

Facultad de Medicina y Ciencias de la Salud

Comparativa de la precisión en la predicción de estructuras de macromoléculas biológicas a partir de secuencias de aminoácidos

Blanca Pueche Granados

Tutores:

Universidad Nebrija: Diego Mañanes Cayero

CNB-CSIC: Carlos Oscar S. Sorzano

Trabajo de Fin de Máster

Máster Universitario en Bioinformática



UNIVERSIDAD
NEBRIJA

Resumen

La predicción de la estructura terciaria de proteínas mediante inteligencia artificial ha revolucionado la biología computacional. Sin embargo, el rápido desarrollo de herramientas como AlphaFold3, Chai-1, Boltz-2, Protenix e IntelliFold ha fragmentado el ecosistema de *software*, generando problemas de interoperabilidad, dependencias complejas y dificultades en la reproducibilidad de los flujos de trabajo.

Para resolver esta limitación, se ha desarrollado e implementado Biofold (scipion-chem-biofold), un nuevo *plugin* integrado en el *framework* Scipion. El *plugin* ofrece una arquitectura modular basada en protocolos de Scipion que automatiza e integra tanto la importación de servidores externos como la ejecución local de las herramientas, abstrayendo la complejidad de la línea de comandos.

La plataforma se validó mediante tres casos de uso con niveles de complejidad biológica incremental (9M1Z, 9NZ9 y 9QS2). El análisis comparativo, que integra criterios de precisión geométrica y eficiencia computacional, demuestra la robustez de Biofold para estandarizar las salidas estructurales y conectarlas con otras herramientas. El análisis evidenció que las divergencias de RMSD localizadas en las regiones terminales no constituyen fallos algorítmicos de los modelos, sino que reflejan la falta de densidad electrónica y coordenadas en los archivos experimentales del PDB. En conclusión, Biofold se consolida como una herramienta reproducible que democratiza y unifica el uso de predictores estructurales.

Palabras clave: Predicción estructural de proteínas, Deep Learning, Scipion, Validación estructural, Pipelines bioinformáticos.

Índice general

Glosario	VIII
1. Introducción	1
1.1. Modelado estructural de proteínas	1
1.2. Scipion como framework	3
1.3. Motivación	4
1.4. Objetivos	4
2. Materiales y métodos	6
2.1. Entorno de trabajo y recursos computacionales	6
2.1.1. Sistema operativo	6
2.1.2. Hardware	6
2.1.3. Entorno de desarrollo y herramientas de soporte	6
2.1.4. Dependencias generales	7
2.2. Framework de Scipion	7
2.3. Herramientas de predicción estructural	9
2.3.1. AlphaFold3	9
2.3.2. Chai-1	11
2.3.3. Protenix	12
2.3.4. Boltz-2	13
2.3.5. IntelliFold	14
2.4. Diseño e implementación del plugin	15
2.4.1. Arquitectura del plugin	15
2.4.2. Protocolo de importación	16
2.4.3. Protocolos de ejecución local	17
Chai-1	18
Protenix	19
Boltz-2	20
IntelliFold	20
2.4.4. Viewers	21

2.4.5. Tests	21
2.5. Casos de uso	22
2.5.1. Estructura del proyecto	23
2.5.2. Configuración de parámetros de ejecución	24
3. Resultados	25
3.1. Implementación del plugin	25
3.1.1. Interfaz de usuario	25
3.1.2. Manejo de objetos	26
3.2. Funcionamiento de protocolos	26
3.3. Casos de uso	27
3.3.1. 9M1Z	27
3.3.2. 9NZ9	31
3.3.3. 9QS2	34
3.4. Análisis y observaciones	36
3.5. Comparativa cuantitativa global de herramientas	37
4. Discusión y perspectivas futuras	40
4.1. Alcance metodológico y evaluación de las capacidades predictivas	40
4.2. Ventajas de la integración en Scipion-chem	41
4.3. Limitaciones actuales	42
4.4. Posibles líneas futuras	42
5. Conclusiones	43
6. Bibliografía	44
7. Anexo A: Repositorios	49
7.1. Código fuente de scipion-chem-biofold	49
7.2. Proyecto de análisis	49
8. Anexo B: Análisis de casos de uso	50
8.1. 9M1Z	50
8.1.1. AlphaFold3	50
8.1.2. Chai-1	50

8.1.3.	Protenix	52
8.1.4.	Boltz-2	53
8.1.5.	IntelliFold	54
8.2.	9NZ9	56
8.2.1.	AlphaFold3	56
8.2.2.	Chai-1	57
8.2.3.	Protenix	58
8.2.4.	Boltz-2	59
8.2.5.	IntelliFold	61
8.3.	9QS2	62
8.3.1.	AlphaFold3	62
8.3.2.	Chai-1	63
8.3.3.	Protenix	65
8.3.4.	Boltz-2	66
8.3.5.	IntelliFold	67

Índice de figuras

1.	Línea temporal con métodos de predicción estructural de proteínas. Elaboración propia basada en la literatura [4].	3
2.	Estructura de extensiones y <i>plugins</i> de Scipion.	8
3.	Interfaz de usuario de Scipion.	8
4.	Arquitectura de AlphaFold3 [27].	10
5.	Arquitectura de Chai-1 [14].	11
6.	Arquitectura de Boltz-2 [13].	13
7.	Arquitectura de IntelliFold [16].	14
8.	Estructura de scipion-chem-biofold.	16
9.	Clasificación de protocolos de Scipion-chem.	17
10.	Estructuras de referencia para el análisis (A: 9M1Z, B: 9NZ9, C: 9QS2).	22
11.	Estructura del análisis para 9M1Z.	23
12.	Árbol de protocolos y formulario de IntelliFold.	25
13.	Objetos entrada y salida de protocolo Protenix.	27
14.	Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9M1Z.	28
15.	Diferencia estructural de los modelos de AlphaFold3 frente a estructura de referencia para 9M1Z.	29
16.	Diferencias en el RMSD por aminoácido de modelos de AlphaFold3 frente a estructura de referencia para 9M1Z.	30
17.	Diferencia estructural de los mejores modelos de 9M1Z frente a estructura de referencia.	30
18.	Diferencias en el RMSD por aminoácido de modelos de 9M1Z frente a estructura de referencia.	31
19.	Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9NZ9.	32
20.	Diferencia estructural de los modelos de 9NZ9 frente a estructura de referencia.	33

21.	Diferencias en el RMSD por aminoácido de modelos de 9NZ9 frente a estructura de referencia.	34
22.	Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9QS2.	35
23.	Diferencia estructural de los modelos de 9QS2 frente a estructura de referencia.	35
24.	Diferencias en el RMSD por aminoácido de modelos de 9QS2 frente a estructura de referencia.	36
25.	Visualización de superposición múltiple para 9M1Z. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores	38
26.	Visualización de superposición múltiple para 9NZ9. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores	39
27.	Visualización de superposición múltiple para 9QS2. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores	39

Índice de tablas

1.	Métricas de AlphaFold3 [28].	10
2.	Parámetros de configuración de Chai-1.	19
3.	Parámetros de configuración de Protenix.	19
4.	Parámetros de configuración de Boltz-2.	20
5.	Parámetros de configuración de IntelliFold.	21
6.	Puntuación de cada herramienta para las estructuras predichas de 9M1Z.	28
7.	Puntuación de cada herramienta para las estructuras predichas de 9NZ9.	32
8.	Puntuación de cada herramienta para las estructuras predichas de 9QS2.	34
9.	Comparativa cuantitativa global de rendimiento y precisión estructural de las herramientas en los tres casos de uso.	38

Glosario

Abreviatura / Acrónimo	Término Completo	Descripción
ADN	Ácido Desoxirribonucleico	Macromolécula biológica que contiene las instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos y algunos virus, estructurada habitualmente en forma de doble hélice.
ARN	Ácido Ribonucleico	Ácido nucleico formado por una cadena de ribonucleótidos, esencial para la transferencia del código genético, la síntesis de proteínas y la regulación de la expresión génica.
CASP	<i>Critical Assessment of Protein Structure Prediction</i> (Evaluación Crítica de la Predicción de la Estructura de Proteínas)	Experimento y competición comunitaria a nivel mundial que evalúa de forma ciega y rigurosa el estado del arte en los métodos de predicción de la estructura de proteínas.
CPU	<i>Central Processing Unit</i> (Unidad Central de Procesamiento)	Componente principal de un ordenador que interpreta las instrucciones y procesa los datos de los programas informáticos.
Cryo-EM	<i>Cryo-Electron Microscopy</i> (Criomicroscopía Electrónica)	Técnica avanzada de microscopía electrónica donde las muestras biológicas se analizan a temperaturas criogénicas, permitiendo la determinación de estructuras macromoleculares a resolución atómica en su estado nativo.

Abreviatura / Acrónimo	Término Completo	Descripción
DockQ	<i>Docking Quality Score</i> (Puntuación de Calidad de Acoplamiento)	Métrica de calidad estandarizada utilizada en biología computacional para evaluar y puntuar la precisión geométrica del acoplamiento en complejos proteína-proteína.
FASTA	Formato FASTA	Formato de archivo de texto plano utilizado para representar secuencias de ácidos nucleicos o de aminoácidos, caracterizado por una línea de cabecera con el símbolo '>' seguido de la secuencia de estudio.
GPU	<i>Graphics Processing Unit</i> (Unidad de Procesamiento Gráfico)	Coprocesador especializado en el cálculo en paralelo. En bioinformática estructural, es un componente crítico indispensable para acelerar la inferencia matemática de los modelos de inteligencia artificial.
HPC	<i>High-Performance Computing</i> (Computación de Alto Rendimiento)	Uso de supercomputadores y clústeres de computación en paralelo para resolver problemas de cálculo intensivo, permitiendo la ejecución masiva de flujos de trabajo.
IA	Inteligencia Artificial	Campo de la informática centrado en el desarrollo de sistemas algorítmicos capaces de realizar tareas complejas, tales como el aprendizaje automático y el análisis predictivo estructural.
IDE	<i>Integrated Development Environment</i> (Entorno de Desarrollo Integrado)	Aplicación informática que proporciona servicios integrales (editor de código, depurador y herramientas de automatización de compilación) para facilitar el desarrollo de software a los programadores.

Abreviatura / Acrónimo	Término Completo	Descripción
JSON	<i>JavaScript Object Notation</i> (Notación de Objetos de JavaScript)	Formato de intercambio de datos ligero, estructurado y legible por humanos, utilizado en el <i>plugin</i> para la configuración de entradas y preparación de parámetros en los protocolos locales.
LVM	<i>Logical Volume Manager</i> (Gestor de Volúmenes Lógicos)	Mecanismo de abstracción del almacenamiento en Linux que permite gestionar de forma flexible y dinámica el espacio en disco combinando múltiples volúmenes físicos.
mmCIF	<i>Macromolecular Crystallographic Information File</i> (Archivo de Información Cristalográfica Macromolecular)	Formato de archivo de datos altamente estructurado adoptado por herramientas modernas (como AlphaFold3 o Chai-1) para almacenar de forma recursiva y fidedigna las coordenadas atómicas tridimensionales.
MSA	<i>Multiple Sequence Alignment</i> (Alineamiento Múltiple de Secuencias)	Alineamiento de tres o más secuencias biológicas (ADN, ARN o proteínas) de longitud similar para identificar regiones homólogas y relaciones evolutivas, base fundamental de la predicción estructural moderna.
NMR	<i>Nuclear Magnetic Resonance</i> (Resonancia Nuclear Magnética)	Técnica biofísica basada en las propiedades magnéticas de los núcleos atómicos, utilizada en biología estructural para determinar la estructura tridimensional y la dinámica de macromoléculas en disolución.

Abreviatura / Acrónimo	Término Completo	Descripción
PAE	<i>Predicted Aligned Error</i> (Error Alineado Predicho)	Métrica de confianza generada por modelos de difusión que estima el error de distancia esperado entre pares de residuos, crucial para estudiar la fiabilidad de la estructura y la posición de dominios o cadenas.
PDB	<i>Protein Data Bank</i> (Banco de Datos de Proteínas)	Archivo global de acceso abierto que recopila y custodia las estructuras tridimensionales de proteínas, ácidos nucleicos y complejos macromoleculares complejos determinadas experimentalmente.
PSP	<i>Protein Structure Prediction</i> (Predicción de la Estructura de Proteínas)	Campo de la bioinformática centrado en el cálculo algorítmico y computacional de la estructura tridimensional de una proteína partiendo exclusivamente de su secuencia primaria de aminoácidos.
RMSD	<i>Root Mean Square Deviation</i> (Desviación Media Cuadrática)	Medida matemática empleada en biología estructural para calcular la distancia promedio entre los átomos (habitualmente los carbonos alfa) de dos estructuras superpuestas, cuantificando su grado de similitud geométrica.
SO	Sistema Operativo	Capa de software base que gestiona directamente los recursos físicos de hardware y proporciona servicios esenciales para la ejecución del entorno de desarrollo y dependencias científicas.

Abreviatura / Acrónimo	Término Completo	Descripción
VRAM	<i>Video Random Access Memory</i> (Memoria de Acceso Aleatorio de Vídeo)	Memoria de acceso aleatorio de velocidad ultra-alta integrada en las tarjetas gráficas (GPU), indispensable para contener la carga matemática de los modelos de aprendizaje profundo y evitar desbordamientos de memoria durante la predicción.
YAML	<i>YAML Ain't Markup Language</i>	Formato de serialización de datos legible por humanos, utilizado para la configuración de parámetros y definición de entradas en herramientas bioinformáticas, como los protocolos de predicción estructural.

1 Introducción

1.1. Modelado estructural de proteínas

Las proteínas son macromoléculas esenciales para la vida y sus funciones son variadas, desde estructurales, hasta enzimáticas, además de ser moléculas esenciales en la regulación de procesos y redes biológicas [1]. Estas funciones vienen establecidas por la estructura terciaria, que varía entre proteínas, y depende de la secuencia de aminoácidos que la constituyen.

Cómo resolver el plegamiento de proteínas a partir de su secuencia es uno de los mayores desafíos de la biología moderna [2]. Desde el punto de vista biofísico, este fenómeno es muy complejo debido a la paradoja de Levinthal, que dice que si una proteína buscara su conformación correcta mediante un muestreo aleatorio de todas las posibles combinaciones, tardaría un tiempo superior a la edad del universo [3]. Sin embargo, en los sistemas biológicos, las proteínas se pliegan en cuestión de milisegundos. Comprender este mapa de energías y ser capaces de predecir estas estructuras partiendo únicamente de la secuencia primaria es el núcleo de estudio del PSP [2, 4].

Para obtener la estructura terciaria se han desarrollado técnicas experimentales, como la cristalografía de rayos X, la espectroscopia por NMR o la cryo-EM [5–7]. Todas estas técnicas permiten obtener la estructura terciaria de proteínas con alta resolución, sin embargo, conllevan costes tanto económicos como temporales, el análisis posterior puede resultar tedioso, y no siempre es aplicable con tipos específicos de proteínas, por lo que no se pueden usar de forma generalizada. La predicción de dicha estructura a partir de la secuencia de aminoácidos evita tener esas limitaciones, pudiendo tener modelos proteicos de forma rápida, escalable y con bajo coste computacional.

Tradicionalmente, los modelos de PSP se dividen en dos grupos, *template-based modeling* y *free modeling*, dependiendo de si usan plantillas o no. Dentro de estos grupos se pueden distinguir además en métodos *ab initio*, por reconocimiento de plegamiento (*threading*) y por homología [4]. Los modelos *ab initio* pueden obtener la estructura de proteínas no conocidas sin necesidad de una plantilla estructural. Una de las herramientas más conocidas dentro de esta metodología es QUARK, que divide la proteína en fragmentos de 20 aminoácidos, busca estructuras en bases de datos y utiliza Monte-Carlo para unir los fragmentos teniendo en cuenta energías y campos de fuerza [8]. Los métodos basados en *threading* asumen que la estructura

terciaria está más conservada que la secuencia de aminoácidos, y que tiene un número finito de conformaciones posibles. Este método utiliza una matriz de puntuaciones para elegir una estructura 3D de pliegues conocidos que encaja con la secuencia de estudio [4]. Un ejemplo es GenTHREADER, que evalúa alineamientos y posteriormente genera los modelos evaluados por redes neuronales para darles puntuaciones de confianza [9]. Por último, los métodos basados en homología se basan en la idea de que secuencias de aminoácidos similares tienen estructuras similares, por lo que, si se conoce la estructura de una secuencia, se puede predecir la de otra similar [4]. De esta forma, se alinean secuencias y se empieza a modelar la estructura terciaria, primero teniendo en cuenta el esqueleto de carbono, y posteriormente los *loops* y las cadenas laterales, finalmente se minimiza la estructura y se evalúa mediante gráficos de Ramachandran. SWISS-MODEL es una herramienta que emplea esta metodología para obtener estructuras terciarias de proteínas mediante su interfaz web [10].

Sin embargo, estos enfoques tradicionales de modelado estructural presentan limitaciones cuando las proteínas de estudio no poseen homólogos detectables o presentan baja similitud de secuencia. La incorporación de técnicas de aprendizaje ha supuesto un cambio radical en las últimas décadas, especialmente con la introducción de AlphaFold2 de DeepMind, que participó en CASP14 obteniendo resultados muy prometedores en el campo de la biología computacional, incluso doblando la puntuación obtenida por otras herramientas participantes en la competición [4, 11, 12].

A partir de este momento, el desarrollo de modelos de predicción estructural ha avanzado rápidamente, desarrollándose nuevas herramientas que van más allá de monómeros, incluyendo complejos proteicos, o interacciones proteína-ligando o proteína-ácidos nucleicos. Entre estos métodos destacan AlphaFold3, así como alternativas recientes como Boltz-2, Chai-1, Protenix o IntelliFold [11, 13–16]. La Figura 1 muestra una línea temporal de los métodos de predicción estructural mencionados desde 1969 hasta la actualidad.

Sin embargo, el éxito y el rápido desarrollo de esta tecnología han trasladado el desafío del plano teórico al operativo. Actualmente, la expansión constante de estas herramientas basadas en IA ha dado lugar a un ecosistema de *software* altamente fragmentado. Cada modelo presenta formatos de entrada y salida distintos, requisitos de *hardware* específicos, dependencias de código complejas y parámetros divergentes, ya sea en servidores públicos o en entornos locales. Estas diferencias no solo dificultan la reproducibilidad de los datos y la integración de múltiples modelos en un mismo trabajo, sino que plantea un problema para la comunidad científica; la

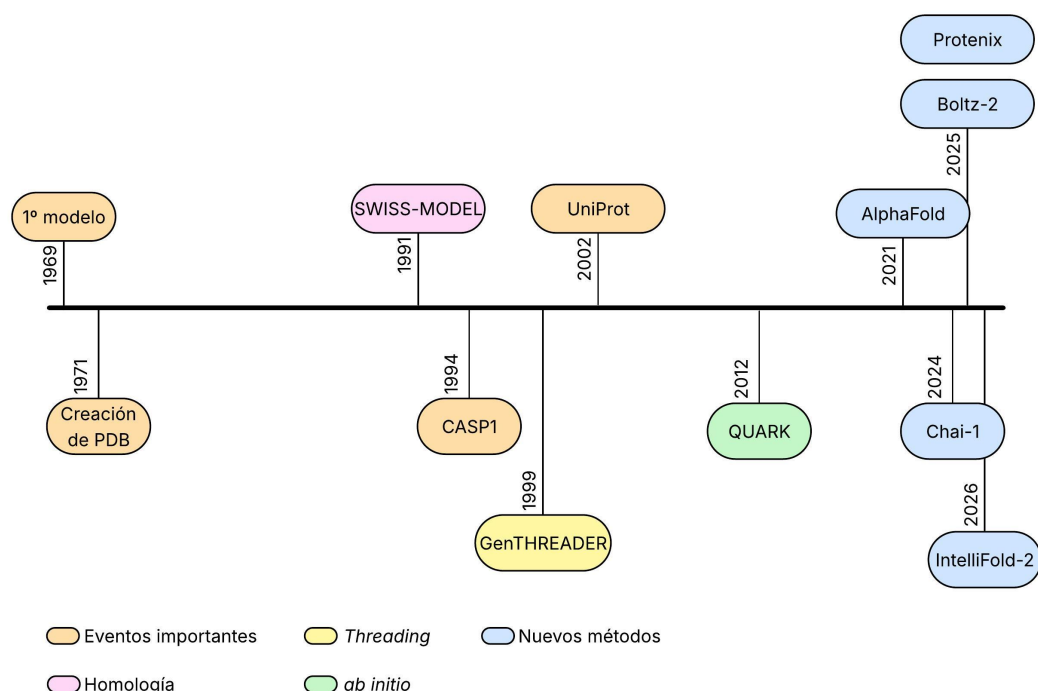


Figura 1: Línea temporal con métodos de predicción estructural de proteínas. Elaboración propia basada en la literatura [4].

dificultad de evaluar, ejecutar y diferenciar qué herramienta ofrece mejor rendimiento ante casos específicos.

Para resolver esta dispersión y permitir una evaluación estandarizada bajo las mismas condiciones, se necesitan plataformas que faciliten la unificación de ejecuciones, la gestión de recursos computacionales y la integración de dichas herramientas dentro de entornos modulares y reproducibles.

1.2. Scipion como framework

Scipion es un *framework* de integración y gestión de flujos orientado al procesamiento de datos en biología estructural, principalmente en la microscopía electrónica de partículas individuales [17]. Su objetivo principal es proporcionar un entorno que permita el uso de distintas herramientas de procesamiento bajo una misma interfaz, creando *workflows* reproducibles, trazables y automatizados. En Scipion, cada *software* se ejecuta de forma independiente, por lo que se pueden construir *pipelines* completos con distintas herramientas de forma estructurada. Esto se logra con una arquitectura basada en protocolos que engloban herramientas externas, estandarizan la entrada y la salida de datos, y permiten su interoperabilidad.

El diseño de Scipion está basado en los principios de integración, reproducibilidad y flexibilidad. La integración permite combinar múltiples programas sin necesidad de adaptar manualmente el formato de los datos, la reproducibilidad asegura que los *workflows* puedan ser ejecutados de forma consistente, y la flexibilidad permite incorporar herramientas de forma sencilla con *plugins* o protocolos.

Dentro de Scipion, surge Scipion-chem, un *plugin* especializado en la química estructural y el cribado virtual de fármacos [18]. Este *plugin* amplía las funcionalidades de Scipion incluyendo herramientas relacionadas con el *docking* molecular, la predicción de interacciones proteína-ligando o el análisis estructural. De esta forma, se pueden construir *workflows* específicos en este dominio sin perder los principios que proporciona Scipion.

1.3. Motivación

Scipion proporciona un entorno que facilita la construcción de *pipelines* reproducibles, evitando la necesidad de cambiar formatos de datos entre las distintas herramientas. Sin embargo, actualmente está enfocado principalmente a metodologías clásicas, sin incorporar herramientas modernas de predicción estructural basadas en *machine learning*.

Pese a la rápida aparición de estos modelos, su integración sigue siendo limitada por la heterogeneidad de sus interfaces, formatos de entrada y salida, y modos de ejecución. Esto obliga a utilizarlos de forma aislada, lo que dificulta su comparación y la integración en *pipelines* reproducibles.

De aquí surge la necesidad de extender Scipion-chem para permitir la integración de estas herramientas, facilitando su uso en un entorno estructurado y su aplicación en análisis estructurales y de cribado virtual.

1.4. Objetivos

El objetivo general de este proyecto consiste en evaluar y comparar de manera sistemática la precisión, el rendimiento y la viabilidad de las herramientas de predicción basadas en *machine learning*. Para lograrlo, se propone como solución técnica el diseño y desarrollo de un *plugin* especializado dentro de Scipion-chem, denominado Biofold.

Para cumplir con el objetivo general, se proponen los siguientes objetivos específicos:

- Desarrollar e integrar el *plugin* Biofold dentro de Scipion-chem, abstrayendo la complejidad

de las herramientas individuales mediante la estructura modular basada en protocolos de Scipion.

- Implementar un protocolo de importación que permita analizar y utilizar las estructuras generadas en servidores web externos dentro de Scipion.
- Diseñar e integrar protocolos de ejecución local para modelos nuevos, especialmente Chai-1, Protenix, Boltz-2 e IntelliFold.
- Validar y contrastar el rendimiento de los modelos implementados mediante casos de uso específicos, analizando sus diferencias estructurales.

2 Materiales y métodos

2.1. Entorno de trabajo y recursos computacionales

El desarrollo del *plugin* y el análisis se realizó en un equipo de alto rendimiento, lo que ha permitido una ejecución eficiente de tareas computacionalmente intensivas.

2.1.1. Sistema operativo

El proyecto se desarrolló en un entorno Linux, concretamente Ubuntu 20.04.6 LTS.

2.1.2. Hardware

- Procesador (CPU): AMD Ryzen Threadripper 3990X 64-Core Processor.
 - 64 núcleos físicos y 128 hilos de procesamiento.
- Memoria RAM: 251 GB.
- Almacenamiento principal:
 - Disco NVMe de 1.8 TB (SO con 1.3 TB en uso).
 - Sistema de almacenamiento adicional basado en volúmenes LVM de hasta 20 TB disponibles.
- Tarjeta gráfica (GPU):
 - 2 x NVIDIA GeForce RTX 3090 (24 GB VRAM cada una).
 - Soporte CUDA 12.8.

2.1.3. Entorno de desarrollo y herramientas de soporte

- PyCharm: PyCharm fue utilizado como el IDE principal para la programación del *plugin* en Python [19]. Su sistema de depuración y análisis facilitó la estructuración de las clases y protocolos de Scipion.
- GitHub: GitHub se utilizó como plataforma para almacenar y gestionar el código fuente basado en Git, permitiendo el control de versiones y garantizando trazabilidad de cada cambio [20].

- LucidChart: Esta herramienta de diseño se empleó para generar diagramas para ayudar a visualizar estructuras de *plugins* y diagramas de flujo [21].
- Overleaf/LaTeX: Overleaf es una plataforma colaborativa que se empleó para la redacción de la memoria del proyecto, asegurando un formato tipográfico homogéneo y una gestión automatizada de la bibliografía [22].
- ChatGPT (OpenAI): Este agente de IA fue utilizado como ayuda para depurar y optimizar código y para la revisión de la memoria final [23].

