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: | 09/04/2025 |
| Repositorio del código fuente: |  |

**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.

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 apoyo a investigaciones en curso en el Instituto de Química-Física "Blas Cabrera" del CSIC, derivado del protocolo experimental empleado y toma como referencia el artículo “*Architectonic Principles of Polyproline II Bundle Protein Domains*” (Segura Rodríguez & Laurents, 2024). 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 la subjetividad 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 han utilizado proteínas modelo como la bien conocida 3BOG, 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 the experimental protocol used and based on the article "Architectonic Principles of Polyproline II Bundle Protein Domains" (Segura Rodríguez & Laurents, 2024). 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 such as the well-known 3BOG 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.

Índice de contenidos

[1. Introducción 1](#_Toc197625471)

[1.1. Motivación 2](#_Toc197625472)

[1.2. Planteamiento del trabajo 3](#_Toc197625473)

[1.3. Estructura del trabajo 3](#_Toc197625474)

[2. Contexto y Estado del Arte 5](#_Toc197625475)

[2.1. Análisis del contexto 5](#_Toc197625476)

[2.1.1. El problema: detección de hélices PPII 5](#_Toc197625477)

[2.1.2. Método manual de hallar PPII con PyMOL 7](#_Toc197625478)

[2.2. Estado del arte 27](#_Toc197625479)

[2.2.1. Comparativa de herramientas existentes para análisis estructural 27](#_Toc197625480)

[2.2.2. Elección de PyMOL 28](#_Toc197625481)

[3. PyMOL como Herramienta para el Análisis Estructural 31](#_Toc197625482)

[3.1. Introducción 31](#_Toc197625483)

[3.2. ¿Qué es PyMOL? 31](#_Toc197625484)

[3.3. Principales funcionalidades utilizadas en este proyecto 31](#_Toc197625485)

[3.4. Procedimiento manual utilizado actualmente en el laboratorio 33](#_Toc197625486)

[3.5. Ejemplo práctico: uso manual de PyMOL sobre la proteína modelo 3BOG 34](#_Toc197625487)

[4. Objetivos y metodología de trabajo 36](#_Toc197625488)

[4.1. Objetivo general 36](#_Toc197625489)

[4.2. Objetivos específicos 36](#_Toc197625490)

[4.3. Metodología de trabajo 37](#_Toc197625491)

[4.4. Contribución de la autora 39](#_Toc197625492)

[4.5. Colaboraciones y contexto experimental 39](#_Toc197625493)

[5. Desarrollo del módulo (ligar con los objetivos del punto 4) 41](#_Toc197625494)

[5.1. Requisitos funcionales y técnicos del sistema 41](#_Toc197625495)

[5.2. Diseño del algoritmo de detección PPII 42](#_Toc197625496)

[5.3. Estructura del código y funciones principales 43](#_Toc197625497)

[5.4. Proceso de integración con PyMOL 45](#_Toc197625498)

[5.5. Validación sobre la proteína modelo 3BOG 46](#_Toc197625499)

[5.6. Estructura técnica de la solución 47](#_Toc197625500)

[5.7. Casos de uso representativos 48](#_Toc197625501)

[6. Resultados y evaluación 50](#_Toc197625502)

[5.1. Resultados obtenidos: detección de hélices PPII en 3BOG 50](#_Toc197625503)

[6.2. Comparación con el análisis manual 51](#_Toc197625504)

[6.3. Valoración de precisión y eficiencia 52](#_Toc197625505)

[7. Conclusiones y trabajo futuro 56](#_Toc197625506)

[7.1. Conclusiones del trabajo 56](#_Toc197625507)

[7.2. Aportaciones del módulo 57](#_Toc197625508)

[7.3. Propuestas de mejora y líneas futuras 57](#_Toc197625509)

[Tabla resumen de tareas 59](#_Toc197625510)

[6. Conclusiones y trabajo futuro 61](#_Toc197625511)

[6.1. Conclusiones del trabajo 61](#_Toc197625512)

[6.2. Líneas de trabajo futuro 61](#_Toc197625513)

[Referencias bibliográficas 62](#_Toc197625514)

[Anexo A. Título del anexo. 66](#_Toc197625515)

[Índice de acrónimos 67](#_Toc197625516)

Índice de figuras

[Figura 1. ángulo diedro Phi. 5](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625517)

[Figura 2. ángulo diedro Psi. 6](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625518)

[Figura 3. Obtención de la proteína en PyMOL 7](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625519)

[Figura 4. Molécula 3BOG (PDB). 8](#_Toc197625520)

[Figura 5. Eliminación de Moléculas de Agua 8](#_Toc197625521)

[Figura 6. Adición de hidrógenos. 9](#_Toc197625522)

[Figura 7. Abriendo la Secuencia. 9](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625523)

[Figura 8. Visualización de la Secuencia en PyMOL. 10](#_Toc197625524)

[Figura 9. Hélices vistas en paralelo. 11](#_Toc197625525)

[Figura 10. Visualización Esquematizada de 3BOG. 11](#_Toc197625526)

[Figura 11. tabla de densidad del empaquetamiento de proteínas. 12](#_Toc197625527)

[Figura 12. Menú de Selección 12](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625528)

[Figura 13. Seleccionando la Cadena. 13](#_Toc197625529)

[Figura 14. Separación de Cadena. 14](#_Toc197625530)

[Figura 15. obj01 en solitario. 14](#_Toc197625531)

[Figura 16. Separación y Coloreado de las hélices seleccionadas. 16](#_Toc197625532)

[Figura 17. Hélices Identificadas. 17](#_Toc197625533)

[Figura 18. Dando nombre a las hélices. 17](#_Toc197625534)

[Figura 19. Visualización de hélices en modo “regaliz”. 18](#_Toc197625535)

