Texto

Descripción generada automáticamente

Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

Automatización de la detección de hélices de Poliprolina II en proteínas

|  |  |
| --- | --- |
| Trabajo fin de estudio presentado por: | Silvia Enma Rodríguez Fernández |
| Director/a: | Marina de la Cruz Echeandía |
| Fecha: | 09/05/2025 |
| Repositorio del código fuente: | <https://github.com/silviaenma/PPIIMoL> |

Resumen

Este Trabajo de Fin de Grado presenta el desarrollo de un módulo en Python integrado en PyMOL, un visor molecular ampliamente utilizado en entornos de investigación, orientado a la detección automática de hélices de poliprolina II (PPII) en estructuras proteicas a partir de archivos PDB. Las hélices PPII, frecuentes en proteínas ricas en glicina y prolina, desempeñan un papel relevante en diversos procesos biomoleculares clave, pero su identificación sigue siendo un proceso manual y laborioso. La iniciativa surge como parte de una línea de investigación desarrollada en el Instituto de Química-Física "Blas Cabrera" del CSIC, basada en un protocolo experimental previamente aplicado para el análisis de dominios de hélice PPII ricos en glicina.

El módulo incluye funciones para el filtrado de residuos, la evaluación de ángulos diedros característicos, la predicción de posibles enlaces de hidrógeno no canónicos y la detección de patrones de plegamiento compatibles con este tipo de hélices. Además, permite visualizar los resultados directamente en PyMOL, facilitando el trabajo de los investigadores. Concebido como una herramienta de acceso libre, será publicado en GitHub con el fin de contribuir a la comunidad científica y fomentar su reutilización y mejora. Uno de los principales valores de esta herramienta reside en su capacidad para automatizar una tarea que hasta ahora requería experiencia previa y tiempo considerable, eliminando el componente subjetivo del análisis visual y permitiendo el estudio sistemático de grandes conjuntos de datos estructurales. El proyecto sigue un modelo incremental, que parte del análisis manual del problema hasta llegar a una herramienta funcional y validada. Como caso de prueba, se ha utilizado un conjunto de proteínas modelo, demostrando la utilidad del módulo como apoyo al análisis estructural de proteínas y al diseño racional de péptidos con estructuras PPII.

**Palabras clave:** Poliprolina II, PPII, PyMOL, glicina, Python, estructuras secundarias.

Abstract

This thesis presents the development of a Python module integrated into PyMOL, a molecular visualization system widely used in research settings, aimed at the automatic detection of polyproline II (PPII) helices in protein structures from PDB files. PPII helices, common in glycine- and proline-rich proteins, play a relevant role in various key biomolecular processes, but their identification remains a manual and laborious process. The initiative was created to support ongoing research at the Blas Cabrera Institute of Physical Chemistry of the CSIC, derived from an experimental protocol previously applied to the analysis of glycine-rich PPII helix domains analysis. The module includes functions for residue filtering, the evaluation of characteristic dihedral angles, the prediction of potential non-canonical hydrogen bonds, and the detection of folding patterns compatible with this type of helix. It also allows the results to be visualized directly in PyMOL, facilitating the work of researchers. Conceived as an open-access tool, it will be published on GitHub to contribute to the scientific community and encourage its reuse and improvement. One of the main advantages of this tool lies in its ability to automate a task that until now required prior experience and considerable time, eliminating the subjectivity of visual analysis, and allowing for the systematic study of large structural data sets. The project is based on an incremental approach, starting with manual analysis of the problem, and leading to a functional and validated tool. Model proteins have been used as a test case, demonstrating the usefulness of the module in supporting the structural analysis of proteins and the rational design of peptides with PPII structures.

**Keywords**: Poliproline II, PPII, PyMOL, Glycine, Python, Secondary structure.

**Agradecimientos**

Este trabajo no habría sido posible sin el acompañamiento y apoyo de distintas personas e instituciones a lo largo del proceso.

En primer lugar, quisiera agradecer a la directora del trabajo, Dra. Marina de la Cruz Echeandía, por su orientación constante, su disponibilidad y sus valiosas aportaciones, tanto en lo técnico como en lo metodológico.

También deseo expresar un agradecimiento especial al Grupo de Estructura, Dinámica e Interacciones de Proteínas por RMN del Instituto de Química-Física “Blas Cabrera” (IQF-CSIC), por compartir el conocimiento experimental que ha servido de base para este desarrollo, así como por su colaboración en la validación funcional del módulo.

Finalmente, agradezco de forma muy personal a mi hijo, que ha sido mi impulso durante todo este tiempo; su acompañamiento y paciencia han sido esenciales para poder llevar a término este proyecto. A él le debo no solo la fuerza para terminar este proyecto, sino también la ilusión de seguir creciendo cada día.

Índice de contenidos

[1. Introducción 1](#_Toc201169871)

[1.1. Motivación 2](#_Toc201169872)

[1.2. Planteamiento del trabajo 3](#_Toc201169873)

[1.3. Estructura del trabajo 4](#_Toc201169874)

[2. Contexto y Estado del Arte 5](#_Toc201169875)

[2.1. Análisis del contexto 5](#_Toc201169876)

[2.1.1. El problema: detección de hélices PPII 5](#_Toc201169877)

[2.1.2. Método manual de hallar PPII con PyMOL 7](#_Toc201169878)

[2.2. Estado del arte 27](#_Toc201169879)

[2.2.1. Comparativa de herramientas existentes para análisis estructural 27](#_Toc201169880)

[2.2.2. Elección de PyMOL 28](#_Toc201169881)

[2.3. PyMOL como Herramienta para el Análisis Estructural 31](#_Toc201169882)

[2.3.1. ¿Qué es PyMOL? 31](#_Toc201169883)

[2.3.2. Principales funcionalidades utilizadas en este proyecto 31](#_Toc201169884)

[2.3.3. Procedimiento manual utilizado actualmente en el laboratorio 33](#_Toc201169885)

[2.3.4. Ejemplo práctico: uso manual de PyMOL sobre la proteína modelo 3BOG 34](#_Toc201169886)

[3. Objetivos y metodología de trabajo 35](#_Toc201169887)

[3.1. Objetivo general 35](#_Toc201169888)

[3.2. Objetivos específicos 35](#_Toc201169889)

[3.3. Metodología de trabajo 36](#_Toc201169890)

[4. Contribución y colaboración 39](#_Toc201169891)

[4.1. Licencia y distribución del software 40](#_Toc201169892)

[4.2. Publicación en repositorios abiertos 40](#_Toc201169893)

[5. Desarrollo del módulo 41](#_Toc201169894)

[5.1. Requisitos funcionales y técnicos del sistema 41](#_Toc201169895)

[5.1.1. Requisitos funcionales. 41](#_Toc201169896)

[5.1.2. Requisitos técnicos. 41](#_Toc201169897)

[5.2. Diseño del algoritmo de detección PPII 42](#_Toc201169898)

[5.3. Publicación y distribución del software 46](#_Toc201169899)

[5.4. Estructura del código y funciones principales 46](#_Toc201169900)

[5.5. Proceso de integración con PyMOL 50](#_Toc201169901)

[5.6. Validación sobre la proteína modelo 3BOG 52](#_Toc201169902)

[5.7. Estructura técnica de la solución 53](#_Toc201169903)

[5.8. Casos de uso representativos 54](#_Toc201169904)

[Caso de uso 1: Carga y preparación de la estructura PDB. 54](#_Toc201169905)

[Caso de uso 2: Análisis geométrico de la proteína 55](#_Toc201169906)

[Caso de uso 3: Identificación visual de hélices PPII 56](#_Toc201169907)

[Caso de uso 4: Análisis comparativo entre cadenas 56](#_Toc201169908)

[Caso de uso 5: Personalización del análisis por parámetros 57](#_Toc201169909)

[Caso de uso 6: Exportación de hélices detectadas para análisis externo 57](#_Toc201169910)

[Caso de uso 7: Análisis geométrico de enlaces potenciales 58](#_Toc201169911)

[Caso de uso 8: Generación automatizada de informes estructurados 59](#_Toc201169912)

[5.9. Resumen del proceso de desarrollo 61](#_Toc201169913)

[5.10. Publicación y contribución al repositorio oficial 64](#_Toc201169914)

[6. Resultados y evaluación 65](#_Toc201169915)

[6.1. Resultados obtenidos: detección de hélices PPII en 3BOG 65](#_Toc201169916)

[6.2. Comparación con el análisis manual 66](#_Toc201169917)

[6.3. Valoración de precisión y eficiencia 67](#_Toc201169918)

[6.4. Limitaciones identificadas 68](#_Toc201169919)

[6.5. Aplicabilidad en otros modelos estructurales 69](#_Toc201169920)

[7. Conclusiones y trabajo futuro 71](#_Toc201169921)

[7.1. Conclusiones del trabajo 71](#_Toc201169922)

[7.2. Aportaciones del módulo 72](#_Toc201169923)

[7.3. Propuestas de mejora y líneas futuras 73](#_Toc201169924)

[7.4. Impacto y transferencia del trabajo 74](#_Toc201169925)

[Referencias bibliográficas 76](#_Toc201169926)

[Anexo A. Fragmentos de código relevantes del módulo 80](#_Toc201169927)

[1. Fragmento A1. Función seleccionar\_archivo 80](#_Toc201169928)

[2. Fragmento A2. Función guardar\_angulos\_phi\_psi 80](#_Toc201169929)

[3. Fragmento A3. Función eliminar\_solventes 80](#_Toc201169930)

[4. Fragmento A4. Función anadir\_hidrogenos 80](#_Toc201169931)

[5. Exportación estructurada de resultados para uso en herramientas externas: 80](#_Toc201169932)

[5.1. Fragmento A5. Función exportar\_helice\_pdb 81](#_Toc201169933)

[Exporta hélices como archivos .pdb desde PyMOL 81](#_Toc201169934)

[5.2. Fragmento A6. Función exportar\_angulos\_csv 81](#_Toc201169935)

[Exporta a .csv los datos de los ángulos φ y ψ 81](#_Toc201169936)

[6. Fragmento A7. Función calcular\_angulos\_ca\_h\_o 82](#_Toc201169937)

[7. Fragmento A8. Función generar\_reporte\_csv() 82](#_Toc201169938)

[8. Fragmento A9. Función calcular\_y\_visualizar\_distancias() 82](#_Toc201169939)

[Anexo B. Guía técnica del módulo de detección de hélices de tipo PPII en PyMOL 84](#_Toc201169940)

[1. Introducción 84](#_Toc201169941)

[2. Requisitos previos 84](#_Toc201169942)

[3. Instalación y configuración 84](#_Toc201169943)

[4. Carga de estructuras 85](#_Toc201169944)

[5. Uso del módulo 85](#_Toc201169945)

[6. Interpretación de resultados 86](#_Toc201169946)

[7. Personalización o ampliación 86](#_Toc201169947)

[8. Créditos y contacto 87](#_Toc201169948)

[Anexo C. Guía del usuario del módulo PPIIMoL para la detección de hélices PPII en PyMOL 89](#_Toc201169949)

[Descripción general 89](#_Toc201169950)

[¿Qué hace esta herramienta? 89](#_Toc201169951)

[Requisitos básicos 89](#_Toc201169952)

[¿Qué necesito para empezar? 89](#_Toc201169953)

[Pasos para usar el módulo 90](#_Toc201169954)

[¿Cómo lo pongo en marcha? 90](#_Toc201169955)

[¿Qué resultados ofrece? 90](#_Toc201169956)

[Visualización e interpretación 91](#_Toc201169957)

[¿Cómo interpreto los resultados? 91](#_Toc201169958)

[Opciones avanzadas 91](#_Toc201169959)

[¿Puedo personalizarlo? 91](#_Toc201169960)

[En caso de error 92](#_Toc201169961)

[¿Qué hacer si algo no funciona? 92](#_Toc201169962)

[Soporte al usuario 92](#_Toc201169963)

[¿A quién puedo contactar si tengo dudas? 92](#_Toc201169964)

[Anexo D. Documentación técnica: Pseudocódigo estructurado del algoritmo principal 93](#_Toc201169965)

[Diagrama lógico del algoritmo de detección de hélices PPII (PPIIMoL) 94](#_Toc201169966)

[Anexo E. Ejemplos de salidas generadas por el módulo PPIIMoL 96](#_Toc201169967)

[E.1. Archivo angulos\_phi\_psi.csv 96](#_Toc201169968)

[E.2. Archivo reporte\_ppii.csv 96](#_Toc201169969)

[E.3. Archivo angulos\_ca\_h\_o.txt 97](#_Toc201169970)

[E.4. Archivo distancias\_colindantes.txt 99](#_Toc201169971)

[Índice de acrónimos 100](#_Toc201169972)

Índice de figuras

[Figura 1. Ángulo diedro Phi. 5](#_Toc201170010)

[Figura 2. Ángulo diedro Psi. 6](#_Toc201170011)

[Figura 3 Obtención de la proteína en PyMOL 8](file:///J:\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_3BORRADOR.docx#_Toc201170012)

[Figura 4. Molécula 3BOG (PDB). 8](#_Toc201170013)

[Figura 5. Eliminación de moléculas de agua 8](#_Toc201170014)

[Figura 6. Adición de hidrógenos. 9](#_Toc201170015)

[Figura 7. Abriendo la Secuencia. 9](file:///J:\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_3BORRADOR.docx#_Toc201170016)

[Figura 8. Visualización de la Secuencia en PyMOL. 10](#_Toc201170017)

[Figura 9. Hélices vistas en paralelo. 11](#_Toc201170018)

[Figura 10. Esquema representativo de la proteína 3BOG. 11](#_Toc201170019)

[Figura 11. Tabla de densidad del empaquetamiento de proteínas. 12](#_Toc201170020)

[Figura 12. Menú de Selección 12](file:///J:\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_3BORRADOR.docx#_Toc201170021)

[Figura 13. Seleccionando la Cadena. 13](#_Toc201170022)

[Figura 14. Separación de Cadena. 14](#_Toc201170023)

[Figura 15. obj01 en solitario. 14](#_Toc201170024)

[Figura 16. Separación y Coloreado de las hélices seleccionadas. 16](#_Toc201170025)

[Figura 17. Hélices Identificadas. 17](#_Toc201170026)

[Figura 18. Dando nombre a las hélices. 17](#_Toc201170027)

[Figura 19. Visualización de hélices en modo regaliz. 18](#_Toc201170028)

[Figura 20. Ángulos Phi, Psi y Omega. 19](#_Toc201170029)

[Figura 21. Activando la herramienta Wizard 19](#_Toc201170030)

[Figura 22. Modo Diedros 20](file:///J:\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_3BORRADOR.docx#_Toc201170031)

[Figura 23. Phi y Psi en PyMOL. 20](#_Toc201170032)

[Figura 24. Selección de c-alphas. 21](#_Toc201170033)

[Figura 25. Selección de oxígenos. 22](#_Toc201170034)

[Figura 26. Posibles localizaciones de puentes de hidrógeno no canónicos*.* 23](#_Toc201170035)

[Figura 27. Posible migraña inducida por mala praxis. 24](#_Toc201170036)

[Figura 28. Selección de ángulos en Wizard. 24](#_Toc201170037)

[Figura 29. Cribado de los posibles puentes de hidrógeno no canónicos. 25](#_Toc201170038)

[Figura 30. Interfaz gráfica del módulo desarrollada en Python con Tkinter e integrada en PyMOL 50](#_Toc201170039)

[Figura 31. Diagrama lógico del algoritmo de detección de hélices PPII (PPIIMoL) 95](#_Toc201170040)

[Figura 32 archivo angulos\_phi\_psi\_7JJV 96](#_Toc201170041)

[Figura 33. Archivo Reporte\_ppii\_7JJV 97](#_Toc201170042)

[Figura 34 Interfaz gráfica del módulo para ajustar los umbrales angulares CA–H···O. 97](#_Toc201170043)

[Figura 35. fichero ángulos\_ca\_h\_o\_7JJV.txt 98](#_Toc201170044)

[Figura 36. fichero ángulos\_ca\_h\_o\_7JJV.txt resultado del cambio de umbral 98](#_Toc201170045)

[Figura 37. Archivo distancias colindantes 7JJV 99](#_Toc201170046)

Índice de tablas

[Tabla 1. Comparativa visores moleculares y su utilidad 28](#_Toc201170047)

[Tabla 2. Comparativa de herramientas respecto al proyecto de detección PPII 30](#_Toc201170048)

[Tabla 3. Índice de moléculas de referencia 38](#_Toc201170049)

[Tabla 4. Casos de uso representativos del módulo de detección de hélices PPII 60](#_Toc201170050)

[Tabla 5. Resumen de tareas técnicas realizadas 63](#_Toc201170051)

[Tabla 6. Funcionalidad de los fragmentos de código incluidos 83](#_Toc201170052)

# Introducción

En el ámbito de la biología molecular, la visualización y el análisis de proteínas juegan un papel esencial en el estudio de procesos celulares, la identificación de dianas terapéuticas y el diseño racional de fármacos. Dentro de la estructura secundaria de las proteínas, las hélices de tipo poliprolina II (PPII) han cobrado especial interés por su implicación en mecanismos neuroquímicos.

Existen herramientas que permiten analizar moléculas a partir de archivos de **Protein Data Bank (PDB),** un formato estándar que caracteriza los componentes atómicos de una molécula en un espacio tridimensional. Cabe destacar que un archivo de este tipo representa una molécula en el momento en que fue capturada su representación, interactuando, por lo tanto, con el medio y, en consecuencia, puede contener otras moléculas, consideradas foráneas, con las que está interactuando, como son el medio y el ligando, caso de estar presentes.

Sin embargo, la detección de las PPII no está automatizada como la de otras estructuras secundarias mejor documentadas, como las hélices α o las láminas β (α y β). Esta carencia justifica el desarrollo de una herramienta propia que facilite a los investigadores la identificación de estas regiones de manera ágil y precisa, especialmente en el análisis de grandes conjuntos de proteínas o en modelos estructurales que aún no han sido completamente caracterizados.

El desarrollo de esta herramienta responde a la necesidad de automatizar la identificación de hélices PPII en estructuras proteicas, tarea que actualmente requiere un análisis manual complejo. Esta iniciativa surge como apoyo a investigaciones en curso en el Instituto de Química-Física "Blas Cabrera" del CSIC, a partir del protocolo experimental utilizado en dicho entorno, y toma como referencia el artículo Architectonic Principles of Polyproline II Bundle Protein Domains (Segura Rodríguez & Laurents, 2024).

Este trabajo de fin de grado se plantea como un proyecto de desarrollo de software cuyo objetivo principal es la creación de un módulo en Python capaz de reconocer automáticamente hélices PPII en estructuras proteicas, utilizando la geometría como criterio principal. El módulo se integrará en PyMOL y se ha diseñado pensando en su aplicabilidad directa en proyectos científicos reales, como los que actualmente se desarrollan en el Instituto de Química-Física "Blas Cabrera" del CSIC (IQF-CSIC).

La elección de PyMOL como entorno de trabajo no ha sido casual. A pesar de la existencia de otras herramientas de análisis estructural como DSSP, STRIDE o ChimeraX, PyMOL destaca por su versatilidad, la posibilidad de personalizar y extender su funcionalidad mediante scripts, y su facilidad de uso por parte de científicos, que no tienen por qué tener experiencia en programación. Además, su integración con lenguaje Python lo convierte en una plataforma ideal para desarrollar complementos orientados a tareas específicas.

A lo largo de este documento se detalla el contexto científico del problema, se revisarán las herramientas existentes y se describe el proceso de desarrollo del módulo, incluyendo su lógica interna, las decisiones técnicas adoptadas, y los resultados obtenidos al aplicarlo sobre proteínas modelo. El objetivo final es desarrollar una herramienta útil que no solo cumpla con los fines académicos del trabajo, sino que también resulte de utilidad para Profesionales del ámbito científico interesados en el estudio de las hélices PPII.

## Motivación

La estructura tridimensional de las proteínas contiene una gran cantidad de información funcional. Entre las distintas conformaciones que pueden adoptar los residuos de una cadena polipeptídica, las hélices de tipo PPII han ganado protagonismo en investigaciones recientes por su implicación en procesos clave como la regulación de la expresión génica, la señalización celular o la consolidación de la memoria. A pesar de su relevancia, todavía no existen herramientas que permitan reconocerlas automáticamente, especialmente si se compara con la detección de otras estructuras secundarias como las α y β.

Esta situación se vuelve especialmente problemática en entornos de investigación donde se requiere analizar múltiples estructuras proteicas o caracterizar plegamientos inusuales. Poder contar con una herramienta capaz de detectar automáticamente las hélices PPII, integradas directamente en un entorno de visualización molecular, permitirá ahorrar tiempo, reducir errores manuales y facilitar la interpretación de los datos por parte de los investigadores.

Este trabajo nace precisamente de esa necesidad y justifica el desarrollo de una herramienta propia que facilite a los investigadores la detección de estas regiones de manera ágil y precisa, especialmente en el análisis de grandes conjuntos de proteínas o en modelos estructurales que aún no han sido completamente caracterizados, integrándose en su herramienta base, PyMOL, para dotarlo de una funcionalidad que actualmente no ofrece. Este desarrollo no solo resulta útil desde el punto de vista práctico, sino que también supone un reto técnico relevante dentro del marco de un trabajo de fin de grado (TFG).

## Planteamiento del trabajo

Este proyecto tiene como objetivo principal el desarrollo de un módulo en Python que detecte de forma automática hélices PPII en estructuras proteicas, utilizando como criterios los ángulos de torsión y la presencia de enlaces de hidrógeno no canónicos. La herramienta se integrará como complemento en PyMOL, permitiendo su ejecución desde la propia interfaz del programa y facilitando la visualización de los resultados.

Para alcanzar este objetivo, se plantean distintas fases. Si bien el desarrollo se centrará en PyMOL, se incluye un análisis inicial de herramientas similares disponibles en el ámbito bioinformático, con el objetivo de evidenciar las ventajas de esta elección. A continuación, se abordará el diseño del algoritmo de detección, el desarrollo del código en Python, la integración con PyMOL, la validación de resultados mediante casos de prueba, y la documentación del uso del módulo. El desarrollo se llevará a cabo con herramientas ya presentes en el entorno habitual de trabajo, como PyMOL, integrándose en Visual Studio Code (VS Code) para facilitar su depuración. Finalmente, el código se alojará en un repositorio público en GitHub, lo que permitirá su reutilización y mantenimiento.

Como caso de prueba, se utilizará la proteína 3BOG, una estructura bien documentada, sencilla y con una elevada proporción de hélices PPII, adecuada para validar la eficacia del algoritmo y su integración con el visor molecular.

Este proyecto aporta la novedad de automatizar la detección de hélices PPII en PyMOL, una funcionalidad que no está disponible de forma nativa ni cubierta por otras herramientas bioinformáticas. La posibilidad de identificar estas estructuras directamente desde el visor molecular, utilizando criterios objetivos basados en parámetros estructurales, contribuye a mejorar la eficiencia del análisis estructural en investigaciones científicas. Asimismo, la publicación del módulo en un repositorio público permitirá su acceso abierto, favoreciendo su uso por parte de otros grupos de investigación y fomentando su mejora colaborativa.

## Estructura del trabajo

El documento se estructura en tres bloques principales. En primer lugar, se presenta el contexto del problema y se analiza el estado del arte, comparando distintas herramientas para justificar el uso de PyMOL. A continuación, se describe el desarrollo del módulo en Python, explicando tanto la lógica del algoritmo como su integración con PyMOL y VS Code. Por último, se muestran los resultados obtenidos al aplicar la herramienta sobre proteínas modelo, evaluando su precisión y eficacia en la identificación de hélices PPII.

Esta fase incluye la validación funcional del módulo por parte del grupo de investigación, así como la depuración del código y la identificación de posibles mejoras para futuras versiones.

Este documento se estructura en seis capítulos, además de los apartados dedicados a bibliografía y anexos. En el Capítulo 1 se presenta la motivación del trabajo, el planteamiento general y una visión global de su contenido. El Capítulo 2 ofrece una revisión del estado del arte, analizando tanto el papel de las hélices PPII como las herramientas existentes para su estudio. En el Capítulo 3 se detallan los objetivos específicos y la metodología empleada, así como el entorno de desarrollo y las tecnologías utilizadas. El Capítulo 4 describe el desarrollo del módulo, explicando su estructura interna, su lógica de funcionamiento y el proceso de integración con PyMOL. En el Capítulo 5 se exponen los resultados obtenidos y se evalúa la funcionalidad del sistema propuesto. Por último, el Capítulo 6 recoge la validación real del trabajo y propone posibles líneas futuras de mejora o ampliación.

La memoria se complementa con un conjunto de anexos donde se incluye parte del código fuente, ejemplos de uso y documentación técnica adicional.

# Contexto y Estado del Arte

## Análisis del contexto

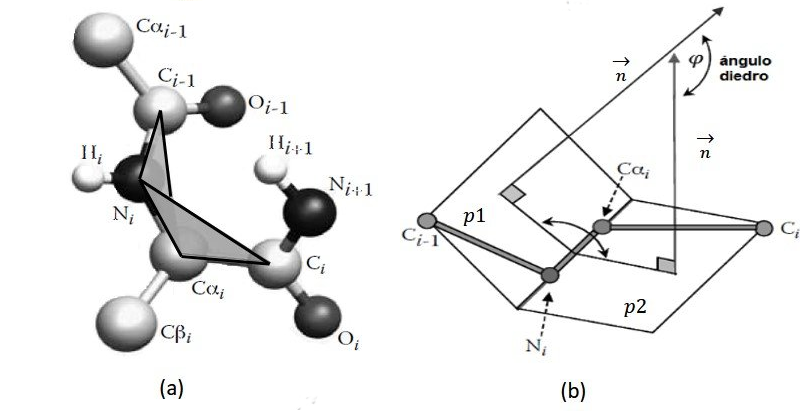
### El problema: detección de hélices PPII

Al comienzo de un estudio de investigación se realiza el cribado de las moléculas que se quieren caracterizar. En este caso, se seleccionan moléculas que potencialmente tengan PPII y que participen en las funciones objeto de análisis, **en función de los objetivos específicos del estudio**. La literatura científica refleja que la PPII está relacionada con procesos neuronales, incluyendo la memoria y el desarrollo de enfermedades neurodegenerativas.

Las proteínas están formadas por cadenas de aminoácidos unidas entre sí. Estas cadenas tienen una estructura flexible que puede girar en ciertos puntos. Cada átomo, dentro de la cadena central puede considerarse un plano con respecto a los enlaces que realiza con los átomos colindantes.

