon vs SVM-SEG	0,3613	1,0839	1,0839	0,7060
9	SVM-SEG vs FoDT	0,4113	1,0839	1,0839	0,7060
10	LRcon vs FoDT	0,7150	1,0839	1,0839	0,7060

Anexo 8.- Resultados del CASP10.

Ordenamiento de los resultados de los algoritmos del CASP10, incluyendo el algoritmo propuesto FoDT, tomando en consideración los valores promedios para la efectividad (Acc) y la calidad de los predictores, basado en la dispersión de las predicciones (Z-Score) de los contactos L/5 de largo alcance.

#	Métodos	Corto alcance (6-12)						Alcance medio (12-24)						Largo alcance (+24)					
		Top5		L/10		L/5		Top5		L/10		L/5		Top5		L/10		L/5	
		Acc	Z	Acc	Z	Acc	Z	Acc	Z	Acc	Z	Acc	Z	Acc	Z	Acc	Z	Acc	Z
1	IGBteam **G305**	37,50	0,59	29,77	0,49	21,47	0,34	20,54	0,67	20,92	0,63	20,54	0,67	16,25	0,34	17,67	0,39	18,02	0,68
2	Distill_roll **G087**	26,25	0,19	22,14	0,14	19,28	0,15	15,12	0,24	18,19	0,33	15,12	0,24	16,25	0,33	15,57	0,40	16,42	0,67
3	SAM-T08-server **G113**	7,69	0,02	8,85	0,00	8,56	0,00	15,52	0,44	14,90	0,42	15,52	0,44	16,67	0,55	13,86	0,38	15,33	0,66
4	RaptorX-Roll **G358**	36,00	0,57	29,21	0,57	18,95	0,32	15,04	0,31	21,10	0,54	15,04	0,31	16,00	0,55	14,61	0,65	14,91	0,63
5	**FoDT**	**26,00**	**0,70**	**20,20**	**0,69**	**13,23**	**0,72**	**11,22**	**0,55**	**13,90**	**0,69**	**13,44**	**0,77**	**8,00**	**0,37**	**14,13**	**0,55**	**8,33**	**0,67**
6	MULTICOM **G489**	45,33	0,76	38,01	1,01	27,29	0,80	25,05	0,92	27,39	0,75	25,05	0,92	20,00	0,62	18,25	0,77	16,94	0,58
7	ZHOU-SPARKS-X **G413**	0,00	0,00	8,85	0,07	7,88	0,02	10,89	0,09	9,86	0,06	10,89	0,09	13,33	0,34	14,90	0,43	12,26	0,62
8	MULTICOM-CONSTRUCT **G222**	43,75	0,60	36,20	0,75	31,43	1,16	21,16	0,46	21,82	0,48	21,16	0,46	26,25	0,81	23,44	0,68	19,15	0,54
9	ProC_S4 **G314**	42,50	0,68	33,46	0,77	22,53	0,36	16,59	0,44	20,11	0,47	16,59	0,44	20,00	0,67	17,24	0,50	16,71	0,55
10	MULTICOM-REFINE **G125**	32,50	0,39	27,71	0,45	24,65	0,59	19,49	0,21	19,97	0,38	19,49	0,21	26,25	0,55	24,66	0,52	19,92	0,48
11	MULTICOM-NOVEL **G424**	36,25	0,40	27,05	0,32	27,88	0,66	19,27	0,23	20,37	0,42	19,27	0,23	28,75	0,71	26,35	0,59	19,12	0,46
12	ICOS **G184**	31,25	0,46	22,64	0,29	20,92	0,30	14,98	0,15	13,59	0,09	14,98	0,15	17,33	0,46	18,63	0,41	16,16	0,37
13	ProC_S5 **G396**	40,00	0,60	28,97	0,57	23,61	0,41	16,11	0,50	17,63	0,36	16,11	0,50	16,25	0,25	19,85	0,40	15,93	0,34
14	CONSIP **G139**	28,00	0,50	18,41	0,20	19,27	0,35	18,05	0,50	20,81	0,57	18,05	0,50	18,00	0,53	12,65	0,36	11,71	0,30
15	PLCT **G332**	27,50	0,23	23,28	0,33	19,60	0,28	14,82	0,43	17,78	0,46	14,82	0,43	10,00	0,22	13,30	0,44	10,46	0,28
16	RBO-CON **G334**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	8,57	0,32	10,38	0,28	8,93	0,22
17	samcha-server **G112**	43,08	0,66	30,07	0,40	21,39	0,29	19,69	0,46	23,70	0,66	19,69	0,46	12,31	0,41	12,62	0,20	13,34	0,17
18	ProC_S3 **G257**	40,00	0,60	28,97	0,57	23,61	0,41	16,11	0,50	17,63	0,36	16,11	0,50	16,25	0,26	14,71	0,21	11,77	0,17
19	Distill **G072**	27,50	0,18	23,71	0,23	21,00	0,39	12,04	0,24	12,40	0,30	12,04	0,24	11,25	0,14	13,10	0,20	0,17	0,17
20	MULTICOM-CLUSTER **G081**	28,75	0,30	23,71	0,29	21,47	0,29	19,53	0,47	18,26	0,37	19,53	0,47	6,67	0,09	7,34	0,08	7,80	0,07
21	Yang-test **G180**	8,33	0,02	1,59	0,00	7,22	0,02	8,39	0,19	10,19	0,20	8,39	0,19	3,64	0,07	6,74	0,16	4,39	0,09
22	SAM-T06-server **G381**	11,67	0,18	12,52	0,05	7,61	0,00	15,06	0,25	15,88	0,25	15,06	0,25	16,67	0,53	10,88	0,37	11,77	0,00
23	FLOUDAS **G077**	10,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	18,75	0,44	0,00	0,00	4,00	0,11	5,72	0,12	3,18	0,00
24	CNIO **G475**	5,00	0,00	8,32	0,00	0,00	0,00	0,00	0,00	10,00	0,00	0,00	0,00	6,67	0,20	7,24	0,32	1,39	0,00
25	confuzzGS **G098**	0,00	0,00	2,38	0,00	6,98	0,00	14,15	0,00	10,71	0,00	14,15	0,00	0,00	0,00	1,23	0,00	1,19	0,00
26	confuzz3d **G462**	0,00	0,00	0,00	0,00	4,65	0,00	11,63	0,00	4,76	0,00	11,63	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Z: valores de Z-Score.