[Figura 20. Ángulos Phi, Psi y Omega. 19](#_Toc197625536)

[Figura 21. Activando la herramienta Wizard 19](#_Toc197625537)

[Figura 22. Modo Diedros 20](file:///D:\Documentos\adaptacion%20al%20grado\TFG\RODIGUEZ_FERNANDEZ_SE_TFG_1BORRADOR_REV_Corr.docx#_Toc197625538)

[Figura 23. Phi y Psi en PyMOL. 20](#_Toc197625539)

[*Figura 24. Selección de c-alphas.* 21](#_Toc197625540)

[Figura 25. selección de oxígenos. 22](#_Toc197625541)

[Figura 26. Posibles localizaciones de puentes de hidrógeno no canónicos*.* 23](#_Toc197625542)

[Figura 27. Posible migraña inducida por mala praxis. 24](#_Toc197625543)

[Figura 28. Selección de ángulos en Wizard. 24](#_Toc197625544)

[Figura 29. Cribado de los posibles puentes de hidrógeno no canónicos. 25](#_Toc197625545)

Índice de tablas

[Tabla 1. Comparativa visores moleculares y su utilidad 28](#_Toc197625546)

[Tabla 2. Comparativa de herramientas respecto al proyecto de detección PPII 30](#_Toc197625547)

[Tabla 3. Índice de moléculas de referencia 38](#_Toc197625548)

[Tabla 4. Casos de uso representativos del módulo de detección de hélices PPII 49](#_Toc197625549)

# 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. Se ha de resaltar que un archivo de este tipo contiene la imagen de 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.

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.

La motivación de este trabajo surge 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 representa también un desafío técnico interesante en el contexto de un trabajo de fin de grado.

## 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 realizará empleando herramientas ya integradas en el entorno habitual de trabajo, como PyMOL, integrándose en Visual Studio 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, ideal para verificar la eficacia del algoritmo propuesto y su correcta 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 trabajo se ha organizado 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 Visual Studio 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

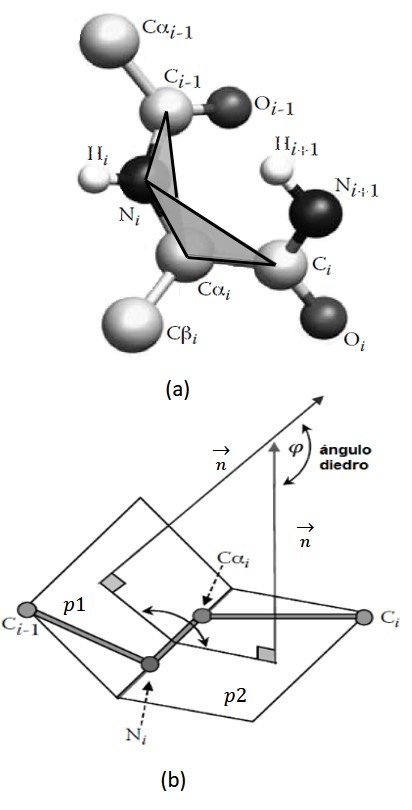
Al comienzo de un estudio de investigación se realiza el cribado de las moléculas que se quieren caracterizar. En este caso, se seleccionan moléculas que potencialmente tengan PPII y que participen en las funciones objeto de análisis, **en función de los objetivos específicos del estudio**. La literatura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

Figura 1. ángulo diedro Phi.

Las proteínas están formadas por cadenas de aminoácidos unidas entre sí. Estas cadenas tienen una especie de "esqueleto" 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.

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.

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

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

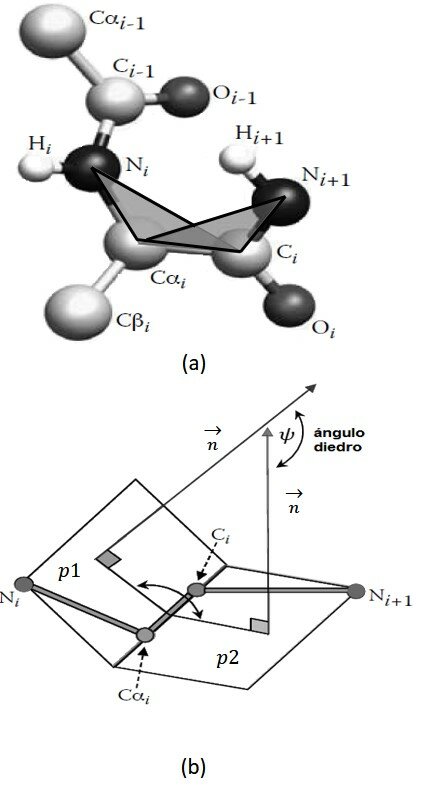
* El ángulo **φ (phi)**, que gira alrededor del enlace entre el nitrógeno (N) y el carbono alfa (Cα).

Figura 2. ángulo diedro Psi.

* 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.

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.

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

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

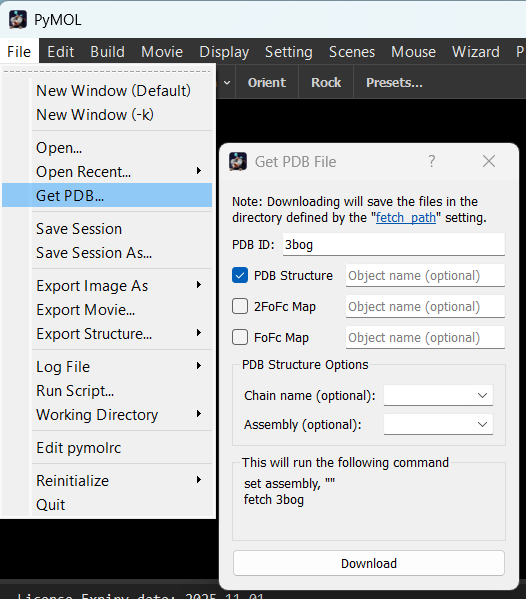
Este proyecto surge directamente del trabajo de investigación recogido en el artículo "Architectonic Principles of Polyproline II Bundle Protein Domains" firmado por Cristian Moisés Segura Rodríguez y Douglas Vison Laurents. En él se describen las características estructurales de los haces de hélices PPII ricas en glicina, incluyendo parámetros geométricos específicos, patrones de enlace y distribución espacial. El algoritmo que aquí se desarrolla toma como referencia esos principios para automatizar su detección y facilitar el trabajo de quienes investigan estructuras similares. El artículo no solo sirve como base teórica, sino que orienta directamente el diseño del sistema y la validación del mismo a partir 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:

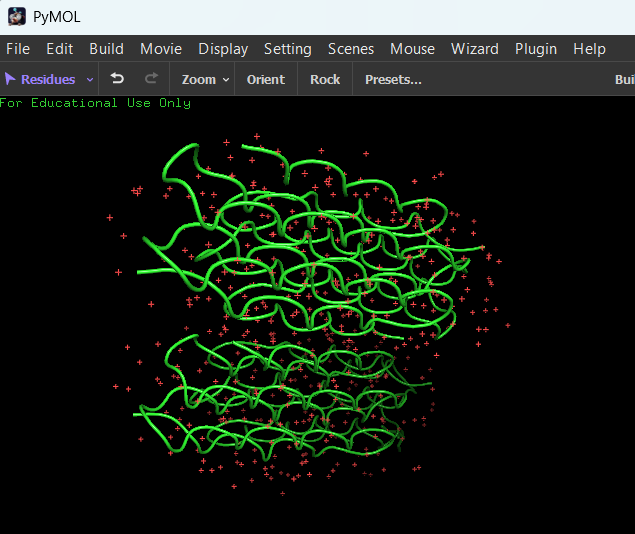
mediante PyMOL.

Figura 3. Obtención de la proteína en 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).

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 puede personalizarse según las necesidades del análisis. (Figura 4).

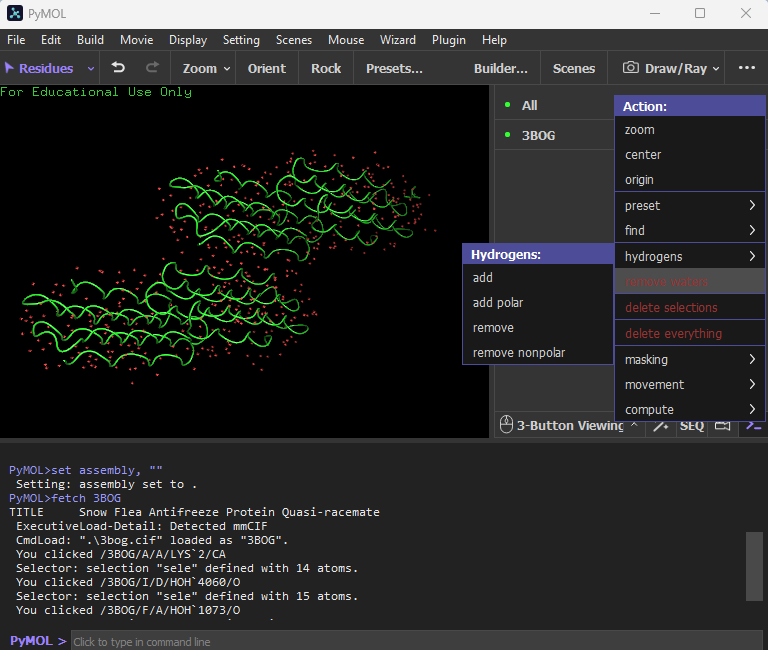
Figura 4. Molécula 3BOG (PDB).



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).

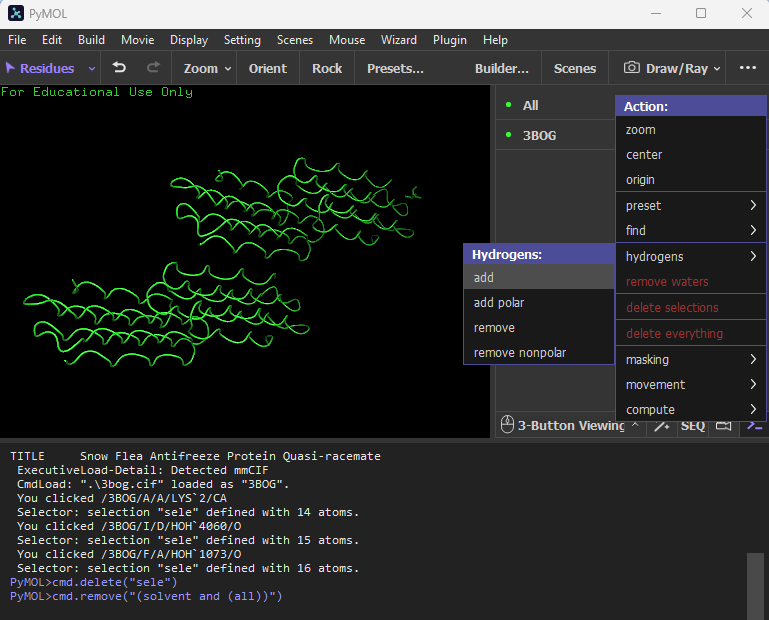
Figura 5. Eliminación de Moléculas de Agua



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

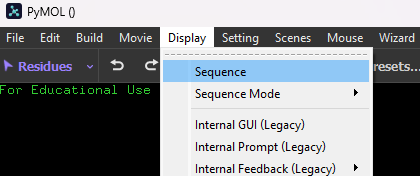
Nota. En otras estructuras se utiliza add polar hydrogens para evitar distorsiones, pero para detectar hélices PPII es recomendable añadir tanto los polares como los apolares.