Un ángulo diedro es el ángulo que se forma entre dos planos que se juntan en una parte de esa cadena. Si se miran los átomos que forman parte de dos aminoácidos seguidos, se puede imaginar un plano antes del giro y otro después; el ángulo entre esos dos planos es el ángulo diedro.

Figura 1. Ángulo diedro Phi.



*Nota.* Adaptada de Blanquel, E., Blanquel, C., & Luna-García, R. (2019).

En la Figura 1 podemos ver el ángulo diedro Phi. En el apartado (a) se representa la disposición tridimensional de los átomos y la formación del ángulo alrededor del enlace N = Cα, y en el apartado (b) se ve el ángulo φ formado por la intersección de la normal del plano Ci+1 Ni, Cαi y la normal al plano Ni, Cαi., Ci.

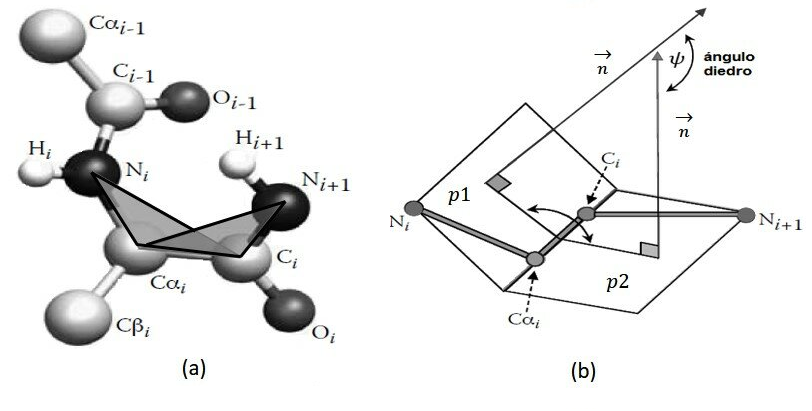
En cada aminoácido, hay dos giros importantes que se pueden medir:

El ángulo φ (phi), que gira alrededor del enlace entre el nitrógeno (N) y el carbono alfa (Cα).

El ángulo ψ (psi), que gira alrededor del enlace entre el carbono alfa (Cα) y el carbono del grupo carboxilo (C).

Estos giros son lo que se conoce como plegamiento y son necesarios para que la proteína adquiera una forma tridimensional específica. Esa forma es fundamental, ya que determina cómo va a funcionar la proteína dentro del organismo, por ejemplo, si va a encajar con otra molécula, catalizar una reacción o formar parte de una estructura celular.

Figura 2. Ángulo diedro Psi.



*Nota.* Adaptada de Blanquel, E., Blanquel, C., & Luna-García, R. (2019).

En la Figura 2 se observa el ángulo diedro psi. En la parte superior (a) se representa la disposición tridimensional de los átomos y la formación del ángulo alrededor del enlace Cα=C, y en la inferior (b), el ángulo ψ formado por la intersección de la normal del plano Ni, Cαi y Ci y la normal al plano Cαi, Ci, Ni+1.

Las hélices de poliprolina tipo II son una estructura secundaria particular, menos conocida que las α y β, pero cada vez más relevante en estudios de proteínas desordenadas, con dominios ricos en glicina. Su conformación cerrada, sus ángulos φ y ψ característicos y la posibilidad de formar enlaces de hidrógeno no canónicos las convierten en una pieza singular dentro del plegamiento proteico. A pesar de su importancia, estas estructuras no aparecen identificadas de forma explícita en los archivos estructurales, lo que obliga a los investigadores a localizarlas manualmente mediante inspección visual y análisis geométrico. Este aspecto complica notablemente la labor investigadora. De ahí surge la necesidad de desarrollar herramientas específicas que permitan detectar hélices PPII de forma automatizada, eficiente y reproducible, en entornos como PyMOL que ya son familiares para la comunidad científica.

#### Nota sobre el artículo base

Este proyecto surge directamente del trabajo de investigación recogido en el artículo *Architectonic Principles of Polyproline II Bundle Protein Domains*, firmado por Cristian Moisés Segura Rodríguez y Douglas Vison Laurents (2024). En él se describen las características estructurales de los haces de hélices PPII ricas en glicina, incluyendo parámetros geométricos específicos, patrones de enlace y distribución espacial.

El algoritmo desarrollado en este proyecto toma como referencia esos principios para automatizar su detección y facilitar el trabajo de quienes investigan estructuras similares. El artículo actúa como base teórica y guía directa en el diseño del sistema y en su validación, llevada a cabo a partir del análisis de la proteína modelo 3BOG.

### Método manual de hallar PPII con PyMOL

Nota. Salvo indicación en contrario, todas las figuras han sido elaboradas por la autora. Actualmente, para identificar las hélices de PPII se realiza el siguiente procedimiento:

mediante PyMOL.

1. **Inicialización del entorno de trabajo**:Se ejecuta la aplicación PyMOL desde el sistema operativo, abriendo la sesión de trabajo.
2. **Obtención de la proteína**: Se Importa la estructura que se desea analizar. Para ello se accede a *File → Get PDB* e introduce el identificador de la proteína, en este caso, 3BOG (ver Figura 3).

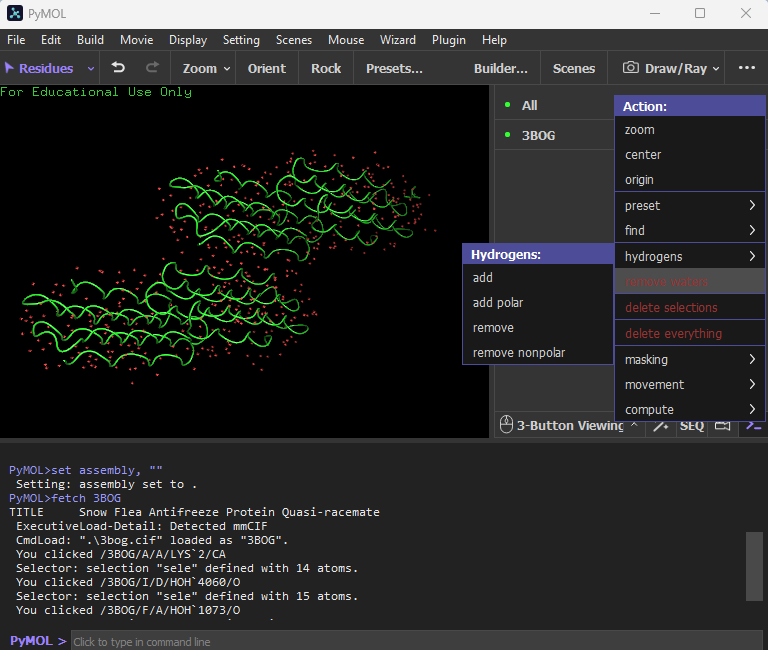
|  |  |
| --- | --- |
| Figura 3 Obtención de la proteína en PyMOL | Figura 4. Molécula 3BOG (PDB). |

Una vez cargado el archivo PDB, la estructura tridimensional de la molécula se muestra en la ventana principal de PyMOL. Se suele representar en modo cartoon o ribbon, aunque se puede personalizar a las necesidades del análisis. (Figura 4).

1. **Limpieza previa del archivo**: Se Retiran los solventes y ligandos no deseados, y se incorporan los átomos de hidrógeno necesarios para el análisis estructural.

Al ejecutar el comando *All → Action → Remove waters*, se eliminan las moléculas de solvente (agua) presentes en el archivo PDB, mostrando únicamente la proteína de interés. (Figura 5).

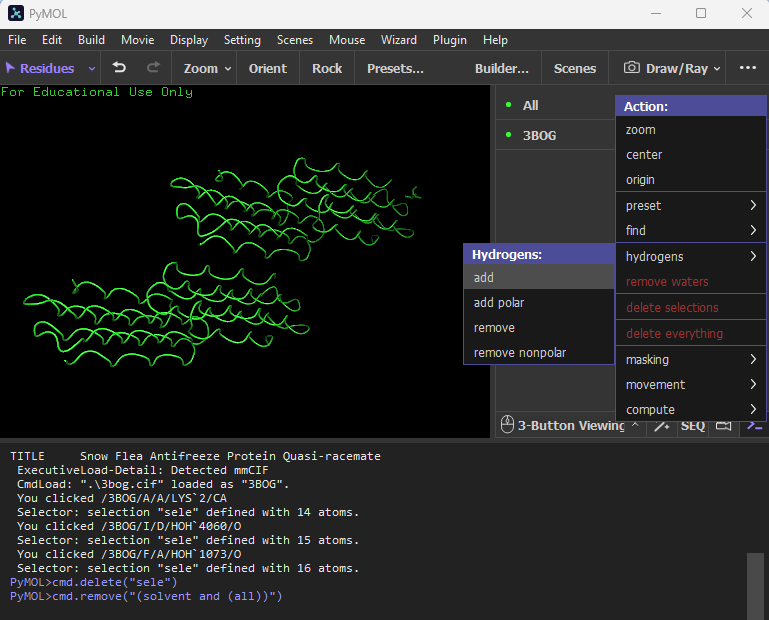
Figura 5. Eliminación de moléculas de agua



Mediante la acción All → Hydrogens → Add, se incorporan átomos de hidrógeno a la molécula. (Figura 6).

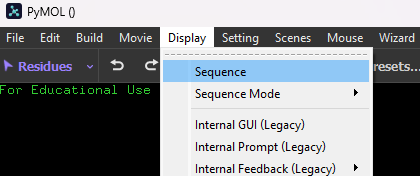
Nota. En otras estructuras se utiliza add polar hydrogens para evitar distorsiones, pero para detectar hélices PPII es recomendable añadir tanto los polares como los apolares.

Figura 6. Adición de hidrógenos.



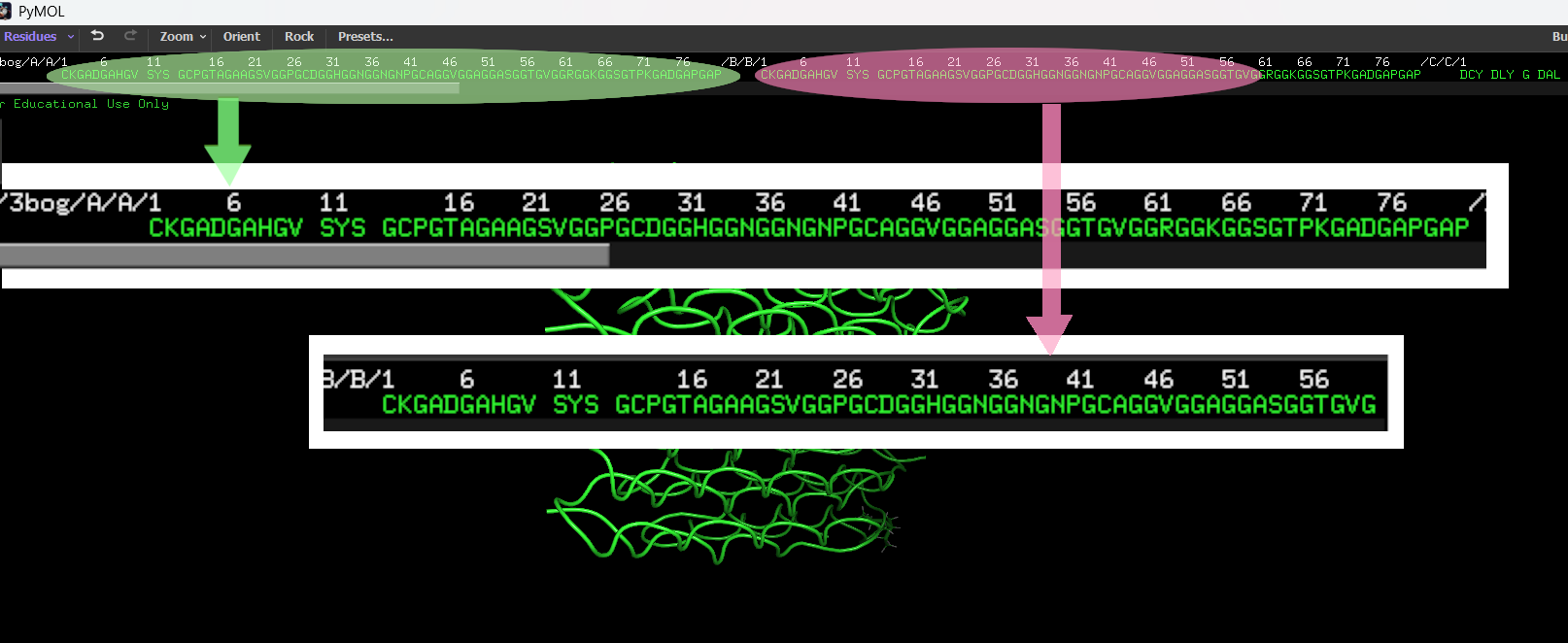
1. **Visualización de la secuencia de aminoácidos**: Se habilita la visualización de la secuencia mediante *Display → Sequence*, permitiendo seleccionar residuos específicos desde la interfaz gráfica. (Figura 7).

Figura 7. Abriendo la Secuencia.



Esta opción muestra la secuencia de aminoácidos en la parte superior de la interfaz de PyMOL. En la Figura 8 se ha ampliado la secuencia, separándola en dos trozos, para poder apreciar su contenido.

Figura 8. Visualización de la Secuencia en PyMOL.



Existe otras opciones de visualización utilizando “Sequence Mode”, pero excepto para moléculas muy grandes o que contenga demasiados aminoácidos modificados, es cómodo utilizar el modo por defecto.

1. **Identificación preliminar de potenciales hélices PPII**: Se inspecciona la proteína en busca de indicios de la presencia de segmentos con conformación PPII. Para agilizar el proceso en casos como 3BOG, en que ya sabemos dónde están las cadenas, se pueden buscar visualmente las hélices triangulares típicas de esta estructura o marcarlas mediante la secuencia indicada en el estudio (Figura 9). Este procedimiento no sirve al analizar moléculas nuevas, en las que la presencia de PPII no está demostrada o su localización dentro de la proteína no está clara. En esos casos se usan otros métodos más empíricos como la resonancia magnética nuclear. En un principio, para codificar el programa solo se utilizan moléculas que ya han sido estudiadas. El posterior empleo con otras moléculas confirmará la validez del mismo. Durante la inspección visual se utiliza el clic izquierdo para cambiar el ángulo de la proteína, el clic derecho para hacer zoom, y el clic central (presionar la rueda del ratón) para arrastrarla.

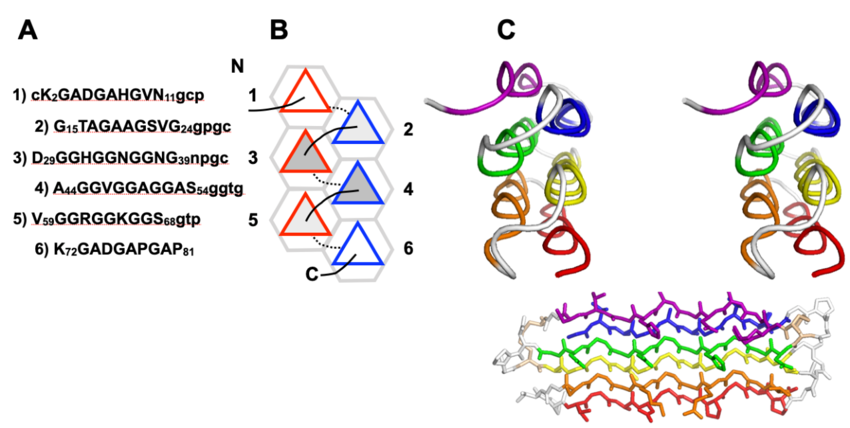
Figura 9. Hélices vistas en paralelo.



En la proteína estudiada (PDB ID: 3BOG), se observa un dominio conformado por seis hélices de PPII, unidas entre sí mediante cinco bucles. Esta organización estructural queda reflejada en la Figura 10, que se divide en tres partes:

1. Muestra las secuencias aminoacídicas correspondientes a cada hélice. Se evidencia la alta presencia de glicinas, las cuales favorecen la formación de la conformación PPII.
2. Representa un esquema del empaquetamiento de las hélices, donde se visualizan las interacciones entre ellas en un patrón hexagonal repetitivo.

Figura 10. Esquema representativo de la proteína 3BOG.



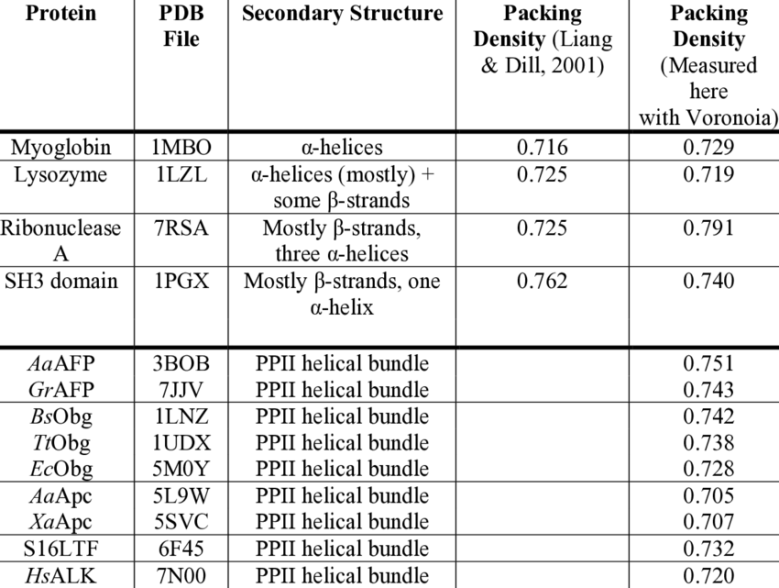
Nota.Reproducido de (Segura Rodríguez & Laurents, 2024).

1. Presenta la estructura tridimensional del péptido, donde cada hélice está coloreada de forma diferente para facilitar su representación.

Para identificar las hélices en 3BOG, se recurrió a los criterios derivados del análisis de densidad de empaquetamiento descritos en la Figura 11, extraída de la publicación de referencia.

Estos criterios consideran entre otros factores los ángulos de torsión φ y ψ característicos de la conformación PPII, la presencia de puentes de hidrógeno no canónicos, y la disposición densa y regular de las cadenas laterales.

Figura 11. Tabla de densidad del empaquetamiento de proteínas.

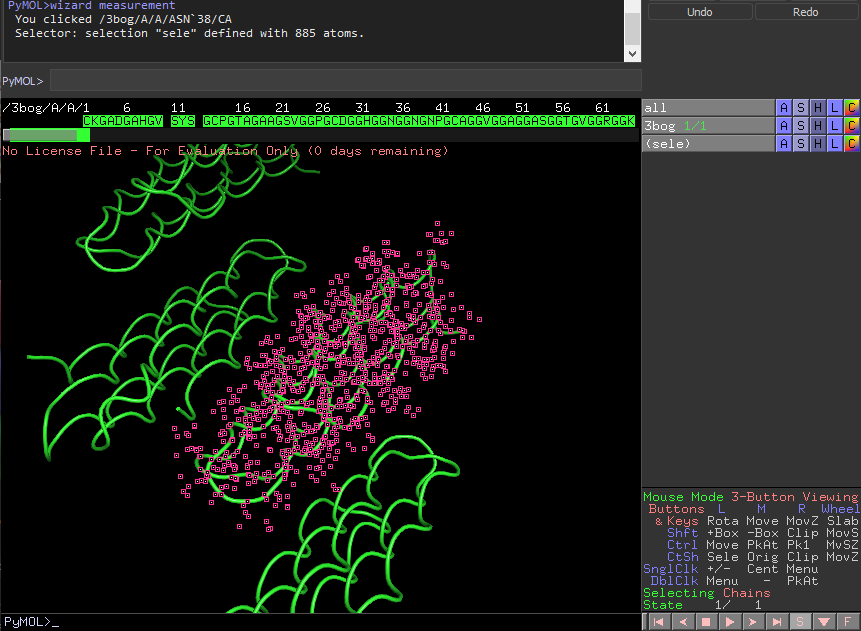


Nota.Reproducido de (Segura Rodríguez & Laurents, 2024).

**6. Aislamiento y coloreado de regiones candidatas:** Con el objetivo de facilitar la diferenciación de regiones PPII respecto al resto de la estructura, se seleccionan las regiones candidatas y se crean nuevos objetos en PyMOL, asignándoles colores específicos para su posterior análisis. Para ello se debe separar una sola de las proteínas de 3BOG, ya que en el archivo vienen varias. Para esto en la esquina inferior derecha, clicamos en la palabra después de *selecting*, que en este caso es *Residues*, y se cambia a *chains*. Una vez esté en el modo correcto, se presiona la cadena que se quiere separar. En este caso sirve cualquiera de ellas, ya que son clones. Esto causará que aparezca en pantalla un punto por cada átomo de cada aminoácido de la cadena seleccionada y se creará una selección denominada *(sele)* en el menú de la derecha, la cual contiene todos estos puntos, tal como se observa en la Figura 13.

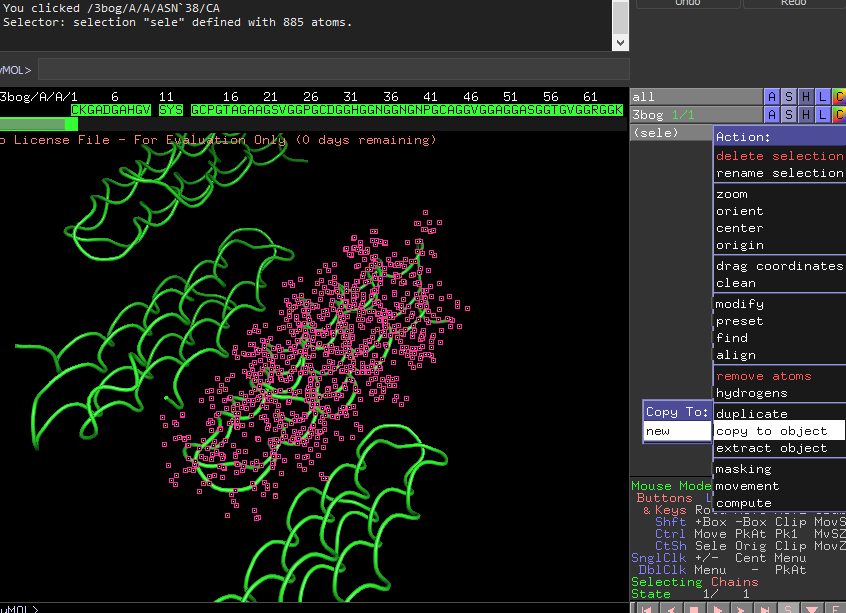
Figura 12. Menú de Selección

Figura 13. Seleccionando la Cadena.



Para copiar la cadena se utiliza (sele) > Action > copy to object > new. Lo cual creara un nuevo objeto que contiene solo esta cadena llamado objxx, donde xx es un número en base a cuantos objetos hayamos creado. En este caso debería ser obj01. Y mostrará una segunda secuencia debajo de la primera. Figura 14. También se podría extraer el objeto en vez de copiarlo, pero eso es más destructivo.

Figura 14. Separación de Cadena.



Para facilitar el análisis, se procede a ocultar la capa principal correspondiente a la estructura original (**3BOG**). De esta manera, se evita que interfiera con la edición y observación del objeto generado (**obj01**). Al realizar esta operación, la vista en PyMOL queda como se observa en la Figura 15.

Figura 15. obj01 en solitario.



Ahora se cambia el selector a *Residues* en la esquina inferior derecha, lo que permite la selección de aminoácidos individuales en lugar de cadenas completas. A continuación, se seleccionan los aminoácidos indicados en la secuencia arrastrando el ratón, siguiendo las referencias extraídas del estudio. En este caso, se identifican seis hélices, correspondientes a los siguientes intervalos:

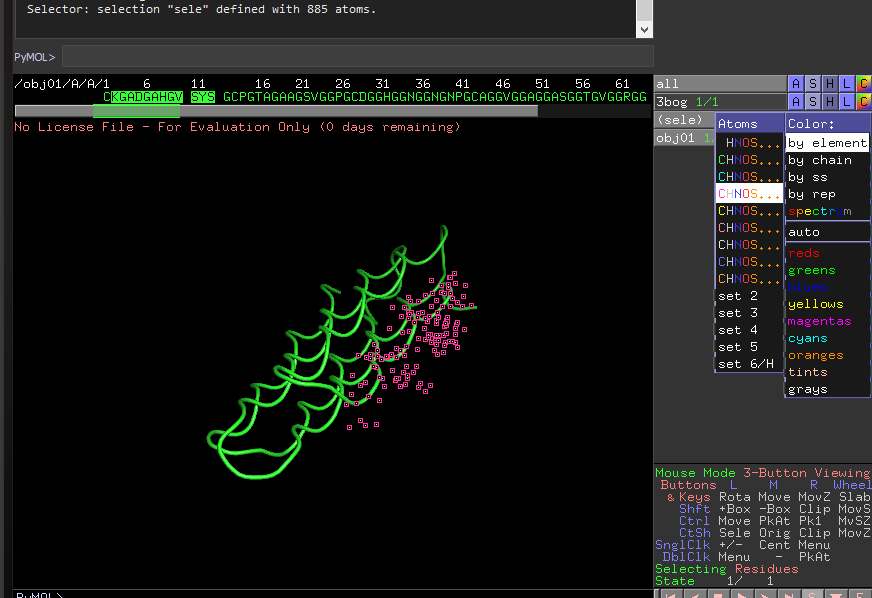
* Desde K2 a N11
* Desde G15 a G24
* Desde D29 a G39
* Desde A44 a S54
* Desde V59 a S68
* Desde K72 a P81

Cada letra representa el aminoácido según el código de una letra, mientras que el número indica su posición en la cadena, la cual suele marcarse en intervalos de cinco unidades sobre la propia secuencia en PyMOL. En caso de trabajar en modo de tres letras o nombres completos, sería necesario traducir las posiciones, aunque resulta más práctico mantener el modo de una letra.

Es importante señalar que, en este archivo, algunos residuos aparecen anotados como *SYS*, una variante modificada del aminoácido *N* (**Asparagina**), lo que explica su representación como N en el estudio de referencia.

Las hélices pueden seleccionarse todas simultáneamente o, como se muestra en la Figura 16, hacerlo de manera individual. Una vez seleccionada cada hélice, se realizan dos acciones principales: en primer lugar, se cambia su color a través del menú (sele) > color > by element > color escogido.

Figura 16. Separación y Coloreado de las hélices seleccionadas.