2.1.4. Dependencias generales

El entorno de ejecución se basa en Python y el *framework* de Scipion, junto con sus extensiones Scipion-em y Scipion-chem [17, 18, 24]. La instalación de estos entornos implica la resolución automática de múltiples dependencias científicas y de procesamiento de datos. Entre los paquetes utilizados se incluyen NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, Pillow, Requests, tqdm, lxml, psutil, setuptools y typing_extensions, entre otras. Además, se emplean paquetes específicos del ámbito bioinformático y de microscopía electrónica, como Biopython, MRCfile, tiffiff, emtable, ncempy y metadata-viewer.

Para el desarrollo del *plugin* Biofold se ha utilizado el *framework* de Scipion, que constituye el núcleo del entorno de procesamiento, junto con la extensión Scipion-em orientada para el análisis de datos de cryo-EM. El nuevo *plugin* se ha desarrollado dentro de la extensión Scipion-chem, especializada en el análisis y modelado en química computacional. Scipion integra componentes como Scipion-app y Scipion-pyworkflow, encargados de la gestión de la interfaz y del flujo de trabajo [25, 26]. Las dependencias específicas de cada módulo se encuentran documentadas en los repositorios de GitHub correspondientes.

2.2. Framework de Scipion

Scipion es un *framework* orientado a la ejecución de *workflows* para el análisis de datos de biología estructural [17]. Su estructura modular permite ejecutar *pipelines* de forma ordenada, reproducible y automatizada. Scipion está compuesto de extensiones, *plugins* y protocolos.

Sus principales extensiones son Scipion-em y Scipion-chem, que se enfocan en el análisis de datos de cryo-EM y análisis de cribado virtual respectivamente [17, 18]. Cada extensión tiene *plugins* que engloban protocolos para realizar tareas sobre los datos. En total Scipion está

Comparativa de la precisión en la predicción de estructuras de macromoléculas biológicas a partir de secuencias de aminoácidos

compuesto por más de 120 *plugins* y más de 1000 protocolos. La Figura 2 muestra un diagrama simplificado de la estructura de Scipion.

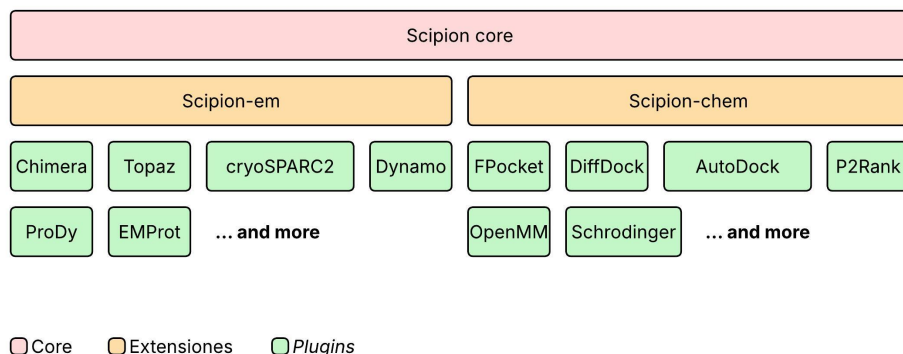


Figura 2: Estructura de extensiones y *plugins* de Scipion.

La Figura 3 muestra la interfaz gráfica de Scipion, que cuenta con tres secciones principales.

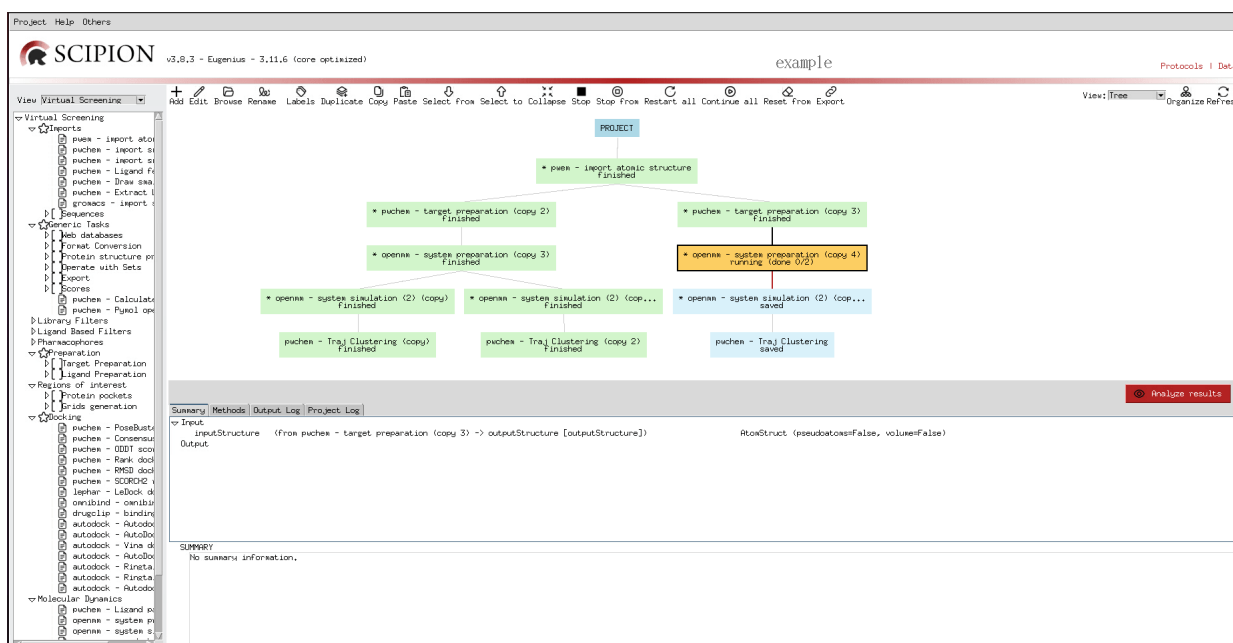


Figura 3: Interfaz de usuario de Scipion.

En la zona central se muestra el *workflow* activo, donde cada caja representa un protocolo. El estado de cada uno se indica mediante un código de colores, que permite identificar si está pendiente de ejecución, en ejecución, finalizado correctamente o con errores. La sección de la izquierda muestra la selección de protocolos descargados, organizados por categorías gracias al desplegable de la parte superior, lo que facilita la búsqueda de nuevas herramientas. La sección inferior muestra información detallada de cada protocolo seleccionado, incluyendo los objetos

de entrada y de salida, y los registros de ejecución. Esta sección permite analizar el comportamiento del protocolo y depurar posibles errores. La parte superior contiene herramientas para el proyecto, como etiquetas, opciones de exportación o una carpeta que almacena información del proyecto y los protocolos individuales.

Los *workflows* de Scipion se basan en la dependencia entre protocolos, donde los resultados de unos actúan como entrada de otro. Otra característica es la reproducibilidad, ya que se almacenan los parámetros, entradas y salidas de cada uno de ellos, lo que facilita la repetición de análisis y la trazabilidad de los resultados. Para gestionar esta información, Scipion utiliza objetos que representan entidades biológicas, como moléculas o secuencias, y datos intermedios, como mapas, que permiten la interoperabilidad de los protocolos en un mismo *workflow*. Por último, el *framework* permite la ejecución en entornos locales o en infraestructuras de computación de alto rendimiento, adaptándose a diferentes necesidades computacionales.

Scipion-chem surge como una extensión de Scipion con el fin de poder realizar análisis de cribado virtual teniendo las facilidades del entorno de Scipion [18]. Esta extensión cuenta aproximadamente con 30 *plugins* que incluyen herramientas que realizan tareas desde la preparación de moléculas hasta la evaluación de las interacciones. El *workflow* típico en cribado virtual dentro de Scipion-chem sigue una secuencia estructurada. Primero se realiza la preparación de la proteína de interés, incluyendo corrección de estructuras o protonación. Paralelamente se preparan los ligandos, que pueden ser filtrados en función de criterios específicos como farmacóforos. Una vez estén las moléculas preparadas, se hace una búsqueda de los posibles sitios de unión o *pockets* de la proteína, lo que permite acotar las regiones donde se ejecutará el *docking*. A continuación se realiza el *docking* proteína-ligando con distintos algoritmos incluidos en Scipion-chem, y por último se pueden realizar dinámicas moleculares para refinar y evaluar la estabilidad de las interacciones. Scipion-chem cuenta adicionalmente con protocolos que se pueden incluir a lo largo del proceso, incluyendo protocolos de *scoring*, consensos o alineamientos.

2.3. Herramientas de predicción estructural

2.3.1. AlphaFold3

AlphaFold3 fue introducido como una evolución de AlphaFold y actualmente es una de las herramientas más avanzadas para la predicción estructural molecular. AlphaFold3 no solo

permite predecir estructuras proteicas, como en versiones anteriores, sino que permite modelar complejos proteicos, con ácidos nucleicos y pequeños ligandos [11, 12]. La arquitectura de AlphaFold3 se divide principalmente en tres módulos, representados en la Figura 4.

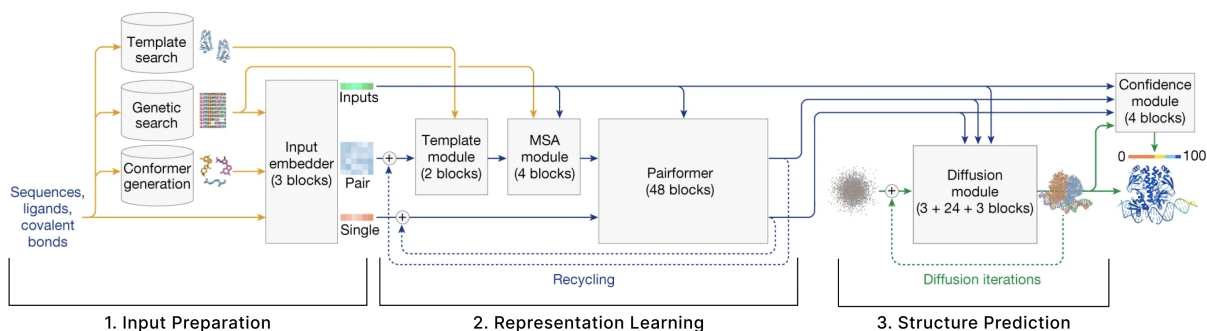


Figura 4: Arquitectura de AlphaFold3 [27].

El primer módulo se encarga de preparar los datos de entrada. Las secuencias son transformadas en representaciones numéricas (*embeddings*), y el modelo incorpora información evolutiva y estructural mediante MSA y plantillas obtenidas de bases de datos. El segundo módulo realiza el aprendizaje representacional. Se utilizan mecanismos de atención y representaciones de pares de residuos para aprender las relaciones espaciales y geométricas. Este módulo contiene el núcleo computacional del modelo. Por último, el tercer módulo realiza las predicciones estructurales. AlphaFold3 utiliza un modelo de difusión condicional para generar coordenadas tridimensionales de átomos, de forma iterada para ir refinando la estructura final.

AlphaFold3 genera por defecto cinco estructuras en formato mmCIF. Además proporciona métricas para estudiar las predicciones, descritas en la Tabla 1.

Métrica	Descripción
<i>Predicted LDDT</i> (pLDDT)	Fiabilidad local de la posición atómica.
<i>Predicted Aligned Error</i> (PAE)	Fiabilidad de la estructura y posición de dominios o cadenas.
<i>Predicted TM</i> (pTM)	Precisión general de la estructura.
<i>Interface-predicted TM</i> (ipTM)	Precisión de un componente de un complejo frente a otros.
<i>Per chain pTM</i> y <i>per-chain pair ipTM</i>	Fiabilidad de cadenas individuales y pares de cadenas.

Tabla 1: Métricas de AlphaFold3 [28].

En CASP14, AlphaFold tuvo los mejores resultados de la competición, obteniendo una precisión media del esqueleto de carbono de aproximadamente 0.96 Å RMSD, superando al siguiente competidor, cuya precisión fue de 2.8 Å. De forma general, la precisión media fue de

aproximadamente 1.5 Å, frente a 3.5 Å del resto de participantes [11].

AlphaFold3 puede ejecutarse mediante el servidor web o localmente, sin embargo, el código fuente completo no se encuentra públicamente accesible, y su instalación local requiere un volumen significativo de almacenamiento, así como una alta capacidad de cómputo [29,30].

2.3.2. Chai-1

Chai-1 es un modelo multimodal de predicción estructural que afirma obtener resultados prometedores, especialmente en predicciones proteína-ligando y de complejos multiméricos [14]. Fue desarrollado por Chai Discovery y, a diferencia de modelos previos, puede trabajar tanto con MSA como en modo de secuencia única, manteniendo una elevada precisión estructural. Su arquitectura sigue la de AlphaFold3, con ligeras modificaciones, como se representa en la Figura 5.

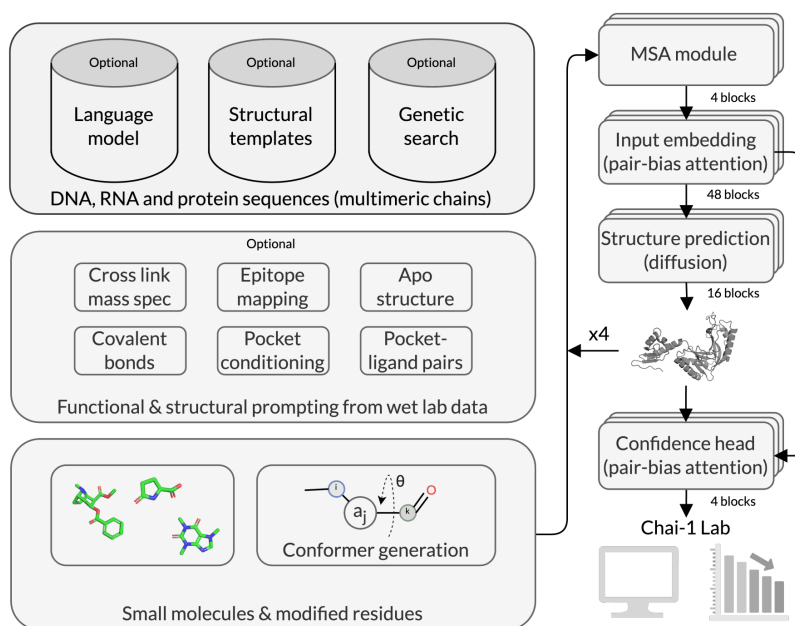


Figura 5: Arquitectura de Chai-1 [14].

Chai-1 recibe como entrada secuencias de proteínas, ADN o ARN, aceptando también complejos multiméricos. Puede incorporar de forma opcional información como plantillas estructurales o información de enlaces covalentes, además de moléculas y residuos modificados generando conformaciones compatibles con el complejo de interés. La predicción estructural se realiza mediante un modelo de difusión iterativo, que genera coordenadas tridimensionales de la estructura. Por último, se lanza un módulo de evaluación de confianza para evaluar la calidad de las predicciones empleando métricas internas.

En predicciones proteína-ligando, Chai-1 alcanza un porcentaje de éxito del 77 % utilizando un umbral de RMSD del ligando inferior a 2 Å, frente al 76 % obtenido por AlphaFold3. Además, Chai-1 muestra un rendimiento superior al de AlphaFold-Multimer 2.3 en predicción de complejos proteína-proteína y anticuerpo-proteína, especialmente en términos de DockQ [14]. Gracias a su capacidad de funcionar sin MSA, Chai-1 mantiene una precisión competitiva respecto a otros modelos, y la incorporación de restricciones experimentales, como contactos entre residuos o información de epítomos, mejora la calidad de las predicciones en complejos difíciles.

Chai-1 puede ejecutarse tanto en su servidor web, como localmente [31]. Su código fuente es público y accesible a través de GitHub [32].

2.3.3. *Protenix*

Protenix surge como alternativa de código abierto a AlphaFold3, alcanzando resultados similares con distintos *benchmarks* [15]. Desarrollado por el equipo de AI4Science de ByteDance, su objetivo principal es hacer accesible el código fuente a la comunidad científica con el objetivo de avanzar en la investigación de predicción estructural, permitiendo las mejoras del modelo por esta comunidad.

Su arquitectura sigue la de AlphaFold3, utilizando módulos tipo Pairformer y un modelo de difusión similar al de Chai-1. Protenix comienza con la entrada de secuencias de proteínas, ADN o ARN, y posibles ligandos y moléculas. El sistema genera características iniciales mediante *embeddings* de secuencia y pares de residuos, además, para proteínas se incorporan MSA obtenidos mediante MMseqs2 y ColabFold, mientras que para ácidos nucleicos no se emplea MSA [33,34]. Las representaciones son procesadas por los bloques Pairformer, que modelan las relaciones espaciales y dependencias entre residuos y cadenas. Finalmente se aplica el modelo de difusión iterativo para obtener las predicciones finales y se aplica un módulo de confianza para calcular las métricas internas. En evaluaciones sobre el *benchmark* de PoseBusters V2 alcanza resultados comparables, e incluso superiores, a AlphaFold3 en predicciones proteína-ligando. Además, supera a AlphaFold-Multimer 2.3 en predicciones proteína-proteína y muestra un rendimiento similar en complejos proteína-ADN y proteína-ARN [15].

Protenix publica el código fuente, pesos entrenados y herramientas de entrenamiento, permitiendo su ejecución local, además de ser accesible a través de un servidor web [35,36].

2.3.4. Boltz-2

Boltz-2 es un modelo de predicción estructural que a diferencia de otros modelos, permite estimar la afinidad proteína-ligando [13]. Desarrollado por investigadores del MIT y Recursion, incorpora un módulo específico para obtener la afinidad proteína-ligando, permitiendo aplicaciones en cribado virtual, descubrimiento de moléculas y optimización de compuestos. Su arquitectura está dividida en cuatro componentes principales, representados en la Figura 6.

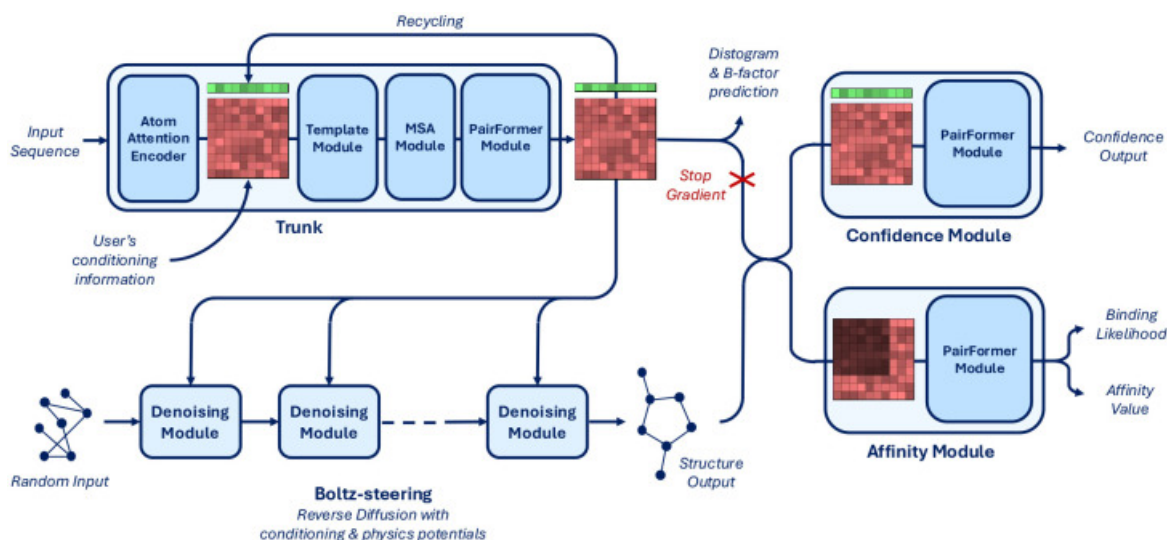


Figura 6: Arquitectura de Boltz-2 [13].

El primer módulo es el de representación, donde las secuencias y moléculas de entrada son transformadas en *embeddings* y procesadas mediante representaciones de pares y mecanismos de atención, además de incorporar MSA y plantillas estructurales. El segundo módulo es el de predicción estructural, en el que se aplica un modelo de difusión iterativo, pudiendo incorporar restricciones de distancia, plantillas y otras condiciones. El tercer módulo es el de confianza, donde se evalúa la calidad de las predicciones. Por último, Boltz-2 aplica un módulo de afinidad, donde se estima la afinidad proteína-ligando, incluyendo probabilidad de unión y valores continuos de afinidad. En *benchmarks* públicos, Boltz-2 alcanza resultados competitivos frente a AlphaFold3 y Chai-1, mejorando la predicción de proteína-ligando y complejos como anticuerpo-antígeno. En predicción de afinidad, Boltz-2 consigue aproximarse al rendimiento de métodos físicos, manteniendo tiempos computacionales considerablemente menores [13].

Boltz-2 puede ejecutarse tanto en su servidor web como localmente, disponiendo de código fuente, pesos entrenados y herramientas de inferencia accesibles públicamente en GitHub [37, 38].

2.3.5. IntelliFold

IntelliFold es un modelo predictivo que mejora la fiabilidad de las predicciones mediante refinamientos arquitectónicos y la incorporación de consistencia estructural multiescala [16]. Su objetivo principal es aumentar la robustez y precisión en las predicciones estructurales, especialmente en interacciones proteína-ligando y anticuerpo-antígeno. Su arquitectura se basa en una serie de innovaciones sobre modelos, incorporando tres mejoras; escalado del espacio latente en los bloques Pairformer, formulación de atención a nivel atómico con atomización estocástica, y un esquema de optimización guiado durante el muestreo por difusión. El diagrama de la arquitectura puede verse en la Figura 7.

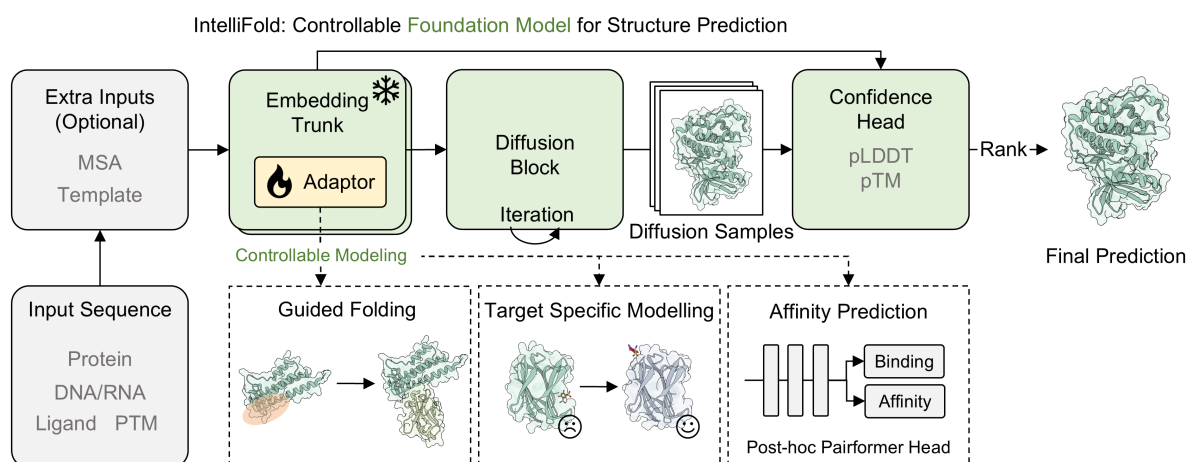


Figura 7: Arquitectura de IntelliFold [16].

En la primera etapa, se transforman las secuencias de entrada en representaciones latentes. Después, los bloques de atención refinan las interacciones entre residuos y átomos, incorporando información estructural implícita. Por último, el modelo de difusión genera las coordenadas tridimensionales antes de aplicar el módulo de confianza. IntelliFold introduce una ponderación de pérdida sensible a la dificultad, mejorando el aprendizaje en regiones estructurales complejas o poco representadas en los datos de entrenamiento. La evaluación sobre *benchmarks* muestra mejoras frente a AlphaFold3 en predicciones proteína-proteína, anticuerpo-antígeno y proteína-ligando [16]. Estos resultados indican una mayor capacidad del modelo para capturar interacciones no triviales en sistemas complejos. La última versión de IntelliFold incluye tres variantes (Flash, v2 y Pro) que permiten ejecutar el modelo con distintos compromisos de velocidad y precisión. La versión Flash está optimizada para eficiencia computacional, mientras que la Pro está orientada para máxima precisión de alto rendimiento.

El modelo se puede ejecutar tanto en el servidor web como localmente, y su código es accesible desde GitHub [39, 40].

2.4. Diseño e implementación del plugin

2.4.1. Arquitectura del plugin

Los *plugins* de Scipion se instalan con `pip`, y aparte de seguir su estructura básica, contienen una carpeta principal, cuyo nombre es el del *plugin*, con distintos archivos y carpetas [41]. Algunos de los archivos principales (no todos son obligatorios) son:

__init__.py: Definición del *plugin*; se encarga de la instalación de binarios y del manejo de variables de entorno.

constants.py: Define las constantes globales utilizadas tanto en el *plugin* como en sus respectivos protocolos.

viewer.py: Implementa las herramientas y visores gráficos necesarios para visualizar y analizar los resultados generados.

wizard.py: Crea los asistentes de ayuda (*wizards*) para guiar al usuario en la configuración de parámetros complejos.

Las carpetas principales de los *plugins* son *protocols*, *scripts*, *tests* y *viewers*. En *protocols* se encuentran los protocolos del *plugin*, en *scripts* están los *scripts* auxiliares para los protocolos, en *tests* se encuentran los *tests* de los protocolos y en *viewers* están los archivos que se encargan de mostrar los resultados de los protocolos. Dentro de cada protocolo, se sigue una estructura similar, dividida en cinco partes. Primero se definen los parámetros que se van a usar en el protocolo, seguido de una lista de pasos que se van a ejecutar en orden para el correcto funcionamiento del protocolo. Posteriormente se describen las funciones mencionadas anteriormente, que pueden ser bloques de código, o llamadas a herramientas. Por último se definen métodos de validación e información, que pueden ser útiles para los usuarios, y otros métodos auxiliares que se puedan necesitar en el protocolo.

Para facilitar la integración de todos los protocolos, los parámetros son instancias de objetos de Scipion. Dependiendo del tipo de dato, pueden ser `StringParam`, `FloatParam` o `EnumParam` entre otros. Si el parámetro de entrada es un objeto de Scipion complejo, como por ejemplo

un set de moléculas, el tipo de parámetro será `PointerParam`, que apunta al objeto deseado cuando se define en el código.