Figura 6. Adición de hidrógenos.



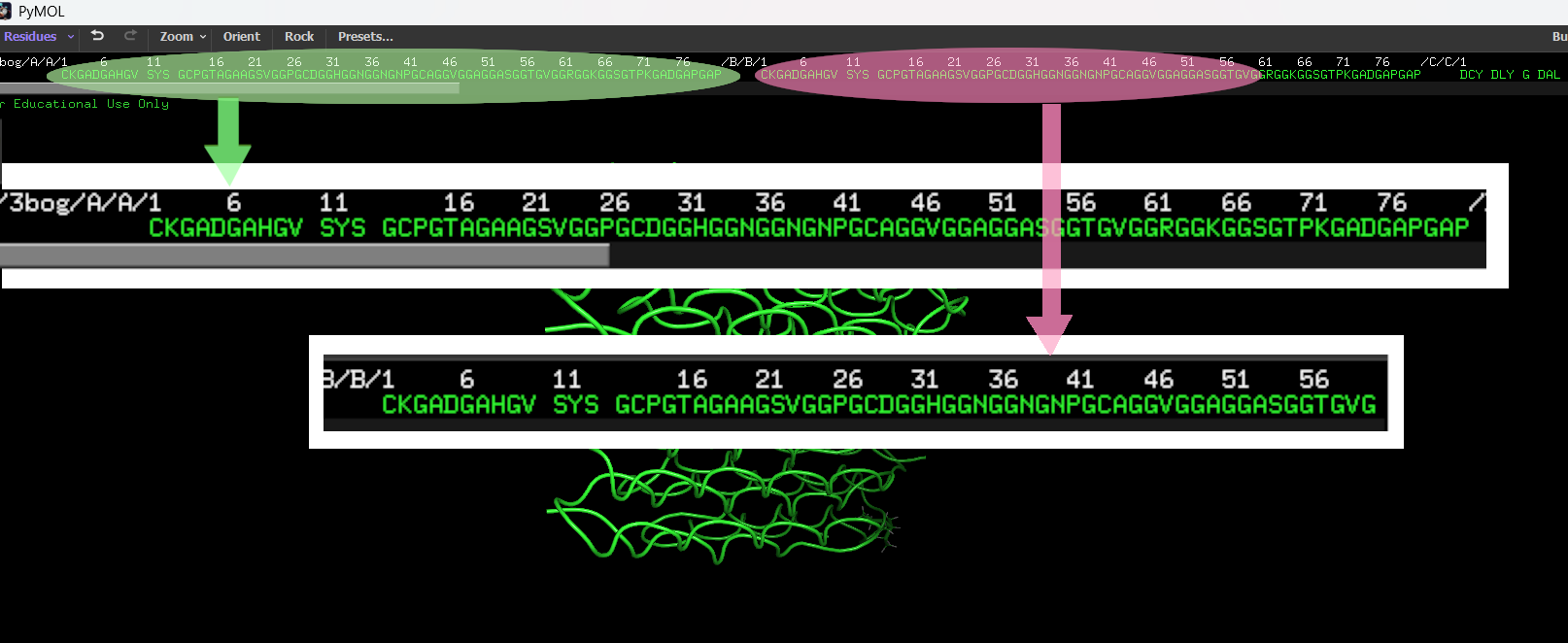
1. **Visualización de la secuencia de aminoácidos**. Se habilita la visualización de la secuencia mediante *Display → Sequence*, permitiendo seleccionar residuos específicos desde la interfaz gráfica. (Figura 7).

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.

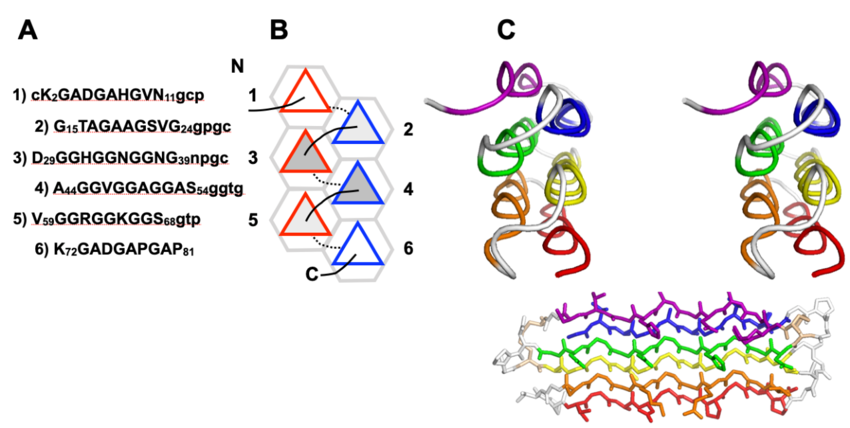
Figura 9. Hélices vistas en paralelo.



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. Visualización Esquematizada de 3BOG.



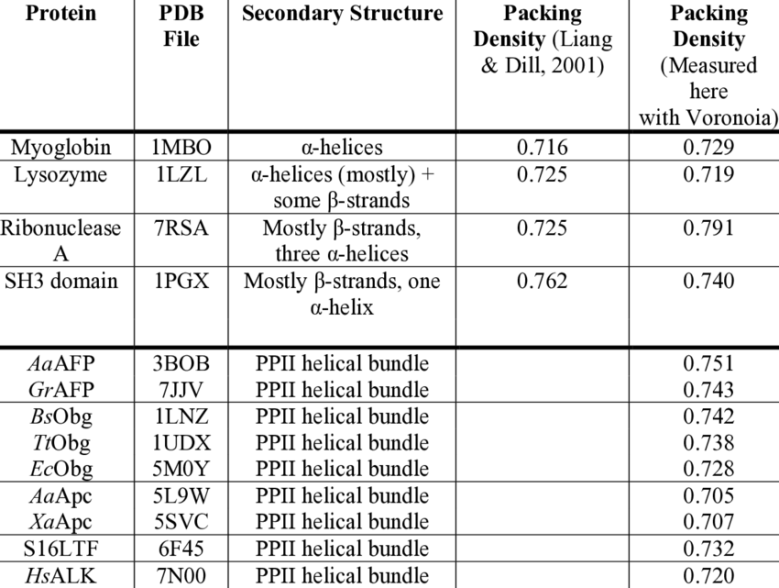
Nota. *Reproducido de (Segura Rodríguez & Laurents, 2024).*

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.