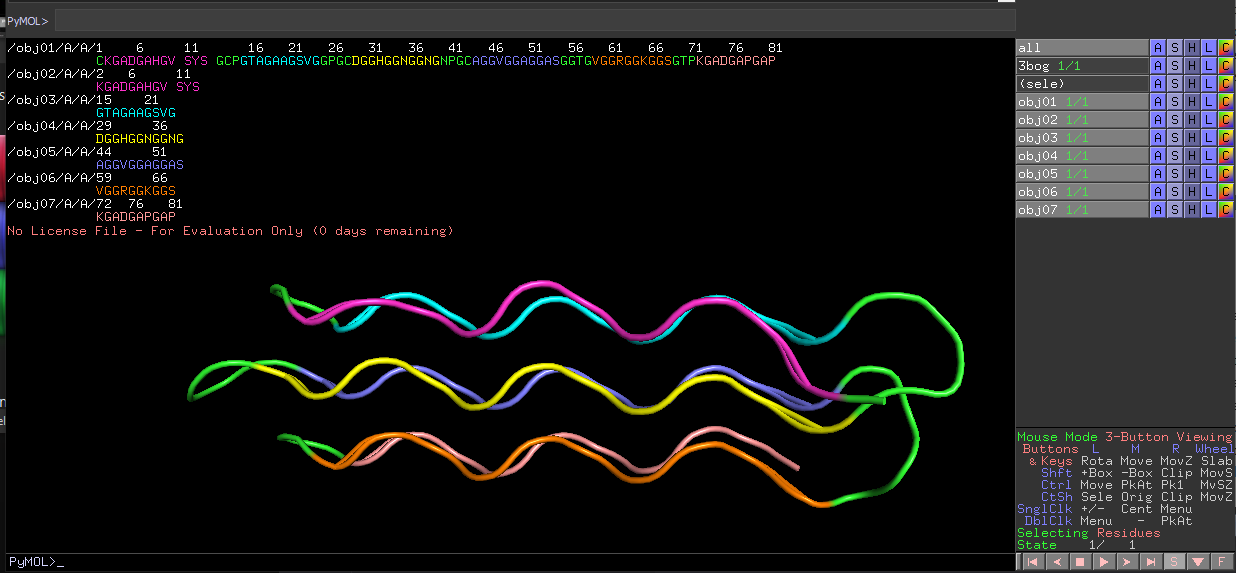


Se utiliza la opción *by element* para aplicar el cambio de color únicamente a los carbonos, de forma que los demás átomos puedan identificarse fácilmente en pasos posteriores. Es recomendable escoger un color distinto al utilizado originalmente para los carbonos del resto de la molécula, a fin de facilitar su diferenciación. Asimismo, resulta conveniente alternar colores entre hélices adyacentes para evitar confusiones durante el análisis.

Una vez coloreada cada hélice, se procede a copiar la selección en un nuevo objeto, utilizando el mismo comando descrito anteriormente. Este procedimiento se repite para cada una de las seis hélices, tal como se ilustra en la Figura 17.

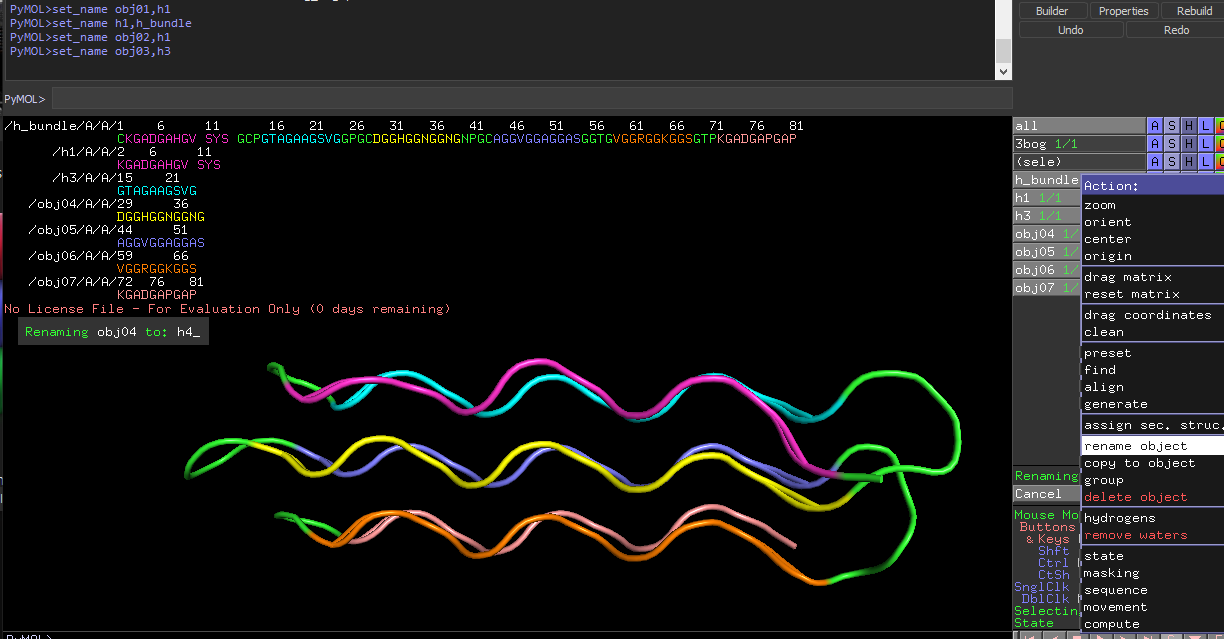
Cabe señalar que, al buscar hélices en nuevos archivos de proteínas, normalmente no se realiza esta separación en objetos hasta haber medido los ángulos phi (φ) y psi (ψ) en el paso 7, ya que la asignación de colores en las etapas iniciales es solo orientativa. Sin embargo, dado que en este caso se trabaja con un archivo previamente conocido, resulta más eficiente realizar la separación desde el principio.

Figura 17. Hélices Identificadas.



Opcionalmente se puede asignar un nombre a cada hélice para facilitar su identificación posterior. Esta acción se realiza seleccionando Action > Rename Object, lo que abre un cuadro de texto en la parte izquierda de la ventana, permitiendo introducir el nuevo nombre asignado al objeto. Este proceso se ilustra en la Figura 18.

Figura 18. Dando nombre a las hélices.



Se utiliza la nomenclatura h\_bundle (hélix bundle) para referirse a la proteína separada del resto de clones, y hx para nombrar a cada hélice individual, donde x representa el número de la hélice según su posición en la cadena.

Posteriormente, con el fin de poder visualizar los átomos individuales, se cambia el modo de visualización de la hélice o hélices deseadas al formato *regaliz*. Para ello, se selecciona el objeto correspondiente (hx) y se accede a *show > as > licorice*, como se muestra en la Figura 19.

Figura 19. Visualización de hélices en modo regaliz.



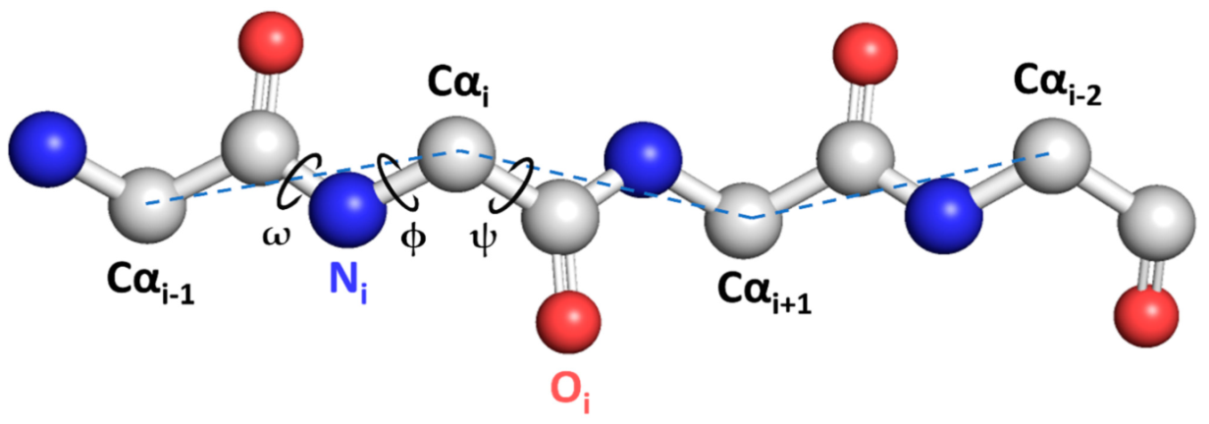
Si se desea volver al modo de visualización anterior, el procedimiento es similar, seleccionando *ribbon* o *cartoon* en lugar de *licorice*. En el estudio la diferencia entre estos modos no resulta relevante para el análisis. Es importante señalar que, si no se utiliza la opción as y se activa directamente *licorice*, se añade la nueva representación sin eliminar la anterior, provocando una superposición de cadenas que dificulta la correcta identificación de los carbonos.

7. **Medición de ángulos diedros phi (φ) y psi (ψ)**:La confirmación de la presencia de una hélice PPII se realiza mediante la medición de los ángulos diedros característicos phi (φ) y psi (ψ). El procedimiento consiste en utilizar las herramientas de medición de PyMOL para calcular estos ángulos en los residuos seleccionados, comparándolos posteriormente con los valores típicos asociados a la conformación PPII.

Recientemente PyMOL incorpora el comando phi\_psi o su variante en script cmd.phi\_psi, que permite calcular directamente estos valores. Sin embargo, en la práctica habitual, se continúa realizando esta tarea de forma manual y visual, dado que el comando únicamente proporciona los valores numéricos sin representación gráfica, dificultando su interpretación directa.

En caso de duda durante el proceso, puede resultar útil consultar la Figura 20.

Figura 20. Ángulos Phi, Psi y Omega.



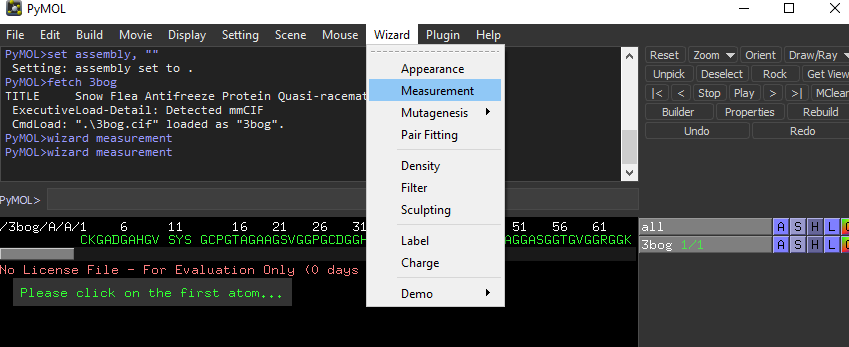
Fuente: Molecules, (Broz, Jukič, & Bren, 2023)

En esta imagen se puede observar como la cadena principal de una proteína está formada por una sucesión de aminoácidos unidos mediante enlaces peptídicos, configurando una secuencia repetitiva:

N--Carbonilo-N--Carbonilo…

En esta cadena, el ángulo Phi φ es el ángulo entre el nitrógeno (N) y su mientras que el psi ψ es el enlace entre un y su Carbonilo (C=O). En la Figura 20, los átomos de nitrógeno (N) están representados como esferas azules, los carbonos como esferas blancas, y los oxígenos del grupo carbonilo como esferas rojas. Los carbonos alfa son los átomos centrales del aminoácido que están entre un N y un Carbonilo y enlazados a sus respectivas cadenas laterales, que no se muestran en la imagen ya que no forman parte de la cadena central. Para medir estos ángulos en PyMOL, se emplea el Wizard accesible desde el menú superior mediante Wizard > Measurement, según revela la Figura 21.

Figura 21. Activando la herramienta Wizard

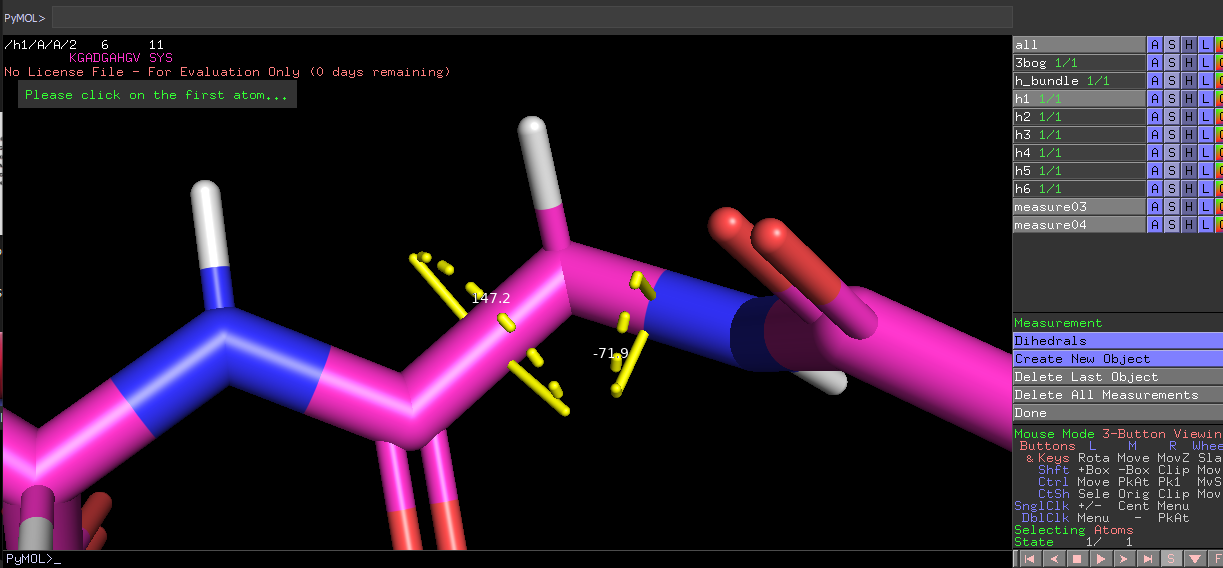


Esto abre el menú wizard en la esquina inferior izquierda. Conviene asegurarse que esté en modo diedro con Measurement Mode > Dihedrals. La opción Angles indica otro tipo de ángulos que se usará para los puentes de hidrógeno h-bonds. Measurement Mode es siempre la primera pestaña del wizard, e indica directamente el modo activo en cada momento. Se puede apreciar que, al cambiar el modo de medición a ángulos diedros, cambia el modo de selección a Atoms, visualizado en la esquina inferior derecha automáticamente, tal como muestra la Figura 22.

Figura 22. Modo Diedros

Una vez en modo wizard el procedimiento consiste en pinchar en orden los cuatro átomos que delimitan el ángulo diedro. Se recomienda ocultar previamente todas las hélices que no se desean analizar. Para ello, se hace clic en el nombre de los objetos correspondientes, desactivando su visualización para que no obstaculicen la trayectoria de selección. A continuación, se coloca la proteína en una posición cómoda para la manipulación, y se señalan los cuatro átomos consecutivos que forman el ángulo diedro, es decir, los dos átomos que forman el ángulo a medir y sus dos átomos colindantes, en concreto, el siguiente y el anterior en la cadena. Ver Figura 23.

Figura 23. Phi y Psi en PyMOL.



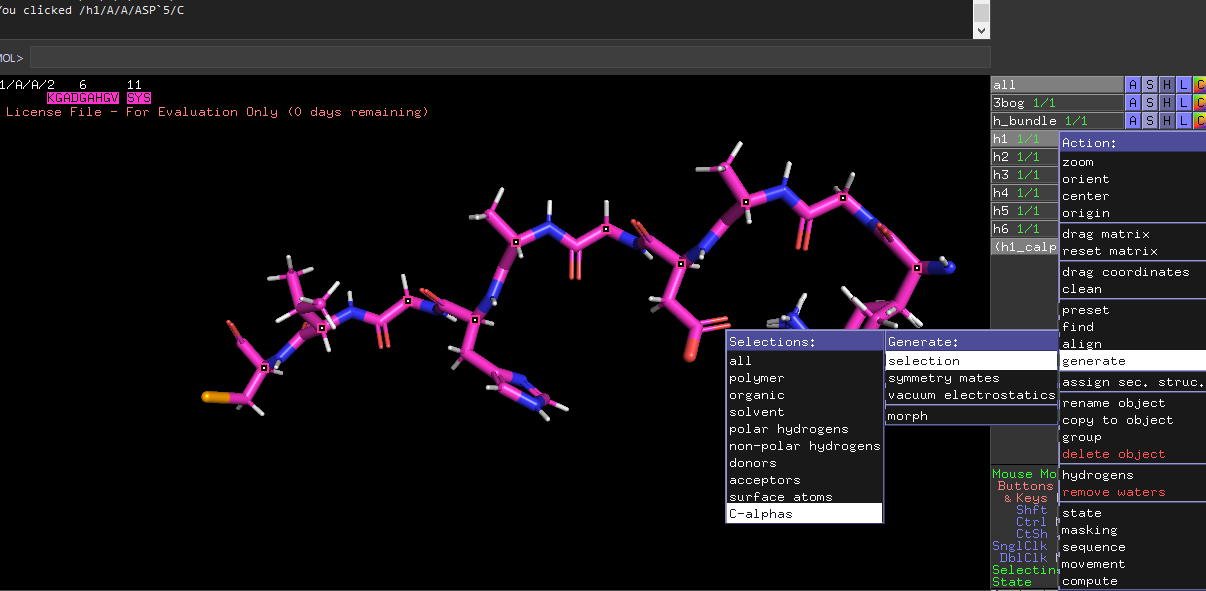
Es fundamental respetar el orden de selección de los átomos, de un extremo a otro de la cadena. Aunque el sentido (izquierda-derecha o derecha-izquierda) no afecta el resultado, un error en el orden de los clics puede provocar una interpretación incorrecta de los puntos espaciales, generando valores de ángulo sin sentido. En caso de duda sobre el proceso de medición, existen guías detalladas disponibles en línea, como por ejemplo el video Practical course: Calculate phi, psi and omega angles of proteins in PyMOL (brooksj4, 2020).

Al disponer de los ángulos deseados, basta con pulsar en *Done* para salir del wizard. A continuación, se anotan los datos en una tabla para su posterior análisis. Se consideran como valores de referencia aquellos **ángulos diedros** cercanos a **ϕ ≈ –75°** y **ψ ≈ 145°**, correspondientes a la conformación típica de hélice PPII. Pero hay que tener en cuenta que los valores de los ángulos no son exactos, por lo que se ha de establecer un margen de tolerancia, mediante una constante ajustable.

1. **Identificación de los Carbonos Alfa y Oxígenos de Carbonilo:** La presencia de enlaces de hidrógeno no canónicos entre carbonos alfa y carbonilos de la cadena central de hélices colindantes es característica de las hélices PPII. Dado que hasta la fecha PyMOL solo identifica enlaces de hidrógeno canónicos, los cuales cuentan con donantes, aceptores y geometría distinta, este proceso es laborioso y requiere de una etapa de preparado previo al análisis.

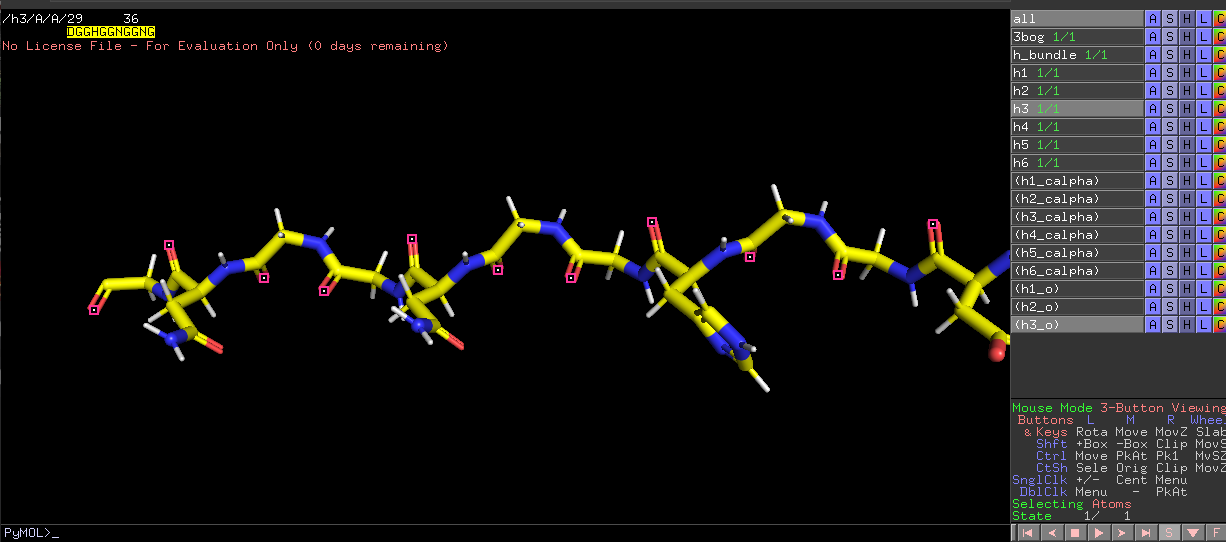
Para este paso se requiere la identificación de los carbonos alfa (cuyos hidrógenos son los donantes de electrones en este enlace) y los oxígenos de carbonilos de la cadena central, que realizan la función de aceptores. Encontrar los es fácil porque ya existe un comando que lo realiza en PyMOL. Tan solo se ha de usar: *hx > generate > selection > c-alphas* como evidencia la Figura 24, repitiendo por cada hélice. Lo cual genera una selección hx\_calpha en un solo clic.

Figura 24. Selección de c-alphas.



Identificar los oxígenos de la cadena central es más complicado dado que, aunque hay un comando para localizar oxígenos, este no discrimina en base a su localización. Por ello, en vez de utilizar ese comando y luego borrar manualmente los oxígenos no deseados, suele ser más fácil seleccionar manualmente los objetivos. Para ello se utiliza el selector en modo *Átomos*, y una a una se seleccionan los oxígenos de carbonilo de cada hélice, tras lo cual se ha de renombrar *(sele)* por *hx\_o*, siendo x el número de la hélice en cuestión, antes de pasar a la siguiente, creando un objeto por cada una como se muestra en la Figura 25.

Figura 25. Selección de oxígenos.



1. **Identificación de enlaces de hidrógeno no canónicos**: Una vez se han creado las selecciones *hx\_calpha* y *hx\_o*, estas pueden utilizarse para realizar las mediciones de distancia y ángulo requeridas para identificar posibles enlaces de hidrógeno no canónicos presentes en la molécula.

Las hélices de PPII solo realizan estos enlaces entre hélices de PPII colindantes. Como se muestra en la Figura 10, las hélices de PPII suelen colindar con entre otras 2 y 6 hélices a su alrededor. Aunque hay excepciones en que solo colindan con 1, y no se descarta que haya alguna excepción con más de 6, pero todavía no se ha detectado ninguna.

Para medir la distancia solo se utilizar el comando *distance* de L, cuya explicación detallada se encuentra disponible en la documentación oficial (Distance - PyMOLWiki, s.f.).

La estructura general del comando es la siguiente:

distance distxoy, (hx\_calpha), (hy\_o), 5

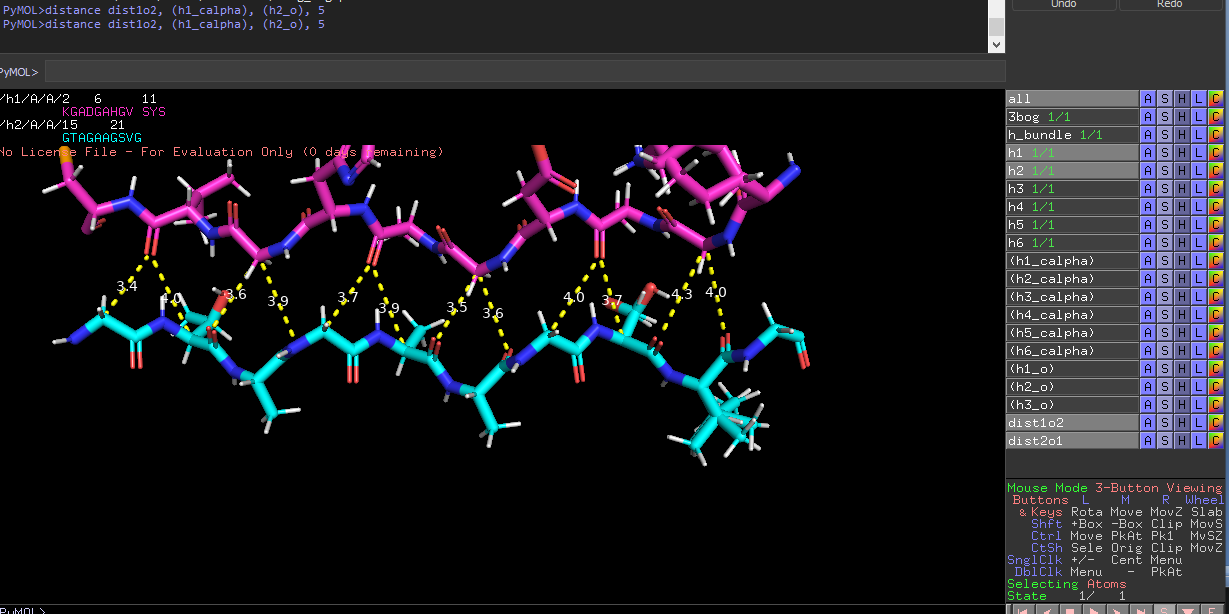
en que distxoy es el nombre que le dará PyMOL al nuevo objeto que almacena las distancias calculadas. *hx\_calpha y hy\_o* son las selecciones de *c-alpha* y *oxigeno* de las dos cadenas, y 5 es el punto de corte de distancia (en angstroms), a partir del cual PyMOL dejará de medir. Se recomienda utilizar 5 **Å**, porque es suficiente para abarcar los enlaces de hidrógeno no canónicos, incluso algo generoso, sin llegar a oxígenos de residuos lejanos. En caso de omitir este parámetro, PyMOL calcularía absolutamente todas las distancias posibles entre los átomos seleccionados, lo que saturaría la visualización y dificultaría la interpretación de los datos. Por ejemplo:

distance dist1o2, (h1\_calpha), (h2\_o), 5

distance dist2o1, (h2\_calpha), (h1\_o), 5

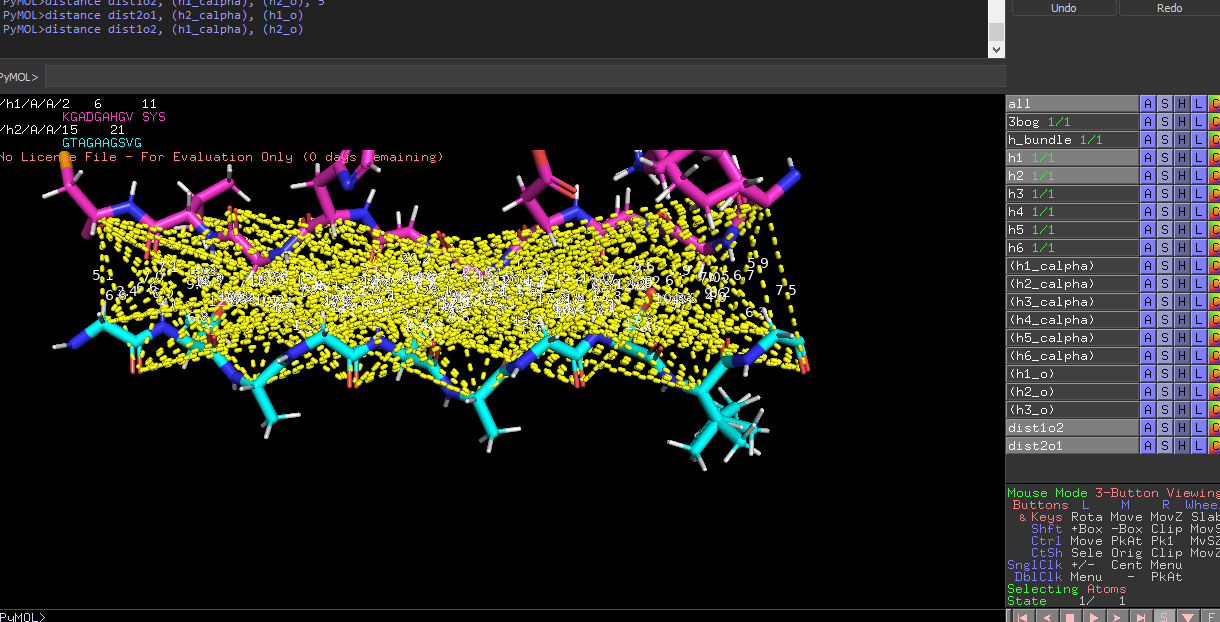
La correcta aplicación de estos comandos genera la Figura 26, una representación limpia y clara de las posibles localizaciones de puentes de hidrógeno no canónicos.