El código del *plugin* de Biofold se puede encontrar en el Anexo A. En la Figura 8 se muestra un esquema de la estructura del *plugin* Biofold. Para este *plugin* se ha añadido un archivo `objects` que contiene objetos específicos usados por los protocolos.

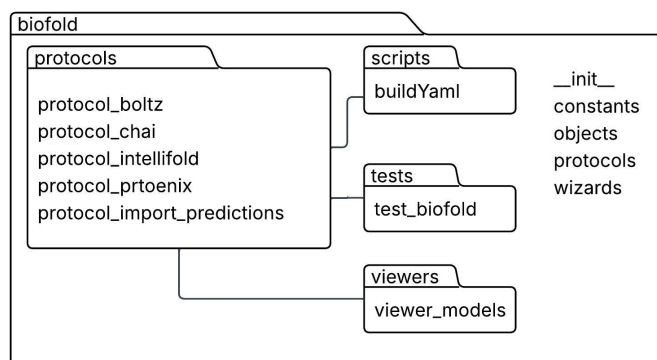


Figura 8: Estructura de scipion-chem-biofold.

Cada protocolo se debe clasificar para que pueda ser encontrado fácilmente en el árbol de protocolos de la interfaz gráfica, descrita en la Figura 3. En este contexto, la Figura 9 muestra un resumen de la clasificación de los protocolos de Scipion-chem, donde se puede ver que los protocolos de Biofold de predicción estructural se han clasificado dentro de `Protein structure prediction`.

2.4.2. Protocolo de importación

El primer protocolo de Biofold tiene como objetivo importar predicciones estructurales generadas en servidores externos y convertirlas en objetos de Scipion para su análisis. El protocolo admite resultados de AlphaFold3, Protenix, Chai-1 y Boltz-2. Como entrada se debe proporcionar el origen de los modelos y el archivo comprimido descargado directamente del servidor web correspondiente. Se soportan formatos `.zip`, `.tar.gz`, `.tgz` y `.tar`.

Primero, se descomprime el archivo de entrada en el directorio de trabajo del protocolo. Debido a que cada herramienta organiza los resultados con una estructura de carpetas diferente, se emplea un sistema de búsqueda de modelos para cada origen. En el caso de Protenix, los modelos se encuentran en la carpeta `'predictions'`, mientras que Boltz-2 los almacena en `'result'`. Para AlphaFold3 y Chai-1 se hace una búsqueda recursiva de archivos mmCIF, ya que el nombre de las carpetas puede variar en función del *seed* empleado.

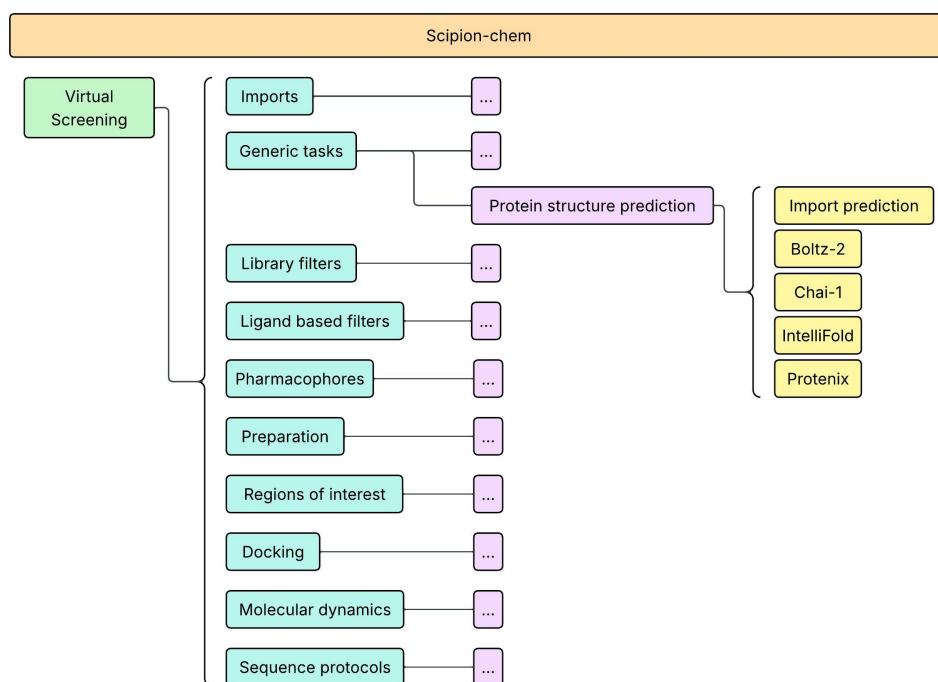


Figura 9: Clasificación de protocolos de Scipion-chem.

Una vez localizadas las estructuras, se calculan las métricas de calidad a partir de los estimadores de confianza almacenados en cada modelo (como los valores de pLDDT o puntuaciones equivalentes según la herramienta). Los resultados se guardan en un archivo resumen que contiene el valor medio de confianza de cada modelo y la estructura seleccionada como mejor predicción. Por último, se convierten las estructuras en objetos `AtomStruct` y `SetOfAtomStructs`, que definen una estructura molecular y un set de estructuras respectivamente.

2.4.3. Protocolos de ejecución local

Todos los protocolos de ejecución local comparten una estructura de parametrización dentro de Scipion. Las secuencias de entrada pueden provenir de distintos orígenes, incluyendo secuencias (`Sequence`), conjuntos de secuencias (`SetOfSequences`), estructuras atómicas (`AtomStruct`) o archivos FASTA externos. Se puede incluir más de una secuencia en la misma ejecución, que pueden provenir de distintos orígenes. En el caso de las estructuras atómicas, se pueden seleccionar cadenas concretas y rangos específicos para extraer subsecuencias. Aunque los cuatro protocolos siguen un modelo similar de entrada de secuencias, Boltz-2 incluye un parámetro que va a permitir definir moléculas cíclicas. Además, para cada secuencia, se debe seleccionar si se trata de un fragmento proteico, ADN o ARN, para el correcto funcionamiento

de las herramientas. Los protocolos incluyen parámetros relacionados con la inferencia del modelo, explicados en las secciones posteriores. Como salida se generan un set de estructuras con el número de estructuras seleccionadas por el parámetro de modelos de salida, y una estructura individual que corresponde a la de mayor puntuación.

Los protocolos de ejecución local se pueden dividir en dos grupos en función del flujo de trabajo requerido por cada herramienta, ya que los formatos de entrada y los pasos intermedios difieren.

Los protocolos de Chai-1 y Protenix siguen una estructura compuesta por cuatro etapas:

1. Conversión y preparación de datos de entrada
2. Ejecución de la herramienta de predicción estructural
3. Extracción de métricas o puntuaciones de confianza
4. Generación de objetos de salida

Para los protocolos de Boltz-2 e IntelliFold se necesitan archivos YAML como entrada, por lo que tiene un paso adicional:

1. Conversión y preparación de datos de entrada
2. Generación del archivo YAML
3. Ejecución de la herramienta de predicción estructural
4. Extracción de métricas o puntuaciones de confianza
5. Generación de objetos de salida

Chai-1

El protocolo de ejecución de Chai-1 incluye cinco parámetros para su ejecución, descritos en la Tabla 2.

El protocolo convierte las secuencias de entrada en un archivo FASTA compatible con el requerido por la herramienta, y posteriormente ejecuta `chai-lab fold` con los parámetros de entrada para obtener los modelos estructurales. Una vez se tienen los modelos, el protocolo calcula la puntuación media a partir de los valores *B_iso_or_equiv* almacenados en los archivos mmCIF, y se generan los objetos de salida, seleccionando automáticamente el modelo con mayor puntuación.

Parámetro	Definición	Valor por defecto	Nivel
MSA	Ejecución con MSA	Sí	Normal
Pasos de reciclaje	Número de ciclos de retroalimentación	3	Avanzado
Pasos de muestreo	Número de pasos de difusión	200	Normal
Muestras del <i>trunk</i>	Número de muestras independientes que procesa en bloque principal	1	Avanzado
Modelos de salida	Número de modelos finales	5	Normal

Tabla 2: Parámetros de configuración de Chai-1.

Protenix

El protocolo de Protenix tiene cuatro parámetros descritos en la Tabla 3. A diferencia del anterior, Protenix permite seleccionar qué modelo emplear para obtener las predicciones.

Parámetro	Definición	Valor por defecto	Nivel
MSA	Ejecución con MSA	Sí	Normal
Pasos de muestreo	Número de pasos de difusión	200	Normal
Modelos de salida	Número de modelos finales	5	Normal
Modelo	Modelo de predicción	<code>protenix-v2</code>	Normal

Tabla 3: Parámetros de configuración de Protenix.

La herramienta cuenta con nueve modelos distintos, sin embargo, para el protocolo se permite seleccionar entre cuatro; `v2`, `default`, `v1`, `20250630` `v1` y `v0.5`, que son los recomendados por los desarrolladores.

El protocolo de Protenix comienza generando un archivo JSON compatible con el formato de entrada de la herramienta. Este JSON incluye la secuencia, cadena y número de veces que se debe incluir. A continuación, se ejecuta la herramienta con el comando `protenix pred` para generar los modelos y extraer posteriormente la puntuación. A diferencia de Chai-1, Protenix genera un archivo para cada modelo donde se incluyen métricas generadas por la herramienta, incluyendo una puntuación de calidad global, que va a ser utilizada para seleccionar el mejor modelo.

Boltz-2

Como se ha mencionado antes, el protocolo de Boltz-2 cuenta con un parámetro adicional al seleccionar las cadenas de entrada que permite especificar si es cíclica o no. El protocolo cuenta con los parámetros descritos en la Tabla 4.

Parámetro	Definición	Valor por defecto	Nivel
GPU	Uso de GPU	Sí	Normal
MSA	Ejecución con MSA del servidor	Sí	Normal
Potenciales de inferencia	Usar potenciales físicos para asegurar poses biológicamente realistas	No	Normal
Pasos de reciclaje	Número de ciclos de retroalimentación	3	Avanzado
Pasos de muestreo	Número de pasos de difusión	200	Normal
Modelos de salida	Número de modelos finales	5	Normal
Tamaño de paso	Escala del proceso de muestreo	1.638	Normal
Corrección por peso molecular	Aplicar factor de corrección para el cálculo de la afinidad	No	Normal
Muestras de difusión	Número de muestras adicionales dedicadas específicamente a la predicción de afinidad	5	Avanzado

Tabla 4: Parámetros de configuración de Boltz-2.

El protocolo de Boltz-2 comienza generando un archivo de entrada JSON, antes de generar un archivo YAML, requerido como formato de entrada. Este archivo incluye información de las secuencias, cadenas y si son cíclicos o no. Para asegurar una correcta conversión del JSON al YAML, se creó un *script* que realiza esta conversión, y un objeto auxiliar BoltzEntity que contiene información sobre las secuencias para simplificar el proceso. Después se ejecutan las predicciones con `boltz predict`, pudiendo ser ejecutadas en GPU. Por último se extraen las puntuaciones de confianza de unos archivos generados por Boltz-2 para cada modelo y se selecciona de forma automática la mejor estructura.

IntelliFold

El protocolo de IntelliFold tiene cinco parámetros, entre los que se incluye la selección del modelo a usar para realizar las predicciones, descritos en la Tabla 5.

Los modelos que se pueden seleccionar para realizar las ejecuciones son: v1, v2 y

Parámetro	Definición	Valor por defecto	Nivel
MSA	Ejecución con MSA	Sí	Normal
Modelos de salida	Número de modelos finales	5	Normal
Pasos de reciclaje	Número de ciclos de retroalimentación	3	Avanzado
Pasos de muestreo	Número de pasos de difusión	200	Normal
Modelo	Modelo de predicción	v2-flash	Normal

Tabla 5: Parámetros de configuración de IntelliFold.

v2-flash, siendo esta última la recomendada por los desarrolladores, ya que fusiona calidad de ejecución con rapidez.

De nuevo, como en el caso de Boltz2, se comienza generando un archivo JSON, del que se va a sacar información para crear el YAML, con la ayuda del *script* auxiliar y un objeto llamado `IntelliFoldEntity`. Posteriormente se ejecuta la herramienta con `intellifold predict` y se obtienen las puntuaciones con unos archivos generados por IntelliFold, a partir de los cuales se selecciona la mejor predicción.

2.4.4. Viewers

Para poder visualizar los resultados de forma correcta, se ha desarrollado un visor específico para Biofold que permite visualizar la mejor estructura en Chimera siguiendo la paleta de colores de AlphaFold. El visor se trata de un método que genera un *script* de Chimera, donde se abre el archivo de la estructura, se selecciona la paleta de colores de AlphaFold, se esconden los átomos y se muestran las estructuras con *ribbon*. Posteriormente se llama a Chimera para abrir la estructura en la pantalla.

2.4.5. Tests

Para asegurar el correcto funcionamiento de los protocolos, se han creado *tests* para los protocolos de ejecución local. Los *tests* se ejecutan sobre una estructura descargada por defecto en Scipion para asegurar que las herramientas se ejecutan de forma correcta y aparecen las estructuras como objetos de salida. El *plugin* incluye un *test* por modelo y todos siguen la misma estructura. Primero se importa la secuencia mencionada previamente, y posteriormente se aplica el protocolo específico de ese *test* sobre ella, reduciendo el número de pasos para que no consuman muchos recursos y se ejecuten de forma eficiente. Estos *tests* se pueden ejecutar a la vez con

el comando `scipion3 tests biofold.tests` o individualmente especificando el nombre del *test* tras el comando anterior (por ejemplo `scipion3 tests biofold.tests.TestChai`).

2.5. Casos de uso

Para probar el *plugin* se han realizado tres casos de uso con distintas estructuras para comparar las predicciones de cada herramienta. El diseño del proyecto consiste en importar una estructura con el protocolo `import atomic structure` para tener una estructura de referencia, aquella que se encuentra en RCSB PDB [42]. De esta misma base de datos se descarga el archivo FASTA de la estructura de estudio para utilizar como entrada en los protocolos. Para la ejecución de las herramientas, AlphaFold3 será ejecutado en el servidor web y posteriormente importado a Scipion con el protocolo de Biofold `import structure predictions`, mientras que las otras herramientas serán ejecutadas localmente en Scipion. Por último se emplea un protocolo del *plugin* de Chimera, llamado `find discrepancies`, que indica diferencias entre estructuras [43].

Para realizar el análisis se han empleado tres estructuras con distintas características para comprobar la eficacia de las herramientas resolviendo distintas estructuras. La primera estructura es MAP2K6 (del PDB 9M1Z), que tiene función transferasa y una única cadena [44]. La segunda estructura se trata de una ADN-liasa OGG1(WT) (con PDB 9NZ9) [45]. Esta estructura fue escogida ya que está unida a un fragmento de ADN, por lo que se utilizará para ver la capacidad de las herramientas para posicionarlo en el surco de unión. Por último, se utilizó la proteasa NS2B-NS3 del virus Zika (en PDB 9QS2), ya que se trata de un complejo, con una cadena de 53 aminoácidos y otra de 176, por lo que nos sirve como complejo proteico simple [46]. La Figura 10 muestra las estructuras de referencia de estas moléculas.

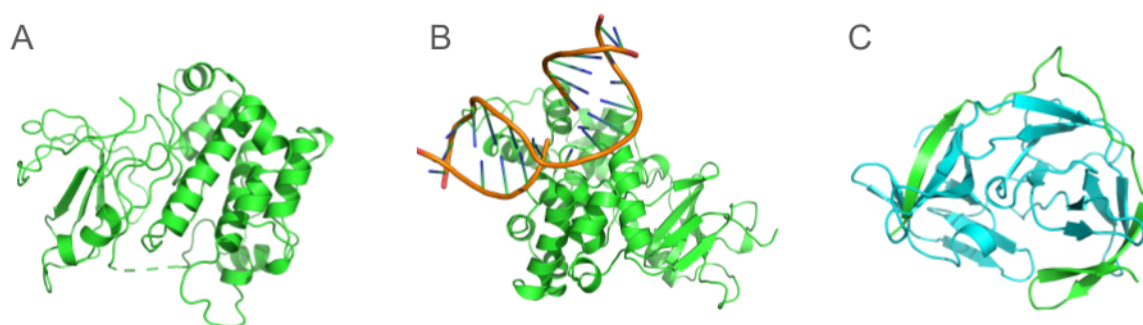


Figura 10: Estructuras de referencia para el análisis (A: 9M1Z, B: 9NZ9, C: 9QS2).

2.5.1. Estructura del proyecto

Como se ha mencionado previamente, en el proyecto se van a emplear los protocolos de Biofold y un par de protocolos adicionales para analizar las estructuras. La Figura 11 muestra el diseño empleado para la estructura 9M1Z, que será repetido para las otras estructuras.

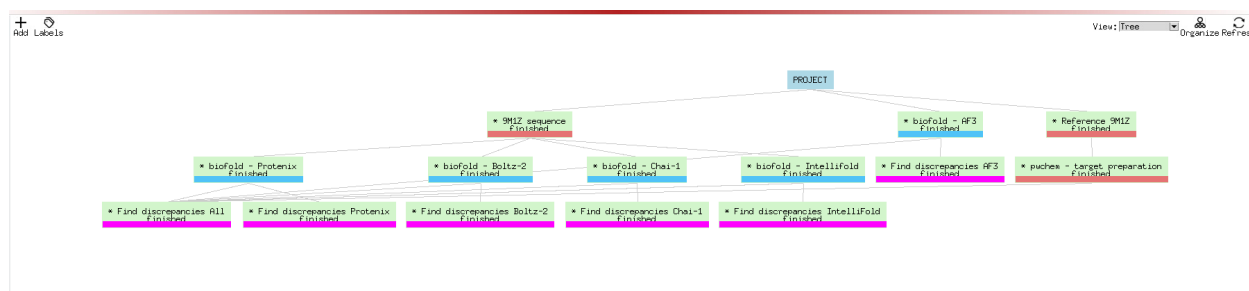


Figura 11: Estructura del análisis para 9M1Z.

El diseño se puede dividir en tres grupos, diferenciados con colores en la Figura 11. El primer grupo (con una banda roja) incluye un protocolo de importación del archivo FASTA (protocolo rojo más a la izquierda), un protocolo que importa la estructura 9M1Z del PDB para tener como referencia (primer protocolo rojo a la derecha), y por último uno que prepara la estructura para poder ser utilizada en protocolos posteriores (segundo protocolo rojo a la derecha). Este último protocolo elimina agua y ligandos de la estructura importada. El segundo grupo (con una banda azul) agrupa los protocolos de Biofold, uno por cada herramienta ejecutada localmente (de izquierda a derecha Protenix, Boltz-2, Chai-1 e Intellifold) y otro que importa los resultados de AlphaFold3 ejecutado externamente (protocolo azul más a la derecha). Por último, el tercer grupo (con una banda rosa) son los protocolos que utilizan el protocolo `find discrepancies` para comparar las estructuras. Este protocolo utiliza Chimera para encontrar diferencias, pudiendo visualizar las estructuras en Chimera y una gráfica de la diferencia de RMSD por aminoácido o nucleótido. El protocolo es ejecutado tanto inter como intra-herramienta, y necesita una estructura de referencia. Para los intra-herramienta se ha seleccionado como referencia la mejor estructura, con el fin de ver cómo varían las predicciones dentro de una misma herramienta, y para el inter-herramienta se ha seleccionado como referencia la estructura importada del PDB (mostradas en la Figura 10) y se han incluido solamente las mejores predicciones de cada herramienta.

Para las estructuras 9M1Z y 9QS2 se ha empleado como entrada un objeto `Sequence`, habiendo importado previamente el archivo FASTA, y para 9NZ9 se ha utilizado directamente el archivo FASTA. De esta forma se comprueba que funcionan correctamente dos tipos de

entrada.

El proyecto completo listo para importar en Scipion se puede encontrar en el Anexo A.

2.5.2. Configuración de parámetros de ejecución

Para establecer una comparativa equitativa entre las distintas herramientas, se definieron parámetros homogéneos. Se fijaron 200 pasos de muestreo y 3 ciclos de reciclaje, representando el estándar de los modelos de difusión, lo que proporciona un equilibrio en la convergencia estructural y el tiempo de computación. Se fijaron un total de 5 modelos de salida en todas las herramientas, incluyendo la ejecución en el servidor de AlphaFold3, lo que va a permitir la identificación de la mejor predicción y la evaluación de la precisión intra-herramienta mediante desviaciones estructurales. Todas las predicciones se realizaron aplicando MSA con el objetivo de comparar con estructuras experimentales del PDB. Por último, en los modelos que lo permiten, se permitió el uso de GPU para una rápida ejecución. En el caso de AlphaFold3, se mantuvieron los parámetros por defecto de la plataforma.

3 Resultados

3.1. Implementación del plugin

El resultado principal de este proyecto es la creación e integración del *plugin* Biofold en el ecosistema Scipion-chem. Se ha logrado una integración funcional que permite a los usuarios acceder a herramientas de predicción estructural avanzadas sin salir de Scipion, eliminando las barreras técnicas en la instalación de cada *software* individual y la gestión de los archivos de salida. Biofold ha sido validado, cumpliendo con los estándares de interoperabilidad de Scipion, y se han implementado con éxito los protocolos de importación y ejecución local, permitiendo su uso adecuado y fluido en los *workflows* de biología estructural.

3.1.1. Interfaz de usuario

Los protocolos se han ordenado de forma lógica dentro del árbol de protocolos para su identificación rápida, proporcionando formularios de parámetros claros que facilitan la configuración básica y el ajuste avanzado de los modelos.

La Figura 12 muestra a la izquierda el árbol de protocolos de Scipion-chem, donde se pueden ver los protocolos de Biofold en Protein structure prediction. A la derecha se muestra un ejemplo de formulario de entrada, específicamente para IntelliFold.

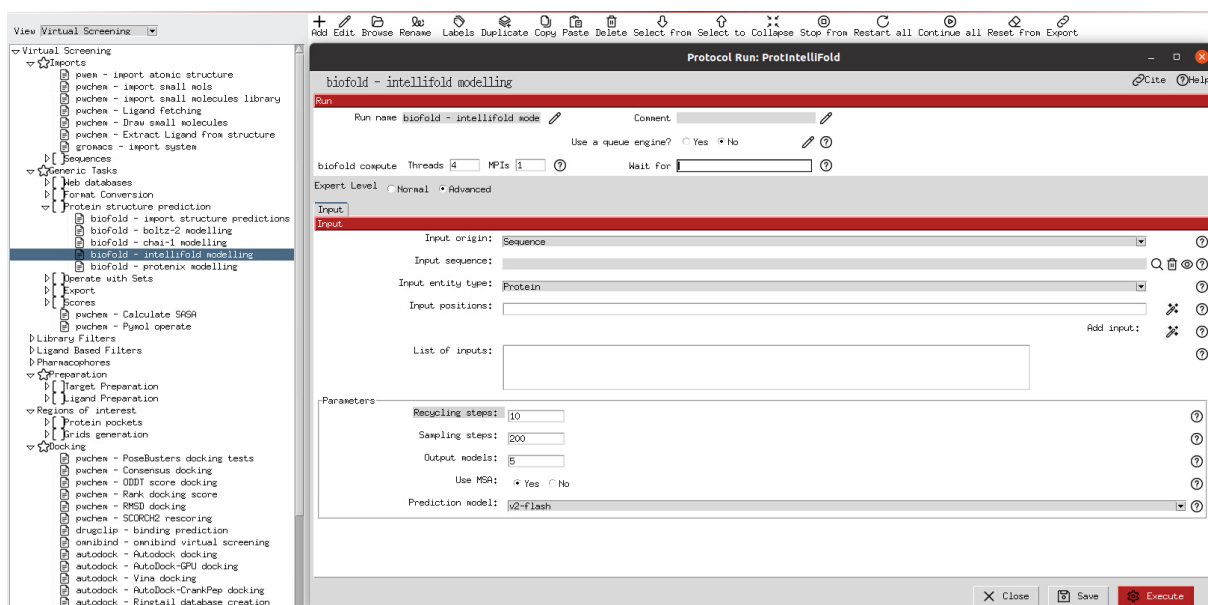


Figura 12: Árbol de protocolos y formulario de IntelliFold.

La parte superior contiene parámetros generales de Scipion, correspondientes al uso de *threads* y, en caso de ser posible, GPU. Además se puede configurar un sistema de colas donde se puede habilitar la ejecución del protocolo en distintos *clusters* en entornos de HPC. La primera sección de entrada permite seleccionar el tipo de objeto del que proviene la cadena (*Sequence*, *SetOfSequence*, *AtomStruct* o archivo FASTA), seguido de la selección del objeto, tipo de molécula y selección de aminoácidos o nucleótidos específicos en caso de querer subsecuencias específicas. El cuadro de texto *List of inputs* almacenará una breve descripción de las secuencias seleccionadas una vez se dé al botón *Add input*, por lo que es posible almacenar más de una secuencia provenientes de distintos objetos.

En la sección de parámetros, se muestran los parámetros básicos (sin resaltar) y los avanzados (resaltados en gris), además de la elección del modelo específico en este caso. Todos los parámetros cuentan con un botón de ayuda (símbolo ? a la derecha), al igual que el protocolo (*Help* en la esquina superior derecha), dónde se muestra una breve descripción para ayudar al usuario.

3.1.2. Manejo de objetos

Tras la ejecución de un protocolo, se muestran en la parte inferior de la pantalla los objetos de entrada y salida, como se menciona en secciones anteriores. Se ha verificado que todos los protocolos proporcionan estos objetos (*AtomStruct* y *SetOfAtomStructs*) para poder utilizar las estructuras en otros protocolos de Scipion.

La Figura 13 muestra un ejemplo, donde se puede ver cómo el protocolo de Protenix proporciona los objetos correspondientes, que pueden ser utilizados como entrada para otros protocolos. Además en la sección *Summary* se puede observar el resumen de la puntuación de cada predicción, como se ha mencionado anteriormente.

3.2. Funcionamiento de protocolos

Para evaluar la robustez de la implementación, se realizaron varias pruebas de ejecución con distintas moléculas. Se prestó especial atención a la gestión de archivos temporales, uso de recursos de *hardware* y capacidad de recuperación ante errores.