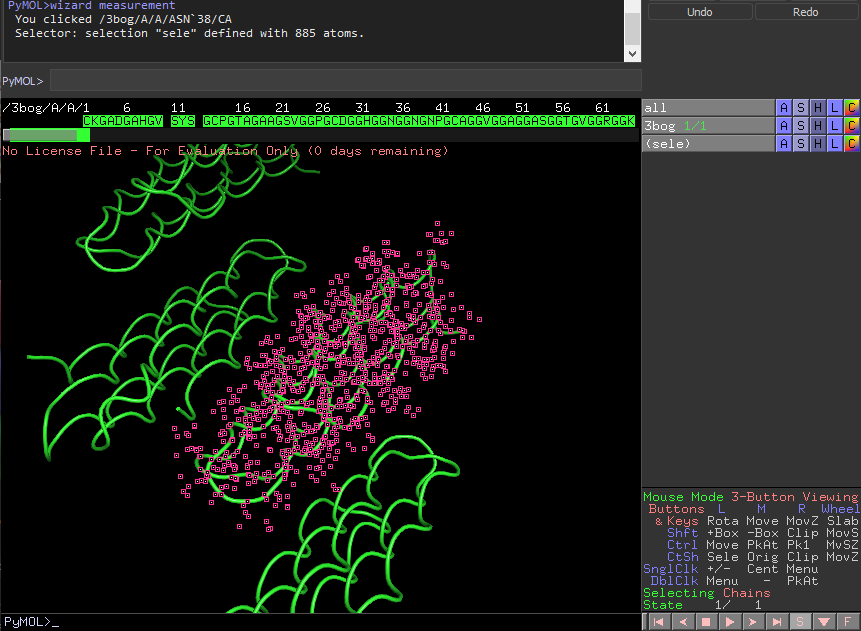


Nota. *Reproducido de (Segura Rodríguez & Laurents, 2024).*

**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. Según representa la Figura 13.

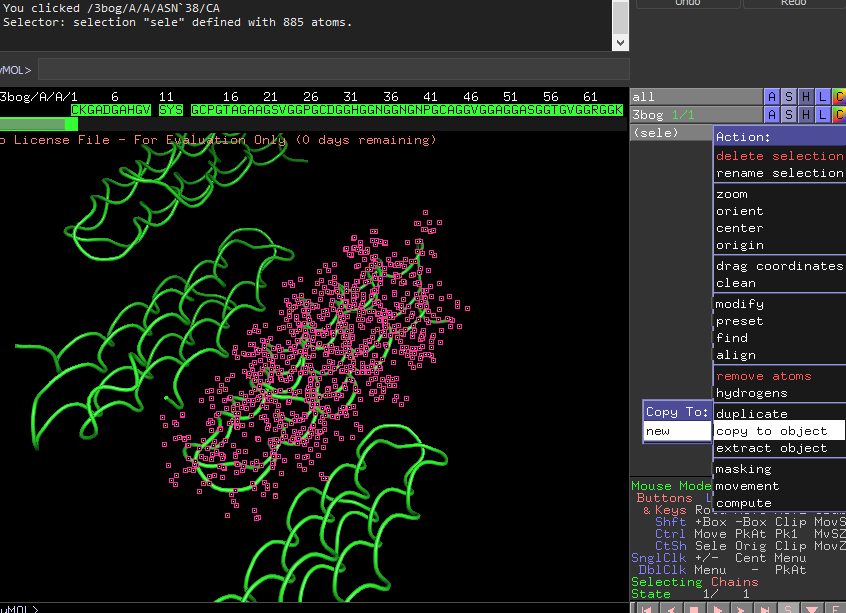
Figura 12. Menú de Selección

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:

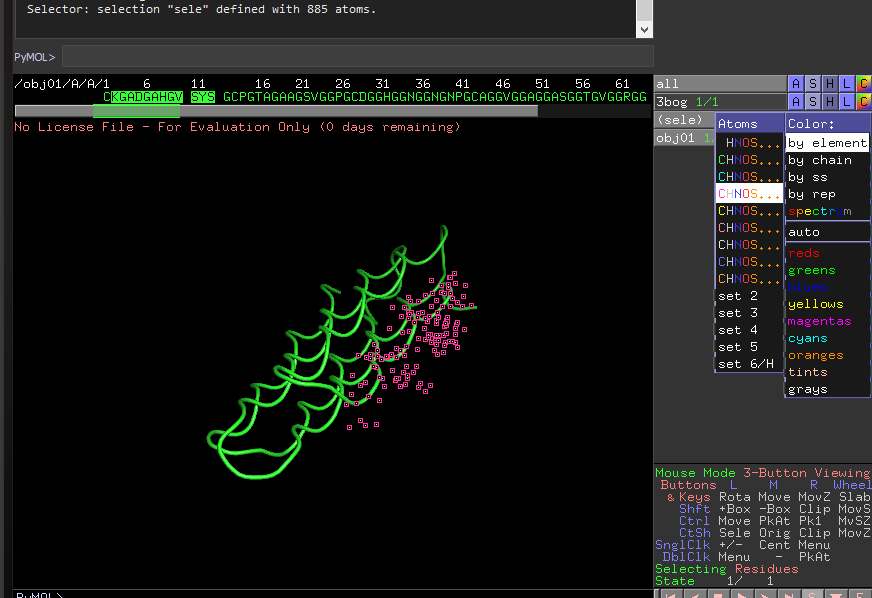
* Desde K2 a N11
* Desde G15 a G24
* Desde D29 a G39
* Desde A44 a S54
* Desde V59 a S68
* Desde K72 a P81

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 (ácido aspártico), 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.