Figura 26. Posibles localizaciones de puentes de hidrógeno no canónicos*.*



En contraste, si se omite el parámetro de distancia máxima, el resultado es una saturación visual que impide el análisis efectivo, tal como se observa en la **Figura 27.**

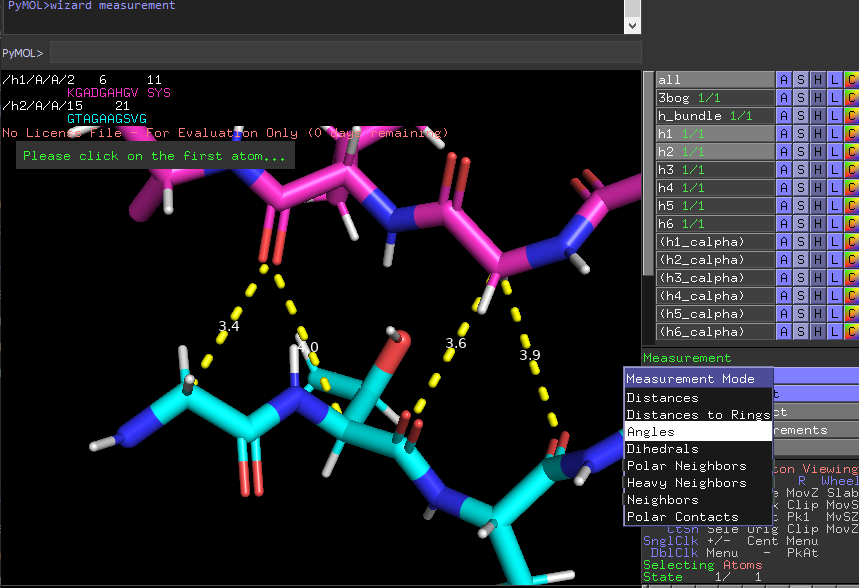
Figura 27. Posible migraña inducida por mala praxis.



Cabe resaltar que, como se ha realizado anteriormente, deben medirse las distancias en, se deben medir las distancias en ambas direcciones, tanto *h1\_calpha > h2\_o* como *h2\_calpha > h1\_o*, dado que ambas interacciones pueden generar puentes de hidrógeno no canónicos.

Una vez tenemos estas distancias, el siguiente paso consiste en volver al wizard, esta vez utilizando el modo *Angles*, como se refleja en la Figura 28.

Figura 28. Selección de ángulos en Wizard.

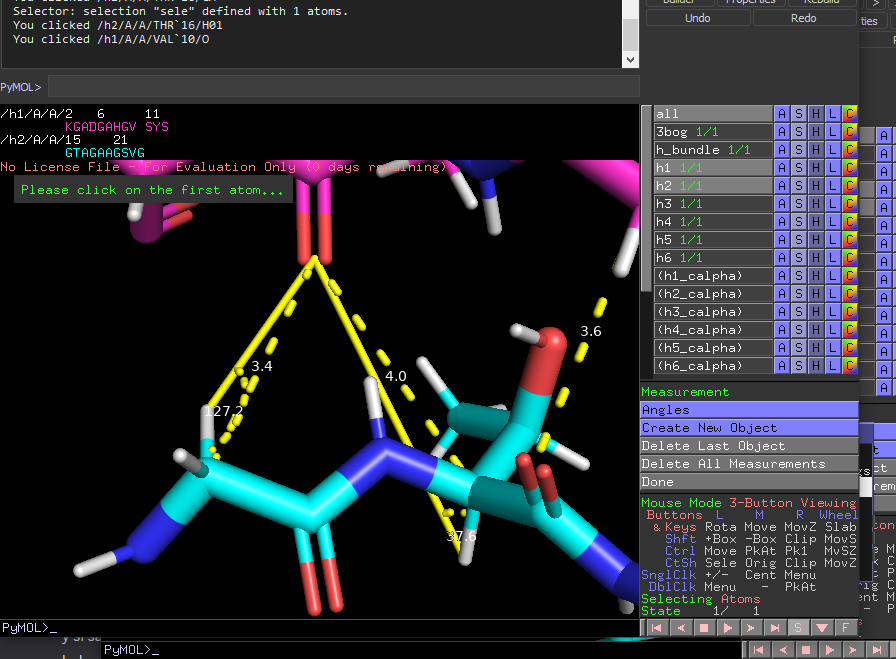


Para medir el ángulo correspondiente a un posible puente de hidrógeno, se debe hacer clic, en orden, primero sobre el **carbono alfa**, después sobre el **hidrógeno**, y por último sobre el **oxígeno**. Alternativamente, también es válido seleccionar primero el **oxígeno**, seguido del **hidrógeno**, y finalmente el **carbono alfa**, siempre que el **hidrógeno quede en la posición central**.

Al igual que sucedía con los diedros, es fundamental respetar el orden de selección, ya que PyMOL no interpreta automáticamente la geometría, y un orden incorrecto podría generar valores de ángulo erróneos.

Tras seleccionar los tres átomos en el orden adecuado, el wizard creará automáticamente un objeto que contiene el ángulo, como se muestra en la Figura 29.

Figura 29. Cribado de los posibles puentes de hidrógeno no canónicos.



A partir de este punto, solo queda repetir esto para todos los ángulos plausibles entre los átomos indicados por las distancias previamente obtenidas.

Algunos ángulos pueden descartarse mediante inspección visual. Por ejemplo, de los dos ángulos mostrados en la Figura 29, el de la derecha tiene el hidrógeno apuntando en la dirección opuesta a la cadena colindante, por lo que el ángulo es sin duda mucho mayor del permitido para la formación de un puente de hidrógeno.

Sin embargo, se recomienda medir todos los ángulos, incluso aquellos que aparentemente se desvían del rango esperado, ya que las perspectivas visuales pueden inducir a error y hacer que ángulos válidos parezcan más abiertos de lo que realmente son, debido a la perspectiva.

#### **Identificación de hélices PPII**: Una vez obtenidas las distancias y los ángulos correspondientes, el siguiente paso consiste en organizarlos en una tabla para identificar qué residuos presentan valores compatibles con la formación de puentes de hidrógeno no canónicos, y cuáles fallan en al menos uno de los dos parámetros analizados.

Si se observa un número suficiente de puentes de hidrógeno y los ángulos están dentro del rango característico, puede considerarse que existe una confirmación estructural de la conformación PPII. No obstante, todavía está en debate cuántos enlaces de hidrógeno son necesarios para validar dicha conformación. Teóricamente, debido a la geometría de las hélices PPII, dos cadenas tienden a formar un puente de hidrógeno cada tres residuos. Sin embargo, en la práctica, este patrón ideal rara vez se cumple por completo.

Esto se debe a que no todos los aminoácidos pueden formar puentes de hidrógeno no canónicos ya que, frecuentemente, los extremos de la cadena rompen la regularidad del patrón. Como consecuencia, el porcentaje real de enlaces de hidrógeno observados en una hélice PPII suele ser inferior al 33 % esperado teóricamente. En cadenas más complejas que el modelo 3BOG, utilizado como ejemplo, esta proporción puede ser considerablemente menor.

Por tanto, aunque el programa sea capaz de proporcionar las distancias y ángulos asociados a enlaces potenciales, la interpretación final de los datos debe quedar en manos del investigador. Es este quien, con criterio experto, debe valorar si la evidencia estructural resulta suficiente para confirmar la presencia de una hélice PPII.

Finalmente, es importante destacar que el presente método **solo considera interacciones entre cadenas PPII**. Las posibles interacciones con cadenas no PPII, que seguirían un procedimiento diferente, quedan fuera del alcance de este análisis.

## Estado del arte

### Comparativa de herramientas existentes para análisis estructural

Existen diversas herramientas bioinformáticas orientadas a la visualización y análisis de estructuras moleculares, cada una con enfoques específicos según el área de aplicación. Por ejemplo, VMD (Theoretical and Computational Biophysics Group, 2025) es especialmente útil en estudios de dinámica molecular debido a su eficiencia en la representación de trayectorias complejas, mientras que Jmol (Jmol Development Team, 2025) es ampliamente empleado en entornos educativos por su ligereza y compatibilidad web. Avogadro (Open Babel, 2025), por su parte, está diseñado para química computacional y modelado basado en mecánica cuántica, siendo menos adecuado para proteínas.

Dentro del análisis estructural clásico, herramientas como DSSP y STRIDE permiten asignar estructuras secundarias de forma automática a partir de ángulos diedros y enlaces de hidrógeno. Sin embargo, estas plataformas no identifican explícitamente las hélices de tipo poliprolina II (PPII), que suelen quedar clasificadas como regiones *coil* o desordenadas. Esto obliga a los investigadores a recurrir a análisis visuales o mediciones adicionales para su detección.

Entre los visores moleculares avanzados destacan UCSF Chimera (University of California, San Francisco, 2025b) y su evolución, ChimeraX (University of California, San Francisco, 2025a), que ofrecen herramientas gráficas potentes para el análisis estructural, interpretación de mapas de densidad electrónica y generación de animaciones. No obstante, su sistema de scripting es menos accesible que el de herramientas como PyMOL, lo que presenta limitaciones a la hora de personalizar funciones específicas, especialmente cuando se requiere trabajar con estructuras no anotadas o introducir criterios geométricos propios.

También existen plataformas comerciales como MOE, Schrödinger Maestro o Discovery Studio, que integran funciones avanzadas de modelado, docking y simulación molecular. Estas soluciones, si bien muy completas, están orientadas a grandes entornos industriales o farmacéuticos, requieren licencias costosas y no siempre permiten adaptar el entorno a tareas especializadas como la automatización del tipo de análisis estructural que se aborda en este trabajo.

La Tabla 1 resume las principales características de las herramientas mencionadas:

Tabla 1. Comparativa visores moleculares y su utilidad

|  |  |  |
| --- | --- | --- |
| **Programa** | **Mejor para** | **Gratuito** |
| PyMOL | Visualización y análisis estructural | Versión open-source disponible |
| ChimeraX | Visualización avanzada y estructuras grandes | SI |
| UCSF Chimera | Análisis detallado de estructuras PDB | SI |
| Jmol | Uso web y enseñanza | SI |
| Avogadro | Química computacional y modelado | SI |
| VMD | Simulación de dinámica molecular | SI |
| RasMol | Visualización ligera y rápida | SI |
| MOE | Diseño de fármacos y modelado | NO |
| Discovery Studio | Modelado molecular y docking | Versión limitada |
| Materials Studio | Modelado de materiales y química cuántica | No |
| Schrödinger Maestro | Diseño de fármacos y simulaciones avanzadas | No |

Fuente: Adaptado sitios web oficiales de cada herramienta.

### Elección de PyMOL

Frente a estas alternativas, PyMOL (Schrödinger, LLC, 2015) se consolida como una de las soluciones más completas para el análisis estructural en biología molecular, gracias a su equilibrio entre potencia gráfica y capacidades de automatización. Aunque no identifica automáticamente las hélices PPII, ofrece todas las funcionalidades necesarias para su detección manual, de forma que permite medir ángulos diedros φ y ψ, analizar la geometría que nos permite predecir la presencia enlaces de hidrógeno no canónicos y manipular visualmente segmentos de interés. Lo más relevante en este contexto es su integración completa con Python, que permite automatizar las funcionalidades claves de este proyecto mediante scripts, reduciendo el tiempo de análisis y eliminando la subjetividad del proceso visual.

PyMOL es una herramienta de código abierto, con una comunidad activa que desarrolla extensiones, módulos y mejoras continuamente. Se ha consolidado como una opción de referencia en la comunidad científica, especialmente en laboratorios académicos y centros de investigación donde la flexibilidad, la transparencia del código y la capacidad de adaptación son criterios fundamentales, convirtiéndose en un recurso clave para la visualización y análisis de estructuras moleculares en 3D.

Entre sus funcionalidades más relevantes, PyMOL permite estudiar de manera detallada la estructura de proteínas, ADN y otras moléculas, mediante diferentes representaciones, como cartoon, sticks o superficies. Del mismo modo posibilita efectuar cálculos y mediciones, superponer y/o comparar diferentes estructuras, seleccionar y manipular moléculas, o sus regiones específicas, de forma independiente. Además, facilita la interpretación de mapas de densidad electrónica y el diseño computacional de fármacos. Cabe destacar su capacidad para integrarse con diversas herramientas complementarias, lo que posibilita análisis más profundos y refuerza su utilidad en investigación. Todo ello lo convierte en una plataforma sólida y eficaz para el análisis molecular. Su potencia radica en que permite que los usuarios puedan modificar el código según sus necesidades, por lo que tiene muchas funcionalidades implementadas en módulos creados por la comunidad científica.

Además, es la plataforma empleada habitualmente para la caracterización molecular en el laboratorio de investigación donde se origina este proyecto, lo que facilita su integración en el flujo de trabajo real y garantiza la validación experimental del módulo. Esta compatibilidad técnica y metodológica refuerza aún más mi elección frente a otras opciones.

A modo de resumen de todo lo anterior, la Tabla 2 muestra que, aunque la mayoría de estas herramientas resultan útiles para tareas generales de visualización y análisis estructural, presentan limitaciones importantes a la hora de identificar estructuras menos comunes como las hélices PPII. Ninguna de ellas, con excepción de PyMOL, ofrece una integración completa con Python ni la flexibilidad necesaria para implementar un algoritmo de detección personalizado basado en ángulos diedros, distancias atómicas y enlaces no canónicos.

Tabla 2. Comparativa de herramientas respecto al proyecto de detección PPII

| **Herramienta** | **Características principales** | **Limitaciones respecto a la detección de hélices PPII** |
| --- | --- | --- |
| DSSP | Asignación automática de estructuras secundarias basada en geometría y enlaces de hidrógeno. | No reconoce hélices PPII como categoría propia. No es personalizable. |
| STRIDE | Añade criterios energéticos a los geométricos. Alta precisión en hélices α y láminas β. | No contempla la conformación PPII de forma explícita. Sin opciones de automatización. |
| Chimera | Visor avanzado con herramientas gráficas y de análisis integradas. | Requiere scripting propio. Menor flexibilidad frente a Python para tareas específicas. |
| ChimeraX | Versión optimizada de Chimera, con mayor rendimiento y visualización moderna. | Curva de aprendizaje elevada. Limitado para desarrollo de módulos personalizados. |
| VMD | Enfocado a simulaciones de dinámica molecular. Visualización eficiente y rápida. | No orientado al análisis geométrico detallado. Escasa personalización. |
| Jmol | Ligero, multiplataforma y muy utilizado en docencia. | Inadecuado para análisis estructural avanzado o desarrollo de herramientas propias. |
| PyMOL | Integra visualización y scripting en Python. Permite análisis atómico detallado y desarrollo de módulos personalizados. | No incluye detección PPII por defecto, pero permite implementarla de forma precisa. |

## PyMOL como Herramienta para el Análisis Estructural

PyMOL es la herramienta elegida para este proyecto debido principalmente a su capacidad de integración con Python, lo que facilita la automatización y personalización de análisis estructurales complejos mediante scripts. Esta flexibilidad, combinada con su uso extendido en laboratorios y grupos de investigación biomolecular, lo convierte en el entorno ideal para desarrollar un módulo específico que automatice la detección de hélices de poliprolina II (PPII), facilitando un proceso hasta ahora manual y laborioso.

### ¿Qué es PyMOL?

PyMOL es un software especializado en la visualización y análisis tridimensional de estructuras biomoleculares, especialmente proteínas y ácidos nucleicos. Desarrollado inicialmente por Warren DeLano y actualmente mantenido por Schrödinger LLC, destaca por su potente interfaz gráfica y capacidades avanzadas de scripting mediante Python, permitiendo realizar análisis detallados y automatizar tareas repetitivas que consumen mucho tiempo en los laboratorios de investigación.

### Principales funcionalidades utilizadas en este proyecto

En los grupos de investigación del Instituto de Química-Física "Blas Cabrera" (CSIC), PyMOL se utiliza de forma práctica y directa en el contexto específico del análisis estructural de proteínas, en particular para la detección de hélices de poliprolina tipo II (PPII). El flujo de trabajo habitual comprende los siguientes pasos:

**Carga y visualización:**

Para comenzar el análisis estructural, se importan las estructuras moleculares directamente desde la base de datos del Protein Data Bank (PDB) mediante identificadores específicos. Un comando típico empleado desde la consola de PyMOL es:

fetch 3bog

aunque esta acción también puede realizarse desde la interfaz gráfica mediante la opción “*Get PDB…”*, que internamente ejecuta el mismo comando.

Esto permite cargar de forma sencilla la estructura deseada. A continuación, el científico suele visualizar inicialmente la proteína en modo *cartoon* para obtener una representación global clara de su organización estructural, y más adelante se cambia al modo *licorice* cuando precisa examinar átomos individuales o realizar mediciones detalladas.

**Preparación estructural:**

Previamente al análisis detallado, se realiza una limpieza automática de la estructura para eliminar moléculas innecesarias y añadir hidrógenos. Los comandos específicos son:

* Eliminación de solventes (agua):

remove solvent

* Adición de todos los hidrógenos (polares y apolares), crucial para detectar enlaces no canónicos:

h\_add

Estas acciones automatizadas dejan preparada la estructura para mediciones más específicas.

**Mediciones clave:**

La identificación de hélices PPII requiere tres tipos principales de mediciones estructurales:

* Ángulos diedros (φ, ψ): Utilizan la herramienta gráfica integrada en PyMOL mediante la opción *Wizard → Measurement → Dihedrals*, seleccionando visualmente los cuatro átomos consecutivos implicados en cada ángulo diedro. Esto permite identificar claramente la conformación característica de las hélices PPII.
* Distancias entre carbonos alfa y oxígenos carbonílicos: Para localizar posibles enlaces de hidrógeno no canónicos se emplea el comando:

distance dist1o2, (h1\_calpha), (h2\_o), 5

Este comando mide automáticamente todas las distancias potenciales dentro de un radio definido (habitualmente 5 Å).

* Medición de ángulos específicos para enlaces de hidrógeno no canónicos: Estos se determinan con la opción Wizard → Measurement → Angles, permitiendo verificar si la geometría entre donador (Cα-H) y aceptor (O=C) se ajusta a los criterios descritos por Segura Rodríguez y Laurents (2024).
* Selección y manipulación:

Para aislar y analizar regiones específicas de la estructura proteica, se utilizan diferentes métodos de selección:

* Para trabajar con una cadena concreta, se activa el selector en modo *Chains* y se genera un nuevo objeto mediante la opción *Action → copy to object → new*.
* Para seleccionar residuos concretos, se emplea el modo *Residues* y se genera el objeto correspondiente con *(sele) → Action → copy to object → new*.

Estas operaciones permiten centrarse en regiones relevantes sin interferencias del resto de la estructura.

**Automatización y scripting:**

Para agilizar el análisis y garantizar su reproducibilidad, pueden emplearse scripts en Python integrados en PyMOL, ejecutables desde su propia interfaz. Un ejemplo típico de preparación automática incluiría los siguientes comandos:

cmd.fetch('3bog')

cmd.remove('solvent')

cmd.h\_add()

cmd.show('cartoon')

Este enfoque facilitaría la ejecución sistemática de tareas recurrentes, optimizando el análisis estructural de proteínas y favoreciendo la identificación eficiente de hélices PPII. No obstante, en la práctica actual, esta preparación no se realiza mediante comandos directos, sino que se encuentra encapsulada dentro de funciones del módulo con interfaz gráfica.

### Procedimiento manual utilizado actualmente en el laboratorio

Resumiendo, y sintetizando, para la detección manual de hélices de poliprolina II (PPII) en PyMOL se siguen los siguientes pasos clave:

1. **Carga de la estructura**: apertura del archivo PDB correspondiente a la proteína de interés.
2. **Preparación inicial**: eliminación de moléculas de agua y adición de hidrógenos para permitir análisis estructural preciso.
3. **Identificación visual preliminar**: localización de regiones con geometría triangular típica de hélices PPII mediante inspección visual.
4. **Selección manual**: aislamiento de cadenas individuales y selección de los residuos candidatos mediante la interfaz gráfica.
5. **Medición de ángulos φ y ψ**: cálculo manual de los ángulos diedros utilizando la herramienta *Wizard → Dihedrals*.
6. **Detección de enlaces no canónicos**: análisis manual de distancias y ángulos entre carbonos alfa e hidrógenos, y oxígenos de carbonilos de hélices vecinas.
7. **Anotación y organización visual**: asignación de colores, etiquetas y creación de objetos independientes para facilitar el análisis.

Estos pasos conforman el procedimiento base que se ha utilizado como referencia para el desarrollo del módulo automatizado descrito en el capítulo 5, el cual implementa esta misma lógica de forma sistemática y replicable.

### Ejemplo práctico: uso manual de PyMOL sobre la proteína modelo 3BOG

A continuación, se resume el procedimiento aplicado a la proteína modelo 3BOG, utilizando PyMOL de forma manual para la identificación de hélices PPII:

* Carga del archivo PDB: se utiliza el identificador 3BOG para cargar la estructura en PyMOL (ver Figura 3).
* Limpieza inicial: eliminación de moléculas de agua y adición de hidrógenos para preparar la estructura (ver Figuras 5 y 6).
* Identificación visual preliminar: se localizan regiones con geometría triangular compatible con hélices PPII (ver Figura 9).
* Medición de ángulos diedros: se determinan los valores φ y ψ en los residuos seleccionados visualmente (ver Figura 23).
* Detección de enlaces no canónicos: se realiza un análisis de distancias y ángulos entre carbonos alfa y oxígenos de hélices colindantes (ver Figuras 26 y 28).

# Objetivos y metodología de trabajo

## Objetivo general

El objetivo principal de este trabajo es desarrollar un módulo en Python, integrado en PyMOL, que permita automatizar la detección de hélices de PPII en estructuras de proteínas a partir de archivos en formato PDB. Esta herramienta busca facilitar el análisis estructural de proteínas que contienen este tipo de hélices, especialmente en el contexto de investigaciones biomoleculares vinculadas a estudios científicos. La automatización de este proceso permitirá ahorrar tiempo, reducir errores humanos y estandarizar la identificación de esta estructura secundaria, lo que refuerza su utilidad como apoyo directo a la labor de especialistas en el estudio de proteínas y estructuras moleculares.

## Objetivos específicos

Para alcanzar el objetivo general, se han definido los siguientes objetivos específicos:

* Diseñar un algoritmo de detección capaz de identificar las hélices PPII mediante la evaluación de parámetros estructurales característicos, como los ángulos diédricos φ (phi) y ψ (psi), así como la ausencia de enlaces de hidrógeno típicos de hélices alfa o estructuras beta.
* Implementar el algoritmo como un módulo en Python, capaz de ejecutarse en PyMOL mediante una interfaz gráfica integrada en PyMOL, sencilla e intuitiva, que integre de forma eficiente el análisis y la visualización de hélices PPII sobre la representación molecular en curso.
* Validar la herramienta con estructuras proteicas reales, incluyendo proteínas previamente estudiadas de forma manual por el grupo de investigación, para comprobar su eficacia, precisión y fiabilidad.
* Documentar el uso del módulo, mediante una guía técnica, accesible para investigadores y usuarios de PyMOL, que facilite su adopción en otros entornos de investigación relacionados con la bioinformática estructural.

## Metodología de trabajo

El desarrollo del módulo se llevó a cabo siguiendo un enfoque iterativo, combinando fases de análisis, diseño, implementación, pruebas y validación. Esta metodología permitió adaptar progresivamente el sistema a las necesidades reales del laboratorio.

Como se describe en el apartado 2.2.2, se seleccionó PyMOL como la herramienta idónea para el desarrollo del módulo, por su capacidad de análisis estructural y su entorno programable mediante scripts en Python.