Durante las ejecuciones, se confirmó la creación automática de los archivos JSON y YAML necesarios para la inferencia, sin errores de formato o sintaxis. Estos archivos se almacenan en el



Figura 13: Objetos entrada y salida de protocolo Protenix.

directorio del trabajo de cada protocolo, accesible a través del botón *Browse* en la parte superior de la interfaz.

Se validó que el sistema de colas de Scipion gestiona de forma adecuada la ejecución gradual de las herramientas. Esto permite que el sistema maneje la carga de trabajo de manera eficaz, evitando el desbordamiento de la memoria VRAM. Por último, la captura de *logs* detallados permite al usuario monitorizar el progreso de la predicción en tiempo real. En caso de fallo en la herramienta externa, Scipion detiene la ejecución, marca el protocolo en rojo y almacena el mensaje de error específico, facilitando así la depuración de la tarea.

3.3. Casos de uso

Para el diseño del proyecto se siguió la estructura descrita en la sección anterior, que incluye protocolos de importación, los protocolos de ejecución local de Biofold y por último la comparación de modelos con *find discrepancies*. El proyecto completo puede encontrarse en el Anexo A.

3.3.1. 9M1Z

La primera estructura analizada es MAP2K6 (PDB 9M1Z), que aparece con una etiqueta roja en el proyecto completo.

La Tabla 6 muestra la puntuación obtenida por cada estructura predicha por los distintos protocolos. La puntuación no es comparable entre herramientas ya que cada una utiliza una métrica interna distinta, sin embargo, puede ser utilizada para comparar estructuras dentro de cada herramienta. En el caso de AlphaFold3, la métrica se basa en el pLDDT medio en una

escala de 0 a 100, mientras que en las otras herramientas se basan en estimaciones del TM-score entre 0 y 1. En todas las herramientas una puntuación más alta indica que el modelo es mejor, subrayadas en la tabla.

	AlphaFold3	Chai-1	Protenix	Boltz-2	IntelliFold
Modelo 1	80.45	0.1700	0.1685	0.829	0.660
Modelo 2	80.07	0.1731	0.1685	0.826	0.645
Modelo 3	80.79	0.1730	0.1684	0.825	0.659
Modelo 4	80.05	0.1714	0.1684	0.823	0.664
Modelo 5	79.86	0.1729	0.1682	0.819	0.684

Tabla 6: Puntuación de cada herramienta para las estructuras predichas de 9M1Z.

Con estos resultados podemos observar que el mejor modelo no siempre es el primero, y para esta estructura la puntuación de los modelos no varía mucho, por lo que se puede concluir que los modelos no varían mucho intra-herramienta.

La Figura 14 muestra las mejores estructuras de cada herramienta con los colores básicos de AlphaFold3 (donde azules indican confianza, mientras que naranjas y rojos indican desconfianza), obtenidos con el *viewer* del protocolo.

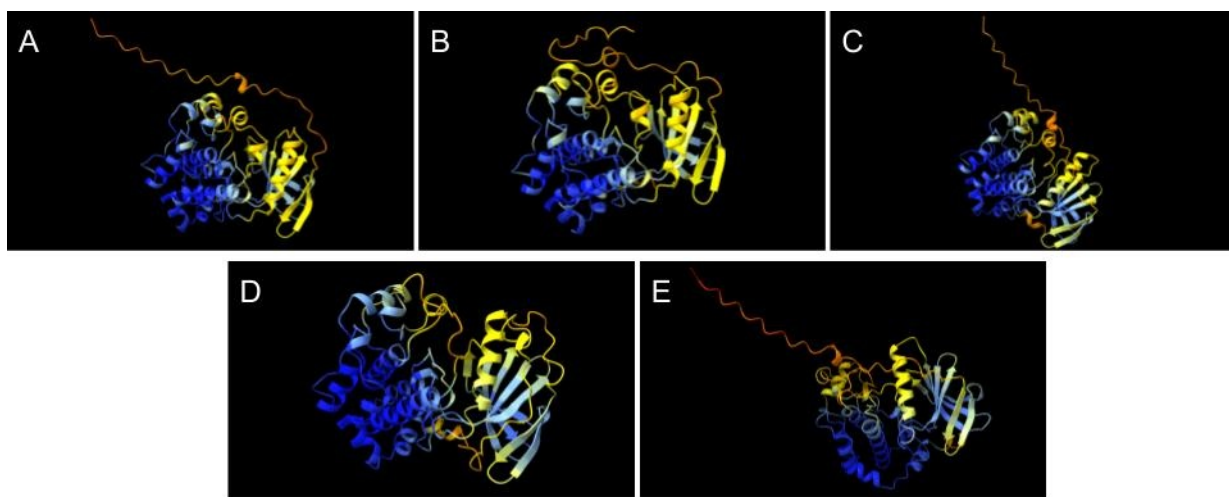


Figura 14: Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9M1Z.

Se puede observar cómo las estructuras A, C y E, correspondientes a las herramientas AlphaFold3, Protenix e IntelliFold, presentan el extremo flexible orientado hacia fuera de la estructura, sin embargo, las figuras B y D presentan este extremo hacia dentro de la estructura, y con mayor confianza, ya que están en amarillo en lugar de en naranja.

La Figura 15 muestra las predicciones de AlphaFold3 alineadas con el protocolo de Chimera, dónde verde oscuro muestra regiones cuyo RMSD entre modelos no varía mucho, y las blancas indican regiones cuyos RMSD varían significativamente. Las comparaciones se han realizado contra la estructura de referencia de AlphaFold3, aquella con mayor puntuación en la Tabla 6.

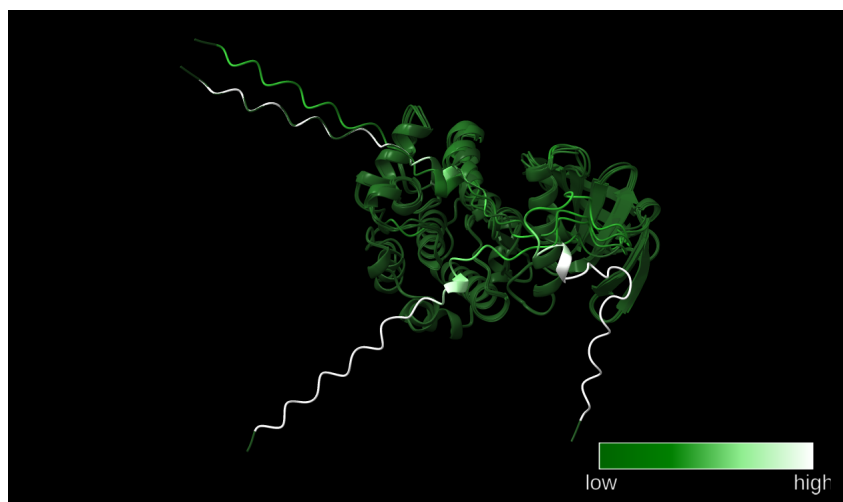


Figura 15: Diferencia estructural de los modelos de AlphaFold3 frente a estructura de referencia para 9M1Z.

La Figura 16 muestra las diferencias de RMSD por aminoácido de las estructuras de AlphaFold3, donde se puede observar que tiene valores bajos en casi toda la estructura, menores de 4 Å, salvo en los extremos (los aminoácidos del 0 al 34 y a partir del 331). En estas regiones se pueden observar valores dispares en cuanto a posición atómica con la estructura de referencia, especialmente en los primeros aminoácidos, donde se alcanza 104 Å, variación que se puede observar también en la Figura 15, correspondiente a la parte blanca de la estructura.

Gracias a la leyenda con colores se puede ver que el modelo 4 (Modelo 5 en la Tabla 6) es el que más difiere en los primeros aminoácidos, seguido del modelo 3 (Modelo 4 en la Tabla 6), mientras que el que más difiere en los últimos aminoácidos es el modelo 3 (Modelo 4 en la 6).

Las figuras y el análisis de las diferencias estructurales de los modelos predichos por las otras herramientas se pueden observar en el Anexo B junto a las gráficas de las diferencias de RMSD. En el análisis intra-herramienta se puede observar cómo las herramientas AlphaFold3, Protenix e IntelliFold generan modelos muy parecidos estructuralmente, salvo los extremos flexibles, que tienen diferencias de RMSD grandes, mientras que el resto de la estructura no llega a tener más de 5 Å de diferencia con la referencia en ningún aminoácido, salvo una zona específica de los modelos de IntelliFold. Chai-1 y Boltz-2 para la estructura 9M1Z genera modelos ligeramente distintos a la referencia en más zonas aparte de los extremos flexibles, especialmente Boltz-2, sin

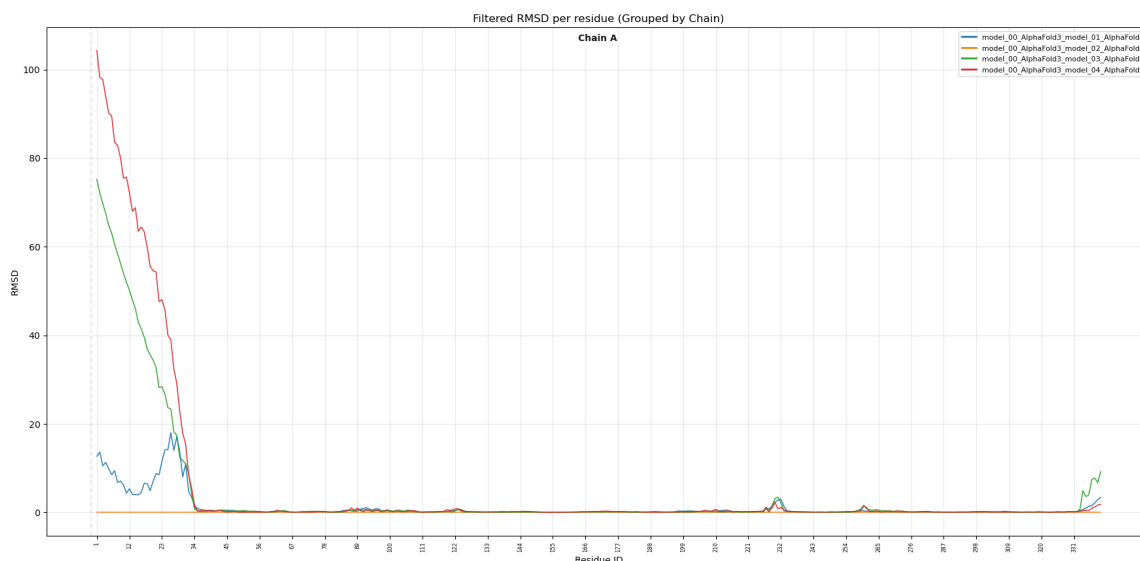


Figura 16: Diferencias en el RMSD por aminoácido de modelos de AlphaFold3 frente a estructura de referencia para 9M1Z.

embargo, las diferencias de RMSD no son mayores de 12 Å. Además, fijándose en la puntuación de los modelos de la Tabla 6, no se encuentran diferencias de puntuación notables entre los modelos dentro de cada herramienta.

Por último se aplicó el protocolo *find discrepancies* a los mejores modelos de cada herramienta utilizando como referencia la estructura descargada del PDB [44]. La Figura 17 muestra las estructuras alineadas en Chimera, donde se puede observar que las zonas flexibles de los extremos parecen ser las que mayor diferencia presentan frente a la estructura de referencia. Además, hay regiones que aparecen pintadas en verde claro en las zonas centrales de la estructura.

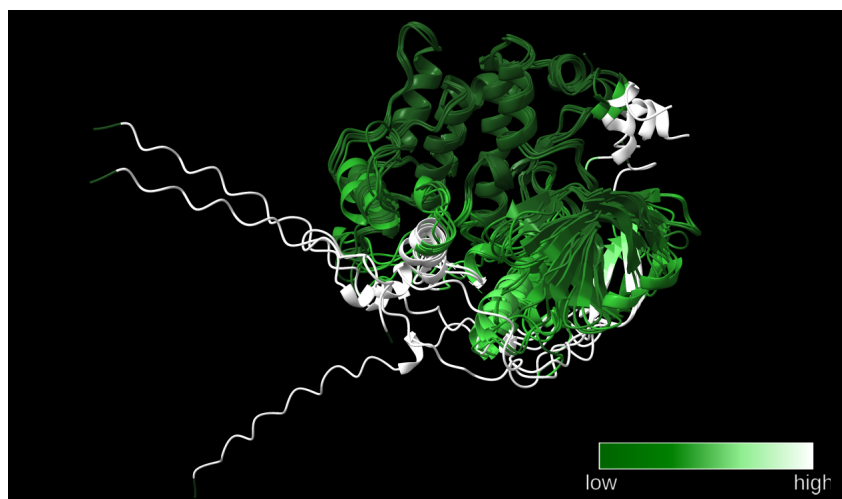


Figura 17: Diferencia estructural de los mejores modelos de 9M1Z frente a estructura de referencia.

La Figura 18 muestra las diferencias de RMSD por aminoácido frente a la estructura de referencia. Los aminoácidos del 0 al 38 y aquellos del final no aparecen con valores de RMSD ya que en las estructuras subidas a PDB, no se suelen añadir las partes flexibles de los extremos debido a que son las más complicadas de obtener y no se tiene una estructura clara de ellas. En la gráfica se puede observar que en los primeros aminoácidos se obtienen valores cercanos a 23 Å, zonas que serían justo la continuación de los aminoácidos blancos de la Figura 17. Además hay dos rangos de aminoácidos con diferencias notables en todos los modelos, del 83 al 101 y del 191 al 236 aproximadamente, donde se alcanzan valores cercanos a 15 Å. Se pueden observar diferencias en casi toda la estructura, salvo algunas regiones que apenas llegan a los 2 Å, y no se podría concluir exactamente cuál es el modelo más similar o más dispar con la estructura de referencia. Hay una zona de interés entre los aminoácidos 200 y 210 aproximadamente donde se ve una línea recta que los une, esto se debe a que esta región de la estructura de referencia no aparece en el archivo PDB, igual que los extremos. Esta región se puede observar de forma fácil en la Figura 10A, donde hay una parte en el centro de la imagen que aparece en discontinua.

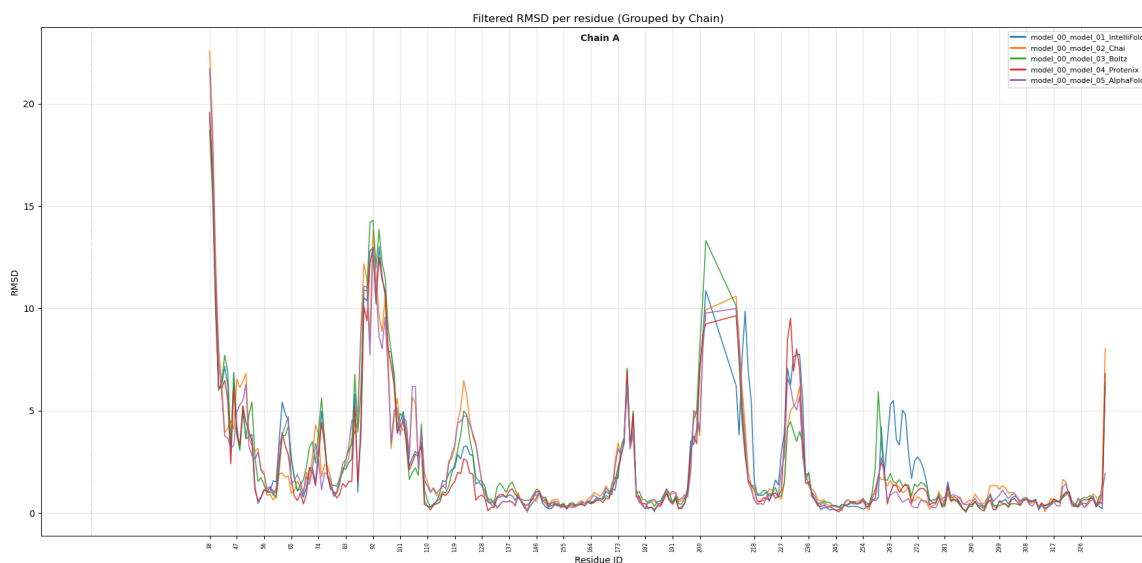


Figura 18: Diferencias en el RMSD por aminoácido de modelos de 9M1Z frente a estructura de referencia.

3.3.2. 9NZ9

La segunda estructura analizada es la ADN-liasa OGG1 (PDB 9NZ9), con una etiqueta naranja en el proyecto completo.

La Tabla 7 muestra las puntuaciones de los modelos para cada herramienta. En este caso se puede observar de nuevo que las mejores estructuras no son siempre las primeras, y para

	AlphaFold3	Chai-1	Protenix	Boltz-2	IntelliFold
Modelo 1	96.86	0.9436	0.9624	0.978	0.914
Modelo 2	96.58	0.9436	0.9623	0.978	0.919
Modelo 3	96.57	0.9440	0.9623	0.977	0.917
Modelo 4	96.30	0.9433	0.9623	0.977	0.920
Modelo 5	96.55	0.9435	0.9623	0.977	0.916

Tabla 7: Puntuación de cada herramienta para las estructuras predichas de 9NZ9.

este caso específico, se repiten puntuaciones entre modelos, especialmente en Protenix, donde cuatro modelos comparten la misma puntuación. Las puntuaciones intra-herramienta no varían mucho, apenas unos decimales de diferencia entre la mejor y la peor puntuación, e incluso en Protenix y Boltz-2 se presentan sólo dos puntuaciones, la máxima y otra que se comparte entre los demás modelos.

La Figura 19 muestra los mejores modelos de cada herramienta. Todas las estructuras muestran confianza en toda la estructura proteica, salvo en los extremos y las regiones flexibles, donde presentan menos confianza. Además la estructura de AlphaFold3 presenta menos confianza en la doble hélice que las otras herramientas y es la que presenta la hélice más distinta de todas.

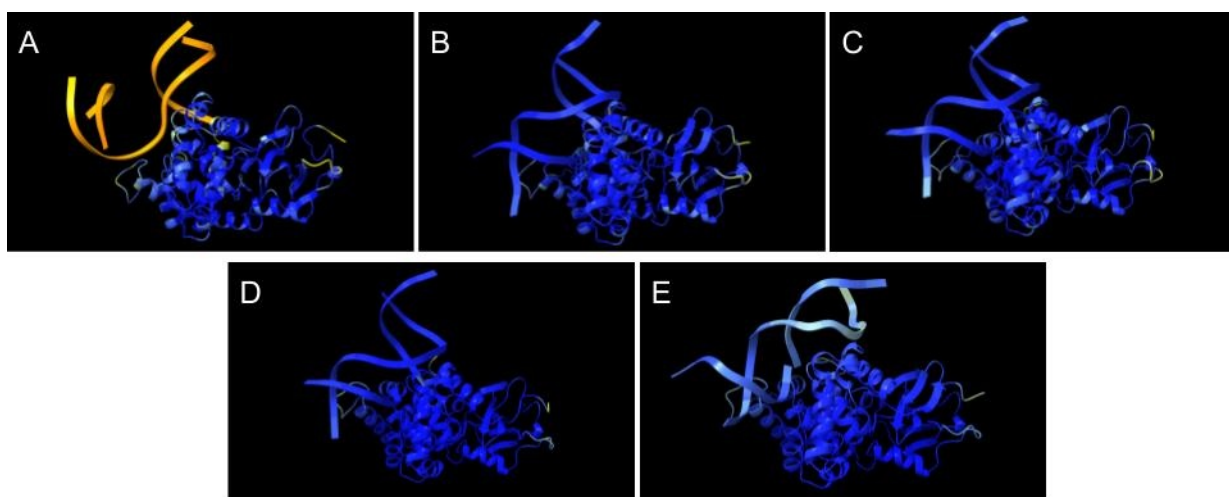


Figura 19: Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9NZ9.

Las figuras y el análisis de las diferencias estructurales intra-herramienta para la estructura 9NZ9 se pueden observar en el Anexo B junto a las gráficas de las diferencias de RMSD. En el análisis intra-herramienta se puede observar que AlphaFold3 presenta las cadenas A, B y C con apenas diferencias de RMSD, sin embargo en la cadena D presenta diferencias de hasta 25 Å.

Las herramientas Chai-1, Boltz-2 e IntelliFold obtienen modelos de salida muy parecidos, salvo regiones puntuales dónde el RMSD puede llegar a alcanzar hasta 5 Å de diferencia. Por último, Protenix presenta modelos muy parecidos donde las diferencias de RMSD apenas alcanzan los 1.2 Å, pese a tener los modelos la misma puntuación.

Al aplicar el protocolo de *find discrepancies* con la estructura del PDB como referencia, junto con las mejores estructuras de cada herramienta, se observan diferencias notables en el posicionamiento de la doble hélice, como se muestra en la Figura 20 [45]. Además se puede observar un nucleótido colocado hacia el exterior de la hélice por Chai-1, coloreado en blanco en el centro de la estructura.

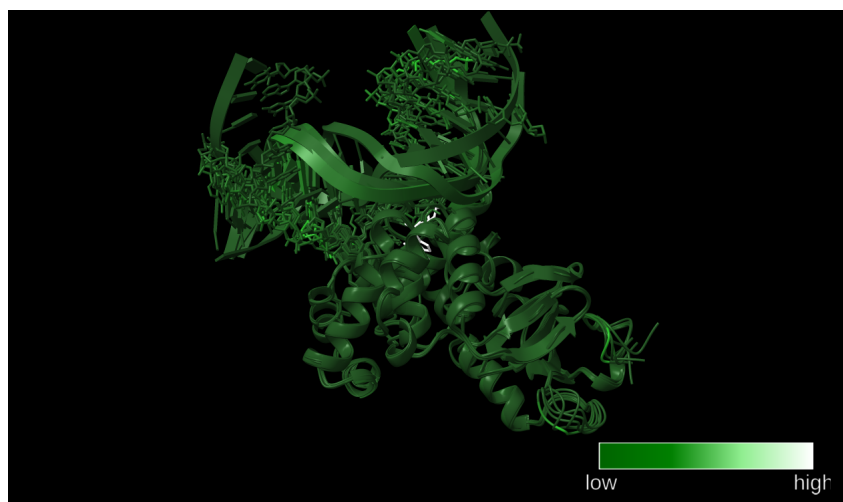


Figura 20: Diferencia estructural de los modelos de 9NZ9 frente a estructura de referencia.

En la Figura 21 se muestran las diferencias de RMSD con la estructura de referencia del PDB. En la cadena A se observan diferencias especialmente en el extremo inferior, con hasta 6.8 Å en el modelo de AlphaFold3, y una región del aminoácido 67 al 78 donde se alcanzan diferencias de 6 Å para el modelo de Chai-1, con Protenix presentando la menor diferencia con menos de 3 Å. En general, la cadena A no presenta diferencias notables, y las diferencias que hay, son compartidas por todos los modelos, salvo la región de los aminoácidos del 276 al 287, donde solo el modelo de IntelliFold presenta diferencias apenas notables con casi 2.5 Å de diferencia. Las cadenas B y C no tienen diferencias altas con la estructura de referencia, sin embargo la cadena D sí presenta diferencias notables en el modelo de AlphaFold3, con hasta 19 Å de diferencia, y el modelo de IntelliFold que alcanza en su diferencia máxima casi 10 Å. Para 9NZ9 se puede concluir que el modelo de AlphaFold3 es el que mayor diferencia tiene con el de referencia, en especial la doble hélice, donde se puede observar en la Figura 19A que es la

herramienta que menor confianza tiene para esta región.

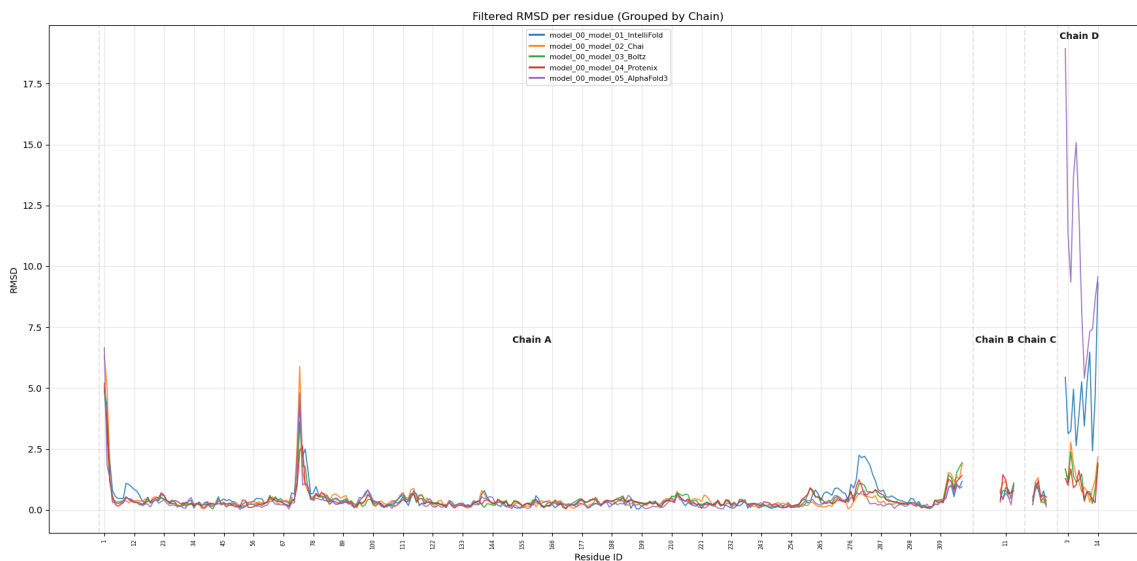


Figura 21: Diferencias en el RMSD por aminoácido de modelos de 9NZ9 frente a estructura de referencia.

3.3.3. 9QS2

La última estructura es la proteasa NS2B-NS3 del virus Zika (PDB 9QS2), con la etiqueta amarilla en el proyecto completo.

En la Tabla 8 se muestra la puntuación de cada modelo por herramienta, donde, de nuevo, no hay mucha diferencia en las puntuaciones, que apenas varía unos decimales.

	AlphaFold3	Chai-1	Protenix	Boltz-2	IntelliFold
Modelo 1	91.87	0.8677	0.8956	0.879	0.807
Modelo 2	91.88	0.8671	0.8956	0.871	0.796
Modelo 3	91.84	0.8675	0.8955	0.871	0.802
Modelo 4	91.78	0.8674	0.8954	0.871	0.798
Modelo 5	91.80	0.8670	0.8954	0.867	0.802

Tabla 8: Puntuación de cada herramienta para las estructuras predichas de 9QS2.