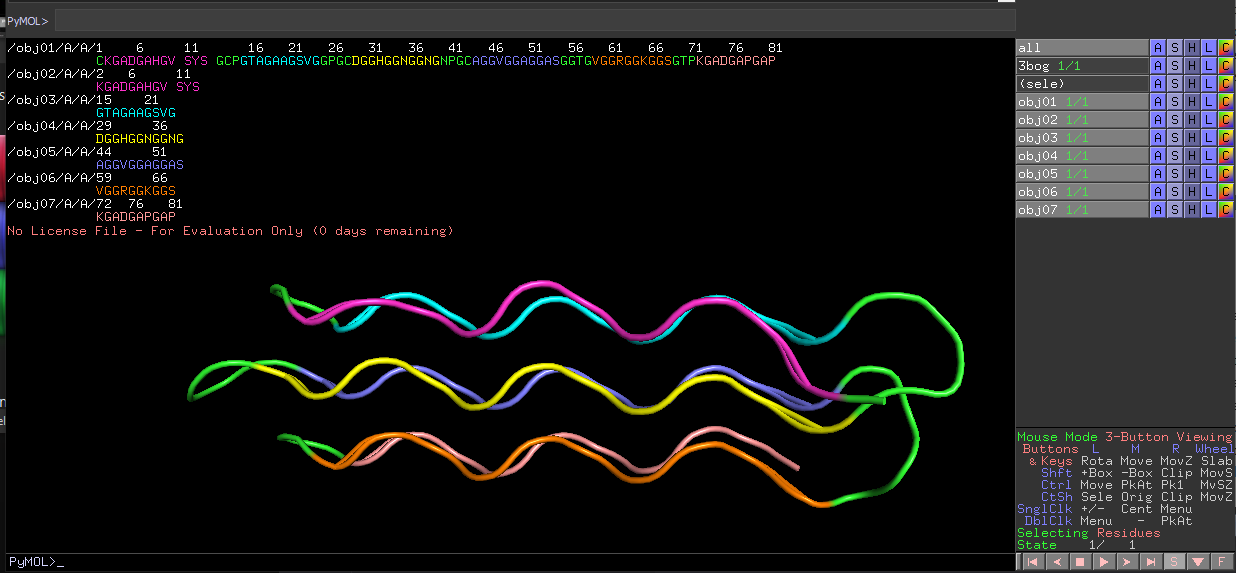


Se utiliza la opción “by element” para aplicar el cambio de color únicamente a los carbonos, de forma que los demás átomos puedan identificarse fácilmente en pasos posteriores. Es recomendable escoger un color distinto al utilizado originalmente para los carbonos del resto de la molécula, a fin de facilitar 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.

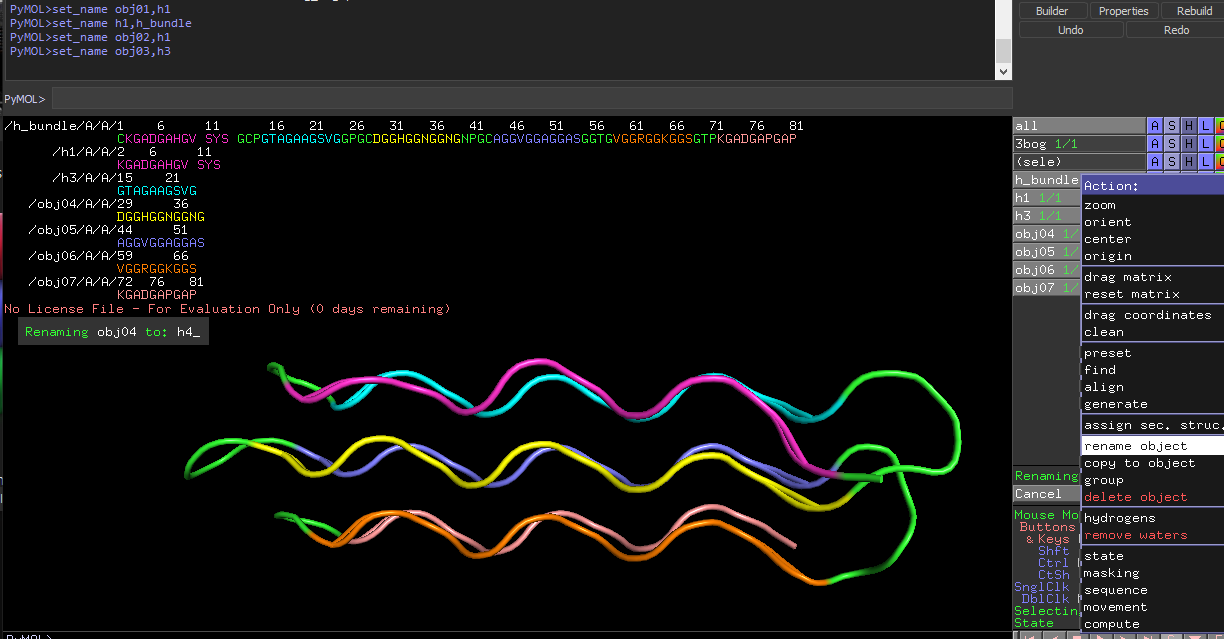
Cabe señalar que, al buscar hélices en nuevos archivos de proteínas, normalmente no se realiza esta separación en objetos hasta haber medido los ángulos phi (φ) y psi (ψ) en el paso 7, ya que la asignación de colores en las etapas iniciales es solo orientativa. Sin embargo, dado que en este caso se trabaja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