Las fases metodológicas seguidas fueron las siguientes:

1. **Análisis del problema**: Se revisaron los procedimientos manuales empleados en el laboratorio para identificar hélices PPII, así como los criterios propuestos en la publicación de referencia (Segura Rodríguez & Laurents, 2024), identificando sus limitaciones y extrayendo los criterios estructurales clave, como los ángulos diedros φ y ψ, la longitud mínima de los tramos y la disposición espacial característica de las hélices PPII.
2. **Diseño del sistema**: Partiendo del análisis previo, se definió una arquitectura modular compuesta por bloques funcionales independientes. Cada bloque replicó una tarea concreta del análisis estructural (cálculo de ángulos, filtrado, detección de enlaces), replicando el flujo de trabajo manual de forma programable. Se establecieron las condiciones geométricas necesarias para la detección automática de hélices PPII, incluyendo los umbrales para los ángulos φ y ψ. También se incorporó la evaluación de enlaces de hidrógeno no canónicos, utilizando las capacidades de medición ofrecidas por PyMOL.
3. **Implementación del módulo en Python**: El código se desarrolló en el entorno de programación Anaconda, empleando VS Code como editor. Se diseñó como script ejecutable desde PyMOL, utilizando librerías como tkinter, para la interfaz gráfica, os, para la gestión de archivos y pymol.cmd como API de interacción estructural. Su diseño modular facilita la modificación y extensión del sistema. El diseño modular evita dependencias externas, garantizando la compatibilidad con PyMOL.
4. **Integración y pruebas funcionales**: El módulo se integró en PyMOL como script ejecutable y se probó con distintas estructuras modelo para verificar el funcionamiento del módulo y ajustar sus parámetros. Estas pruebas permitieron identificar errores, refinar el comportamiento del sistema, ajustar los umbrales de detección y asegurar la estabilidad del sistema.
5. **Documentación y validación experimental**: Se elaboró una guía de uso con instrucciones, ejemplos y recomendaciones de uso. El módulo fue validado utilizando estructuras reales seleccionadas para comprobar su fiabilidad y su capacidad de reproducir los resultados manuales. Los detalles completos del proceso de validación y los resultados obtenidos se describen en los capítulos 5 y 6.

Este enfoque, que integra conocimientos de bioinformática estructural, desarrollo de software y análisis científico, permitió construir una herramienta técnica robusta, flexible y alineada con las necesidades reales del entorno de investigación estructural, aportando una solución reproducible, funcional y útil para entornos de investigación.

Tabla 3. Índice de moléculas de referencia

| **Proteína** | **Código PDB** | **Tipo (IP/PD)** | **Características estructurales** | **Referencia** |
| --- | --- | --- | --- | --- |
| Hypogastrura harveyi antifreeze protein (HhAFP), Fig. 2 | 3BOG | IP | Seis hélices PPII dispuestas en dos capas | Pentelute et al., 2008 |
| Granisotoma rainieri antifreeze protein (GrAFP), Fig. 3 | 7JJV | IP | Nueve hélices PPII dispuestas en dos capas | Scholl et al., 2021 |
| Bacillus subtilis Obg GTP-binding protein (BsObg), Fig. 4 | 1LNZ | PD | Seis hélices PPII dispuestas en dos capas | Buglino et al., 2002 |
| Thermus thermophilus Obg GTP-binding protein (TtObg) | 1UDX | PD | Seis hélices PPII dispuestas en dos capas | Kukimoto-Niino et al., 2004 |
| Escherichia coli Obg GTP-binding protein (EcObg) | 5M0Y | PD | Seis hélices PPII dispuestas en dos capas | Gkekas et al., 2017 |
| Aromatoleum aromaticum acetophenone carboxylase (AaApc), Fig. 5 | 5L9W | PD | Siete hélices PPII dispuestas en tres capas | Warkentin et al., 2017 |
| Xanthobacter autotrophicus acetone carboxylase (XaAtc) | 5SVC | PD | Siete hélices PPII dispuestas en tres capas | Mus et al., 2017 |
| Salmonella φS16 long tail fiber (S16LTF), Fig. 6 | 6F45 | PD | Diez hélices PPII dispuestas en tres capas | Dunne et al., 2018 |
| Homo sapiens anaplastic leukemia kinase (HsALK), Fig. 7 | 7N00 | PD | Catorce hélices PPII dispuestas en cuatro capas | Reshetnyak et al., 2021 |

1. Nota. IP = Independent Protein; PD = Protein Domain. Adaptado de Segura Rodríguez y Laurents (2024).

# Contribución y colaboración

El desarrollo del módulo de detección de hélices PPII ha sido llevado a cabo íntegramente por la autora de este trabajo, desde la concepción inicial hasta la implementación final, incluyendo el diseño de la interfaz, la lógica del algoritmo, la integración con PyMOL y la documentación técnica del sistema. La contribución se ha centrado en transformar en código automatizado el procedimiento manual desarrollado previamente por el grupo de investigación, con el fin de facilitar su aplicación, aumentar su reproducibilidad y adaptarlo a entornos computacionales.

Para asegurar su correcto funcionamiento, se ha estructurado el código en bloques funcionales independientes, lo que ha facilitado su validación por partes, la depuración progresiva y la posibilidad de reutilizar componentes en futuras versiones. Asimismo, se ha optado por una interfaz ligera implementada en tkinter, priorizando la compatibilidad con entornos científicos sin necesidad de dependencias externas complejas.

Durante el desarrollo se tomaron decisiones técnicas clave, como reemplazar funciones internas de PyMOL por cálculos geométricos directos mediante el módulo math, aplicar un filtrado previo de residuos para reducir la carga computacional, y estructurar la salida del sistema en formatos interoperables (.csv y .pdb), lo que permite el uso posterior de los datos en otras herramientas bioinformáticas.

Las decisiones relacionadas con la lógica del algoritmo, la segmentación del código, los criterios geométricos de selección, la validación de los ángulos y distancias, y la construcción del sistema de interacción con el usuario han sido desarrolladas íntegramente por la autora, aplicando conocimientos adquiridos en el Grado en Ingeniería Informática. También se ha elaborado la documentación técnica y de uso para facilitar la reutilización del módulo por parte de otros investigadores.

Este trabajo se ha realizado en colaboración con el Grupo de Estructura, Dinámica e Interacciones de Proteínas por RMN del Instituto de Química-Física “Blas Cabrera” (CSIC), que ha proporcionado la base teórica, el procedimiento manual original y las estructuras empleadas en la validación del módulo. En concreto, el doctorando Cristian Segura Rodríguez y el Dr. Douglas V. Laurents han participado en la definición de los criterios geométricos, en la documentación del procedimiento manual y en la validación de los resultados obtenidos mediante el módulo automatizado. Esta colaboración ha sido clave para asegurar que el desarrollo respondiera a necesidades reales de investigación y se adaptara a los protocolos vigentes en el laboratorio.

## Licencia y distribución del software

El módulo desarrollado para la detección automática de hélices PPII ha sido publicado en el repositorio público GitHub, en la dirección: <https://github.com/silviaenma/PPIIMoL>, bajo los términos de la Licencia Pública General GNU, versión 3.0 (GPLv3).

Esta licencia garantiza que el software pueda ser utilizado, estudiado, modificado y redistribuido libremente, siempre que las versiones derivadas conserven la misma licencia.

De este modo, se asegura que tanto esta herramienta como sus futuras adaptaciones permanezcan abiertas y accesibles para toda la comunidad.

El uso de una licencia copyleft como la GPLv3 refuerza el compromiso con la ciencia abierta y fomenta la colaboración en entornos académicos y científicos, permitiendo que otros investigadores reutilicen y mejoren el código en beneficio de proyectos relacionados.  
El texto completo de la licencia puede consultarse en: <https://www.gnu.org/licenses/gplv3-the-program.es.html>

## Publicación en repositorios abiertos

Como parte del compromiso con el software libre y la ciencia abierta, el módulo desarrollado ha sido publicado en un repositorio público de GitHub, accesible en la siguiente dirección:

https://github.com/silviaenma/PPIIMoL

Esta decisión responde a la voluntad de facilitar su reutilización, adaptación y mejora por parte de otros grupos de investigación, especialmente aquellos que trabajan con proteínas desordenadas o con estructuras ricas en glicina y prolina. El repositorio incluye el código fuente completo, ejemplos de uso, documentación en castellano e inglés, e información sobre la licencia aplicada (GPLv3).

La publicación abierta del módulo refuerza el carácter colaborativo del trabajo y permite que su impacto trascienda el marco del presente proyecto académico.

# Desarrollo del módulo

En línea con los objetivos específicos expuestos en el capítulo anterior, este capítulo detalla el desarrollo técnico de la herramienta, desde el diseño del algoritmo hasta su implementación y validación práctica en el entorno PyMOL (versión 3.1.4.1), que ha servido como base estable para las pruebas realizadas. Se detallan las decisiones de diseño, las funcionalidades implementadas y el proceso de automatización llevado a cabo.

## Requisitos funcionales y técnicos del sistema

El desarrollo del módulo parte de una serie de requisitos definidos desde dos vertientes: funcional (capacidades del sistema) y técnica (integración y entorno de ejecución).

### Requisitos funcionales.

El sistema debe ser capaz de:

* Integrarse en el programa PyMOL y desde este entorno, ejecutar sus funciones.
* Cargar estructuras proteicas en formato PDB, tanto desde archivos locales como directamente desde el repositorio Protein Data Bank.
* Preparar el archivo para su caracterización.
* Analizar las cadenas presentes e identificar regiones cuyos residuos presenten ángulos diédricos φ y ψ compatibles con la conformación PPII.
* Permitir ajustar el margen de tolerancia de los ángulos φ y ψ para adaptarse a la variabilidad de las moléculas reales.
* Detectar posibles enlaces de hidrógeno no canónicos entre hélices adyacentes.
* Segregar visualmente en PyMOL las regiones detectadas, marcándolas específicamente y generando nuevos objetos identificables.
* Generar un archivo de texto que liste las posibles hélices PPII detectadas, incluyendo su posición en la secuencia y sus características principales.

### Requisitos técnicos.

Desde el punto de vista técnico, el módulo debe cumplir con los siguientes criterios:

* Estar desarrollado en Python 3, utilizando el sistema de scripting interno de PyMOL y su API cmd.
* Ser compatible con sistemas operativos Windows, macOS y Linux, en consonancia con la portabilidad de PyMOL.
* Integrarse sin complicaciones en los flujos de trabajo habituales del laboratorio, sin requerir herramientas externas ni dependencias complejas.
* Poder ejecutarse directamente desde la interfaz de PyMOL mediante el comando run o cargando un archivo .py.
* Estar estructurado en funciones modulares documentadas y organizadas, facilitando su mantenimiento, ampliación y reutilización por parte de otros usuarios.

En función de estos requisitos, se ha diseñado una solución modular, dividida en bloques funcionales independientes, como el filtrado de residuos, el análisis geométrico, la detección de enlaces y la visualización. Cada uno de estos bloques puede ejecutarse de forma individual o conjunta, lo que facilita tanto su depuración como su futura ampliación sin comprometer la estabilidad ni la usabilidad general del sistema.

## Diseño del algoritmo de detección PPII

El algoritmo implementado automatiza los pasos del procedimiento manual descrito en el apartado 2.1.2 e incorpora criterios geométricos definidos en el artículo base. Estos criterios, extraídos del trabajo de Segura Rodríguez y Laurents (2024), establecen los valores típicos de los ángulos diedros y la distribución espacial de residuos en dominios ricos en hélices PPII, lo que ha servido como guía para definir los rangos de selección y el enfoque estructural del análisis.

El objetivo ha sido trasladar a código una secuencia de análisis que tradicionalmente dependía de la inspección visual, la medición manual de ángulos diedros y la evaluación subjetiva de posibles enlaces estructurales.

La lógica del sistema se ha estructurado en bloques funcionales independientes, que actúan de forma secuencial sobre la estructura proteica cargada en PyMOL. Cada bloque refleja un paso equivalente del procedimiento manual, lo que garantiza coherencia metodológica y facilita su validación por parte del equipo investigador.

Los pasos principales que componen el diseño del algoritmo son:

1. Carga del archivo PDB y visualización inicial: el usuario selecciona un archivo .pdb, ya sea desde el propio equipo o descargándola directamente desde la base de datos pública **Protein Data Bank,** mediante su identificador. Una vez cargada, la estructura se importa en PyMOL, donde se representa en modo licorice para posibilitar la visualización de los átomos del péptido.
2. Preparación de la estructura: se eliminan las moléculas de solvente y se añaden hidrógenos a todos los residuos mediante funciones nativas de PyMOL, dejando el modelo listo para su análisis estructural.
3. Selección y segmentación de cadenas: el módulo identifica automáticamente las cadenas presentes en la estructura y las separa en objetos independientes, lo que permite analizarlas de forma individual en los siguientes pasos del proceso.
4. Cálculo y exportación de ángulos φ y ψ: el sistema calcula automáticamente dichos ángulos para cada residuo mediante la función *get\_phipsi()* de PyMOL. Los resultados se almacenan ordenadamente junto con el identificador del residuo y su tipo *(resn)*, y se exportan a un archivo *.txt* para su posterior análisis. Alternativamente, también pueden mostrarse directamente en pantalla mediante una ventana emergente con *scroll*.
5. Filtrado de residuos candidatos: se consideran como candidatos PPII aquellos residuos cuyos ángulos φ y ψ se encuentren dentro de un rango tolerado en torno a φ ≈ –75° y ψ ≈ 145°. A partir de estos residuos compatibles, el algoritmo identifica tramos consecutivos de al menos tres posiciones que cumplan dicho criterio, considerándolos como posibles hélices de tipo PPII y facilitando así su selección en etapas posteriores.
6. Creación de objetos independientes por hélice PPII detectada: cada hélice identificada se separa como un nuevo objeto independiente en PyMOL, lo que permite aplicar operaciones específicas sobre ella —como coloreado, aislamiento o medición— sin afectar al resto del modelo.
7. Selección de carbonos α: para cada hélice identificada, se utiliza el comando name CA (select c\_alphas, name CA) para seleccionar automáticamente los átomos de carbono alfa, que actuarán como donantes de enlace en el análisis posterior.
8. Identificación de oxígenos carbonílicos: para analizar posibles enlaces de hidrógeno no canónicos, es necesario localizar los átomos de oxígeno del grupo carbonilo, que actuarán como aceptores potenciales. Los átomos de oxígeno pueden seleccionarse mediante la expresión name O, y luego limitar la búsqueda a los oxígenos relevantes filtrando solo los que pertenecen a la cadena principal de la proteína, mediante la condición backbone. También es posible restringir el análisis solo a los residuos que forman parte de las hélices PPII detectadas.
9. Análisis de enlaces de hidrógeno no canónicos: una vez seleccionados los carbonos alfa (donantes) y los oxígenos carbonílicos (aceptores), el sistema calcula las distancias entre cada par de átomos mediante el comando distance, aplicando un umbral configurable (habitualmente 5 Å). Alternativamente, para optimizar el cribado previo, pueden emplearse comandos como select within o neighbor, que permiten identificar directamente los átomos cercanos sin tener que comparar todas las combinaciones posibles.
10. **Medición del ángulo de enlace (Cα–H–O):** para cada par de átomos que cumple el criterio de distancia, se mide el ángulo formado por el carbono alfa, el hidrógeno enlazado y el oxígeno carbonílico. Esta medida permite distinguir enlaces reales frente a simples proximidades espaciales sin relevancia estructural. Solo se consideran válidos aquellos pares cuyo ángulo se sitúe dentro de un rango geométrico compatible con un enlace de hidrógeno no canónico, habitualmente entre 110° y 180°, aunque este umbral angular puede ajustarse en el código para adaptarse a proteínas atípicas o contextos experimentales concretos. Además, el módulo registra automáticamente estos ángulos en un archivo de texto, lo que facilita su análisis posterior y refuerza la validación estructural del modelo.
11. Visualización y coloreado de hélices PPII: una vez identificadas las hélices PPII y, en su caso, validados sus posibles enlaces, cada una se representa como un objeto independiente en PyMOL, y se le asigna un color distintivo. Este coloreado, que puede realizarse de forma automática desde el script utilizando una paleta predefinida o aleatoria, facilita la interpretación visual y mejora la presentación en pantalla, diferenciando cada hélice.
12. Generación de tabla resumen: el sistema genera finalmente un archivo de salida en formato .txt que recoge, para cada hélice detectada, el rango de residuos que la componen, los valores φ y ψ correspondientes a cada uno, las distancias medidas entre carbonos alfa y oxígenos carbonílicos, y los ángulos Cα–H–O obtenidos durante el análisis. Esta tabla constituye un documento estructurado que puede utilizarse para análisis posteriores, comparación entre estructuras o inclusión directa en informes científicos.
13. Como parte del desarrollo del módulo, se ha incorporado una funcionalidad de exportación que permite guardar las hélices PPII detectadas en formatos estándar como .pdb y .csv. Esto no solo permite visualizar los resultados, sino que habilita su uso directo en herramientas bioinformáticas avanzadas. Por ejemplo, **Biopython** permite automatizar comparaciones estructurales sobre las hélices detectadas, **MDTraj** puede analizar dinámicas conformacionales a partir de los fragmentos exportados, y **PyRosetta** permite evaluar su estabilidad energética o simular mutaciones. Esta interoperabilidad amplía el alcance del módulo más allá de PyMOL e integra sus resultados en flujos de análisis profesionales.

El sistema permite guardar individualmente las hélices como archivos .pdb, conservando su contexto estructural, y generar archivos .csv que recogen los residuos implicados, los ángulos φ y ψ de cada uno y, en su caso, los enlaces de hidrógeno no canónicos detectados. Esta capacidad permite su integración en análisis estructurales más amplios, tanto individuales como por lotes. Estos archivos pueden integrarse fácilmente en flujos de trabajo científicos, bases de datos estructurales o materiales complementarios de publicaciones académicas, lo que amplía la visibilidad y reutilización de los resultados generados.

El diseño modular del sistema permite ejecutar cada bloque de forma independiente y ajustar los parámetros de análisis según el contexto, lo que facilita su depuración, mantenimiento y adaptación a otros tipos de estudios estructurales. El módulo permite realizar una caracterización precisa y rápida de proteínas candidatas con conformación PPII potencial, reproduciendo fielmente el procedimiento manual empleado en el laboratorio y transformándolo en una herramienta automatizada, reproducible y fácilmente integrable en flujos de trabajo científicos reales.

**Optimización del cálculo de distancias**

Durante el desarrollo del módulo, se identificó que la función de cálculo de distancias basada en los métodos internos de PyMOL resultaba ineficiente para estructuras grandes o para análisis repetitivos. En ciertos casos, el sistema se ralentizaba notablemente o incluso se bloqueaba temporalmente durante el procesamiento.

Para evitar este cuello de botella, se optó por implementar manualmente el cálculo de distancias entre átomos utilizando operaciones matemáticas básicas del módulo math de Python. Esta decisión, aunque a priori más rudimentaria, resultó ser significativamente más eficiente en la práctica, reduciendo el tiempo de ejecución y mejorando la estabilidad del sistema. Gracias a esta modificación, el análisis de posibles enlaces de hidrógeno no canónicos y otras comprobaciones geométricas se realiza ahora de forma más fluida, sin comprometer la precisión de los resultados.

## Publicación y distribución del software

El módulo desarrollado se publica bajo la Licencia Pública General GNU, versión 3 (GPLv3), lo que permite su uso, distribución y modificación libre, conforme a los términos de dicha licencia.

El código fuente completo, junto con la documentación de instalación y uso, está disponible en el siguiente repositorio de GitHub: <https://github.com/silviaenma/PPIIMoL>. Las condiciones de la licencia pueden consultarse en: https://www.gnu.org/licenses/gpl-3.0.html.

## Estructura del código y funciones principales

El módulo se ha desarrollado en Python utilizando el scripting interno de PyMOL y la biblioteca gráfica *Tkinter*, lo que permite su integración directa en sesiones de trabajo habituales sin necesidad de herramientas externas adicionales. La estructura del código se ha organizado en funciones independientes, cada una de ellas asociada a una operación concreta dentro del proceso de detección y análisis de hélices PPII.

El script se carga como un archivo .py y activa una interfaz gráfica interactiva dentro de PyMOL, con botones que activan las distintas funcionalidades implementadas. A continuación, se describen las funciones principales del módulo:

* seleccionar\_archivo(): permite al usuario cargar un archivo .pdb mediante un cuadro de diálogo. La estructura seleccionada se importa a PyMOL bajo el nombre *proteina* y se representa en estilo *licorice* para facilitar la visualización de átomos individuales.
* descargar\_molecula(): permite al usuario descargar directamente una estructura desde el Protein Data Bank (PDB) mediante su identificador. La molécula descargada se carga en la sesión de PyMOL con estilo licorice.
* eliminar\_solventes()*:* elimina las moléculas de solvente presentes en el archivo PDB, como el agua, que podrían interferir en el análisis estructural.
* anadir\_hidrogenos(): ejecuta la adición automática de átomos de hidrógeno a toda la proteína, lo cual es un paso necesario para poder evaluar posteriormente los posibles enlaces de hidrógeno no canónicos.
* ocultar\_side\_chains()*:* oculta las cadenas laterales de los aminoácidos, dejando visible solo el esqueleto principal de la proteína (*backbone*), lo que facilita la evaluación de la geometría y los ángulos diedros.
* separar\_cadenas(): recorre la proteína cargada e identifica las distintas cadenas presentes, creando un nuevo objeto independiente por cada una de ellas. Esto permite trabajar de forma aislada sobre cada subunidad estructural.
* obtener\_angulos\_phi\_psi\_por\_cadena(): calcula internamente los ángulos diedros φ y ψ de cada residuo utilizando el nuevo método get\_phipsi() de PyMOL y los agrupa por cadena.
* guardar\_angulos\_phi\_psi(): calcula los ángulos diedros φ y ψ para cada residuo de cada cadena, utilizando el método *get\_phipsi()* de PyMOL, y guarda los resultados en un archivo de texto.
* detectar\_segmentos\_ppii(): identifica segmentos consecutivos de la proteína cuyos ángulos diedros se ajustan al patrón geométrico característico de hélices PPII. Los segmentos se resaltan visualmente y se guardan como objetos en PyMOL.
* localizar\_atomicos\_clave\_segmentos(): obtiene los átomos clave (Cα, oxígeno carbonílico y átomos de hidrógeno próximos) para cada residuo de los segmentos detectados, generando pseudoátomos en PyMOL (representaciones visuales sin base química real) que permiten el análisis geométrico de posiciones clave.
* calcular\_distancias\_colindantes(): calcula las distancias entre átomos Cα y O de segmentos consecutivos. Si se encuentran por debajo de un umbral predefinido, se consideran candidatos a formar enlaces de hidrógeno.
* calcular\_angulos\_ca\_h\_o(): a partir de las tripletas formadas por Cα, H y O, calcula el ángulo geométrico entre ellos y filtra los valores que se encuentran dentro del rango característico de enlaces H no canónicos (110°–180°). Los resultados se almacenan en un archivo de texto.
* calcular\_y\_guardar\_angulos(): función de alto nivel que automatiza el proceso de detección de segmentos, obtención de átomos clave, cálculo de distancias y registro de ángulos.
* calcular\_y\_visualizar\_distancias(): combina la obtención de tripletas espaciales con su visualización directa en PyMOL mediante líneas punteadas de color cian, lo que permite una evaluación inmediata de la proximidad entre segmentos.
* visualizar\_distancias\_pares(): dibuja en PyMOL las distancias entre pares de átomos seleccionados, representándolos mediante líneas codificadas por color y grosor.
* generar\_reporte\_csv(): exporta un archivo estructurado en formato .csv que empareja átomos Cα–O y sus correspondientes distancias y ángulos, permitiendo un análisis sistemático de las posibles interacciones no canónicas.
* guardar\_segmentos\_ppii\_pdb(): exporta cada segmento identificado como archivo .pdb individual, permitiendo su análisis posterior de forma externa.
* convertir\_a\_selecciones\_pymol(): función auxiliar que convierte los pares de átomos detectados en selecciones compatibles con PyMOL, facilitando posibles ampliaciones o personalizaciones.
* obtener\_carpeta\_resultados(): gestiona la organización de los archivos de salida, generando automáticamente una carpeta llamada Resultados\_PPIIMoL dentro del directorio de documentos del usuario.
* lanzar\_interfaz(): inicializa la interfaz gráfica, creando una ventana principal con botones para ejecutar cada una de las funciones anteriores. Esta interfaz mejora la usabilidad del módulo, especialmente para usuarios sin experiencia en scripting.

El diseño modular del script facilita tanto su uso como su mantenimiento y evolución. Cada función puede ejecutarse de manera independiente, lo que permite modificar o ampliar su funcionalidad sin afectar al resto del sistema. Además, se ha mantenido un enfoque centrado en la claridad y la reutilización del código, siguiendo buenas prácticas de programación en bioinformática estructural.

La Figura 30 muestra la interfaz gráfica completa del módulo, compuesta por un menú adicional integrado en la interfaz de PyMOL mediante la biblioteca *Tkinter*. A través de él, el usuario puede seleccionar archivos PDB, ejecutar análisis geométricos, detectar segmentos PPII, ajustar umbrales de ángulo y distancia, y exportar los resultados generados, todo ello sin necesidad de conocimientos avanzados de programación.

Figura 30. Interfaz gráfica del módulo desarrollada en Python con Tkinter e integrada en PyMOL



## Proceso de integración con PyMOL

Una de las principales ventajas de PyMOL es su integración nativa con scripts en Python, ejecutables directamente desde su entorno gráfico, lo que permite extender su funcionalidad sin necesidad de modificar el código fuente del programa. En este proyecto, ha aprovechado esta característica para integrar el módulo de detección de hélices PPII como un complemento ejecutable desde el propio visor molecular, sin alterar la experiencia de usuario habitual.

El proceso de integración se ha diseñado para que sea lo más simple y transparente posible. El usuario únicamente debe cargar el archivo *PPIIMoL.py* del módulo desde PyMOL mediante el comando *run*, que activa automáticamente la interfaz gráfica del sistema. Desde esa ventana, es posible acceder a todas las funciones del módulo a través de botones interactivos, sin necesidad de introducir comandos manuales.

La interfaz se ha construido utilizando *Tkinter*, una biblioteca estándar de Python que permite generar ventanas, botones y cuadros de diálogo. Esta elección garantiza la compatibilidad con sistemas operativos como Windows, macOS y Linux, siempre que se utilice una instalación estándar de Python que incluya soporte para Tkinter. De este modo, se facilita su distribución entre usuarios con diferentes configuraciones sin necesidad de instalar dependencias adicionales.

Además, el módulo aprovecha las funciones internas del objeto *cmd* de PyMOL, que permite manipular objetos moleculares, realizar mediciones, seleccionar átomos, modificar estilos de representación y extraer propiedades estructurales como los ángulos diedros. Gracias a esta integración nativa, el sistema interactúa directamente con la estructura cargada en pantalla, mostrando los resultados del análisis en tiempo real sobre el modelo tridimensional.

Otra ventaja de esta arquitectura es que el módulo puede ejecutarse tanto en modo interactivo como desde la línea de comandos, lo que permite su uso en análisis por lotes o ejecución sobre múltiples estructuras automáticamente mediante scripts externos o bucles de ejecución.