La Figura 22 muestra los mejores modelos de cada herramienta. Todos presentan una región central que parece tener mayor confianza (en azul oscuro) mientras que a medida que se llega a los extremos, presentan menos confianza (en amarillo). De nuevo se observa que Boltz-2 trata de integrar los extremos en la estructura en vez de colocarlos hacia el exterior.

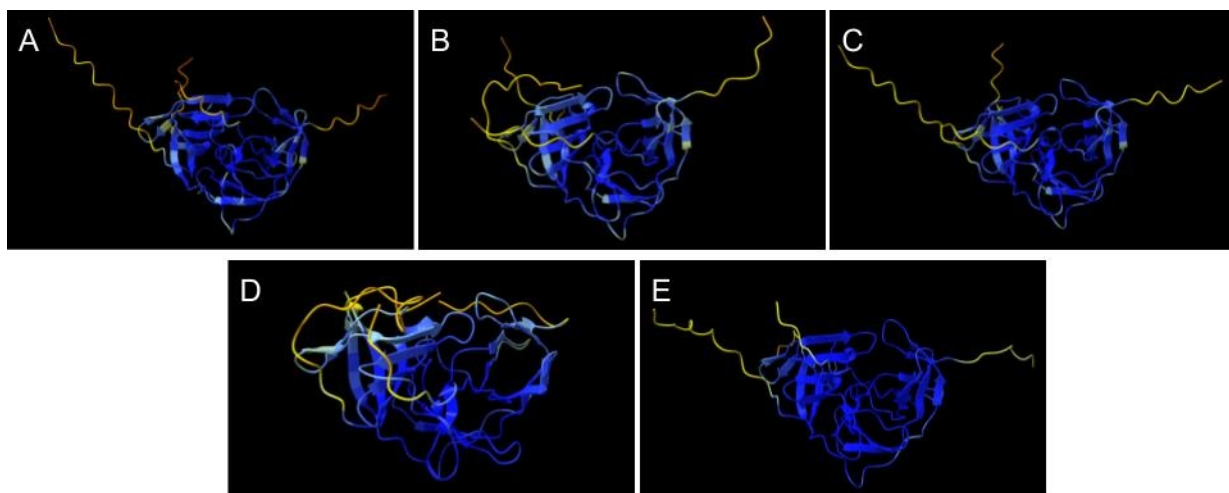


Figura 22: Mejores estructuras de cada herramienta (A: AlphaFold3, B: Chai-1, C: Protenix, D: Boltz-2, E: IntelliFold) para 9QS2.

Las figuras y el análisis de las diferencias estructurales de las otras estructuras para 9QS2 se pueden observar en el Anexo B junto a las gráficas de las diferencias de RMSD. El análisis intra-herramienta demostró que la zona dónde más difieren los modelos son los extremos, alcanzando hasta 60 Å en uno de los modelos de IntelliFold. Todas las herramientas presentan la misma tendencia, donde la mayoría de la estructura apenas alcanza 3 Å de diferencia salvo en los extremos.

Al aplicar `find discrepancies` sobre las mejores estructuras de cada herramienta con la referencia de PDB, se obtuvieron los resultados de las Figuras 23 y 24 [46].



Figura 23: Diferencia estructural de los modelos de 9QS2 frente a estructura de referencia.

En la Figura 23 se muestra la representación en Chimera, donde los extremos aparecen en blanco, indicando que difieren de la estructura de referencia, sin embargo, esto se puede deber

a que estos extremos no se incluyen en la estructura de referencia, ya que en la Figura 24 la cadena A comienza en el aminoácido 7, y la cadena B en el 17 aproximadamente. La cadena A apenas muestra diferencias, ya que no llega a alcanzar 1 Å ni en los extremos, sin embargo, la cadena B sí que muestra más diferencias. El extremo inferior muestra un RMSD de hasta 6 Å en los aminoácidos del 27 al 33, especialmente en los modelos de AlphaFold3, Protenix y Boltz-2, mientras que Chai-1 e IntelliFold no llegan a 3 Å. A lo largo de la cadena y en el extremo superior no se alcanzan más de 1.6 Å, por lo que los modelos sí se parecen a la referencia en estas zonas.

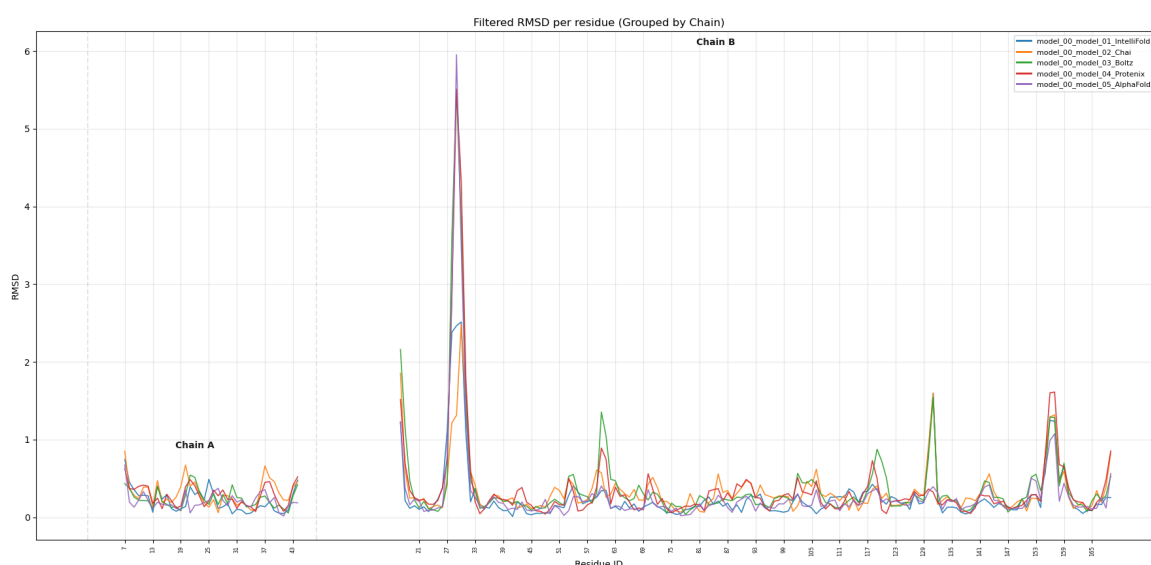


Figura 24: Diferencias en el RMSD por aminoácido de modelos de 9QS2 frente a estructura de referencia.

3.4. Análisis y observaciones

El análisis de las tres estructuras permite extraer conclusiones sobre el rendimiento y el comportamiento predictivo de las herramientas integradas en el *plugin* Biofold. En todas las ejecuciones se ha observado que los cinco modelos generados por una misma herramienta presentan una topología central casi idéntica, con variaciones mínimas (inferiores a 2-3 Å). Las variaciones se concentran en regiones desordenadas, *loops* y extremos terminales, demostrando la estabilidad de los algoritmos de muestreo, donde el ruido de la difusión altera la flexibilidad local pero mantiene el plegamiento principal.

Se demuestra que las puntuaciones internas no muestran oscilaciones abruptas dentro de un mismo grupo de modelos, sin embargo, en herramientas como Chai-1 y Boltz-2, una estructura

idéntica puede recibir menor puntuación por las penalizaciones probabilísticas latentes del estimador de confianza.

Mientras que para el monómero (9M1Z) o el complejo proteico (9QS2) las herramientas obtuvieron modelos muy próximos a la realidad del PDB, para la ADN-liasa (9NZ9) se obtuvieron modelos más divergentes. AlphaFold3 e IntelliFold obtuvieron desviaciones de RMSD más altas (hasta 19-25 Å) en la doble hélice, regiones donde la confianza descendió en los índices locales. Sin embargo, Protenix o Chai-1 demostraron mayor robustez en el modelado de interfaces proteína-ácido nucleico.

Estos casos de uso han destapado un factor crítico en la biología estructural, las discrepancias de RMSD masivas en los extremos no siempre implican un error de la inteligencia artificial. En la mayoría de ocasiones, se debe a que estas regiones carecen de densidad electrónica en los experimentos, por lo que están ausentes en los archivos depositados en el PDB.

3.5. Comparativa cuantitativa global de herramientas

La Tabla 9 muestra una serie de parámetros obtenidos para cada herramienta al comparar las mejores predicciones con las estructuras de referencia descargadas del PDB. Los bloques en los que se han obtenido diferencias de 0 Å se han omitido para realizar esta comparativa, ya que significan ausencia de datos. La columna de tiempo refleja el tiempo total que tarda cada protocolo en procesar todos los pasos. Para AlphaFold3 se ha puesto una raya (—) ya que, al usar su servidor web, no se tiene control total de tiempo ni listas de espera.

En el caso de 9M1Z, los valores medios de RMSD se encuentran entre 2.28 y 2.53 Å, por lo que no varían mucho. Sin embargo, el RMSD máximo alcanza los 25 Å en Chai-1, fenómeno asociado a la flexibilidad de los extremos. La coincidencia de los residuos con valores de RMSD mínimos en AlphaFold3 y Chai-1, y en Boltz-2 e IntelliFold sugiere una convergencia robusta en la estructura secundaria. Esta concordancia geométrica se aprecia en la Figura 25, donde el núcleo globular de todos los modelos se acopla casi a la perfección sobre la referencia en gris, mientras que los extremos desordenados divergen de forma aleatoria en el espacio. AlphaFold3 aparece en azul, Chai-1 en rosa, Protenix en amarillo, Boltz-2 en salmón e IntelliFold en verde.

Para 9NZ9, que incluye múltiples cadenas incluyendo la doble hélice, se observan valores bajos de RMSD medio salvo para AlphaFold3, ya que si recordamos la Figura 21, es el modelo que presenta mayor diferencia en la cadena D, afectando a este valor medio. Las otras herramientas presentan valores similares en los demás parámetros, localizándose el residuo mínimo siempre

Caso	Herramienta	RMSD Med (Å)	RMSD Máx (Å)	RMSD Mín (Å)	Res. Mín	Tiempo
9M1Z	AlphaFold3	2.2858	21.7021	0.1016	143	–
	Chai-1	2.2891	25.4921	0.0847	143	4' 27"
	Boltz-2	2.5298	18.7008	0.0822	288	2' 36"
	Protenix	2.3789	19.5879	0.0545	246	2' 40"
	IntelliFold	2.5221	19.3212	0.0387	288	3' 34"
9NZ9	AlphaFold3	1.2582	18.9397 (D:1)	0.0131 (A:51)	A:51	–
	Chai-1	0.4851	6.3454 (A:1)	0.0459 (A:275)	A:275	6' 13"
	Boltz-2	0.4856	4.9964 (A:1)	0.0397 (A:41)	A:41	3' 16"
	Protenix	0.4184	5.1999 (A:1)	0.0376 (A:171)	A:171	3' 20"
	IntelliFold	0.6974	9.3405 (D:13)	0.0380 (A:198)	A:198	3' 38"
9QS2	AlphaFold3	0.2688	5.9533 (B:29)	0.0157 (A:41)	A:41	–
	Chai-1	0.3564	2.4836 (B:30)	0.0593 (A:27)	A:27	6' 15"
	Boltz-2	0.3204	5.5003 (B:29)	0.0365 (B:78)	B:78	2' 41"
	Protenix	0.3410	5.5120 (B:29)	0.0410 (B:48)	B:48	2' 13"
	IntelliFold	0.2943	2.5132 (B:30)	0.0099 (B:41)	B:41	4' 00"

Tabla 9: Comparativa cuantitativa global de rendimiento y precisión estructural de las herramientas en los tres casos de uso.

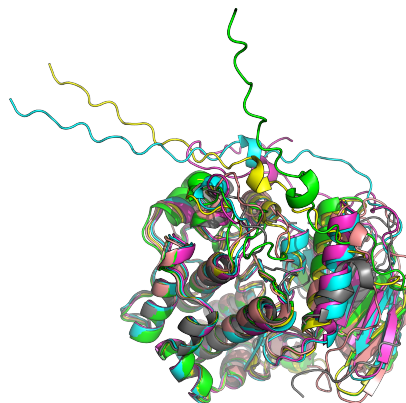


Figura 25: Visualización de superposición múltiple para 9M1Z. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores

en la cadena A. Estos valores indican que las herramientas predicen la estructura proteica con mayor precisión y confianza que las interfaces de ácidos nucleicos. De nuevo se pueden observar estas diferencias superpuestas sobre la estructura de referencia en la Figura 26 con los mismos colores que el caso anterior.

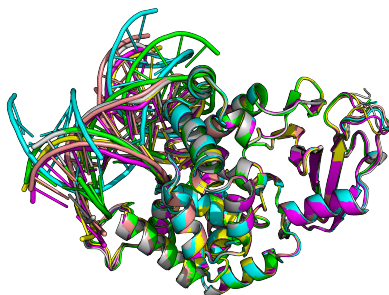


Figura 26: Visualización de superposición múltiple para 9NZ9. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores

Por último, la estructura 9QS2 muestra los valores medios de RMSD más bajos de los tres casos de uso. Todas las herramientas presentan la mayor diferencia de RMSD concentrada en los residuos B:29 o B:30, exhibiendo valores máximos contenidos en comparación con los presentados en 9M1Z o 9NZ9. Los residuos con valores mínimos de RMSD varían entre las herramientas: mientras que AlphaFold3 y Chai-1 lo presentan en la cadena A, las demás herramientas lo localizan en la cadena B, aunque ningún algoritmo coincide en el residuo específico. La Figura 27 muestra estas diferencias con distintos colores sobre la estructura de referencia en gris.

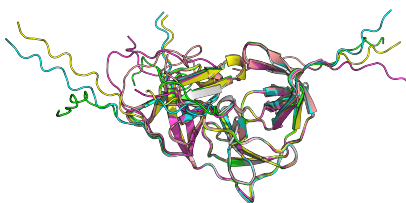


Figura 27: Visualización de superposición múltiple para 9QS2. Estructura de referencia del PDB en gris y los mejores modelos de cada herramienta superpuestos en distintos colores

4 Discusión y perspectivas futuras

4.1. Alcance metodológico y evaluación de las capacidades predictivas

Los resultados obtenidos por los tres casos de uso ofrecen datos suficientes para sacar conclusiones sobre las herramientas integradas en Biofold. Sin embargo, hay que recalcar que las observaciones discutidas son exclusivas para este proyecto, y no deben ser generalizadas como leyes universales aplicables a cualquier familia de proteínas y macromoléculas.

Uno de los hallazgos más importantes al analizar el RMSD por residuo en estructuras de IntelliFold o Boltz-2 (con picos superiores a 25-60 Å en los extremos), es que estas desviaciones no representan un fallo exclusivamente de la IA, sino que refleja un choque metodológico. Las herramientas predictivas tratan de predecir la cadena completa, mientras que las estructuras depositadas en el PDB a menudo carecen de las coordenadas en los extremos debido a la ausencia de la densidad electrónica en los experimentos. Debido a esto, los usuarios deben interpretar con cautela estos valores, ya que la baja correspondencia puede ser reflejo de un acierto del modelo al capturar la flexibilidad y no un error algorítmico.

El comportamiento observado en el caso de la ADN-liasa (PDB 9NZ9) plantea un debate técnico. El hecho de que herramientas consolidadas como AlphaFold3 muestren grandes desviaciones en la doble hélice (19-25 Å), frente a valores estables observados en otros modelos como Protenix o Chai-1, demuestra que el rendimiento de estas herramientas viene influenciado por sesgos moleculares y conceptuales de sus datos de entrenamiento. Esto sugiere que actualmente no existe un predictor universal, sino que algunos modelos han madurado en el reconocimiento de las interacciones con ligandos o ácidos nucleicos mientras que otros destacan en el núcleo globular peptídico.

De igual manera, el análisis del complejo de la proteasa del Zika (PDB 9QS2) destaca un fenómeno de consenso algorítmico de gran interés. El hecho de que tres de las herramientas (AlphaFold3, Boltz-2 y Protenix) concentren su máxima desviación exactamente en el residuo 29 de la cadena B, mientras que las otras dos (Chai-1 e IntelliFold) lo ubiquen en el residuo contiguo 30 bajando el error a la mitad, evidencia cómo regiones intrínsecamente flexibles en las interfaces moleculares pueden bifurcar las soluciones topológicas de las redes neuronales.

Esto demuestra que la consistencia de estas topologías centrales convive con las penalizaciones en los estimadores de confianza. Esto es crítico, ya que los usuarios no deben guiar su toma de decisiones únicamente por la puntuación numérica, sino que deben visualizar las estructuras para validar una predicción.

4.2. Ventajas de la integración en Scipion-chem

Hasta el momento, Scipion estaba limitado al uso de estructuras ya conocidas y depositadas en bases de datos experimentales. La incorporación de herramientas de predicción estructural soluciona el problema cuando se desea trabajar con proteínas o complejos cuya estructura tridimensional es desconocida, partiendo únicamente de su secuencia.

El *plugin* de Biofold supone una gran ventaja, no solo para poder trabajar en Scipion con estructuras desconocidas, sino que al integrar protocolos de ejecución local y de importación, permite el uso sencillo de distintas herramientas predictivas sin necesidad de cambiar constantemente de entornos o de importar y exportar archivos específicos. La instalación es sencilla y rápida, integrándose adecuadamente en el entorno de Scipion para facilitar el uso de los usuarios.

Precisamente, la ventaja diferencial de Biofold radica en mitigar la incertidumbre predictiva descrita en los casos de uso; al unificar los cinco protocolos bajo un *workflow* idéntico, se concede al investigador la capacidad de realizar validaciones cruzadas y comparativas ortogonales sin coste técnico, explotando las fortalezas específicas de cada herramienta (como la robustez de Protenix en ácidos nucleicos o la afinidad local de IntelliFold en *loops*) en función de la naturaleza de su sistema de estudio.

En el caso del protocolo de importación, solo es necesario descargar el archivo comprimido de los servidores web, facilitando a los usuarios el uso de estas estructuras evitando la búsqueda de archivos en directorios locales complejos. Adicionalmente, los protocolos de ejecución local solamente requieren de secuencias de entrada y parámetros simples para poder realizar las predicciones, eliminando la necesidad de interactuar con la línea de comandos o conocer la sintaxis exacta de los ejecutables originales. Finalmente, la estandarización de salidas en objetos de Scipion permite que los modelos puedan ser empleados en análisis posteriores.

4.3. Limitaciones actuales

A pesar de que estos modelos de inteligencia artificial pueden ejecutarse utilizando solamente CPU, tardan más tiempo. Para una inferencia eficiente, se requiere de GPUs con gran potencia, además de un almacenamiento masivo para los modelos descargados de cada herramienta. Si el usuario no tiene acceso a estos recursos, el potencial de los protocolos de ejecución local es limitado, sin embargo, el protocolo de importación permite seguir utilizando predicciones estructurales directamente importadas desde servidores locales, por lo que Biofold no limita su uso exclusivamente a usuarios con recursos de *hardware*.

Una limitación de estos modelos es su funcionamiento interno, pese a que en la mayoría de casos el código fuente es público, el funcionamiento interno de las redes neuronales no es sencillo de interpretar por la mayoría de usuarios. Este problema se acentúa en herramientas como AlphaFold3, cuyas restricciones comerciales restringen el acceso libre a su código y limitan su uso local.

4.4. Posibles líneas futuras

Para seguir mejorando el funcionamiento de Biofold, se plantea la inclusión de herramientas que faciliten la comparación directa de los modelos, como un protocolo de puntuación propio de Scipion que unifique las métricas de confianza internas bajo un mismo baremo matemático. Además, la estructura modular de Scipion permite la inclusión de nuevas herramientas predictivas, así como la compatibilidad con otros servidores web de predicción.

Actualmente, los protocolos no permiten la entrada de ligandos para las predicciones, pese a que las herramientas lo permiten. Esto se debe a que Scipion-chem ya incluye una gran variedad de herramientas de *docking* molecular, por lo que para este proyecto, el principal interés era la predicción estructural, no el *docking*. Sin embargo, sería interesante añadir la opción de incluir la entrada de ligandos para comparar con otras herramientas de *docking* tradicionales.

Por último, a nivel académico y metodológico, resulta de interés generar un *benchmark* masivo utilizando estructuras del PDB frente a estas cinco herramientas. Esto permitiría evaluar de forma estadística el potencial predictivo y los sesgos de cada algoritmo ante diferentes familias de proteínas.

5 Conclusiones

Se ha diseñado, programado e implementado con éxito el *plugin* Biofold dentro de Scipion-chem, logrando la unificación de cinco herramientas de predicción estructural basadas en inteligencia artificial más avanzadas del estado del arte: AlphaFold3, Chai-1, Protenix, Boltz-2 e IntelliFold.

La interfaz gráfica de Scipion ha permitido el desarrollo eficaz del *plugin*, gestionando archivos auxiliares, la configuración de parámetros y la sintaxis de ejecución. Esto democratiza el acceso a herramientas bioinformáticas complejas, eliminando la necesidad de interactuar con la línea de comandos o configuración de entornos virtuales independientes.

Gracias a la dualidad del *plugin* (importaciones o ejecución local), Biofold se consolida como una solución versátil. Permite el máximo rendimiento computacional en entornos de alta computación, y ofrece una alternativa viable para usuarios con recursos de *hardware* limitados.

Se ha validado la integración del flujo de datos en Scipion mediante la conversión de las salidas en objetos estándar de la plataforma. Esto garantiza que los modelos generados por las herramientas de predicción puedan encadenarse con otros protocolos. La combinación de los protocolos de Biofold con el protocolo `find discrepancies` permite discriminar y mapear con precisión las diferencias sutiles de puntuación y las variaciones de RMSD frente a estructuras experimentales del PDB.

Por último, los casos de uso reales confirman la solidez del *plugin* al procesar desde monómeros sencillos a estructuras complejas con ácidos nucleicos. Además, el *workflow* demostró una gran madurez metodológica al mostrar que las discrepancias grandes de RMSD en los extremos se deben a limitaciones de los archivos del PDB y no a fallos en la predicción de la herramienta, evidenciando la necesidad de contar con una plataforma con múltiples herramientas que permita identificar puntos de consenso estructural e inestabilidad local en interfaces macromoleculares complejas.

6 Bibliografía

- [1] Wuyun, Q., Chen, Y., Shen, Y., Cao, Y., Hu, G., Cui, W., Gao, J., & Zheng, W. (2024). Recent Progress of Protein Tertiary Structure Prediction. *Molecules*, 29(4), 832. DOI: <https://doi.org/10.3390/molecules29040832>
- [2] AlQuraishi, M. (2021). Machine learning in protein structure prediction. *Current Opinion in Chemical Biology*, 65, 1-8. DOI: <https://doi.org/10.1016/j.cbpa.2021.04.005>
- [3] Honig, B. (1999). Protein folding: from the levinthal paradox to structure prediction. *Journal of Molecular Biology*, 293(2), 283-293. DOI: <https://doi.org/10.1006/jmbi.1999.3006>
- [4] Bertoline, L. M. F., Lima, A. N., Krieger, J. E., & Teixeira, S. K. (2023). Before and after AlphaFold2: An overview of protein structure prediction. *Frontiers in Bioinformatics*, 3. DOI: <https://doi.org/10.3389/fbinf.2023.1120370>
- [5] Glusker, J. P. (1994). X-ray crystallography of proteins. *Methods of Biochemical Analysis*, 37, 1–72. DOI: <https://doi.org/10.1002/9780470110584.ch1>
- [6] Cavanagh, J., Fairbrother, W. J., & Skelton, N. J. (1996). *Protein NMR Spectroscopy: Principles and Practice*. Academic Press. ISBN: 978-0121644901
- [7] Cheng, Y. (2015). Single-Particle Cryo-EM at Crystallographic Resolution. *Cell*, 161(3), 450–457. DOI: <https://doi.org/10.1016/j.cell.2015.03.049>
- [8] Xu, D., & Zhang, Y. (2013). Toward optimal fragment generations for ab initio protein structure assembly. *Proteins: Structure, Function, and Bioinformatics*, 81(2), 229-239. DOI: <https://doi.org/10.1002/prot.24179>
- [9] Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology*, 287(4), 797-815. DOI: <https://doi.org/10.1006/jmbi.1999.2583>

- [10] Kiefer, F., Arnold, K., Kunzli, M., Bordoli, L., & Schwede, T. (2009). The SWISS-MODEL Repository and associated resources. *Nucleic Acids Research*, 37(suppl_1), D387-D392. DOI: <https://doi.org/10.1093/nar/gkn750>
- [11] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zidek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596, 583–589. DOI: <https://doi.org/10.1038/s41586-021-03819-2>
- [12] CASP Organizing Committee. (2020). *CASP14 Abstracts*. Critical Assessment of protein Structure Prediction (CASP14), Prediction Center. https://predictioncenter.org/casp14/doc/CASP14_Abstracts.pdf
- [13] Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., & Barzilay, R. (2025). Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction. *bioRxiv*. DOI: <https://doi.org/10.1101/2025.06.14.659707>
- [14] Boitreaud, J., Dent, J., McPartlon, M., Meier, J., Reis, V., Rogozhnikov, A., & Wu, K. (2024). Chai-1: Decoding the molecular interactions of life. *bioRxiv*. DOI: <https://doi.org/10.1101/2024.10.10.615955>
- [15] Chen, X., Zhang, Y., Lu, C., Ma, W., Guan, J., Gong, C., Yang, J., Zhang, H., Zhang, K., Wu, S., Zhou, K., Yang, Y., Liu, Z., Wang, L., Shi, B., Shi, S., & Xiao, W. (2025). Protenix - Advancing Structure Prediction Through a Comprehensive AlphaFold3 Reproduction. *bioRxiv*. DOI: <https://doi.org/10.1101/2025.01.08.631967>
- [16] Qiao, L., Yan, H., Liu, G., Guo, G., & Sun, S. (2026). IntelliFold-2: Surpassing AlphaFold 3 via Architectural Refinement and Structural Consistency. *bioRxiv*. DOI: <https://doi.org/10.64898/2026.02.09.704787>
- [17] Conesa, P., Fonseca, Y. C., Jimenez de la Morena, J., Sharov, G., de la Rosa-Trevin, J. M., Cuervo, A., Garcia Mena, A., Rodriguez de Francisco, B., del Hoyo, D., Herreros,