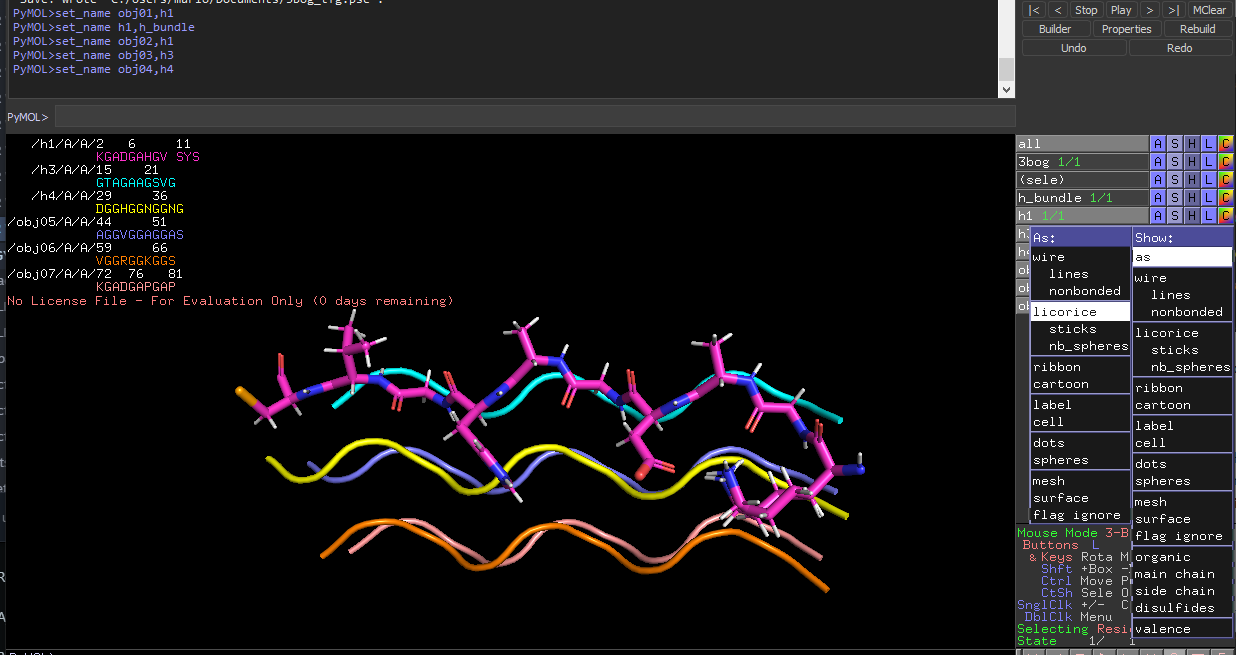
Figura 18. Dando nombre a las hélices.



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”.



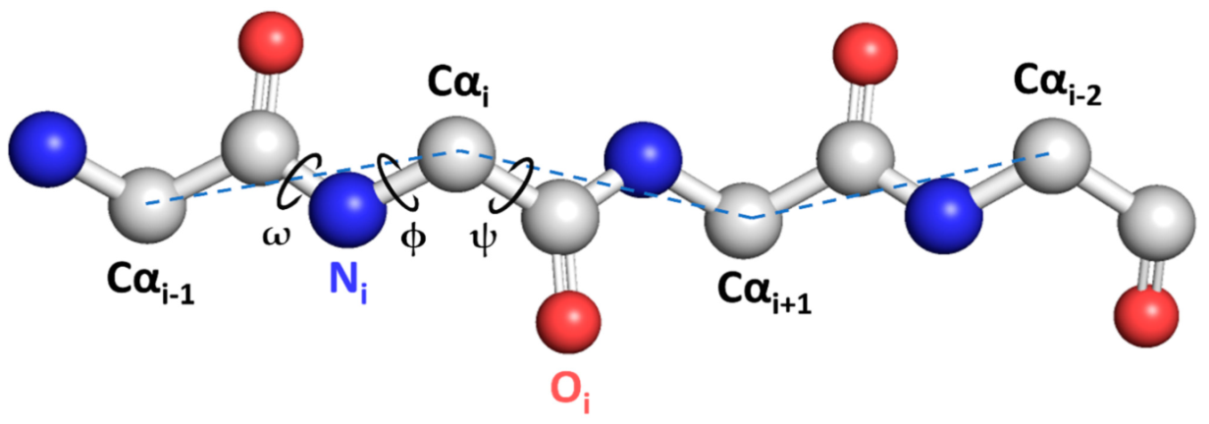
Si se desea volver al modo de visualización anterior, el procedimiento es similar, seleccionando “ribbon” o “cartoon” en lugar de “licorice”. En el estudio la diferencia entre estos modos no resulta relevante para el análisis. Es importante señalar que, si no se utiliza la opción as y se activa directamente “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.