La versión final del módulo incluye nuevas funciones que amplían notablemente sus capacidades, manteniendo la integración nativa con PyMOL. Entre ellas se encuentran el cálculo y visualización de distancias entre átomos colindantes, la estimación de ángulos CA–H···O y la exportación estructurada de resultados en formatos .csv, .txt y .pdb. Todos estos procesos pueden activarse desde nuevos botones añadidos a la interfaz gráfica, que permiten ejecutar análisis geométricos avanzados sin necesidad de escribir una sola línea de código. Los resultados se representan en la ventana de PyMOL mediante pseudoátomos y líneas de distancia, y se almacenan automáticamente en una carpeta organizada dentro del sistema del usuario, lo que contribuye a una trazabilidad clara y ordenada del análisis realizado.

En definitiva, la integración con PyMOL no solo hace que la herramienta sea accesible y funcional para usuarios sin conocimientos avanzados de programación, sino que también permite su uso en flujos de trabajo reales dentro del laboratorio, cumpliendo así uno de los objetivos principales de este trabajo.

## Validación sobre la proteína modelo 3BOG

Para comprobar la funcionalidad del módulo desarrollado, se ha llevado a cabo una verificación de resultados con la proteína modelo 3BOG, ampliamente estudiada en el artículo base del proyecto (Segura Rodríguez & Laurents, 2024) y utilizada previamente en el laboratorio como referencia para el análisis manual de hélices de tipo PPII.

Esta proteína presenta un dominio estructural formado por seis hélices PPII unidas mediante bucles, cuya geometría y secuencia han sido caracterizadas con detalle en investigaciones anteriores. Por tanto, constituye un caso ideal para verificar si el algoritmo implementado es capaz de reproducir los resultados esperados y detectar correctamente los segmentos conformacionales definidos como PPII.

El procedimiento de validación ha seguido los siguientes pasos:

* Se ha cargado el archivo PDB de la proteína 3BOG en PyMOL y se ha ejecutado el módulo mediante la interfaz desarrollada.
* Se han aplicado las funciones de filtrado, cálculo de ángulos φ y ψ, y selección de residuos compatibles con la conformación PPII.
* Se han detectado automáticamente los seis segmentos correspondientes a las hélices conocidas, con una coincidencia completa respecto a la localización identificada previamente mediante inspección manual.
* Adicionalmente, se han evaluado posibles enlaces de hidrógeno no canónicos entre cadenas colindantes, identificando patrones espaciales coherentes con la disposición geométrica descrita en el artículo base.
* Los resultados han sido visualizados directamente en el modelo molecular mediante colores y etiquetas que distinguen cada hélice, lo que ha permitido una validación visual rápida y precisa.

Como resultado de este proceso, se ha confirmado que el módulo reproduce de forma automatizada las tareas anteriormente realizadas de manera manual, con una precisión equivalente y una considerable reducción del tiempo requerido. El sistema no solo detecta correctamente las hélices ya identificadas manualmente, mostrando una coincidencia total en los tramos reconocidos, sino que proporciona una salida clara, exportable y reutilizable para estudios posteriores.

Esta validación constituye una prueba de concepto sólida que respalda el uso del módulo en futuras investigaciones. Tras la verificación inicial sobre 3BOG, se ha ampliado la evaluación a otras proteínas con dominios desordenados o regiones enriquecidas en glicina y prolina, incluidas en la Tabla 3, obteniendo resultados coherentes con la conformación PPII descrita en el artículo base.

## Estructura técnica de la solución

Con el objetivo de fomentar la reutilización del módulo y facilitar su adopción por parte de otros grupos de investigación, el código fuente ha sido publicado en un repositorio abierto en GitHub (https://github.com/silviaenma/PPIIMoL/). Este repositorio contiene tanto el archivo principal del script como la documentación técnica necesaria para su uso e integración.

La publicación en formato abierto permite a la comunidad científica examinar, modificar y adaptar el módulo a sus propias necesidades, además de contribuir con mejoras o reportar posibles errores a través de las herramientas colaborativas que ofrece la plataforma. El módulo se distribuye bajo la Licencia Pública General GNU v3.0 (GPLv3), que garantiza que cualquier versión derivada mantenga la misma licencia, protegiendo así la libertad del software y evitando su apropiación privativa. Esta decisión está alineada con los principios de ciencia abierta y desarrollo reproducible, cada vez más valorados en el ámbito de la bioinformática estructural.

Además del código fuente, el repositorio incluye:

• Una guía de uso paso a paso.

• Ejemplos de uso con capturas de pantalla.

• Archivos de prueba (incluyendo la estructura 3BOG).

• Instrucciones para contribuir con sugerencias o nuevas funcionalidades.

Además de su publicación en GitHub, el módulo se ha enviado para su inclusión en el repositorio oficial de scripts de PyMOL (PyMOL-script-repo), una plataforma mantenida por la comunidad que recopila extensiones útiles para ampliar las funcionalidades del visor molecular. Esta acción tiene como objetivo facilitar su visibilidad, promover su uso en otros laboratorios y garantizar su disponibilidad dentro del ecosistema habitual de trabajo de los usuarios de PyMOL.

A nivel estructural, el módulo se ha dividido en bloques funcionales claramente diferenciados: gestión de entrada (seleccionar\_archivo(), descargar\_molecula()), limpieza y preparación de la estructura (eliminar\_solventes(), anadir\_hidrogenos(), ocultar\_side\_chains()), análisis estructural (obtener\_angulos\_phi\_psi\_por\_cadena(), detectar\_segmentos\_ppii()), visualización (visualizar\_distancias\_pares(), localizar\_atomicos\_clave\_segmentos()), y exportación de resultados (guardar\_angulos\_phi\_psi(), guardar\_segmentos\_ppii\_pdb(), generar\_reporte\_csv()). Esta organización modular permite mantener una separación lógica de responsabilidades dentro del código, lo que facilita tanto su depuración como su extensión futura.

Para garantizar una gestión ordenada de los datos generados, todos los archivos de salida —como listas de ángulos, distancias, reportes y estructuras parciales— se almacenan automáticamente en una carpeta llamada Resultados\_PPIIMoL, creada dentro del directorio de documentos del sistema. Este enfoque mejora la trazabilidad del análisis y evita la dispersión de archivos, especialmente en entornos compartidos o colaborativos.

## Casos de uso representativos

A continuación, se describen algunos casos de uso que ilustran la interacción esperada entre el usuario y el módulo desarrollado, así como las funciones principales que este proporciona dentro del entorno PyMOL. Estos casos permiten visualizar cómo el sistema responde ante distintos escenarios reales, desde operaciones básicas hasta ajustes personalizados para investigación avanzada.

### Caso de uso 1: Carga y preparación de la estructura PDB.

* **Actor**: Usuario (investigador)
* **Objetivo**: Importar una proteína en formato PDB y preparar el modelo para su análisis estructural.
* **Pasos**:
  1. Ejecutar PyMOL.
  2. Cargar el módulo desde la interfaz con el comando *run*.
  3. Utilizar la opción *Descargar proteína* para importar el archivo PDB.
  4. Pulsar sucesivamente *Eliminar solvente* y *añadir hidrógenos.* Lo hará automáticamente.
  5. Confirmar visualmente que la estructura se encuentra limpia y lista para análisis.
* **Resultado esperado**: La estructura aparece correctamente cargada en PyMOL, sin solventes, con hidrógenos añadidos y preparada para ser procesada por el módulo de detección.

Este paso inicial permite estandarizar el modelo de entrada y garantizar la fiabilidad del análisis posterior.

### Caso de uso 2: Análisis geométrico de la proteína

* **Actor**: Usuario (investigador)
* **Objetivo**: Calcular los ángulos diedros φ y ψ de todos los residuos de una cadena concreta.
* **Pasos**:
  1. Seleccionar la cadena de interés desde la interfaz del módulo.
  2. Ejecutar la función *Guardar ángulos φ y ψ*.
  3. Revisar el archivo de texto generado con los valores por residuo.
  4. Comparar estos datos con estudios anteriores o usarlos como entrada para análisis externos.
* **Resultado esperado:** Se genera un archivo de texto con los ángulos calculados para cada residuo, lo que permite una evaluación cuantitativa de la conformación de la cadena.

Este análisis permite verificar la validez estructural de la cadena, lo que resulta útil como paso previo a la detección de conformaciones PPII.

### Caso de uso 3: Identificación visual de hélices PPII

* **Actor**: Usuario (investigador)
* **Objetivo**: Detectar y visualizar regiones de tipo PPII en una proteína modelo.
* **Pasos**:
  1. Ejecutar el análisis completo desde la interfaz.
  2. Visualizar las hélices detectadas directamente en la estructura tridimensional.
  3. Revisar la información asociada a cada hélice: residuos implicados, valores angulares, enlaces potenciales.
  4. Exportar el resumen de hélices a un archivo para su documentación o publicación.
* **Resultado esperado**: Las hélices candidatas se representan en PyMOL con colores diferenciados y se genera un informe exportable con los parámetros clave de cada una.

El análisis visual agiliza la validación experimental y favorece la presentación de resultados en publicaciones.

### Caso de uso 4: Análisis comparativo entre cadenas

• Actor: Usuario (investigador)

• Objetivo: Comparar regiones candidatas a hélice PPII entre dos cadenas de la misma proteína.

• Pasos:

1. Separar las cadenas de la proteína mediante la función correspondiente.
2. Ejecutar el análisis individualmente para cada una.
3. Visualizar las hélices detectadas en ambas cadenas con códigos de color distintos.
4. Comparar la posición y geometría de las hélices mediante inspección visual y lectura de resultados.

• Resultado esperado: Se identifican similitudes y diferencias entre las cadenas analizadas, facilitando estudios de simetría estructural o evaluación de variabilidad conformacional.

Esta funcionalidad es útil en estudios evolutivos o de diseño racional de proteínas.

### Caso de uso 5: Personalización del análisis por parámetros

• Actor: Usuario avanzado (investigador)

• Objetivo: Modificar los parámetros de análisis (ángulos, longitud mínima, distancias) para adaptarse a proteínas atípicas o condiciones experimentales específicas.

• Pasos:

1. Acceder a las opciones avanzadas del módulo desde la interfaz.
2. Modificar los umbrales de φ y ψ, la longitud mínima de hélice y la distancia máxima para enlaces.
3. Ejecutar el análisis con los nuevos parámetros.
4. Evaluar si los nuevos resultados permiten detectar regiones no identificadas con la configuración estándar.

• Resultado esperado: El sistema detecta nuevas hélices compatibles con los criterios modificados, demostrando su flexibilidad para su uso en estudios no convencionales.

Tanto los umbrales de los ángulos φ y ψ como el rango geométrico de enlace Cα–H–O son configurables, lo que aporta flexibilidad para adaptarse a diferentes contextos moleculares. Este ajuste permite al investigador adaptar el análisis a proteínas no canónicas o condiciones experimentales específicas.

### Caso de uso 6: Exportación de hélices detectadas para análisis externo

* Objetivo: guardar los resultados del análisis estructural en formatos .pdb y .csv para su reutilización en otras herramientas bioinformáticas.
* Estructura analizada: proteína 3BOG (modelo)
* Pasos realizados:
  1. Ejecución del módulo sobre la proteína cargada en PyMOL.
  2. Detección automática de hélices PPII según criterios geométricos definidos.
  3. Selección de cada hélice como objeto independiente en PyMOL.
  4. Exportación de las hélices detectadas a archivos .pdb individuales con cmd.save().
  5. Generación de una tabla .csv con los residuos implicados, sus ángulos φ y ψ, y los enlaces de hidrógeno no canónicos identificados.
* Archivos generados:
  1. H1\_3bog.pdb, H2\_3bog.pdb, etc. (una hélice por archivo).
  2. helices\_ppii.csv (tabla resumen con parámetros estructurales clave).
* Aplicación posterior:
  1. Análisis estructural con Biopython (Bio.PDB)
  2. Medidas dinámicas con MDTraj (md.load())
  3. Evaluación energética o mutagénesis con PyRosetta (pose\_from\_pdb())
* Resultado esperado: Los archivos exportados permiten reutilizar los resultados del análisis estructural en otros entornos, integrándolos en flujos de trabajo bioinformáticos más amplios.

Esta funcionalidad refuerza la interoperabilidad del sistema y favorece la publicación y reproducibilidad de los resultados.

### Caso de uso 7: Análisis geométrico de enlaces potenciales

* **Actor:** Usuario (investigador)
* **Objetivo:** Analizar la geometría entre átomos implicados en enlaces de hidrógeno no canónicos (Cα–H···O).
* **Pasos:**
  1. Ejecutar el módulo desde la interfaz de PyMOL.
  2. Detectar los segmentos PPII con la función correspondiente.
  3. Calcular automáticamente las distancias entre átomos Cα y oxígenos de segmentos colindantes.
  4. Calcular los ángulos formados por tripletas Cα–H···O.
  5. Revisar los resultados generados y visualizados en PyMOL.
* **Resultado esperado:** Se identifican pares de átomos cuya disposición geométrica es compatible con enlaces de hidrógeno no canónicos. Las distancias y ángulos se visualizan directamente sobre la estructura y se almacenan en archivos .txt para su análisis posterior.  
  Esta funcionalidad permite validar la conformación PPII desde un punto de vista geométrico más preciso.

### Caso de uso 8: Generación automatizada de informes estructurados

* **Actor:** Usuario (investigador)
* **Objetivo:** Obtener un informe estructurado con los datos del análisis en formato .csv para su documentación o reutilización.
* **Pasos:**
  1. Ejecutar el análisis completo del módulo.
  2. Pulsar el botón correspondiente a “Generar informe estructurado”.
  3. Guardar el archivo generado (reporte\_ppii.csv) en la carpeta automática de resultados.
* **Resultado esperado:** Se obtiene un archivo .csv que incluye las distancias Cα–O, los ángulos Cα–H···O y la información de los segmentos PPII detectados. Este informe es compatible con herramientas de análisis externas y facilita la trazabilidad del trabajo científico.

Estos casos de uso ilustran la versatilidad del módulo, que permite tanto un uso básico como un análisis estructural avanzado, adaptándose a diferentes necesidades dentro de la investigación bioinformática.

Tabla 4. Casos de uso representativos del módulo de detección de hélices PPII

| **Nº** | **Caso de uso** | **Objetivo principal** | **Resultado esperado** |
| --- | --- | --- | --- |
| 1 | Carga y preparación de la estructura PDB | Importar una proteína en formato PDB y preparar el modelo para su análisis estructural | La estructura aparece correctamente cargada en PyMOL, sin solventes, con hidrógenos añadidos y lista para ser procesada por el módulo de detección. |
| 2 | Análisis geométrico de la proteína | Calcular los ángulos diedros φ y ψ de todos los residuos de una cadena concreta | Se genera un archivo estructurado con los ángulos calculados para cada residuo, lo que permite una evaluación cuantitativa de la conformación de la cadena. |
| 3 | Identificación visual de hélices PPII | Detectar y visualizar regiones de tipo PPII en una proteína modelo | Las hélices candidatas se representan en PyMOL con colores diferenciados, y se genera un informe exportable con los parámetros clave de cada una. |
| 4 | Análisis comparativo entre cadenas | Comparar regiones candidatas a hélice PPII entre dos cadenas de la misma proteína | Se identifican similitudes y diferencias entre las cadenas analizadas, facilitando estudios de simetría estructural o evaluación de variabilidad conformacional. |
| 5 | Personalización de parámetros de análisis | Modificar los parámetros de análisis (ángulos, longitud mínima, distancias) para adaptarse a proteínas atípicas | El sistema detecta nuevas hélices compatibles con los criterios modificados, demostrando su flexibilidad para su uso en estudios no convencionales. |
| 6 | Exportación de hélices detectadas para análisis externo | Permitir la reutilización de resultados estructurales en otros entornos mediante archivos .pdb y .csv. | Los archivos generados pueden utilizarse en Biopython, MDTraj o PyRosetta, ampliando la interoperabilidad del módulo |
| 7 | Análisis geométrico de enlaces potenciales | Analizar la geometría entre átomos implicados en enlaces de hidrógeno no canónicos (Cα–H···O) | Se identifican pares de átomos con geometría compatible con enlaces H no canónicos. Los resultados se visualizan en PyMOL y se guardan en archivos .txt. |
| 8 | Generación automatizada de informes estructurados | Obtener un informe estructurado en .csv con los datos del análisis geométrico | Se genera automáticamente un archivo reporte\_ppii.csv con distancias, ángulos y segmentos PPII, compatible con otras herramientas de análisis. |

## Resumen del proceso de desarrollo

Este capítulo ha presentado el desarrollo técnico del módulo diseñado para la detección semiautomática de hélices PPII en estructuras proteicas, detallando las decisiones adoptadas, las funcionalidades incorporadas y el proceso completo de automatización llevado a cabo.

La creación del módulo responde directamente a una necesidad identificada en el laboratorio: automatizar un procedimiento que hasta ahora se realizaba manualmente mediante herramientas visuales como PyMOL. Si bien este enfoque manual ha resultado útil para el análisis puntual de ciertas proteínas, presenta limitaciones importantes en términos de escalabilidad, reproducibilidad y objetividad. El presente trabajo propone una solución computacional que formaliza esos criterios en un algoritmo programable, capaz de aplicarse de manera sistemática y replicable.

El proceso de diseño e implementación ha incluido las siguientes tareas:

* Automatización del proceso de identificación de hélices PPII, mediante un sistema que reduce la intervención manual y permite obtener resultados consistentes, reproducibles y escalables sobre un gran número de estructuras proteicas.
* Análisis y formalización del procedimiento manual empleado por el grupo de investigación, que ha servido como base para traducir los pasos heurísticos en reglas computacionales claras. Esta tarea implicó la revisión de protocolos internos, la observación directa del uso de PyMOL en el laboratorio y la sistematización de criterios basados en la literatura científica.
* Diseño e implementación del algoritmo de detección, incluyendo la definición de umbrales de ángulos diédricos (φ y ψ), la creación de filtros para eliminar estructuras secundarias no deseadas y la evaluación de enlaces de hidrógeno no canónicos. Esta lógica fue implementada en Python, empleando el API de PyMOL para interactuar directamente con las estructuras proteicas cargadas.
* Desarrollo de una interfaz gráfica integrada en PyMOL, accesible mediante comandos personalizados, que permite lanzar el análisis de forma intuitiva para el usuario. Esta funcionalidad fue diseñada para integrarse sin alterar el flujo de trabajo habitual del laboratorio.
* Validación experimental del módulo, empleando como prueba estructuras proteicas ya caracterizadas de forma manual, lo que facilitó la comparación con los resultados del sistema automatizado y permitió validar su precisión.
* Elaboración de la documentación técnica y guía de uso, orientada a facilitar la adopción de la herramienta por parte de otros miembros del grupo de investigación y de la comunidad científica en general.

Cabe destacar que, aunque el módulo automatiza gran parte del análisis, el sistema permite la intervención manual en cualquier fase del proceso. Esto ofrece a los investigadores la flexibilidad necesaria para introducir ajustes específicos o realizar comprobaciones adicionales directamente en PyMOL.

Tabla 5. Resumen de tareas técnicas realizadas

| **Tarea principal** | **Responsable** |
| --- | --- |
| Adiestramiento inicial y descripción del procedimiento manual | Cristian M. Segura Rodríguez (IQF-CSIC) |
| Apoyo técnico y validación funcional del módulo | Cristian M. Segura Rodríguez (IQF-CSIC)  Douglas V. Laurents (IQF-CSIC) |
| Análisis del procedimiento manual y sistematización | Silvia Enma Rodríguez Fernández (autora) |
| Diseño e implementación del algoritmo en Python | Silvia Enma Rodríguez Fernández (autora) |
| Automatización del proceso de detección en PyMOL | Silvia Enma Rodríguez Fernández (autora) |
| Integración con la interfaz de PyMOL | Silvia Enma Rodríguez Fernández (autora) |
| Pruebas y validación con proteínas reales | Silvia Enma Rodríguez Fernández (autora), validación con apoyo del CSIC |
| Redacción de la guía de uso y documentación técnica | Silvia Enma Rodríguez Fernández (autora) |

La colaboración con el Grupo de Estructura, Dinámica e Interacciones de Proteínas por RMN del IQF-CSIC ha sido clave en la fase de diseño conceptual y validación, si bien todas las tareas técnicas y de desarrollo han sido realizadas íntegramente por la autora en el marco del Trabajo Fin de Grado.

Aunque el diseño conceptual del algoritmo responde a una necesidad identificada conjuntamente con el equipo investigador, el desarrollo técnico, las decisiones de implementación y la programación del módulo han sido realizadas íntegramente por la autora. El trabajo ha contado con la supervisión de la dirección académica del TFG y con el apoyo puntual del equipo de laboratorio en aspectos concretos de interpretación estructural.

Este apartado define el marco técnico del trabajo y concreta la aportación realizada, diferenciando con claridad entre los recursos ya existentes que han sido utilizados —como PyMOL o estructuras públicas del PDB— y los componentes desarrollados específicamente para este proyecto.

## Publicación y contribución al repositorio oficial

Tras la validación funcional del módulo, se ha preparado su incorporación al repositorio oficial de scripts de PyMOL, con el objetivo de facilitar su difusión y adopción por parte de la comunidad científica.

Siguiendo las instrucciones establecidas por el equipo de mantenimiento del repositorio (https://www.pymolwiki.org/index.php/Git\_authors), se ha realizado un fork del proyecto original, es decir, una copia personal del repositorio en GitHub para poder modificarlo sin afectar al original, adaptado la estructura del código, creado un directorio específico para el módulo, y generado la documentación necesaria (`README.md`) en inglés y castellano.

Una vez completada esta preparación, se ha creado un pull request proponiendo su integración oficial, incluyendo una descripción técnica de la herramienta, sus funcionalidades principales y su utilidad potencial en el análisis automatizado de hélices PPII.

La contribución se alinea con la línea de trabajo del Instituto de Química-Física “Blas Cabrera” (IQF-CSIC), donde ha sido validada experimentalmente, y responde al objetivo de transferir el desarrollo tecnológico al entorno científico mediante mecanismos abiertos y sostenibles.

# Resultados y evaluación

## Resultados obtenidos: detección de hélices PPII en 3BOG

La ejecución del módulo sobre la proteína modelo 3BOG ha permitido verificar el correcto funcionamiento de cada una de las funciones implementadas, así como su capacidad para identificar hélices de tipo PPII de manera automatizada y fiable.

Tras cargar la estructura PDB de 3BOG en PyMOL e iniciar el módulo mediante la interfaz desarrollada, se aplicaron los pasos del algoritmo en el orden establecido: filtrado de residuos, cálculo de ángulos diedros φ y ψ, y selección de tramos consecutivos compatibles con los valores característicos de PPII. La salida generada mostró seis segmentos coincidentes con las hélices previamente identificadas mediante análisis manual por parte del equipo investigador, lo que confirma la validez de los criterios empleados y del enfoque automatizado.

Cada hélice detectada se representó en PyMOL con colores distintivos y etiquetas, lo que facilitó la validación visual directa sobre el modelo tridimensional. Además, se generó una salida en formato texto con la posición de los residuos implicados, sus valores de φ y ψ, y el nombre asignado a cada hélice. Esta información puede exportarse en formatos estructurados (.txt, .csv) para su análisis cuantitativo o integración en otros estudios.

Como valor añadido, el módulo permitió identificar los posibles enlaces de hidrógeno no canónicos entre hélices colindantes. Para ello, se midieron distancias entre carbonos alfa y oxígenos de carbonilo, así como los ángulos formados, utilizando umbrales definidos por el usuario. Estos resultados complementan el análisis angular y refuerzan la fiabilidad de la detección, especialmente en estructuras complejas donde la conformación no siempre es evidente a simple vista.

El tiempo total de análisis se redujo significativamente respecto al procedimiento manual, pasando de más de una hora a menos de dos minutos para una estructura de tamaño medio como 3BOG. Este ahorro, sumado a la eliminación de errores humanos por interpretación visual, representa una mejora sustancial para el flujo de trabajo en estudios estructurales.

Tras validar el correcto funcionamiento del módulo sobre 3BOG, se amplió el análisis a otras proteínas incluidas en la Tabla 3 del artículo base, todas ellas caracterizadas por contener regiones desordenadas ricas en glicina y prolina. En estos casos, el sistema fue capaz de detectar segmentos compatibles con hélices PPII y generar los informes geométricos correspondientes, confirmando así su aplicabilidad a distintos contextos estructurales más allá del modelo de referencia.

## Comparación con el análisis manual

La principal motivación de este proyecto ha sido automatizar un proceso que hasta ahora se realizaba de forma completamente manual, requiriendo un conocimiento profundo de la estructura proteica y de PyMOL.

El análisis manual de una proteína como 3BOG implica cargar la estructura, limpiar el modelo (eliminar moléculas foráneas y añadir hidrógenos), identificar visualmente los posibles tramos helicoidales, medir ángulos diedros uno por uno, comprobar la geometría y, si se desea, buscar puentes de hidrógeno no canónicos entre hélices. Este procedimiento es laborioso, propenso a errores y poco reproducible entre diferentes usuarios.

En cambio, el módulo automatiza cada uno de estos pasos, aplicando criterios geométricos de forma objetiva y sistemática. La detección de ángulos φ y ψ se realiza sobre todos los residuos sin necesidad de intervención del usuario, y la selección de tramos candidatos sigue reglas predefinidas que pueden ajustarse mediante parámetros. Además, la identificación de posibles enlaces de hidrógeno se lleva a cabo mediante el cálculo directo de distancias y ángulos, sin depender de la inspección visual.

Durante la validación, se verificó que el módulo detecta las mismas hélices que el análisis manual, con una coincidencia total en los tramos detectados. Esta comparación se ha extendido a la totalidad de estructuras incluidas en el estudio original **de Segura Rodríguez y Laurents (2024),** utilizando las estructuras detalladas en la Tabla 3. En todos los casos, el módulo reprodujo con exactitud los tramos previamente identificados y generó salidas estructuradas que mejoran la trazabilidad y facilitan su reutilización en contextos bioinformáticos.

En consecuencia, el sistema automatizado no solo iguala la precisión del método manual, sino que lo supera en eficiencia, reproducibilidad y aplicabilidad a gran escala.

## Valoración de precisión y eficiencia

La evaluación del módulo desarrollado ha puesto de manifiesto una alta precisión en la detección de hélices PPII, respaldada por la coincidencia total con los tramos identificados manualmente en la proteína modelo 3BOG. Los valores de los ángulos diedros φ y ψ calculados por el sistema se sitúan dentro de los rangos esperados para esta conformación, lo que indica que el algoritmo implementado cumple con los criterios geométricos definidos en el marco teórico y aplicados previamente en el laboratorio.