- D., Marchan, D., Strelak, D., Fernandez-Gimenez, E., Ramirez-Aportela, E., de Isidro-Gomez, F. P., Sanchez, I., Krieger, J., Vilas, J. L., del Cano, L., ... Sanchez Sorzano, C. O. (2023). Scipion3: A workflow engine for cryo-electron microscopy image processing and structural biology. *Acta Crystallographica Section D: Structural Biology*, 79(9), 847–859. DOI: <https://doi.org/10.1017/S2633903X23000132>
- [18] Del Hoyo, D., Salinas, M., Lomas, A., Ulzurrun, E., Campillo, N. E., & Sorzano, C. O. (2023). Scipion-Chem: An Open Platform for Virtual Drug Screening. *Journal of Chemical Information and Modeling*, 63(24), 7873-7885. DOI: <https://doi.org/10.1021/acs.jcim.3c01085>
- [19] JetBrains. (2026). *PyCharm: The Python IDE for Professional Developers*. <https://www.jetbrains.com/pycharm/>
- [20] GitHub. (2026). *GitHub: Where the world builds software*. <https://github.com/>
- [21] Lucid Software Inc. (2026). *Lucidchart: Intelligent diagramming platform*. <https://www.lucidchart.com/pages>
- [22] Overleaf. (2026). *Overleaf: Online LaTeX Editor and Collaboration Platform*. <https://www.overleaf.com/>
- [23] OpenAI. (2026). *OpenAI*. <https://openai.com/es-ES/>
- [24] Scipion development team. (2026). *Scipion EM: open-source software platform for cryo-electron microscopy*. Repositorio GitHub. <https://github.com/scipion-em>
- [25] Scipion development team. (2026). *Scipion App: Plugin manager for Scipion EM (cryo-EM workflow framework)*. Repositorio GitHub. <https://github.com/scipion-em/scipion-app>
- [26] Scipion development team. (2026). *scipion-pyworkflow: Workflow engine for scientific applications (Scipion framework)*. Repositorio GitHub. <https://github.com/scipion-em/scipion-pyworkflow>
- [27] Pearl, E. (2024). *The Illustrated AlphaFold*. Entrada de blog. <https://elanapearl.github.io/blog/2024/the-illustrated-alphafold/>

- [28] EMBL-EBI Training. (2025). *How does AlphaFold 3 work?*. Curso online EMBL-EBI. <https://www.ebi.ac.uk/training/online/courses/alphafold/alphafold-3-and-alphafold-server/introducing-alphafold-3/how-does-alphafold-3-work/>
- [29] DeepMind & Isomorphic Labs. (2024). *AlphaFold Server*. Plataforma web. <https://alphafoldserver.com/>
- [30] Google DeepMind. (2024). *AlphaFold 3: Inference pipeline codebase*. Repositorio GitHub. <https://github.com/google-deepmind/alphafold3>
- [31] Chai Discovery. (2026). *Chai Discovery*. Plataforma web. <https://www.chaidiscovery.com/>
- [32] Chai Discovery. (2026). *chai-lab*. Repositorio GitHub. <https://github.com/chaidiscovery/chai-lab>
- [33] Steinegger, M., & Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11), 1026–1028. DOI: <https://doi.org/10.1038/nbt.3988>
- [34] Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., & Steinegger, M. (2022). ColabFold: making protein folding accessible to all. *Nature Methods*, 19(6), 679–682. DOI: <https://doi.org/10.1038/s41592-022-01488-1>
- [35] ByteDance AI4Science Team. (2026). *Protenix Server*. Plataforma web. <https://protenix-server.com/>
- [36] ByteDance AI4Science Team. (2026). *Protenix: Toward High-Accuracy Open-Source Biomolecular Structure Prediction*. Repositorio GitHub. <https://github.com/bytedance/Protenix>
- [37] ProteinIQ. (2026). *Boltz-2 Online Server*. Plataforma web. <https://proteiniq.io/app/boltz-2>
- [38] Wohlwend, J., Corso, G., Passaro, S., Barzilay, R., & Jaakkola, T. (2026). *Boltz: Biomolecular Interaction Prediction Models*. Repositorio GitHub. <https://github.com/jwohlwend/boltz>

- [39] University of Reading Bioinformatics Group. (2026). *IntFOLD Server: Integrated Protein Structure and Function Prediction*. Servidor web. <https://server.intfold.com/>
- [40] IntelliGen-AI. (2026). *IntelliFold: Controllable Foundation Model for Biomolecular Structure Prediction*. Repositorio GitHub. <https://github.com/IntelliGen-AI/IntelliFold/>
- [41] Scipion Development Team. (2023). *Scipion 3 Architecture: Workflow Engine for Cryo-EM Image Processing and Structural Biology*. Documentación técnica. <https://scipion-em.github.io/docs/release-3.0.0/docs/developer/architecture.html>
- [42] RCSB Protein Data Bank. (2026). *RCSB PDB: Protein Data Bank Archive of 3D Macromolecular Structures*. <https://www.rcsb.org/>
- [43] Scipion Development Team. (2024). *scipion-em-chimera: Plugin to use UCSF ChimeraX within the Scipion framework*. Repositorio GitHub. <https://github.com/scipion-em/scipion-em-chimera>
- [44] RCSB Protein Data Bank. (2026). *Crystal Structure of MAP2K6 complexed with 5Z-7-oxozeaenol (PDB ID: 9M1Z)*. Base de datos PDB. <https://www.rcsb.org/structure/9M1Z>
- [45] RCSB Protein Data Bank. (2026). *Crystal structure of product-bound human OGG1 (PDB ID: 9NZ9)*. Base de datos PDB. <https://www.rcsb.org/structure/9NZ9>
- [46] RCSB Protein Data Bank. (2026). *Crystal structure of Zika virus NS2B-NS3 protease in complex with Z100582766 (PDB ID: 9QS2)*. Base de datos PDB. <https://www.rcsb.org/structure/9QS2>

7 Anexo A: Repositorios

7.1. Código fuente de *scipion-chem-biofold*

El código fuente del *plugin* está disponible de forma pública en GitHub, dentro de la organización de Scipion-chem, al cual se puede acceder a través de este enlace¹.

7.2. Proyecto de análisis

La carpeta con el proyecto y los datos de los casos de uso está disponible en un repositorio de GitHub².

La estructura del proyecto se organiza en directorios siguiendo el estándar de los proyectos de Scipion. El propósito de cada carpeta se detalla a continuación:

data: Contiene los archivos de datos utilizados en el proyecto, incluyendo los archivos comprimidos descargados del servidor de AlphaFold3 y las secuencias de las estructuras en formato FASTA.

Logs: Almacena los registros de ejecución (*logs*) del sistema y de los procesos asociados al proyecto.

Tmp: Destinada al almacenamiento de archivos y datos temporales generados durante las ejecuciones.

Uploads: Espacio reservado para almacenar de forma flexible archivos auxiliares aportados por el usuario.

Runs: Constituye el núcleo de ejecución, ya que almacena todos los protocolos contenidos en el proyecto. El nombre de cada subdirectorio combina un identificador único (ID) junto con el nombre del protocolo correspondiente. En su interior se guardan los archivos específicos de cada ejecución, tales como datos auxiliares, metadatos de los resultados e información crítica definida por los desarrolladores.

¹URL completa: <https://github.com/scipion-chem/scipion-chem-biofold>

²URL completa: https://github.com/blanca-pueche/Biofold_tfm

8 Anexo B: Análisis de casos de uso

Este anexo recopila el análisis estructural intra-herramienta detallado y las métricas de desviación para los diferentes casos de uso abordados en la memoria.

Antes de proceder con la evaluación, se deben tener en cuenta dos criterios metodológicos:

1. **Selección de referencia:** Se utiliza la estructura con mayor puntuación de confianza (*score*) de cada herramienta. En caso de empate, se selecciona el primer modelo según su orden de salida (por ejemplo, entre el modelo 2 y 3, se toma el 2).
2. **Numeración de los modelos:** La numeración en las figuras y gráficos de RMSD corresponde al identificador secuencial de Scipion (*model_1*, *model_2*, etc.). En cambio, en las tablas se ordenan de mayor a menor según su puntuación. Este cambio de criterio genera un desfase numérico entre imágenes y tablas que se detalla entre paréntesis en cada sección.

8.1. 9M1Z

8.1.1. AlphaFold3

Las Figuras 15 y 16 muestran las diferencias estructurales de los modelos predichos con el servidor de AlphaFold3 frente a la estructura de referencia descargada del PDB. Estas gráficas y visualizaciones ya han sido analizadas y comentadas detalladamente en la sección de Resultados (Capítulo 3, página 29).

8.1.2. Chai-1

Las Figuras 1 y 2 muestran las diferencias en los modelos predichos por Chai-1 para 9M1Z.

Se puede observar de nuevo que los extremos parecen ser los que más diferencias tienen, ya que aparecen en un verde más claro. Esta suposición puede ser confirmada con el gráfico de RMSD, donde se puede observar que las mayores diferencias se encuentran antes del aminoácido 45, llegando a un valor aproximado de RMSD de 50 Å en el modelo 1 (Modelo 1 en la Tabla 6). Las mayores diferencias se encuentran con el modelo 1 y el modelo 3 (Modelos 1 y 4 respectivamente en la Tabla 6), y a diferencia del caso de AlphaFold3, se pueden ver diferencias en toda la estructura, no solo en los extremos. Además, el modelo 4 (Modelo 5 en la Tabla 6) es exactamente igual que el modelo de referencia, pese a tener menor puntuación. Esto se

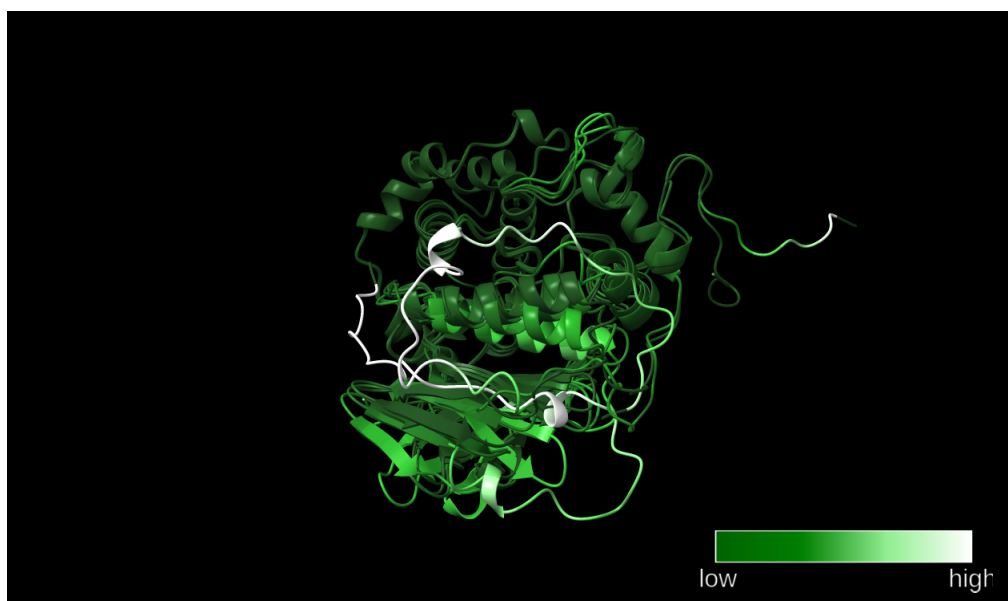


Figura 1: Diferencia estructural de los modelos de Chai-1 frente a estructura de referencia para 9M1Z.

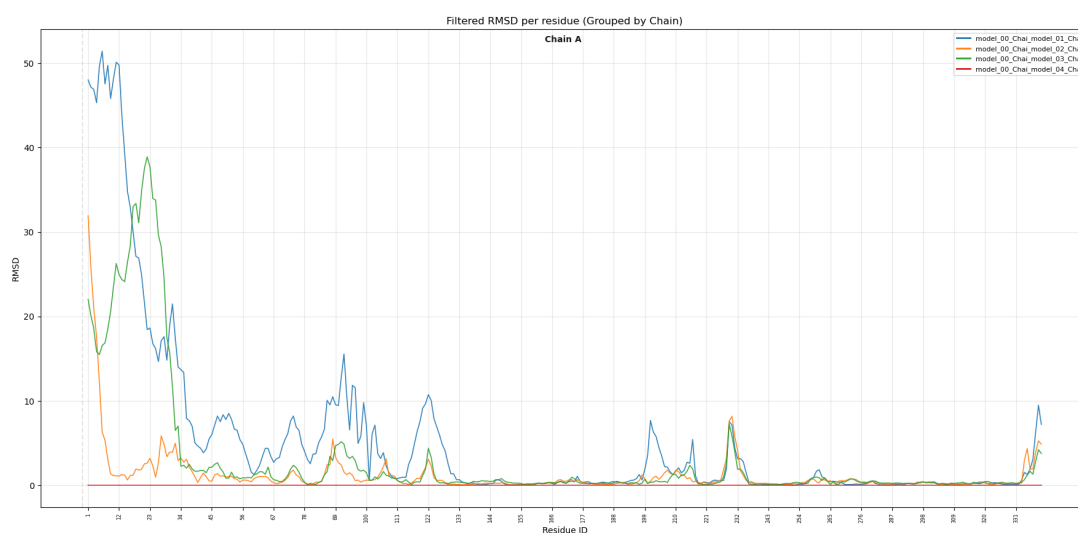


Figura 2: Diferencias en el RMSD por aminoácido de modelos de Chai-1 frente a estructura de referencia para 9M1Z.

debe a que el módulo de predicción geométrica y el módulo de confianza de Chai-1 actúan de forma independiente. Aunque la herramienta generó dos estructuras idénticas, el estimador de confianza introdujo pequeñas variaciones matemáticas debido al ruido propio del proceso de difusión, alterando la puntuación final en los decimales sin que existan cambios reales en la proteína.

8.1.3. *Protenix*

Para este caso se obtuvo la misma puntuación (0.1685) para los modelos 1 y 2 de la Tabla 6, por lo que se escogió como referencia el modelo 1, siguiendo la regla mencionada previamente.

Las Figuras 3 y 4 muestran estas diferencias estructurales, donde se obtienen resultados similares a los de AlphaFold3, con los extremos apuntando hacia fuera de la estructura. El extremo de los primeros aminoácidos vuelve a tener una diferencia de RMSD mayor que los finales, con un RMSD de aproximadamente 93 Å, siendo el modelo 2 (Modelo 3 en la Tabla 6) el que más difiere en esta parte, y en general parece que es la que más diferencias tiene con la estructura de referencia. Además, aparece de nuevo un caso similar al anterior donde el modelo 3 (Modelo 4 en la Tabla 6) parece ser la misma estructura que la referencia pero tiene una puntuación menor.

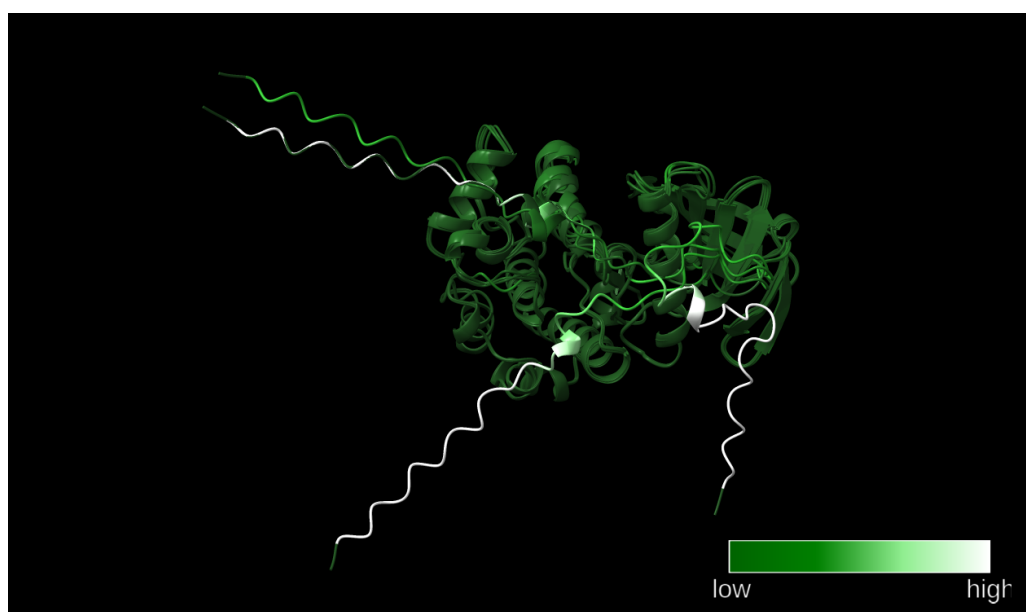


Figura 3: Diferencia estructural de los modelos de Protenix frente a estructura de referencia para 9M1Z.

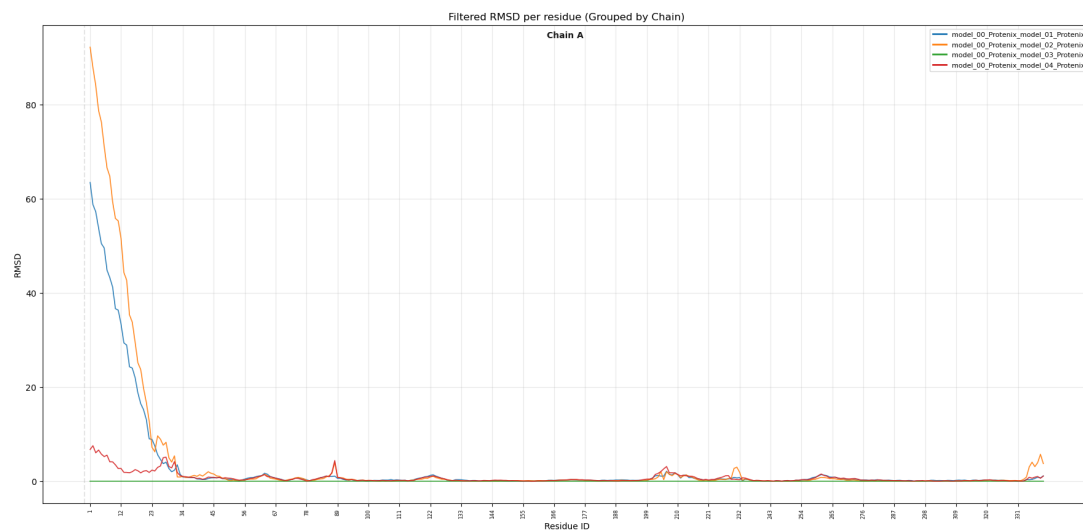


Figura 4: Diferencias en el RMSD por aminoácido de modelos de Protenix frente a estructura de referencia para 9M1Z.

8.1.4. Boltz-2

Las Figuras 5 y 6 muestran las diferencias en los modelos predichos por Boltz-2.

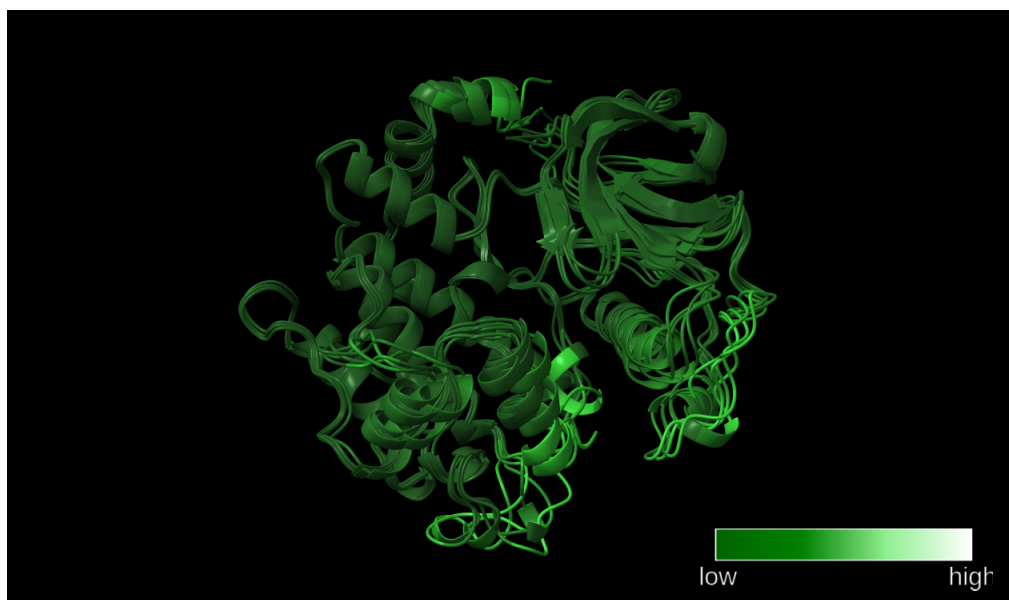


Figura 5: Diferencia estructural de los modelos de Boltz-2 frente a estructura de referencia para 9M1Z.

En la Figura 5 se puede observar que las estructuras parecen estar más unificadas, sin partes flexibles orientadas hacia el exterior, y que tiene un color uniforme, sin partes excesivamente claras o blancas. Si nos fijamos en la gráfica de la Figura 6 se puede ver que tiene más diferencia de RMSD en toda la estructura, siendo los aminoácidos aproximadamente en los rangos del 23

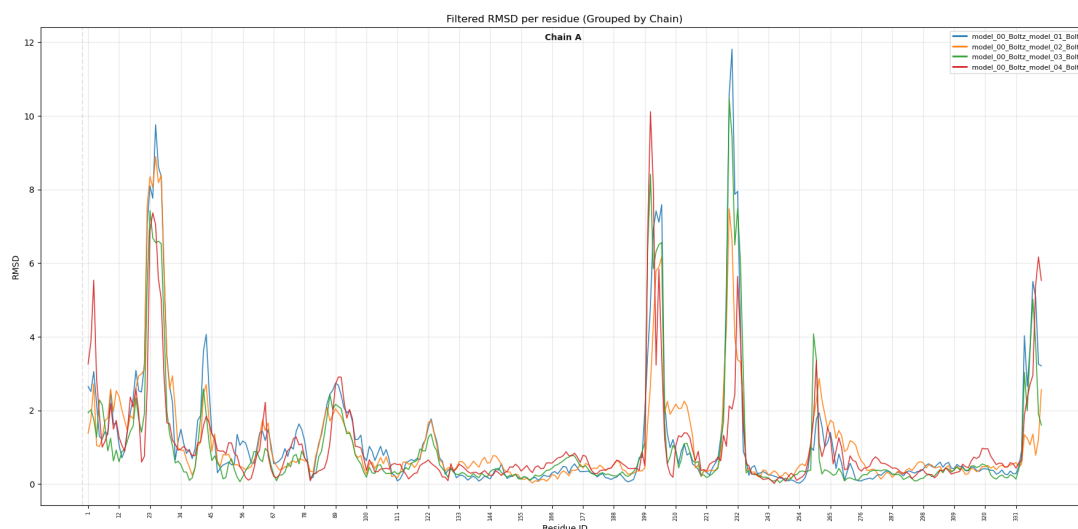


Figura 6: Diferencias en el RMSD por aminoácido de modelos de Boltz-2 frente a estructura de referencia para 9M1Z.

al 34 y del 199 al 240 los que presentan mayor RMSD, llegando a aproximadamente 10 Å y 12 Å respectivamente. Estos valores son muy inferiores a los obtenidos en las estructuras previas, lo que concuerda con las estructuras de la Figura 5. En este caso hay diferencias de RMSD con la estructura de referencia en todos los aminoácidos, sin haber un modelo que parezca idéntico a la referencia o uno que difiera de ella de forma clara.

8.1.5. *IntelliFold*

Las Figuras 7 y 8 muestran las diferencias de los modelos predichos por IntelliFold.

En la Figura 7 se vuelve a ver que hay una parte flexible que se aleja de la estructura, similar a los casos de AlphaFold3 y Protenix, y en la Figura 8 se observa que de nuevo se trata de los aminoácidos del 0 al 34, que llegan a obtener un RMSD de 96 Å en el caso del modelo 4 (Modelo 4 en la Tabla 6). Además, los aminoácidos finales no presentan las diferencias con el modelo de referencia como lo hacían en las otras herramientas, sin embargo sí que hay diferencias en los aminoácidos del rango del 199 al 220 aproximadamente, llegando a un RMSD de alrededor de 30 Å en todos los modelos.

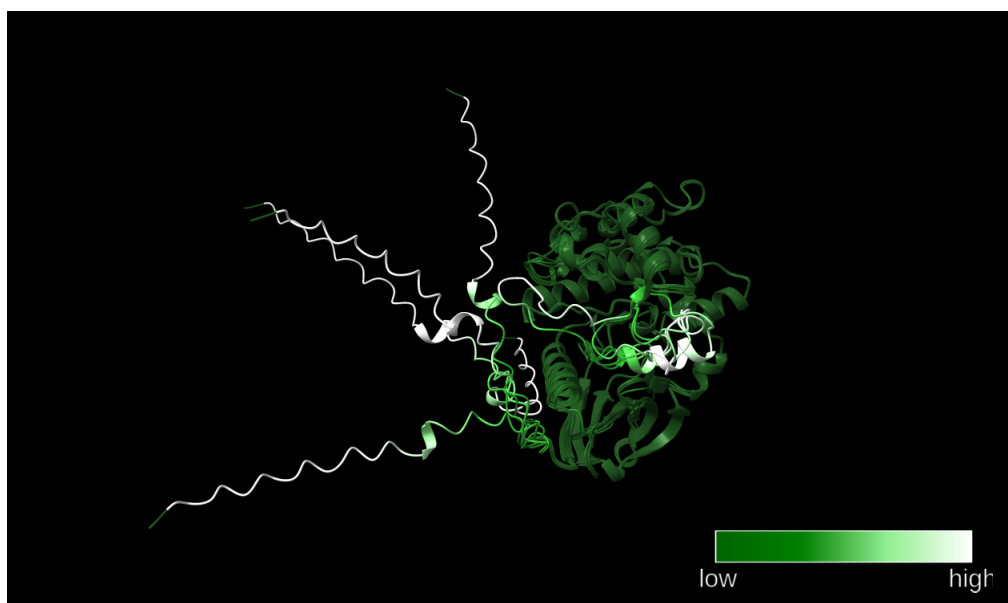


Figura 7: Diferencia estructural de los modelos de IntelliFold frente a estructura de referencia para 9M1Z.

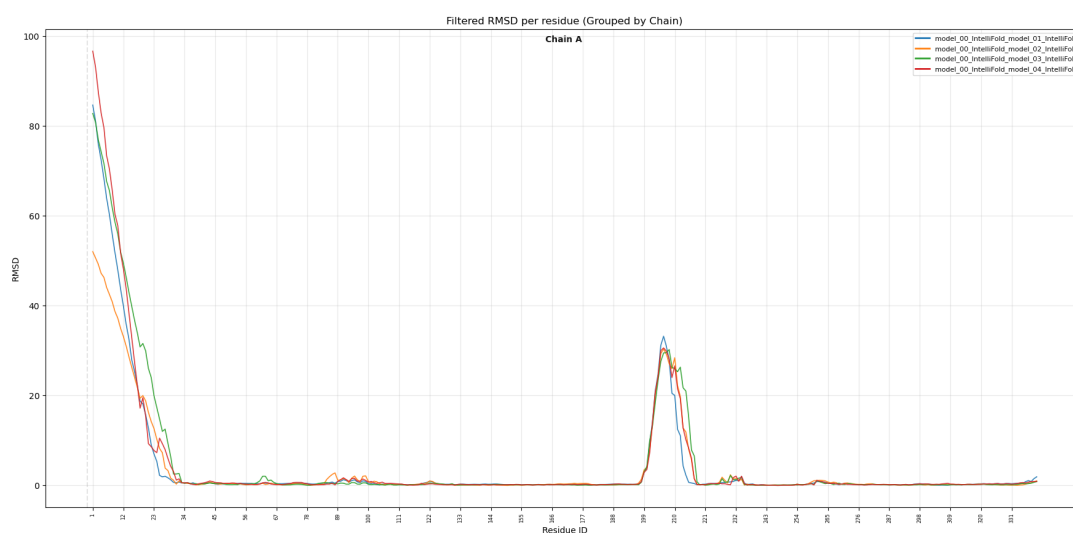


Figura 8: Diferencias en el RMSD por aminoácido de modelos de IntelliFold frente a estructura de referencia para 9M1Z.

8.2. 9NZ9

8.2.1. AlphaFold3

Las Figuras 9 y 10 muestran las diferencias estructurales entre los modelos predichos por AlphaFold3 para 9NZ9.

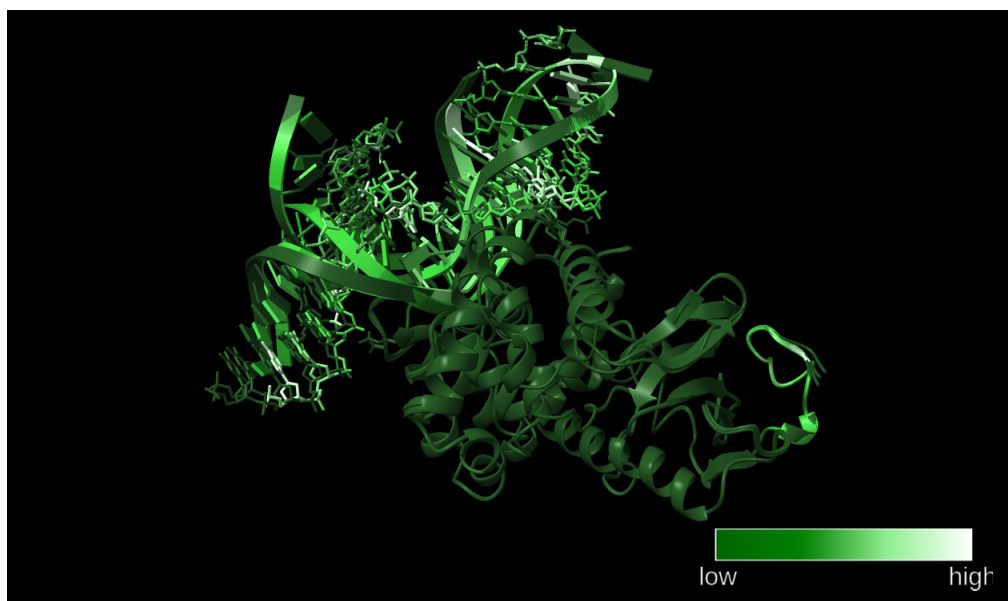


Figura 9: Diferencia estructural de los modelos de AlphaFold3 frente a estructura de referencia para 9NZ9.

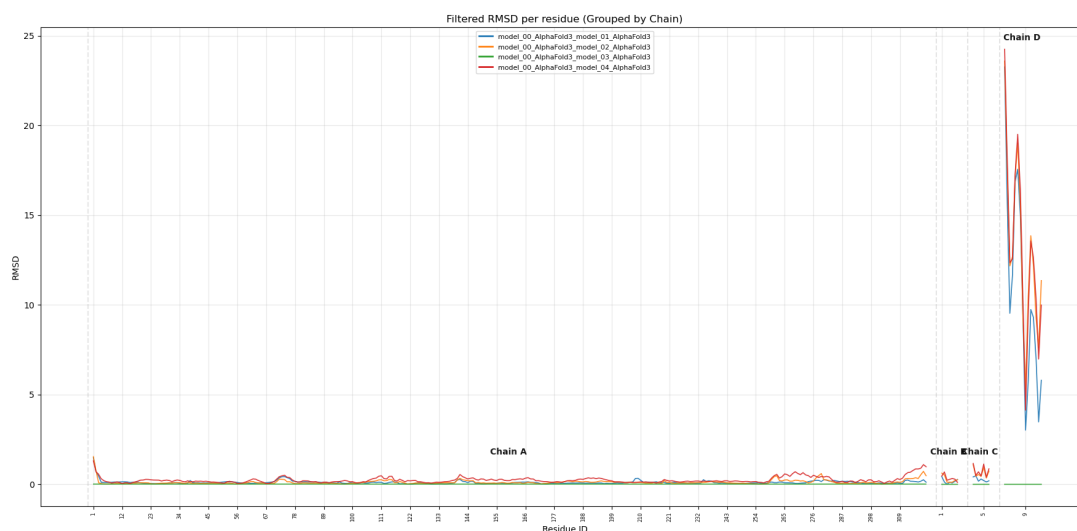


Figura 10: Diferencias en el RMSD por aminoácido de modelos de AlphaFold3 frente a estructura de referencia para 9NZ9.

Esta molécula contiene tres cadenas de ácidos nucleicos, que son las que más varían con la estructura de referencia. Observando sólo la estructura proteica, todo parece ser similar salvo

uno de los extremos que presenta un tono más claro de verde, indicando más diferencia en el RMSD.

En la Figura 10 se muestran las diferencias separadas por cadena, siendo la cadena A la estructura proteica y las B, C y D los ácidos nucleicos. En la cadena A la diferencia de RMSD es muy baja para todas las estructuras, teniendo el modelo 4 (Modelo 5 en la Tabla 7) la mayor diferencia, sin embargo, el modelo 3 (Modelo 4 en la Tabla 7) de nuevo muestra una diferencia de 0 en toda la estructura. Además se puede ver claramente que los extremos son los que presentan valores más dispares, sin embargo apenas alcanzan valores de 2 Å. Las cadenas B y C apenas muestran diferencias con la estructura de referencia, ni siquiera alcanzando 1.5 Å de diferencia. La cadena D, que es la más larga de las cadenas de ácidos nucleicos, muestra diferencias de hasta 20 Å en algunos nucleótidos en todos modelos salvo el modelo 3 (Modelo 4 en la Tabla 7).

8.2.2. Chai-1

Las Figuras 11 y 12 muestran las diferencias de los modelos devueltos por Chai-1.

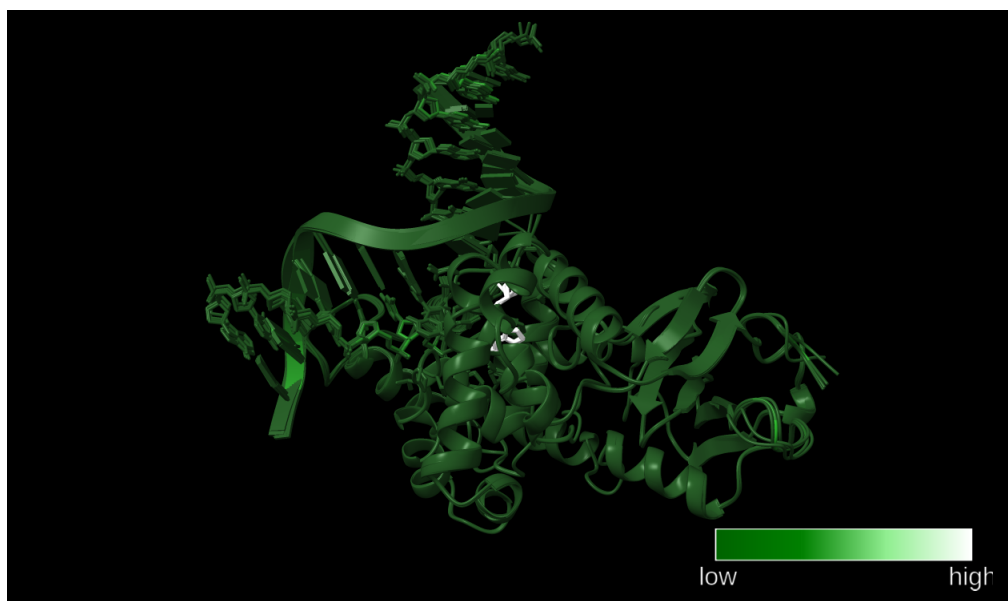


Figura 11: Diferencia estructural de los modelos de Chai-1 frente a estructura de referencia para 9NZ9.

Se puede observar que a simple vista no parece haber mucha diferencia, ya que los colores de Chimera muestran toda la estructura en un verde oscuro, salvo un ácido nucleico que aparece en blanco, ya que está orientado hacia el exterior de la hélice, y una región de la cadena B que aparece en verde ligeramente más claro. En la gráfica, se puede ver que la cadena A presenta diferencias de RMSD en los extremos y en el rango de aminoácidos del 67 al 78, llegando a

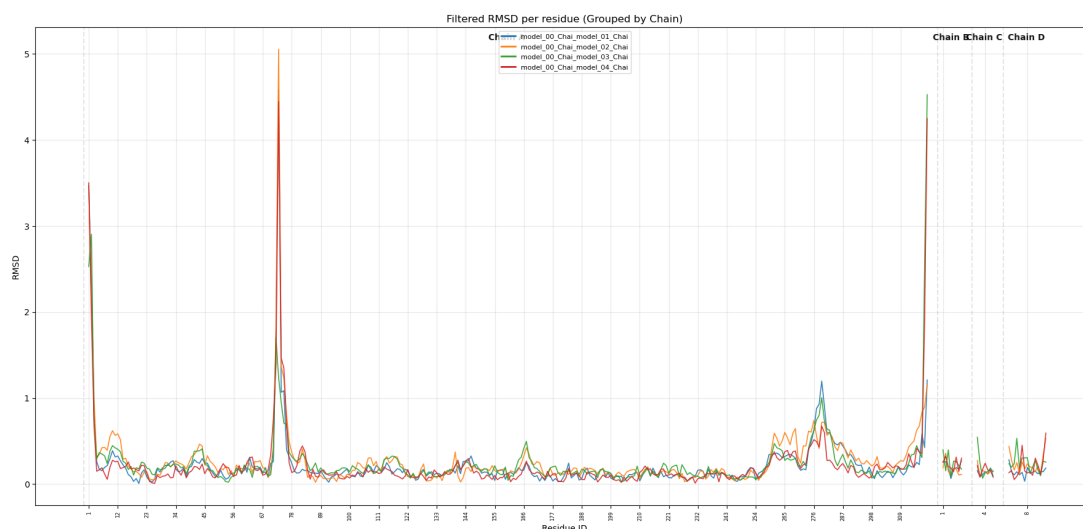


Figura 12: Diferencias en el RMSD por aminoácido de modelos de Chai-1 frente a estructura de referencia para 9NZ9.

presentar un RMSD de 5 Å en estas zonas frente a apenas 2 Å en el resto de la estructura. Las cadenas B, C y D apenas presentan diferencias, no llegando a alcanzar 1 Å de diferencia, lo que puede ser observado también en la Figura 11 a simple vista.

8.2.3. Protenix

Las Figuras 13 y 14 muestran las diferencias estructurales de los modelos de Protenix.

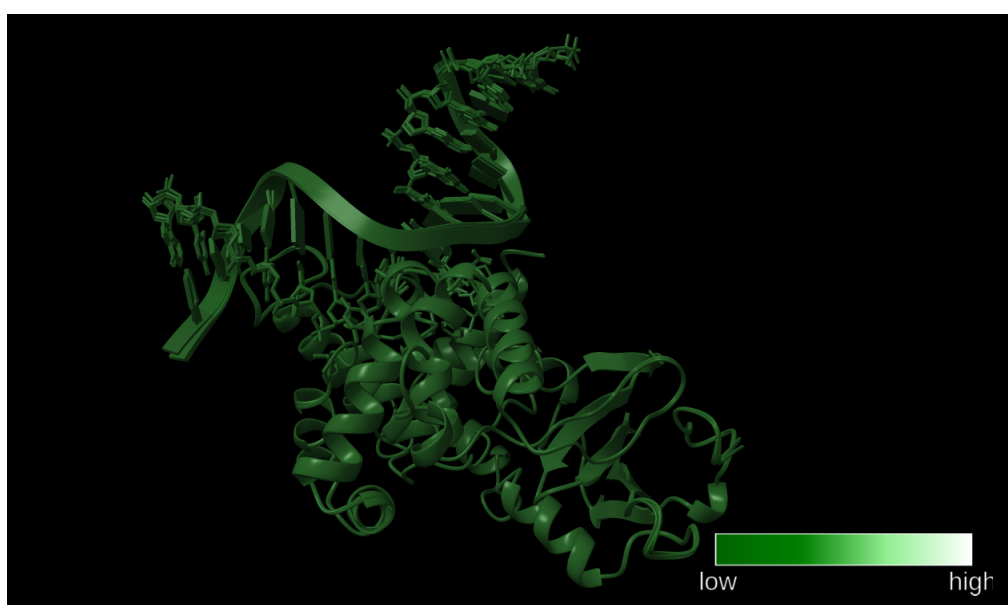


Figura 13: Diferencia estructural de los modelos de Protenix frente a estructura de referencia para 9NZ9.

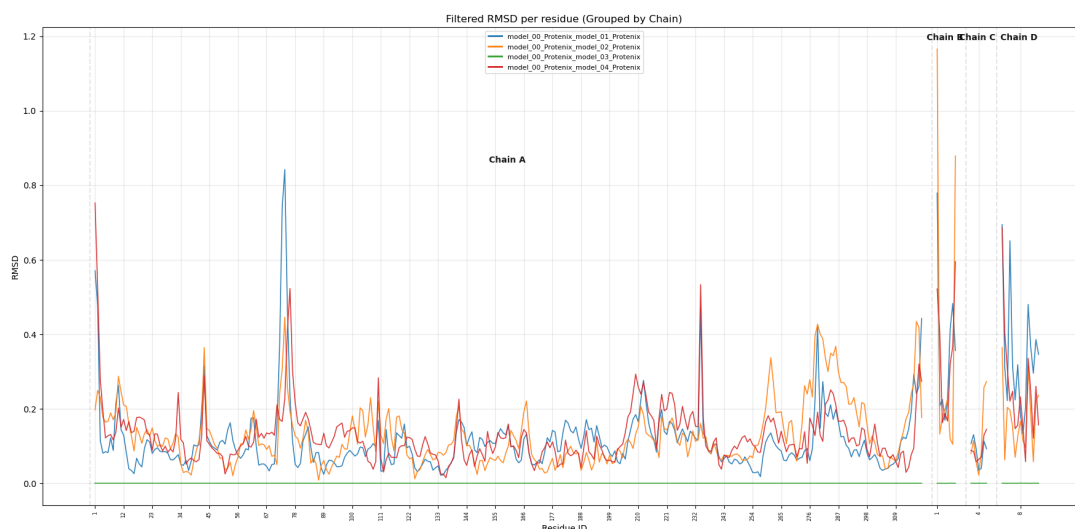


Figura 14: Diferencias en el RMSD por aminoácido de modelos de Protenix frente a estructura de referencia para 9NZ9.

No se observan diferencias notables ni en la cadena A, ni en las cadenas de la hélice. Los valores de RMSD apenas alcanzan los 1.2 Å, por lo que los modelos son muy parecidos. Además, en este caso se vuelve a ver que el modelo 3 (Modelo 4 en la Tabla 7) es exactamente igual que la referencia, con RMSD de 0 Å en toda la estructura. La cadena B es la que mayor diferencia presenta, especialmente en el modelo 2 (Modelo 3 en la Tabla 7) presentando una diferencia de RMSD de 1.2 Å en los primeros nucleótidos.

8.2.4. Boltz-2

Las Figuras 15 y 16 muestran las diferencias de los modelos de Boltz-2 para 9NZ9.

En Chimera no se observan diferencias notables en la estructura proteica, sin embargo en la hélice sí que se puede observar un ligero cambio entre los modelos, aunque no es excesivo, ya que todo tiene el mismo tono de verde, indicando que la diferencia de RMSD con la referencia no es excesiva. En la Figura 16 se puede ver, sin embargo, que estas diferencias de posición en la doble hélice apenas llegan a 1 Å de diferencia en la cadena D, ya que en la B y en la C no alcanzan los 0.5 Å.

El modelo que más difiere en esta parte de la estructura es el modelo 2 (Modelo 3 la Tabla 7), seguido del modelo 3 (Modelo 4 en la Tabla 7). En la cadena A, las diferencias más claras se encuentran en los extremos de la cadena, alcanzando los 4.2 Å aproximadamente en el extremo superior. Además se observan dos regiones en las que hay diferencias claras, en el rango de

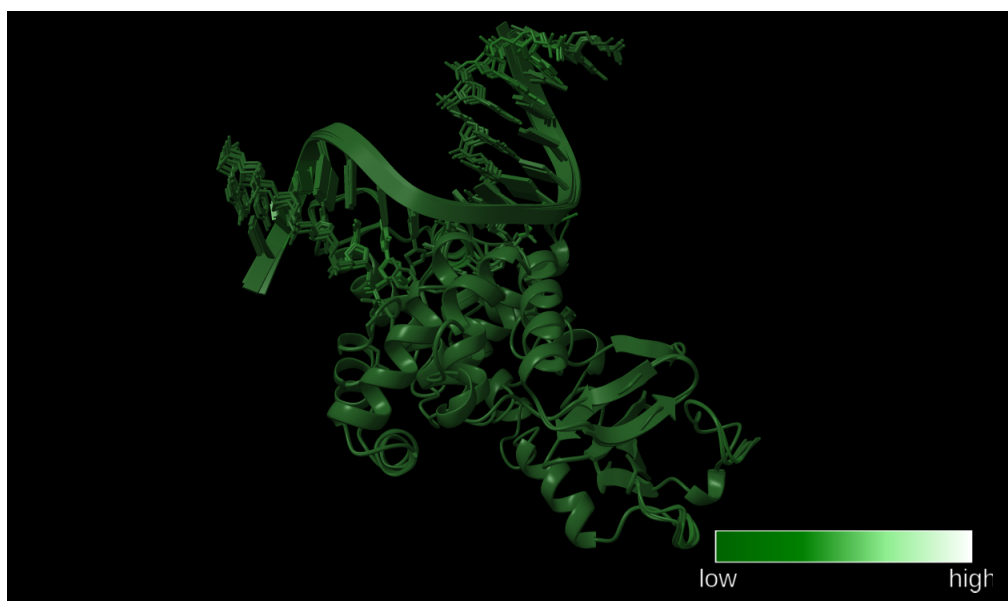


Figura 15: Diferencia estructural de los modelos de Boltz-2 frente a estructura de referencia para 9NZ9.

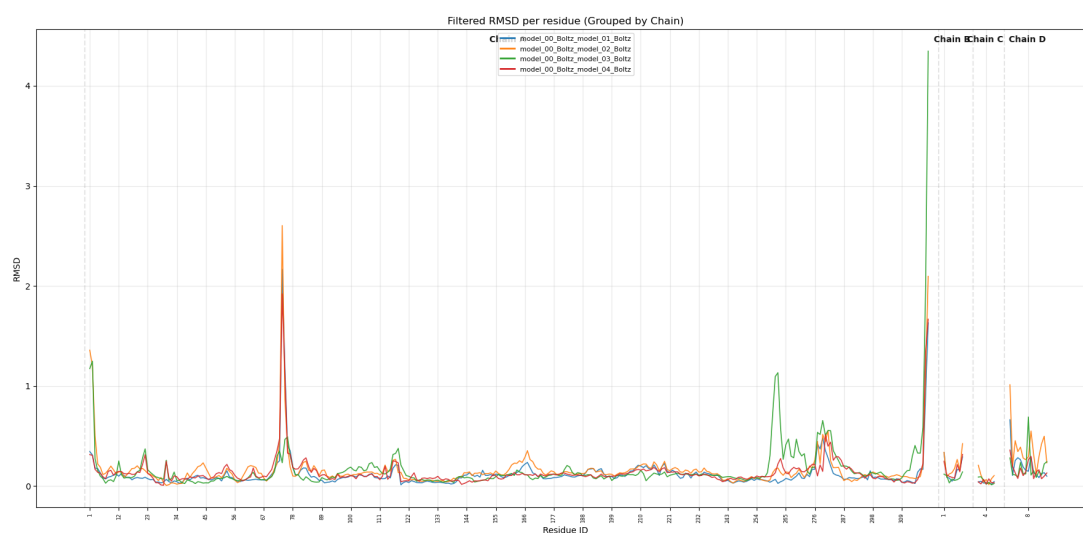


Figura 16: Diferencias en el RMSD por aminoácido de modelos de Boltz-2 frente a estructura de referencia para 9NZ9.

aminoácidos del 67 al 78, y en el rango del 254 al 287 aproximadamente. En el primero, todos los modelos salvo el modelo 3 (Modelo 4 en la Tabla 7) presentan una diferencia de hasta 2.5 Å, mientras que en el segundo rango es el modelo 3 (Modelo 4 en la Tabla 7) el que presenta una mayor diferencia, alcanzando apenas 1.2 Å.

Como se observa en la Tabla 7, hay dos modelos que presentan una puntuación máxima igual, y como se ha explicado previamente, en este caso se ha escogido como referencia el modelo 1 de esa tabla, mientras que el modelo 2 se ha empleado para realizar el gráfico. El modelo que presenta la misma puntuación que la referencia es el modelo 1 (Modelo 2 en la Tabla 7), que presenta un color azul. En la Figura 16 se puede ver que este modelo no parece diferir mucho del modelo de referencia, alcanzando apenas una diferencia de 2.15 Å en el rango de aminoácidos del 67 al 78, y tan sólo 0.6 Å en la cadena D. Esto demuestra que modelos que han sido asignados la misma puntuación no tienen por qué ser exactamente iguales.

8.2.5. *IntelliFold*

Las Figuras 17 y 18 muestran las diferencias en los modelos de IntelliFold.

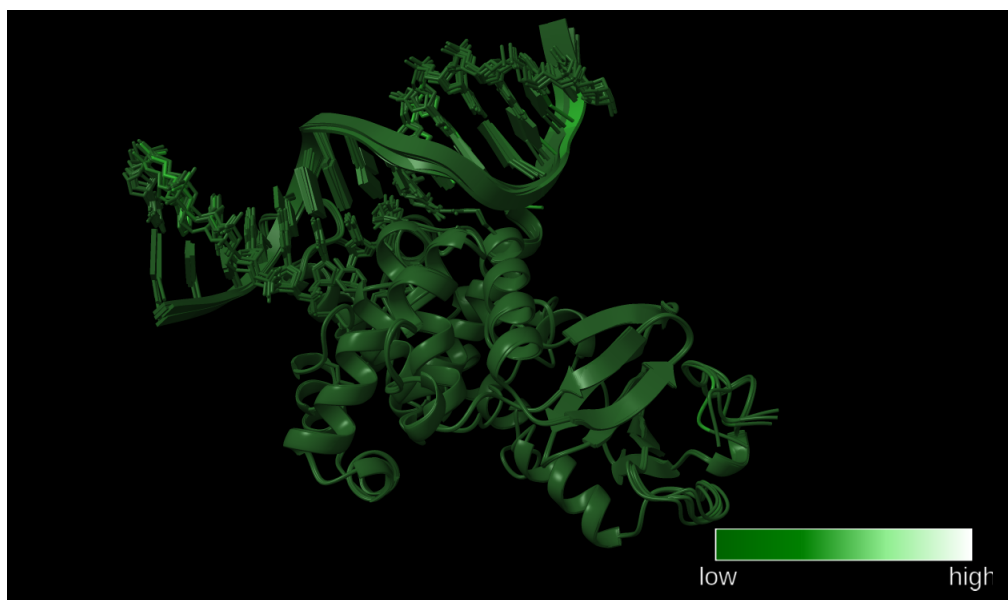


Figura 17: Diferencia estructural de los modelos de IntelliFold frente a estructura de referencia para 9NZ9.

En Chimera no se observan diferencias notables, sin embargo, en la Figura 18 se alcanza una diferencia de hasta 10.5 Å en uno de los extremos. Todos los modelos parecen ser similares, y solo muestran una ligera diferencia en los aminoácidos del 67 al 78, alcanzando solo 2 Å de diferencia con el modelo 2. Las cadenas B, C y D no presentan diferencias notables.