En cuanto a eficiencia, el módulo permite completar el análisis de una estructura media en cuestión de segundos o pocos minutos, según el número de cadenas y la complejidad de la proteína. Esta mejora operativa representa un avance considerable frente al proceso manual tradicional, que requiere una intervención prolongada y repetitiva para medir ángulos, evaluar geometría y comprobar interacciones. Esta reducción en el tiempo de análisis resulta especialmente útil cuando se trabaja con múltiples estructuras o se necesitan realizar pruebas bajo distintas configuraciones de parámetros.

Además, el sistema permite configurar umbrales de tolerancia y criterios de detección, lo que amplía su flexibilidad y adaptabilidad a distintos contextos de investigación. Esta capacidad de ajuste, combinada con la automatización del análisis y la generación de salidas exportables, convierte al módulo en una herramienta eficaz tanto para estudios exploratorios como para análisis sistemáticos de grandes volúmenes de datos estructurales.

Desde el punto de vista de la reproducibilidad, el módulo garantiza resultados consistentes independientemente del usuario que lo ejecute, eliminando así la variabilidad subjetiva propia del análisis visual. Esta estandarización es esencial en entornos de investigación donde se requiere comparar resultados entre diferentes experimentos, equipos o condiciones.

En conjunto, el sistema desarrollado destaca por su equilibrio entre precisión geométrica, eficiencia operativa y facilidad de uso, consolidándose como una solución robusta, replicable y escalable para la detección automatizada de hélices PPII en estructuras proteicas.

En suma, el sistema permite un análisis preciso, rápido y reproducible, reduciendo drásticamente la carga manual y facilitando su integración en flujos de trabajo automatizados.

## 6.4. Limitaciones identificadas

Aunque el módulo desarrollado ha demostrado ser eficaz para la detección de hélices de tipo poliprolina II (PPII) en proteínas modelo, durante su implementación se han identificado algunas limitaciones relacionadas con el rendimiento y la escalabilidad del sistema, especialmente al trabajar con estructuras de gran tamaño o que contienen múltiples cadenas. El uso intensivo de funciones internas de PyMOL, aunque cómodo, puede ralentizar notablemente la ejecución cuando se analizan muchas combinaciones posibles entre átomos, como sucede durante el análisis de enlaces de hidrógeno no canónicos.

Para mitigar esta situación, se ha rediseñado el cálculo de distancias y ángulos empleando operaciones matemáticas básicas del módulo math, evitando así las funciones internas más pesadas. Esta optimización ha permitido reducir notablemente los tiempos de procesamiento y mejorar la estabilidad del programa en análisis prolongados.

Aun así, el rendimiento podría seguir viéndose afectado al analizar estructuras de gran complejidad o cuando se procesan múltiples cadenas de forma simultánea. Como propuesta de mejora, se plantea en futuras versiones incorporar técnicas como:

* Paralelización del análisis mediante multiprocessing, aprovechando los núcleos del procesador.
* Análisis por bloques o por cadenas, en lugar de procesar toda la estructura en un solo ciclo.
* Uso de bibliotecas como NumPy para realizar cálculos vectorizados de forma más eficiente.
* Posible desacoplamiento de algunas partes del análisis del entorno PyMOL, empleando herramientas externas como MDTraj o Biopython.

Estas estrategias permitirían solventar las limitaciones derivadas del entorno de ejecución y escalar el análisis a estructuras más complejas, facilitando el análisis de grandes volúmenes de datos estructurales de forma más ágil y robusta.

Aunque se considera razonable el uso de bibliotecas como NumPy, en esta versión inicial se ha optado por no integrarlas para garantizar la compatibilidad directa con el entorno de ejecución de PyMOL, donde no siempre está disponible por defecto. Además, las operaciones requeridas pueden resolverse eficientemente mediante funciones básicas del módulo math, lo que ha permitido mantener el sistema ligero, portable y sin dependencias externas adicionales.

Además, se ha identificado una limitación relacionada con la detección de enlaces de hidrógeno no canónicos. El algoritmo actual marca como posibles interacciones aquellas en las que un hidrógeno (donante) presenta distancia y ángulo compatibles con más de un oxígeno (aceptor) cercano. Esto da lugar a que, en algunos casos, un mismo aminoácido aparezca vinculado a dos enlaces no canónicos simultáneamente, lo cual no es físicamente viable, ya que un único donante no puede transferir densidad electrónica a dos aceptores distintos al mismo tiempo.

A pesar de esta imprecisión, se ha optado por mantener este comportamiento en la versión actual, dado que la discriminación final entre interacciones plausibles y artefactos geométricos depende del patrón estructural completo, el cual puede requerir cierta interpretación subjetiva por parte del investigador. En consecuencia, el módulo debe entenderse como un sistema de preselección, que facilita un cribado automatizado eficaz, pero cuya interpretación **requiere necesariamente la intervención de un investigador con criterio científico,** capaz de contextualizar los resultados y validar cuáles de esas interacciones son estructuralmente relevantes.

## 6.5. Aplicabilidad en otros modelos estructurales

Aunque la validación del módulo comenzó con el análisis de la proteína modelo 3BOG, luego se extendió al resto de las proteínas modelo del estudio, y su diseño permite su aplicación directa sobre cualquier estructura proteica disponible en formato PDB. Gracias a su arquitectura modular y a la posibilidad de ajustar los parámetros de detección, el sistema puede adaptarse a proteínas de distinto tamaño, complejidad y nivel de anotación estructural.

Esta versatilidad resulta especialmente útil en estudios centrados en regiones desordenadas, dominios ricos en glicina y prolina, o secuencias peptídicas no estructuradas susceptibles de adoptar conformaciones PPII en determinados entornos fisiológicos o experimentales. En estos casos, donde las herramientas convencionales tienden a clasificar estas regiones como coil o desorden, el módulo ofrece un enfoque alternativo basado en criterios geométricos objetivos.

Además, el sistema puede integrarse en scripts externos, lo que permite realizar análisis masivos sobre bases de datos estructurales. Esta capacidad es particularmente valiosa en proyectos de cribado estructural o en investigaciones orientadas al diseño racional de péptidos con conformaciones específicas.

También puede resultar útil en estudios evolutivos o comparativos, donde se analicen familias de proteínas con variaciones sutiles en su plegamiento. El uso del módulo sobre diferentes estructuras permite detectar patrones comunes de conformación PPII o identificar mutaciones que alteren dicha conformación.

Por último, al no depender de una base de datos concreta ni de herramientas externas, puede emplearse con estructuras obtenidas mediante cristalografía, espectroscopía de RMN o predicción por inteligencia artificial, siempre que estén disponibles en el formato adecuado. Esto refuerza su papel como herramienta versátil y adaptable para distintos contextos de análisis estructural.

En conjunto, el módulo desarrollado no solo ha demostrado su utilidad en el análisis de las proteínas modelo recogidas en el estudio de referencia, sino que presenta el potencial técnico y funcional necesario para integrarse en flujos de trabajo más amplios dentro de la biología estructural, la bioinformática y el diseño computacional de biomoléculas contribuyendo así a una caracterización más eficiente y reproducible de este tipo de estructuras, incluso en entornos experimentales poco definidos o en etapas tempranas de desarrollo de fármacos o péptidos funcionales.

**Los archivos generados automáticamente por el módulo durante el análisis** (valores angulares, segmentos detectados, distancias y enlaces potenciales) **se recogen como ejemplo en el Anexo E,** donde se muestran fragmentos representativos de los ficheros angulos\_phi\_psi.csv, reporte\_ppii.csv, angulos\_ca\_h\_o.txt y distancias\_colindantes.txt. Estos ficheros ilustran el tipo de salida que puede obtenerse tras aplicar el sistema a una proteína modelo y permiten al investigador revisar y reutilizar los resultados obtenidos.

# Conclusiones y trabajo futuro

Este capítulo presenta las conclusiones derivadas del desarrollo del proyecto, valorando el grado de cumplimiento de los objetivos planteados. Asimismo, se proponen posibles líneas de trabajo futuro orientadas a ampliar, mejorar o profundizar los resultados obtenidos.

## 7.1. Conclusiones del trabajo

El objetivo general de este proyecto consistía en desarrollar un módulo integrado en PyMOL capaz de automatizar la detección de hélices de tipo PPII a partir de estructuras proteicas en formato PDB. A lo largo del trabajo se han abordado y cumplido los distintos objetivos específicos planteados para alcanzar dicha meta.

En primer lugar, se analizó en profundidad la problemática asociada a la identificación manual de hélices PPII, recopilando tanto los criterios geométricos descritos en la literatura como el protocolo utilizado en el laboratorio de referencia. Esta fase permitió definir con claridad los parámetros estructurales clave para la detección.

A partir de este análisis, se desarrolló un algoritmo capaz de identificar automáticamente segmentos compatibles con la conformación PPII, mediante el cálculo de los ángulos diedros φ y ψ, así como la evaluación de posibles enlaces de hidrógeno no canónicos entre hélices colindantes. El sistema fue implementado en Python e integrado en PyMOL a través de una interfaz gráfica intuitiva, lo que facilita su uso por investigadores sin experiencia en programación y permite su incorporación directa en sesiones de trabajo habituales.

La validación experimental, realizada sobre la proteína modelo 3BOG, confirmó que el sistema reproduce con precisión los resultados obtenidos mediante análisis manual, reduciendo significativamente el tiempo de ejecución y eliminando la subjetividad inherente a los métodos visuales. El módulo permite detectar, visualizar y documentar regiones candidatas a hélices PPII de forma clara y reproducible, reforzando así su utilidad como herramienta en análisis estructural. Además, los parámetros geométricos utilizados en la detección, como los umbrales de los ángulos φ y ψ o el rango angular del enlace Cα–H···O, pueden ajustarse fácilmente, lo que permite adaptar el análisis a estructuras no canónicas o condiciones experimentales concretas.

En conjunto, puede afirmarse que el proyecto ha cumplido satisfactoriamente sus objetivos, aportando una solución técnica que automatiza un proceso complejo y permite su integración en flujos de trabajo reales. Además, sienta las bases para futuras ampliaciones orientadas al análisis masivo de estructuras y al desarrollo de herramientas más avanzadas en el ámbito de la bioinformática estructural. El impacto potencial del módulo, tanto en entornos científicos como docentes, se analiza en detalle en el apartado 7.4.

Este proyecto demuestra que es posible convertir un análisis estructural complejo en una herramienta práctica y reutilizable, acercando la automatización a la rutina investigadora sin perder rigor ni precisión.

## 7.2. Aportaciones del módulo

El módulo desarrollado en este trabajo representa una aportación técnica y funcional relevante en el ámbito del análisis estructural de proteínas, al automatizar la detección de hélices de tipo PPII mediante criterios geométricos. Su implementación permite ejecutar de forma reproducible, rápida y objetiva un proceso que hasta ahora se realizaba manualmente, eliminando la dependencia del análisis visual y reduciendo el margen de error humano.

Desde el punto de vista práctico, la principal aportación del módulo radica en su capacidad para calcular automáticamente los ángulos diedros φ y ψ en cada residuo, filtrar aquellos que cumplen con los criterios de conformación PPII y representar visualmente los segmentos detectados directamente sobre el modelo tridimensional en PyMOL. Además, incorpora una funcionalidad opcional para detectar posibles enlaces de hidrógeno no canónicos entre hélices colindantes, lo que refuerza la validación estructural de los resultados obtenidos.

Otra aportación destacable es su integración en un entorno ya familiar para los investigadores en bioinformática estructural: PyMOL. Esta característica permite su adopción inmediata en flujos de trabajo reales, sin necesidad de herramientas externas ni conocimientos avanzados de programación. La interfaz gráfica permite su uso incluso por investigadores sin experiencia en programación, ampliando así el alcance potencial de la herramienta.

Finalmente, la publicación del código en un repositorio abierto garantiza su accesibilidad, fomenta su reutilización por otros grupos de investigación y abre la puerta a futuras mejoras colaborativas. En conjunto, el módulo constituye una herramienta útil, versátil y alineada con los principios de la ciencia abierta, con potencial para integrarse en análisis estructurales de mayor escala o en líneas de investigación orientadas al diseño racional de péptidos con estructura PPII. Además, el diseño modular y el código abierto permiten ajustar los rangos angulares empleados, facilitando su adaptación a distintos tipos de proteínas o condiciones experimentales.

## 7.3. Propuestas de mejora y líneas futuras

A partir del trabajo desarrollado, se identifican diversas líneas de mejora que podrían abordarse en proyectos posteriores para ampliar el alcance y la utilidad del módulo.

En primer lugar, se plantea mejorar la precisión del algoritmo mediante el ajuste fino de sus parámetros o mediante el uso de modelos computacionales más avanzados. A medio plazo, las salidas generadas por este módulo podrían emplearse como base de datos de entrenamiento para sistemas de inteligencia artificial orientados a la predicción estructural, como AlphaFold o herramientas similares. Esta estrategia podría conducir a una detección automática de hélices PPII en modelos estructurales más complejos, especialmente en regiones desordenadas o poco caracterizadas. Asimismo, sería interesante extender el análisis a interacciones entre hélices PPII y otras regiones estructurales no PPII, actualmente fuera del alcance de este proyecto.

Aunque las herramientas convencionales como DSSP o STRIDE no detectan hélices de tipo poliprolina II de forma específica, muchas de las regiones que clasifican como “coil” o desordenadas podrían corresponder a tramos con geometría PPII. Por ello, **una posible línea de mejora** sería incorporar una funcionalidad que permita comparar los resultados del módulo con esas asignaciones estructurales generales. Este enfoque podría ayudar a validar el comportamiento del algoritmo en diferentes contextos y a identificar patrones repetitivos que, aunque pasen desapercibidos para los métodos estándar, revelen patrones estructurales basados en conformaciones PPII.

Aunque el módulo ya permite configurar el umbral utilizado para detectar enlaces de hidrógeno no canónicos, podría afinarse aún más mediante una calibración basada en evidencia experimental, ajustando con mayor precisión los valores de distancia y ángulo C–H···O. Aunque el módulo ya incorpora esta medición, la diversidad de criterios existentes en la literatura científica sobre los valores mínimos aceptables para este tipo de enlaces en hélices PPII limita su optimización. Una vez que la comunidad científica establezca criterios más definidos, podría afinarse la detección para reducir falsos positivos y mejorar la fiabilidad estructural del análisis.

Como contribución a la comunidad científica, ya se contempla el envío de las hélices PPII detectadas mediante este módulo a bases de datos públicas como PolyprOnline (Chebrek et al., 2014), con el objetivo de enriquecer los recursos disponibles y favorecer la validación cruzada de estructuras poco caracterizadas. Dado que el sistema permite exportar las hélices en formatos estructurados y legibles por terceros, esta colaboración puede llevarse a cabo fácilmente tras la validación del módulo, reforzando así su visibilidad y su utilidad práctica en futuros proyectos de investigación. La base de datos PolyprOnline está disponible públicamente en: <https://www.dsimb.inserm.fr/dsimb_tools/polyproline>.

A largo plazo, podría incorporarse una funcionalidad de análisis por lotes, permitiendo la detección automática de hélices PPII en múltiples estructuras cargadas desde un directorio. Esta capacidad facilitaría estudios de cribado estructural, minería de datos o análisis masivos en proyectos de alto rendimiento, ampliando el alcance del módulo más allá del uso individual.

En conjunto, estas líneas futuras abren la puerta al desarrollo de versiones más robustas, precisas y adaptables del módulo, con potencial para consolidarse como una herramienta de referencia en el análisis automatizado de estructuras proteicas.

## 7.4. Impacto y transferencia del trabajo

El desarrollo del módulo no solo ha permitido automatizar un proceso complejo y mejorar su reproducibilidad, sino que también abre la puerta a su integración en iniciativas científicas y formativas más amplias. Gracias a su carácter abierto, su diseño modular y su compatibilidad con formatos estándar, el sistema puede ser adoptado fácilmente por otros grupos de investigación.

Entre las acciones previstas tras la finalización del proyecto se encuentra el envío de los resultados generados a bases de datos públicas como PolyprOnline, especializada en hélices PPII. Esta colaboración contribuiría a enriquecer los recursos disponibles sobre este tipo de estructuras y a validar experimentalmente las detecciones realizadas por el módulo, reforzando su impacto científico.

Asimismo, la publicación del código en un repositorio abierto bajo licencia GPLv3 garantiza su accesibilidad, fomenta su reutilización por parte de la comunidad y permite su mejora colaborativa en futuras versiones.

Desde el punto de vista docente, se han elaborado casos de uso guiados que podrían utilizarse como material formativo en asignaturas de bioinformática estructural o como soporte práctico para investigadores no expertos en programación. Este enfoque favorece la transferencia del conocimiento generado y amplía el alcance del proyecto más allá del entorno técnico.

En conjunto, el módulo desarrollado no solo ha demostrado su utilidad en el análisis de las proteínas modelo recogidas en el estudio de referencia, sino que presenta el potencial técnico y funcional necesario para integrarse en flujos de trabajo más amplios dentro de la biología estructural, la bioinformática y el diseño computacional de biomoléculas, contribuyendo así a una caracterización más eficiente y reproducible de este tipo de estructuras.

Además, la posibilidad de ajustar fácilmente los parámetros geométricos —como los ángulos φ y ψ, o el rango angular del enlace Cα–H···O— amplía su aplicabilidad en contextos experimentales diversos y estructuras no convencionales.

En definitiva, este trabajo no solo ha resuelto un problema técnico concreto, sino que abre un camino hacia la automatización eficiente de análisis estructurales avanzados, accesibles y reproducibles en contextos científicos y docentes.

Referencias bibliográficas

Anaconda, Inc. (2023). *Anaconda Distribution* (Version 2023.03) [Computer software]. <https://www.anaconda.com/>

Berka, K., et al. (2023). MDTraj: *A Python library for the analysis of molecular dynamics trajectories*. <https://www.mdtraj.org/>

Blanquel, E., Blanquel, C., & Luna-García, R. (2019). Hybrid evolutionary algorithm for molecular geometric optimization*. Computación y Sistemas, 23* (2), 291–300. <https://doi.org/10.13053/cys-23-2-2541>

brooksj4. (2020, July 16). Practical course: Calculate phi, psi and omega angles of proteins in PyMOL [Video]. YouTube. <https://www.youtube.com/watch?v=1wzVbIzqA2M>

Broz, M., Jukič, M., & Bren, U. (2023). Naive prediction of protein backbone phi and psi

dihedral angles using deep learning. *Molecules, 28* (20), 7046.

<https://doi.org/10.3390/molecules28207046>

Buglino, J. A., Shen, V., Hakimian, P., & Lima, C. D. (2002). Structural and biochemical analysis of the Obg GTP binding protein. *Structure*, 10(11), 1581–1591. <https://doi.org/10.1016/S0969-2126(02)00878-5>

Chebrek, R., Leonard, S., de Brevern, A. G., & Gelly, J. C. (2014). PolyprOnline: Polyproline helix II and secondary structure assignment database. *Database, 2014*, bau102. <https://doi.org/10.1093/database/bau102>

Chemical Computing Group. (2024). *MOE: Molecular Operating Environment*. ) [Computer software]. <https://www.chemcomp.com/MOE-Molecular_Operating_Environment.htm>

Dassault Systèmes. (2025). *BIOVIA Discovery Studio 2025*. ) [Computer software]. <https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-discovery-studio/>

DeLano, W. L. (2002). *The PyMOL molecular graphics system* [Computer software]. DeLano Scientific. <https://pymol.org/>

Dunne, M., Rupf, B., Tala, M., Qabrati, X., Ernst, P., Shen, Y., ... & van Raaij, M. J. (2018). Salmonella phage S16 encodes a long tail fiber that recognizes OmpC. *Nature Microbiology*, 3(11), 1174–1182. <https://doi.org/10.1038/s41564-018-0246-0>

Frishman, D., & Argos, P. (1995). Knowledge-based protein secondary structure assignment. Proteins: Structure, Function, and Bioinformatics, 23(4), 566–579. <https://doi.org/10.1002/prot.340230412>

Free Software Foundation. (2007). *GNU General Public License v3.0*.

<https://www.gnu.org/licenses/gpl-3.0.html>

Gkekas, C., Garg, J., Sakuraba, H., & Ohshima, T. (2017). Crystal structure of E. coli Obg in complex with GDP*. Biochemical and Biophysical Research Communications,* 484(3), 537–542. <https://doi.org/10.1016/j.bbrc.2017.01.095>

Hanwell, M. D., Curtis, D. E., Lonie, D. C., Vandermeersch, T., & Zurek, E. (2024). *Avogadro* (Version 1.100.0) [Computer software]. <https://avogadro.cc/>

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering,* 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>

Jmol Development Team. (n.d.). *Jmol: An open-source Java viewer for chemical structures in 3D*. [Computer software]. <http://www.jmol.org/>

Kukimoto-Niino, M., Murayama, K., Inoue, M., Terada, T., Tame, J. R. H., Kuramitsu, S., & Yokoyama, S. (2004). Crystal structure of the GTP-binding protein Obg from Thermus thermophilus*. Structure*, 12(7), 1223–1230. <https://doi.org/10.1016/j.str.2004.04.015>

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference*, 51–56.

<https://doi.org/10.25080/Majora-92bf1922-00a>

Microsoft. (2023). *Visual Studio Code* [Computer software]. <https://code.visualstudio.com/>

Mus, F., Woyke, T., Kirton, E., Ivanova, N., Lykidis, A., & McInerney, M. J. (2017). Comparative genomics and functional analysis of acetone carboxylase*. Environmental Microbiology*, 19(12), 5072–5088. <https://doi.org/10.1111/1462-2920.13928>

Pentelute, B. L., Gates, Z. P., Tereshko, V., Dashnau, J. L., Vanderkooi, J. M., Kossiakoff, A. A., & Kent, S. B. H. (2008*). X-ray structure of snow flea antifreeze protein determined by racemic crystallization of synthetic protein enantiomers*. Journal of the American Chemical Society, 130(30), 9695–9701. <https://doi.org/10.1021/ja802980u>

Pettersen, E. F., et al. (2021). UCSF ChimeraX: *Structure visualization for researchers, educators, and developers*. Protein Science, 30(1), 70–82. <https://doi.org/10.1002/pro.3943>

PyMOLWiki. (n.d.). PyMOL. <https://pymolwiki.org/index.php/Main_Page>

Python Software Foundation. (2023). *Python* (Version 3.10) [Computer software]. <https://www.python.org/>

Python Software Foundation. (2023). *Tkinter* [**Computer software**]**.** <https://docs.python.org/3/library/tkinter.html>

Reshetnyak, A. V., Murray, P. B., Shi, X., Mo, E. S., Mohanty, J., Tome, F., ... & Nicolis di Robilant, B. (2021). Augmented structural dynamics of anaplastic lymphoma kinase underlie the mechanism of activation*. Nature Communications*, 12(1), 2083. <https://doi.org/10.1038/s41467-021-22322-1>

Rodríguez Fernández, S. E. (2024). *PPIIMoL: Automatización de la detección de hélices PPII* [**Computer software].**

GitHub. <https://github.com/silviaenma/PPIIMoL>

Sayle, R. A., & Milner-White, E. J. (2025). *RasMol: Molecular Graphics Program*. [**Computer software**]**.** <http://www.rasmol.org>

Segura Rodríguez, C. M., & Laurents, D. V. (2024). Architectonic principles of polyproline II helix bundle protein domains. Archives of Biochemistry and Biophysics, 741, 109981. <https://doi.org/10.1016/j.abb.2024.109981>

Schrödinger, LLC. (2015). *The PyMOL molecular graphics system* (Version 2.0) [**Computer s**oftware]. <https://pymol.org/>

Schrödinger, LLC. (2025). Maestro (Version 2025-2)[**Computer s**oftware]. <https://www.schrodinger.com/maestro>

Scholl, J. P., Yang, F., Marsili, E., & Liu, Y. (2021). Structure of antifreeze protein from Granisotoma rainieri. *Journal of Structural Biology*, 213(4), 107749. <https://doi.org/10.1016/j.jsb.2021.107749>

The pandas development team. (2023). *pandas-dev/pandas: Pandas* (v2.0) [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.3509134>

Theoretical and Computational Biophysics Group. (2025). *VMD - Visual Molecular Dynamics*. [Computer software].<https://www.ks.uiuc.edu/Research/vmd/>

University of California, San Francisco. (2025a). UCSF ChimeraX (Version 1.10 RC) ) [Computer software]. <https://www.cgl.ucsf.edu/chimerax/>

University of California, San Francisco. (2025b). UCSF Chimera (Version 1.19) ) [Computer software]. <https://www.cgl.ucsf.edu/chimera/>

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace Independent Publishing Platform.

Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

Warkentin, E., Müh, U., Demmer, U., Moll, J., Stöckigt, J., & Ermler, U. (2017). Structure and function of an acetophenone carboxylase. Proceedings of the National Academy of Sciences, 114(5), E413–E420. <https://doi.org/10.1073/pnas.1619885114>

1. **Fragmentos de código relevantes del módulo**

A continuación, se presentan fragmentos seleccionados del código desarrollado para el módulo de detección de hélices PPII. Estos extractos representan las funciones principales que componen la lógica del sistema.

## Fragmento A1. Función seleccionar\_archivo

def seleccionar\_archivo():

archivo = filedialog.askopenfilename(filetypes=[("Archivos PDB", "\*.pdb")])

if archivo:

cmd.load(archivo, "proteina")

cmd.show("licorice", "proteina")

## Fragmento A2. Función guardar\_angulos\_phi\_psi

def guardar\_angulos\_phi\_psi():

with open("angulos\_phi\_psi.txt", "w") as archivo:

cadenas = cmd.get\_chains("proteina")

for cadena in cadenas:

angulos = cmd.get\_phipsi(f"proteina and chain {cadena}")

for residuo, valores in angulos.items():

archivo.write(f"{cadena},{residuo[1]},{valores[0]},

{valores[1]}\n")

## Fragmento A3. Función eliminar\_solventes

def eliminar\_solventes():

cmd.remove("solvent")

## Fragmento A4. Función anadir\_hidrogenos

def anadir\_hidrogenos():

cmd.h\_add()

## Exportación estructurada de resultados para uso en herramientas externas:

A continuación, se presentan los fragmentos de código relacionados con la exportación estructurada de los resultados generados por el módulo. Estos permiten guardar tanto los tramos helicoidales detectados en formato .pdb como generar tablas en .csv con los valores de ángulos φ y ψ y los enlaces de hidrógeno no canónicos identificados. Estas salidas están pensadas para su uso posterior en herramientas externas como Biopython, MDTraj o PyRosetta.