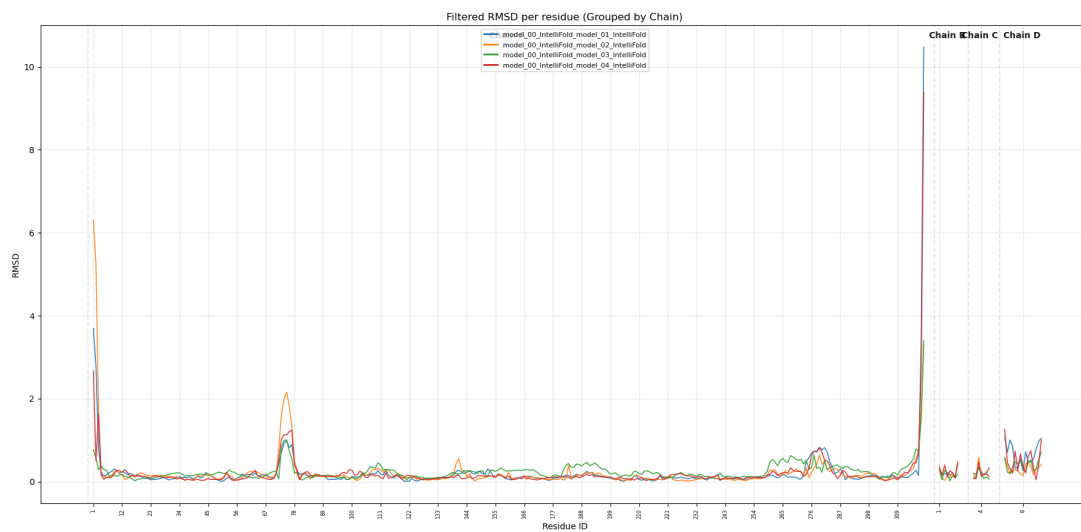


Figura 18: Diferencias en el RMSD por aminoácido de modelos de IntelliFold frente a estructura de referencia para 9NZ9.

8.3. 9QS2

8.3.1. AlphaFold3

Los modelos de AlphaFold3 se pueden observar abiertos en Chimera en la Figura 19. La mayor parte de la estructura parece ser igual, salvo en los extremos flexibles, donde sí se puede ver que hay diferencias en la posición de los aminoácidos.

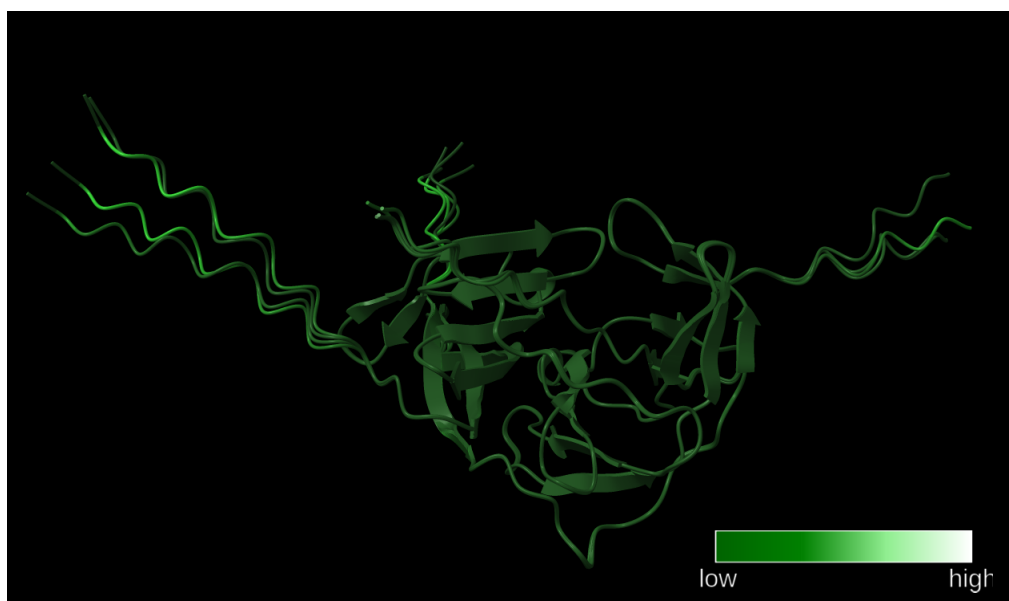


Figura 19: Diferencia estructural de los modelos de AlphaFold3 frente a estructura de referencia para 9QS2.

Estas diferencias se pueden observar mejor en la Figura 20, donde hay apenas diferencias,

tanto en la cadena A como la B, salvo en los extremos, que se alcanzan diferencias de hasta 11 Å en la cadena B. Se puede observar de nuevo que el modelo 4 (Modelo 5 en la Tabla 8) parece no tener ninguna diferencia con el mejor modelo de AlphaFold3, aun presentando menor puntuación. En la cadena A el modelo que parece diferir más es el modelo 3 (Modelo 4 en la Tabla 8), pese a no presentar tanta diferencia en los extremos. Sin embargo, en la cadena B, este modelo es el que presenta mayor diferencia en el extremo inferior y apenas diferencias a lo largo del resto de la cadena.

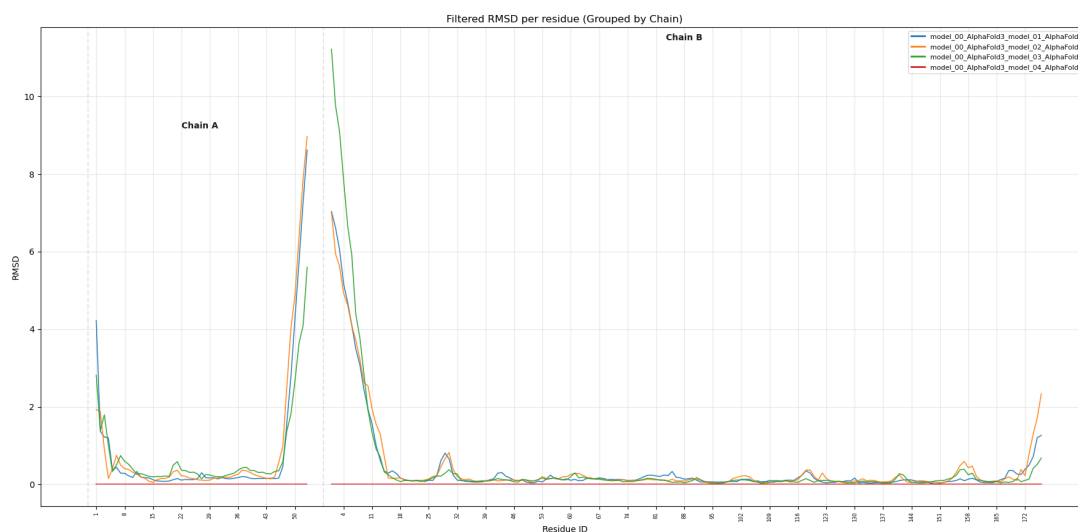


Figura 20: Diferencias en el RMSD por aminoácido de modelos de AlphaFold3 frente a estructura de referencia para 9QS2.

8.3.2. Chai-1

Las figuras 21 y 22 muestran las diferencias de los modelos de Chai-1 para 9QS2.

En la representación en Chimera se puede observar que la estructura difiere en los extremos, especialmente en los extremos que aparecen a la izquierda, cuyo color es más claro. La Figura 22 muestra las diferencias de RMSD, donde de nuevo, el modelo 1 (Modelo 2 en la Tabla 8) aparece con RMSD de 0 Å en toda la estructura. La cadena A presenta valores altos de RMSD en los extremos, en especial los modelos 2 y 4 (Modelos 3 y 5 en la Tabla 8 respectivamente), que alcanzan valores de 16 y 37.6 Å en el extremo superior respectivamente. Para el valor de RMSD del modelo 4 (Modelo 5 en la Tabla 8), se puede ver este cambio en la Figura 21, donde en la parte derecha, hay una cadena que difiere con el resto, ya que se encuentra hacia abajo, en vez de hacia arriba. En la cadena B no se muestran diferencias notables salvo en los extremos,

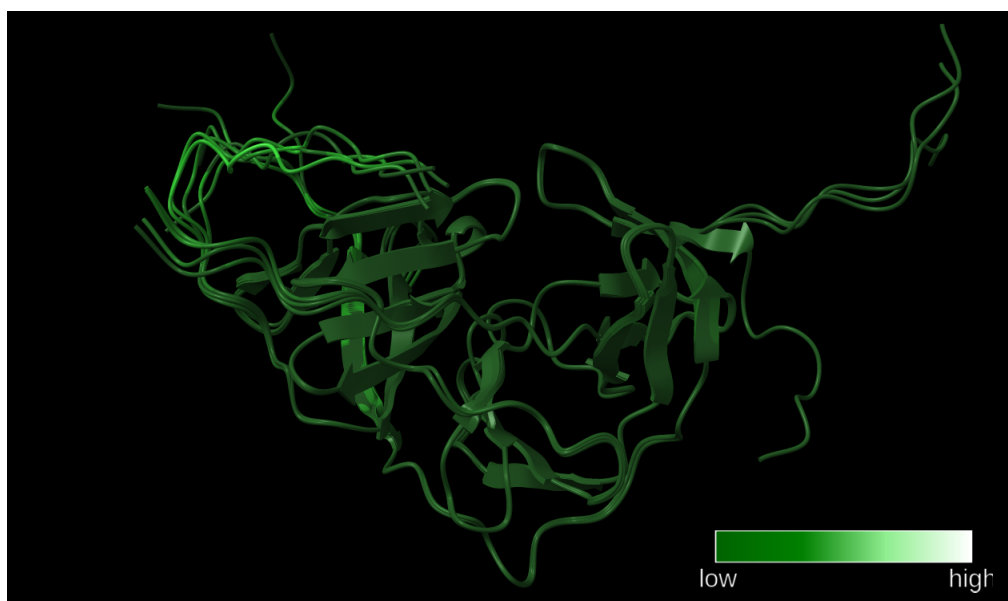


Figura 21: Diferencia estructural de los modelos de Chai-1 frente a estructura de referencia para 9QS2.

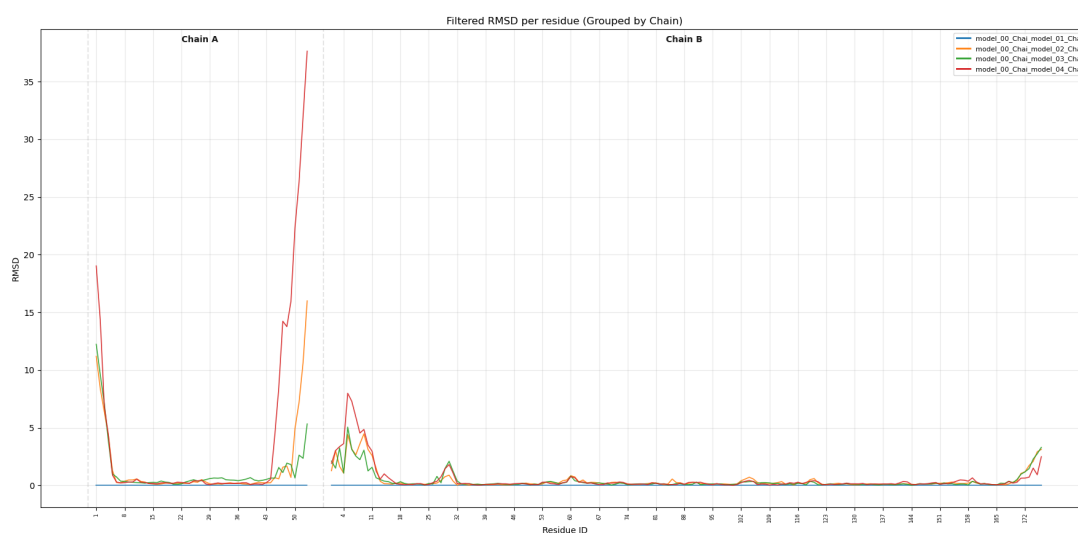


Figura 22: Diferencias en el RMSD por aminoácido de modelos de Chai-1 frente a estructura de referencia para 9QS2.

donde se llega a los 8 Å por el modelo 4 (Modelo 5 en la Tabla 8).

8.3.3. Protenix

Las Figuras 23 y 24 muestran las diferencias en los modelos de Protenix.

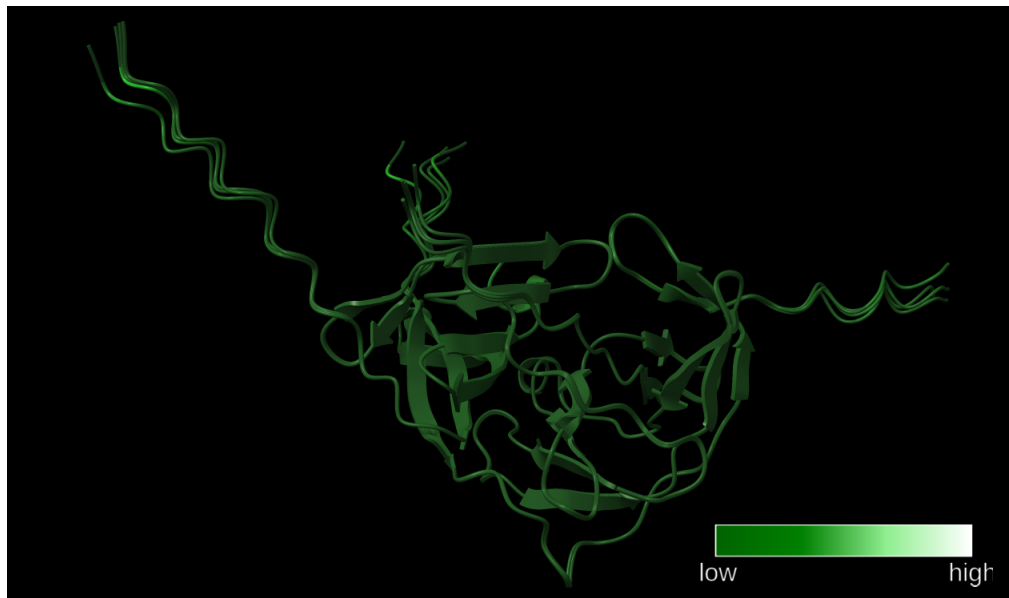


Figura 23: Diferencia estructural de los modelos de Protenix frente a estructura de referencia para 9QS2.

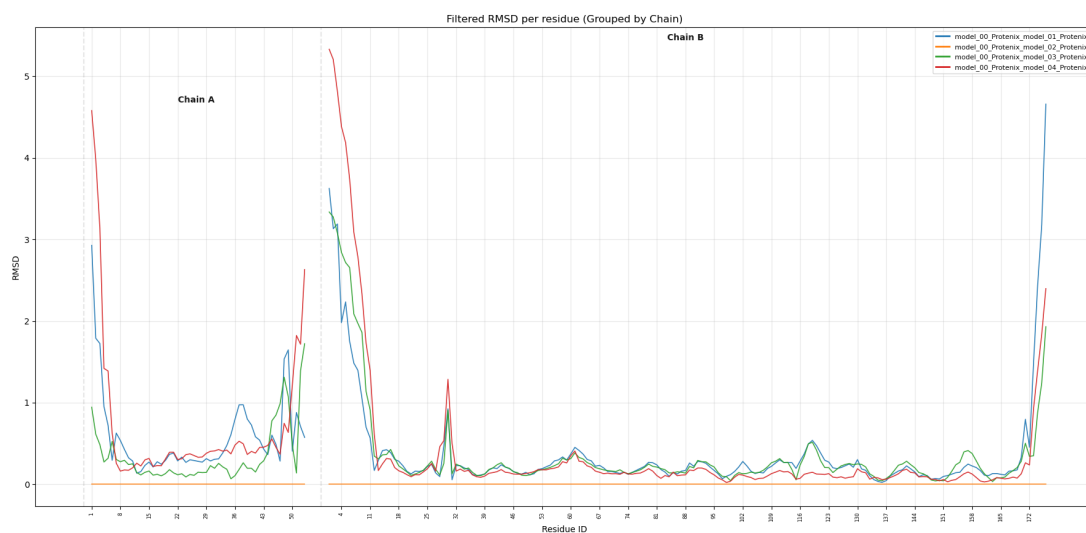


Figura 24: Diferencias en el RMSD por aminoácido de modelos de Protenix frente a estructura de referencia para 9QS2.

En la Figura 23 apenas se notan diferencias, salvo ligeros cambios en los extremos flexibles, que vienen explicados en la gráfica de la Figura 24, donde apenas llegan a alcanzar valores

de RMSD de 4.6 Å en la cadena A y 5.3 Å en la cadena B. Ambos valores los presenta el modelo 4 (Modelo 5 en la Tabla 8), que es uno de los modelos que presenta más diferencia de puntuación con el modelo de referencia. En la cadena A, excluyendo los extremos, el valor más alto de RMSD es tan solo de 1 Å en el modelo 1 (Modelo 2 en la Tabla 8), por lo que es casi imperceptible, al igual que en la cadena B, donde el modelo 4 (Modelo 5 en la Tabla 8) alcanza 1.3 Å. De la misma forma que ha ocurrido en otras predicciones, en este caso el modelo 2 (Modelo 3 en la Tabla 8) presenta un RMSD de 0 Å en toda la estructura.

8.3.4. Boltz-2

Las Figuras 25 y 26 muestran los modelos predichos por Boltz-2.

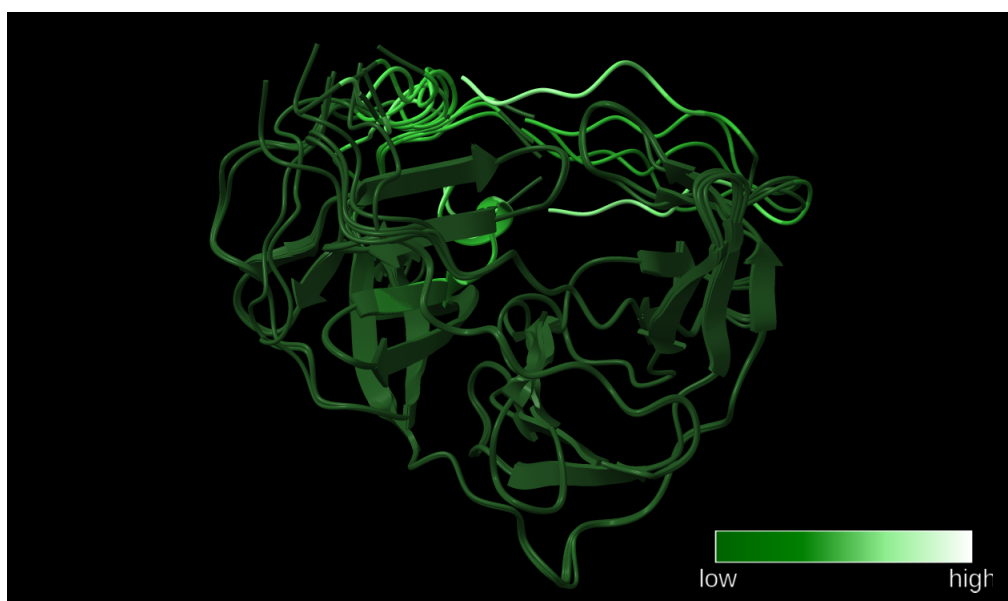


Figura 25: Diferencia estructural de los modelos de Boltz-2 frente a estructura de referencia para 9QS2.

En la visualización en Chimera se ven diferencias en los extremos flexibles, donde en este caso aparecen en un tono más claro que los extremos de otros modelos anteriores. Estas diferencias se observan mejor en la Figura 26, donde los extremos alcanzan valores de RMSD de hasta 19.5 Å en el caso de la cadena A. El modelo 3 (Modelo 4 en la Tabla 8) parece ser el que más diferencias tiene con el modelo de referencia en los extremos de ambas cadenas, sin embargo, a lo largo de la cadena B, los modelos 1 y 4 (Modelos 2 y 5 en la Tabla 8 respectivamente) alcanzan picos de hasta 3.4 Å en el rango de aminoácidos del 25 al 32.

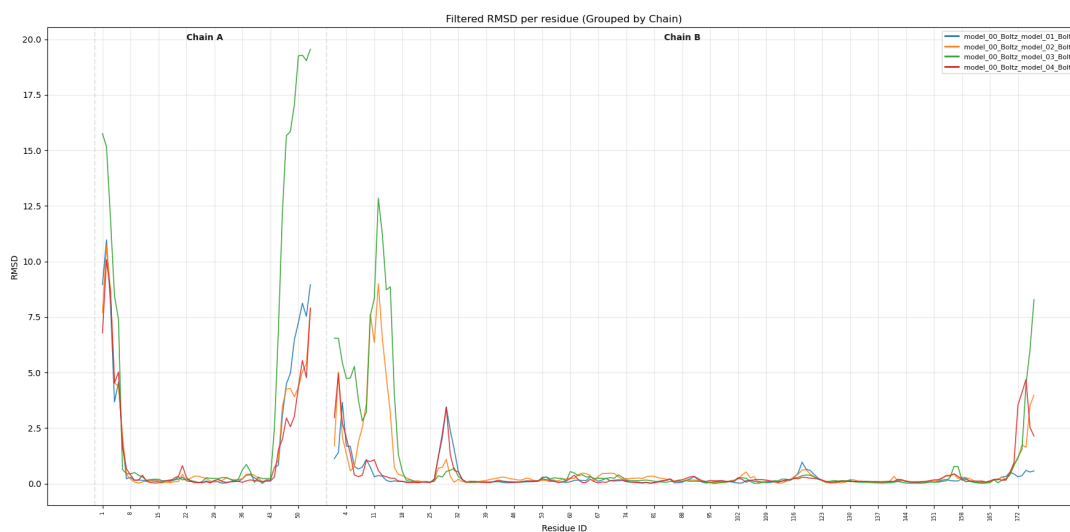


Figura 26: Diferencias en el RMSD por aminoácido de modelos de Boltz-2 frente a estructura de referencia para 9QS2.

8.3.5. IntelliFold

Las Figuras 27 y 28 muestran los modelos de IntelliFold, que parece que son los que mayor diferencia tienen en los extremos.

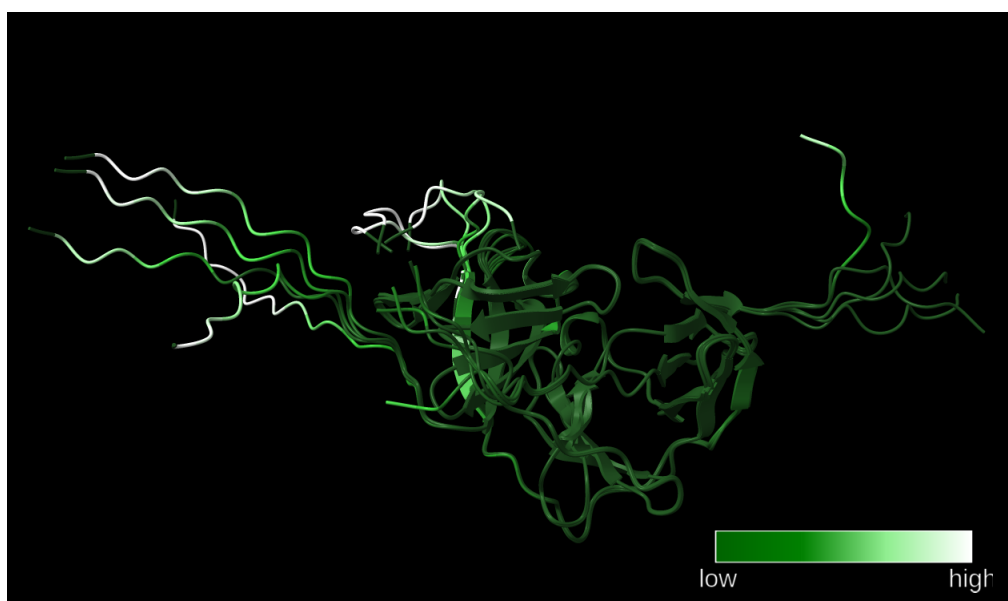


Figura 27: Diferencia estructural de los modelos de IntelliFold frente a estructura de referencia para 9QS2.

En la Figura 27 se puede ver cómo los extremos de ambas cadenas aparecen en color más claro, incluso blanco, en algunas zonas. Esto se ve reflejado en la Figura 28, donde para la cadena A todos los modelos alcanzan más de un valor de RMSD de 12 Å en el extremo inferior,

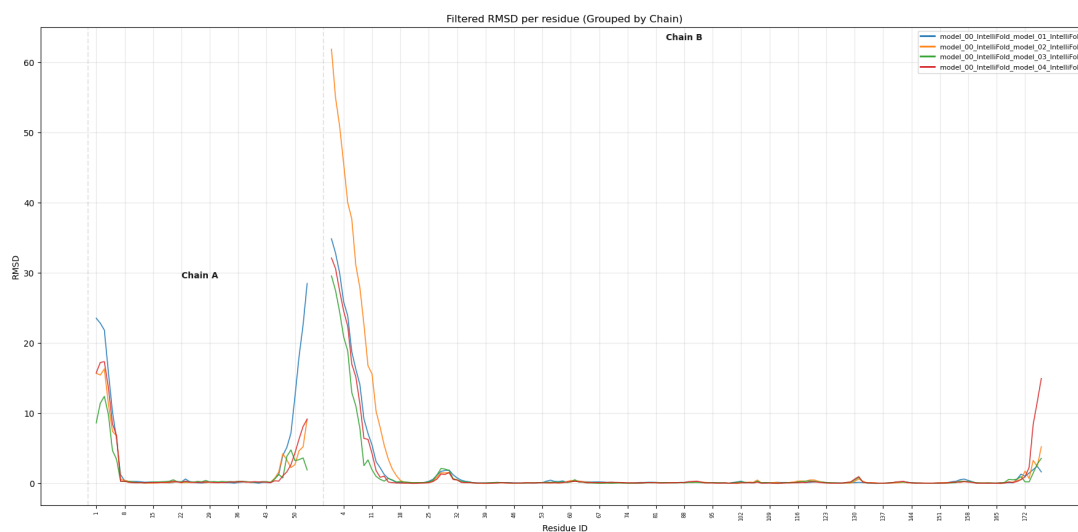


Figura 28: Diferencias en el RMSD por aminoácido de modelos de IntelliFold frente a estructura de referencia para 9QS2.

y de 4.6 Å en el superior, con el modelo 1 (Modelo 2 en la Tabla 8) alcanzando 28.5 Å en uno de los extremos. En la cadena B todos los modelos alcanzan al menos 29.5 Å de RMSD en el extremo inferior, con el modelo 2 (Modelo 3 en la Tabla 8) llegando a más de 60 Å de RMSD. Sin embargo, en el resto de la cadena apenas hay diferencia con el modelo de referencia, al igual que en el extremo superior, donde solo uno de modelos alcanza más de 5.5 Å de RMSD.