## Fragmento A5. Función exportar\_helice\_pdb

## Exporta hélices como archivos .pdb desde PyMOL

def exportar\_helice\_pdb(nombre\_seleccion, nombre\_archivo):

"""

Exporta la selección de PyMOL como un archivo PDB.

"""

cmd.save(nombre\_archivo, selection=nombre\_seleccion)

Ejemplo de uso dentro del módulo:

exportar\_helice\_pdb("helice\_ppii\_1", "H1\_3bog.pdb")

## Fragmento A6. Función exportar\_angulos\_csv

## Exporta a .csv los datos de los ángulos φ y ψ

import csv

def exportar\_angulos\_csv(nombre\_archivo, lista\_angulos):

with open(nombre\_archivo, 'w', newline='') as f:

writer = csv.writer(f)

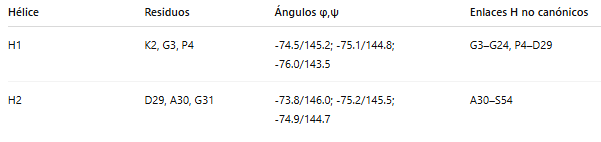
writer.writerow(["Residuo", "Ángulo φ", "Ángulo ψ"])

for res, phi, psi in lista\_angulos:

writer.writerow([res, f"{phi:.2f}", f"{psi:.2f}"])

Ejemplo de salida .csv generada por el módulo

Figura A1. Fragmento del archivo .csv generado con los datos estructurados de hélices PPII detectadas.



*Nota*. Se incluyen los residuos implicados, sus ángulos φ y ψ, y los enlaces de hidrógeno no canónicos detectados.

## Fragmento A7. Función calcular\_angulos\_ca\_h\_o

Calcula ángulos geométricos Cα–H···O para detectar enlaces no canónicos.

def calcular\_angulos\_ca\_h\_o(tripletas):

resultados = []

for ca, h, o in tripletas:

v1 = h.coord - ca.coord

v2 = o.coord - h.coord

cos\_ang = np.dot(v1, v2) / (np.linalg.norm(v1) \* np.linalg.norm(v2))

angulo = np.degrees(np.arccos(cos\_ang))

if 110 <= angulo <= 180:

resultados.append((ca.index, h.index, o.index, angulo))

return resultados

## Fragmento A8. Función generar\_reporte\_csv()

Genera un archivo .csv con distancias y ángulos de enlaces potenciales.

def generar\_reporte\_csv(nombre\_archivo, datos):

with open(nombre\_archivo, 'w', newline='') as f:

writer = csv.writer(f)

writer.writerow(["Cα", "H", "O", "Ángulo CA-H···O", "Distancia CA-O"])

for ca, h, o, ang, dist in datos:

writer.writerow([ca, h, o, f"{ang:.2f}", f"{dist:.2f}"])

## Fragmento A9. Función calcular\_y\_visualizar\_distancias()

Calcula distancias relevantes entre segmentos y las muestra gráficamente.

def calcular\_y\_visualizar\_distancias(segmentos):

for i in range(len(segmentos) - 1):

for atomo1 in segmentos[i]:

for atomo2 in segmentos[i+1]:

dist = cmd.get\_distance(atomo1, atomo2)

if dist < 5.0:

cmd.distance(f"dist\_{atomo1}\_{atomo2}", atomo1, atomo2)

El código completo, actualizado y comentado, se encuentra disponible en el repositorio público del proyecto en GitHub, junto con los archivos de ejemplo y documentación técnica complementaria:  
[https://github.com/silviaenma/ PPIIMoL](https://github.com/silviaenma/PPIIdetector)

A continuación, la tabla 6 resume la funcionalidad de los fragmentos de código incluidos:

Tabla 6. Funcionalidad de los fragmentos de código incluidos

| **Fragmento** | **Función principal** | **Relevancia** |
| --- | --- | --- |
| A1 | seleccionar\_archivo() | Carga del modelo inicial |
| A2 | guardar\_angulos\_phi\_psi() | Registro de datos clave |
| A3 | eliminar\_solventes() | Limpieza inicial del modelo para análisis estructural |
| A4 | anadir\_hidrogenos() | Preparación de la geometría para detección de enlaces. |
| A5 | exportar\_helice\_pdb() | Exportación estructural |
| A6 | exportar\_angulos\_csv() | Análisis cuantitativo |
| A7 | calcular\_angulos\_ca\_h\_o() | Detección de enlaces Cα–H···O no canónicos |
| A8 | generar\_reporte\_csv() | Generación de informes estructurados para documentación |
| A9 | calcular\_y\_visualizar\_distancias() | Visualización de interacciones entre segmentos consecutivos |
|  |  |  |

1. **Guía técnica del módulo de detección de hélices de tipo PPII en PyMOL**

## 1. Introducción

Este módulo ha sido desarrollado para facilitar la detección automática de hélices de tipo poliprolina II (PPII) en estructuras de proteínas visualizadas con PyMOL. La herramienta está dirigida a investigadores, docentes y usuarios de bioinformática estructural que deseen identificar de forma rápida este tipo de estructura secundaria, especialmente en el contexto del análisis de regiones desordenadas o ricas en prolina y glicina, habituales en proteínas funcionalmente relevantes. Las hélices PPII son relevantes por su papel en la interacción proteína-proteína y en la conformación de regiones desordenadas estructuralmente, lo que las convierte en un objetivo de interés en el análisis estructural.

## 2. Requisitos previos

- Tener instalada una versión reciente de PyMOL (recomendado: PyMOL 2.5 o superior).

- Python 3.7 o superior (el módulo ha sido desarrollado y probado con Python 3.9.13).

- Módulo del proyecto: `PPIIMoL.py`.

- Se recomienda trabajar en un entorno gestionado con Anaconda para facilitar la integración de PyMOL y Python.

Biblioteca gráfica tkinter (incluida por defecto en Python en la mayoría de sistemas).

## 3. Instalación y configuración

1. Descargar el archivo PPIIMoL.py desde el repositorio oficial del proyecto: <https://github.com/silviaenma/PPIIMoL>, y copiar en el mismo directorio donde se ejecutará PyMOL, o en una ruta accesible desde el terminal de comandos.

2. Abrir PyMOL con interfaz de comandos.

3. En la consola de PyMOL, importar el módulo escribiendo:

run PPIIMoL.py

4. Si se trabaja en un entorno gestionado por Conda, asegurarse de activarlo antes de ejecutar PyMOL:

conda activate pymol\_env

## 4. Carga de estructuras

Para analizar una proteína, primero debe cargarse el archivo .pdb en PyMOL:

load ruta/al/archivo/nombre-residuo.pdb

Sustituya ruta/al/archivo/ por la ubicación real del archivo .pdb en su equipo. Por ejemplo:

load C:/Users/silvia/Documents/proteinas/3bog.pdb

También pueden utilizarse archivos descargados directamente desde el PDB online:

fetch nombre-residuo

Por ejemplo, para descargar la estructura con ID 1abc, el comando sería:

fetch 1abc

También puede usarse la opción “Descargar proteína” incluida en la interfaz gráfica del módulo, que permite introducir un código PDB sin necesidad de línea de comandos.

## 5. Uso del módulo

- Se activa una interfaz gráfica integrada en PyMOL.

- Desde la interfaz, el usuario puede seleccionar las cadenas a analizar.

- El módulo analiza automáticamente los ángulos diedros φ y ψ de cada residuo y detecta tramos consecutivos que cumplan los criterios geométricos de la hélice PPII. También permite calcular distancias entre segmentos, estimar ángulos Cα–H···O, y exportar los resultados en diferentes formatos estructurados.

- Las regiones detectadas se muestran resaltadas en color, y se genera una selección llamada `ppii\_segmento\_cadena\_n1\_n2 `, por ejemplo: `ppii\_segmento\_A\_3\_8`, donde n1 es el primer aminoácido que forma parte de la selección y n2 el último aminoácido de dicha selección.

## 6. Interpretación de resultados

- Los tramos identificados como hélices PPII se representan con un color distintivo (por ejemplo, rojo).

- Cada hélice detectada se guarda como una selección independiente en PyMOL, con el nombre: `ppii\_segmento\_[cadena]\_[inicio]\_[fin]` El usuario puede activar, ocultar o modificar la selección desde la interfaz de PyMOL.

- Se puede realizar zoom sobre cada región, activar o desactivar selecciones y modificar su estilo de visualización.

**Archivos generados típicos:**

Tras el análisis, los resultados se almacenan automáticamente en una carpeta llamada Resultados\_PPIIMoL, ubicada en el mismo directorio desde el que se ejecutó el script.

Los archivos típicos que se generan son:

* angulos\_phi\_psi.txt: listado detallado de los ángulos φ y ψ para cada residuo
* helices\_ppii.csv: tabla resumen de hélices detectadas, su longitud, cadena, posición y conformación.
* H1\_3bog.pdb, H2\_3bog.pdb, etc.: archivos .pdb individuales con las hélices exportadas.
* reporte\_ppii.csv: contiene las distancias Cα–H···O y ángulos geométricos detectados en los posibles enlaces no canónicos.

## 7. Personalización o ampliación

El código está modularizado para facilitar su modificación y extensión. Entre las opciones personalizables se incluyen:

* **Ajuste de umbrales**: los valores umbral para los ángulos φ y ψ pueden modificarse directamente desde el script para adaptarse a distintos criterios estructurales.
* **Estilo visual**: es posible cambiar el color, el grosor o el tipo de representación gráfica de las hélices detectadas. El color por defecto (rojo) puede sustituirse fácilmente desde la línea que define la selección.
* **Detección de otras estructuras**: el algoritmo puede adaptarse para buscar otros motivos secundarios no canónicos, como giros, láminas irregulares o segmentos ricos en glicina.
* **Exportación de resultados**: el módulo puede ampliarse para generar salidas personalizadas. Actualmente exporta archivos .pdb con las hélices detectadas, así como ficheros .txt y .csv con los ángulos y los residuos identificados. Estos formatos pueden ajustarse para integrarse con otros programas de análisis estructural o bases de datos.
* **Análisis de interacciones**: puede añadirse funcionalidad para analizar interacciones entre distintas regiones estructurales o detectar patrones de agrupamiento entre hélices PPII.

Gracias al diseño modular del código, todas estas modificaciones pueden implementarse de forma relativamente sencilla si se dispone de conocimientos básicos de Python y PyMOL.

## 8. Créditos y contacto

Este módulo ha sido desarrollado como parte de un Trabajo Fin de Grado en Ingeniería Informática de la Universidad Internacional de La Rioja por Silvia Enma Rodríguez Fernández, en colaboración con el Grupo de Estructura, Dinámica e Interacciones de Proteínas por RMN del Instituto de Química-Física “Blas Cabrera” (IQF-CSIC).

Dentro del grupo de investigación, la colaboración del doctorando Cristian Moisés Segura Rodríguez y el Dr. Douglas V. Laurents ha sido fundamental. Ambos investigadores han proporcionado el marco teórico, los protocolos de referencia empleados en laboratorio y las estructuras modelo utilizadas en la validación experimental. Además, han verificado los resultados generados por el módulo, asegurando su coherencia con el procedimiento manual tradicional. Además, se han encargado de verificar los resultados generados por el módulo, asegurando la coherencia y precisión del análisis automatizado respecto al procedimiento manual original.

Contactos:

* Silvia Enma Rodríguez Fernández ([silviaenma.rodriguez@gmail.com](mailto:silviaenma.rodriguez@gmail.com))
* Cristian M. Segura Rodríguez (csegura@iqf.csic.es)
* Douglas V. Laurents ([dlaurents@iqfr.csic.es](mailto:dlaurents@iqfr.csic.es))

1. **Guía del usuario del módulo PPIIMoL para la detección de hélices PPII en PyMOL**

**Versión del módulo:** PPIIMoL v1.0  
**Fecha de última actualización:** junio de 2025

## Descripción general

Esta guía está pensada para facilitar el uso del módulo PPIIMoL.py a cualquier persona que trabaje con estructuras proteicas en PyMOL, con el fin de detectar hélices de tipo poliprolina II (PPII) en proteínas. La herramienta cuenta con una interfaz gráfica intuitiva, por lo que no es necesario tener conocimientos de programación para utilizarla.

### ¿Qué hace esta herramienta?

Este módulo permite **identificar, analizar y visualizar automáticamente hélices de tipo poliprolina II (PPII)** en proteínas representadas en PyMOL. Calcula los ángulos φ y ψ y aplica criterios geométricos para detectar segmentos compatibles con esta conformación. Además, permite evaluar enlaces de hidrógeno no canónicos entre ellos. Una vez ejecutado, resalta en color las hélices encontradas, **genera selecciones individuales**, y exporta los datos estructurales de las hélices detectadas en formatos .pdb, .txt, .csv. En resumen, permite una **inspección visual y cuantitativa** de dichas estructuras directamente desde la interfaz gráfica del visor molecular.

## Requisitos básicos

### ¿Qué necesito para empezar?

• Tener **PyMOL** instalado (se recomienda versión 2.5 o superior).  
• Tener **Python 3.7 o superior** operativo (probado con Python 3.9.13).

* Disponer del archivo PPIIMoL.py.
* Contar con un archivo .pdb de la proteína a analizar, o conexión a internet para descargarlo de Protein Data Bank.
* Tener instalada la biblioteca tkinter.
* **Recomendado:** usar un entorno gestionado con **Anaconda o Miniconda**, ya que facilita la compatibilidad entre PyMOL, Python y las bibliotecas necesarias.

## Pasos para usar el módulo

### ¿Cómo lo pongo en marcha?

1. Abrir PyMOL.
2. Ejecutar el módulo con el comando:

run ruta\_del\_archivo/PPIIMoL.py

Sustituye ruta\_del\_archivo por la ruta real del script en tu equipo.

Este comando activará automáticamente la interfaz gráfica del módulo, desde la que se podrá realizar todo el análisis sin necesidad de escribir más comandos.

1. Cargar la estructura proteica:

Puedes trabajar con una estructura que ya tengas descargada previamente o acceder directamente a la base de datos PDB, siempre que conozcas el **ID de la proteína** (por ejemplo, 3BOG). Para ello basta con activar el botón correspondiente.

Una vez cargada la proteína, utiliza los botones de la interfaz para iniciar el análisis.

1. Ejecutar el análisis:

Pulsa los diferentes botones de la interfaz para preparar la proteína, y posteriormente el botón **“Detectar hélices PPII”**. El módulo calculará los ángulos φ y ψ, detectará regiones compatibles con PPII y generará visualizaciones y archivos con los datos de dicho análisis.

### ¿Qué resultados ofrece?

* **Visualización directa en PyMOL:**  
  Las hélices detectadas se muestran en rojo (configurable).
* **Selecciones creadas automáticamente:**  
  Se crea un objeto con cada hélice y se nombra como: ppii\_segmento\_[cadena]\_[inicio]\_[fin]
* **Archivos exportados:**
  + .pdb: hélices seleccionadas.
  + .txt y .csv: ángulos y residuos detectados

## Visualización e interpretación

### ¿Cómo interpreto los resultados?

* Las hélices detectadas se visualizan con un color específico, fácilmente configurable en el código. Se escogió el rojo por defecto, porque no se utiliza en estos análisis para otro propósito.
* Las selecciones se nombran como ppii\_segmento\_[cadena]\_[inicio]\_[fin], que indica la cadena a la que pertenece el segmento, y el primer y último aminoácido que forma parte de la hélice. Esto permite aislar visualmente las diferentes regiones encontradas y analizarlas como cualquier otra selección de PyMOL.
* Desde la interfaz puedes hacer zoom, aislar regiones, hacer selecciones y ajustar la vista para facilitar la inspección y hacer cualquier tipo de análisis posterior en PyMOL a partir de los resultados.
* Mediante la extracción de los csv también pueden ser exportados a otras aplicaciones para posteriores análisis.

## Opciones avanzadas

### ¿Puedo personalizarlo?

Sí, el módulo permite:

* Modificar el color directamente en el script.
* Ajustar los valores umbral para los ángulos φ y ψ.
* Modificar el estilo visual de las hélices detectadas (estilo, grosor, etc.).
* El código puede modificarse fácilmente para ajustar parámetros o incluir la detección de otras estructuras secundarias si el usuario tiene conocimientos básicos de Python.
* Modificar el formato de salida o la lógica de selección.

## En caso de error

### ¿Qué hacer si algo no funciona?

* Verifica que el módulo ha sido cargado correctamente en PyMOL. Si no se carga el módulo, revisa la ruta al archivo y asegúrate de que termina en .py
* Asegúrate de que la estructura .pdb esté visible en pantalla.
* Si usas Conda, activa previamente el entorno correspondiente:

conda activate nombreDelEntorno.

* Para configuraciones técnicas detalladas, consulta el Anexo B (Guía Técnica).

## Soporte al usuario

### ¿A quién puedo contactar si tengo dudas?

Para consultas sobre el uso del módulo:

Contacto: [silviaenma.rodriguez@gmail.com](mailto:silviaenma.rodriguez@gmail.com)

Esta guía ofrece una visión general del uso del módulo. Para personalizaciones más avanzadas, se recomienda consultar el código fuente comentado o contactar directamente con la autora.

1. **Documentación técnica: Pseudocódigo estructurado del algoritmo principal**

|  |
| --- |
| **INICIAR módulo de análisis PPII**   1. CREAR carpeta de resultados en el sistema (Resultados\_PPIIMoL). 2. CARGAR archivo .pdb o DESCARGAR desde el PDB   → Mostrar en PyMOL en modo "licorice   1. ELIMINAR solventes 2. AÑADIR hidrógenos 3. SEPARAR las cadenas en objetos independientes 4. PARA cada cadena: a. CALCULAR ángulos φ y ψ de todos los residuos b. GUARDAR en archivo y a­­macenar internamente:   → Número de residuo (resi)  → Tipo de residuo (resn)  → φ y ψ en grados   1. DETECTAR hélices PPII: a. DEFINIR rango aceptado para PPII:   → φ ∈ [–85°, –65°]  → ψ ∈ [135°, 155°]  b. IDENTIFICAR tramos de ≥ **3 residuos consecutivos** que cumplan los valores c. GUARDAR tramos como hélices candidatas (objeto en PyMOL).   1. PARA cada hélice candidata: a. CREAR nuevo objeto PyMOL con esos residuos b. COLOREAR con un color distinto 2. EXPORTAR segmentos PPII como archivos .pdb individuales 3. IDENTIFICAR átomos implicados en enlaces de hidrógeno: a. Seleccionar Cα y O entre segmentos colindantes b. Medir distancia entre cada par Cα–O   → SI distancia < 5 Å   * + - * → AÑADIR a la lista de pares candidatos  1. C**ALCULAR** ángulo **Cα–H···O** para cada par candidato:    * SI ángulo ∈ [110°, 180°]: (Este rango es ajustable)      + MARCAR como posible enlace de hidrógeno no canónico 2. . VISUALIZAR las distancias y ángulos detectados en PyMOL mediante líneas de conexón. 3. GENERAR tabla de salida Para cada hélice:   →Rango de residuos  → φ y ψ por residuo  → Distancias Cα–O válidas  → Ángulos H–O válidos   1. GUARDAR resultados en archivo .txt, .pdb y.csv con distancias y ángulos geométricos válidos 2. MOSTRAR resumen del análisis en la interfaz gráfica 3. **ALMACENAR** automáticamente todos los archivos generados en la carpeta Resultados\_PPIIMoL   **FINALIZAR** |

## Diagrama lógico del algoritmo de detección de hélices PPII (PPIIMoL)

El diagrama de flujo general que resume el funcionamiento del módulo desarrollado para la detección automatizada de hélices de tipo poliprolina II (PPII) en estructuras proteicas. Este esquema refleja las principales etapas del proceso, desde la carga y preparación del archivo PDB hasta la identificación de posibles enlaces de hidrógeno no canónicos asociados.

Figura 31. Diagrama lógico del algoritmo de detección de hélices PPII (PPIIMoL)

Diagrama

Descripción generada automáticamente

1. **Ejemplos de salidas generadas por el módulo PPIIMoL**

Este anexo recoge ejemplos representativos de los archivos generados automáticamente por el módulo de detección de hélices PPII desarrollado en Python para PyMOL. Cada uno de estos archivos forma parte del sistema de salida estructurada que permite registrar los resultados obtenidos durante el análisis, así como exportarlos para su revisión o procesamiento posterior.

## E.1. Archivo angulos\_phi\_psi.csv

Contiene los valores calculados de los ángulos diedros φ (phi) y ψ (psi) para cada residuo, organizados por cadena. Este archivo sirve de base para la identificación de tramos compatibles con la geometría característica de las hélices PPII.

Ejemplo de contenido:

Figura 32 archivo angulos\_phi\_psi\_7JJV

Tabla

Descripción generada automáticamente

## E.2. Archivo reporte\_ppii.csv

Incluye los segmentos identificados como hélices PPII, indicando para cada uno de ellos los residuos implicados, su cadena, y los valores φ/ψ medios asociados. Representa la salida final del algoritmo de detección.

Ejemplo de contenido:

Figura 33. Archivo Reporte\_ppii\_7JJV

Tabla

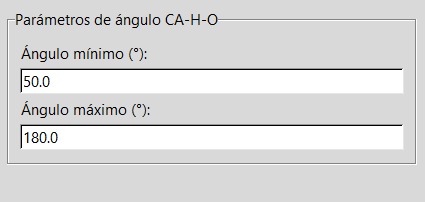
Descripción generada automáticamente

## E.3. Archivo angulos\_ca\_h\_o.txt

Este archivo recoge los ángulos formados entre el carbono alfa (Cα), el hidrógeno enlazado y el oxígeno carbonílico de residuos cercanos. Estos valores permiten estimar la viabilidad geométrica de enlaces de hidrógeno no canónicos del tipo Cα–H···O, presentes en algunas regiones de hélice PPII.

En la versión actual del módulo, los umbrales angulares utilizados se corresponden con criterios recogidos en la literatura. Sin embargo, estos valores pueden modificarse fácilmente desde la interfaz gráfica, lo que permite al usuario adaptar el análisis a diferentes contextos estructurales.

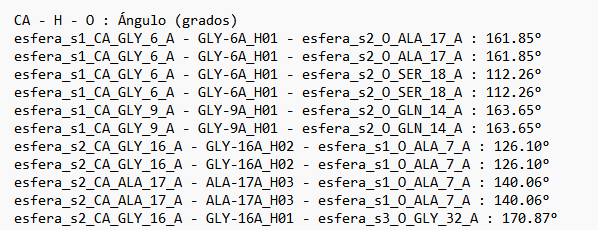
Figura 34 Interfaz gráfica del módulo para ajustar los umbrales angulares CA–H···O.



Con fines ilustrativos, se ha generado una salida alternativa ampliando el rango de detección a ángulos entre 50° y 180°, lo cual permite observar cómo influye este cambio en la selección de enlaces. Este ajuste se realizó mediante el siguiente cuadro de entrada:

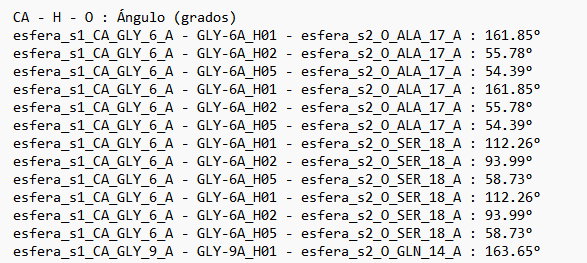
* Figura 33. Distribución de ángulos Cα–H···O generada con los valores por defecto.

Figura 35. fichero ángulos\_ca\_h\_o\_7JJV.txt



* Figura 34. Distribución obtenida tras ajustar el rango angular a 50°–180°, únicamente para fines de ejemplo.

Figura 36. fichero ángulos\_ca\_h\_o\_7JJV.txt resultado del cambio de umbral



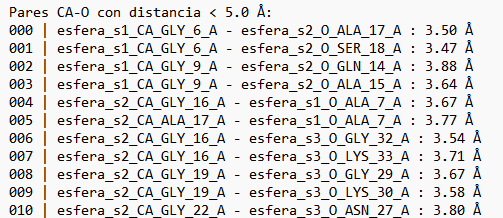
Este tipo de modificación puede resultar útil durante la fase de exploración o ajuste fino del algoritmo, ya que permite **adaptar el comportamiento del módulo a diferentes tipos de moléculas** o contextos estructurales concretos. No obstante, **no forma parte del funcionamiento estándar del sistema** y debe aplicarse con criterio por parte del investigador, valorando la relevancia de los resultados en función del caso de estudio.

## E.4. Archivo distancias\_colindantes.txt

Muestra las distancias entre átomos clave (Cα, H y O) de residuos potencialmente implicados en enlaces de hidrógeno no canónicos. Estas medidas permiten evaluar qué tripletas cumplen con los criterios espaciales definidos por el usuario.

Ejemplo de contenido:

Figura 37. Archivo distancias colindantes 7JJV



Estos archivos, junto con la visualización en PyMOL, constituyen el conjunto completo de resultados que el módulo ofrece al investigador para facilitar un análisis estructural más preciso, sistemático y reproducible.

**Índice de acrónimos**

|  |  |
| --- | --- |
| **Acrónimo** | **Significado** |
| Anaconda | Distribución de Python para ciencia de datos y desarrollo científico |
| API | Application Programming Interface (Interfaz de Programación de Aplicaciones) |
| Biopython | Librería de Python para bioinformática estructural y análisis de datos |
| cmd | Comando interno de PyMOL (pymol.cmd) |
| Conda | Gestor de entornos y paquetes incluido en Anaconda |
| CSIC | Consejo Superior de Investigaciones Científicas |
| CSV | Comma-Separated Values (valores separados por comas) |
| DSSP | Define Secondary Structure of Proteins |
| GPLv3 | GNU General Public License versión 3 |
| GUI | Graphical User Interface (Interfaz Gráfica de Usuario) |
| IQF-CSIC | Instituto de Química-Física “Blas Cabrera” – CSIC |
| MDTraj | Molecular Dynamics Trajectories (librería para análisis de dinámica molecular) |
| NMR | Nuclear Magnetic Resonance (Resonancia Magnética Nuclear) |
| PDB | Protein Data Bank |
| PPII | Hélice de poliprolina tipo II |
| PyMOL | Python Molecular Graphics Tool |
| **Acrónimo** | **Significado** |
| PyRosetta | Interfaz de Python para el paquete de modelado estructural Rosetta |
| TFG | Trabajo de Fin de Grado |
| Tkinter | Toolkit interface (Interfaz de desarrollo de GUIs en Python) |
| UCSF | University of California, San Francisco |
| VMD | Visual Molecular Dynamics |
| VS Code | Visual Studio Code |