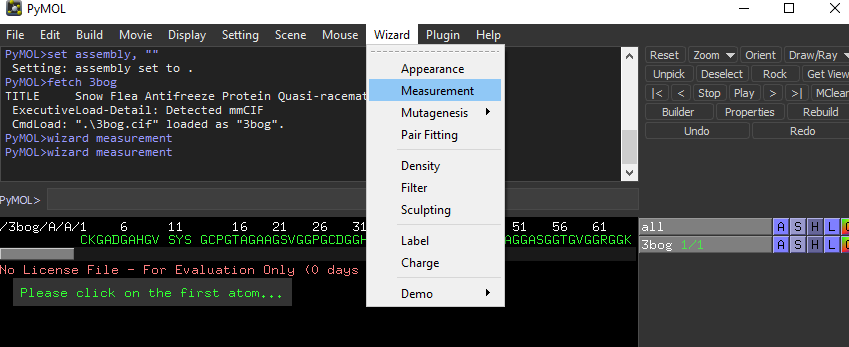
Fuente: Molecules, (Broz, Jukič, & Bren, 2023)

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

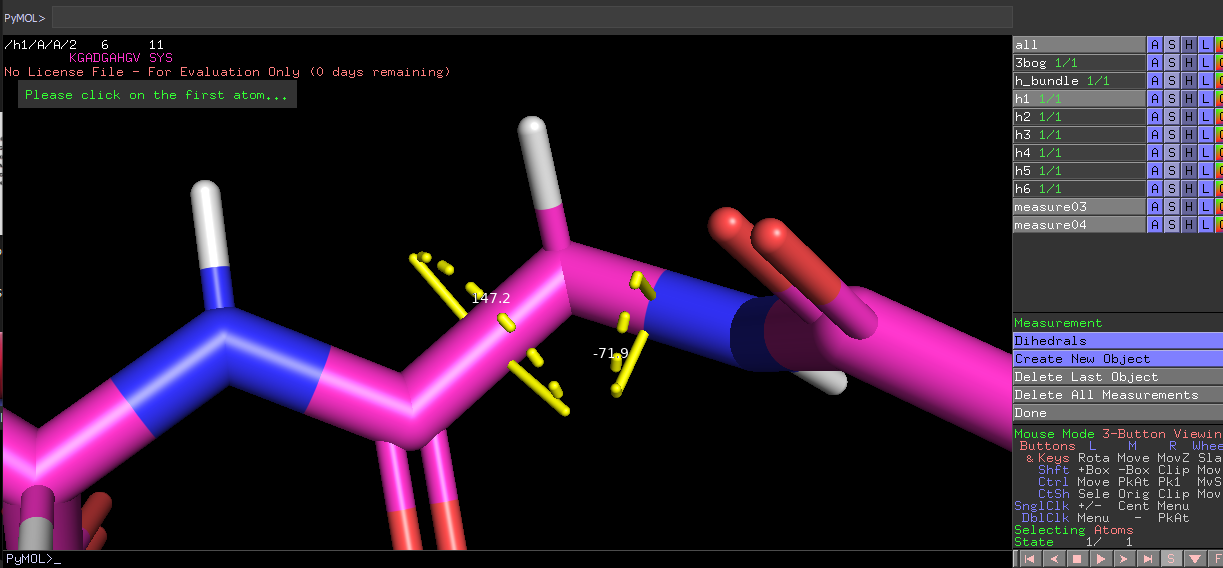


Esto abre el menú wizard en la esquina inferior izquierda. Conviene asegurarse que esté en modo diedro con “Measurement Mode > Dihedrals”. La opción “Angles” indica otro tipo de ángulos que se usará para los puentes de hidrógeno h-bonds. “Measurement Mode” es siempre la primera pestaña del wizard, e 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.



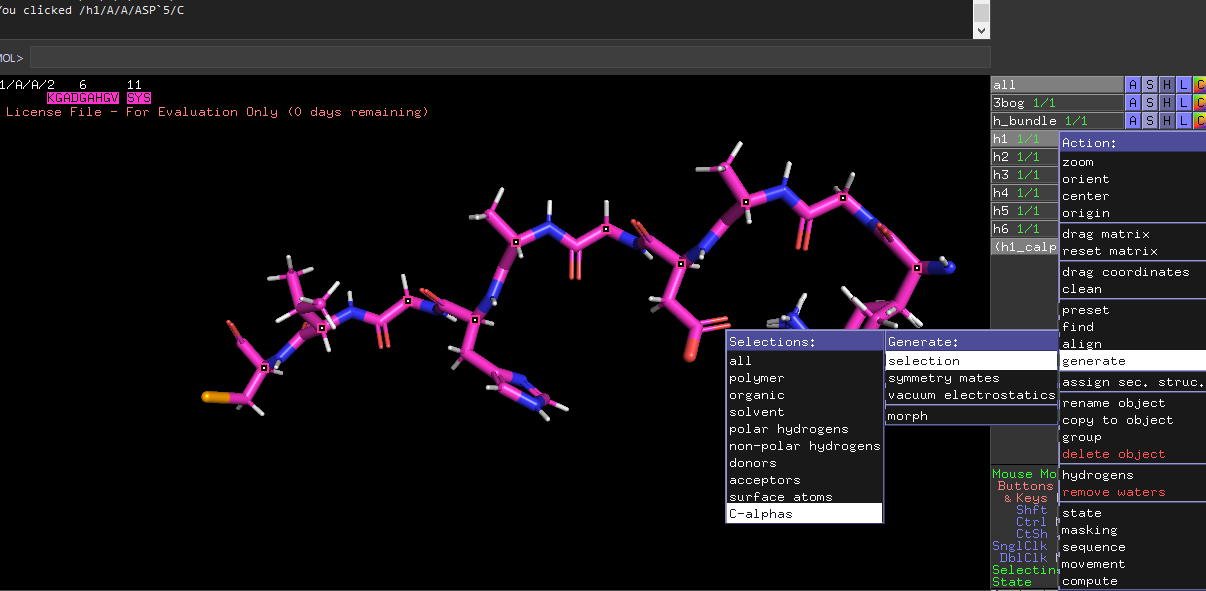
Es fundamental respetar el orden de selección de los átomos, de un extremo a otro de la cadena. Aunque el sentido (izquierda-derecha o derecha-izquierda) no afecta el resultado, un error en el orden de los clics puede provocar una interpretación incorrecta de los puntos espaciales, generando valores 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 hay 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.

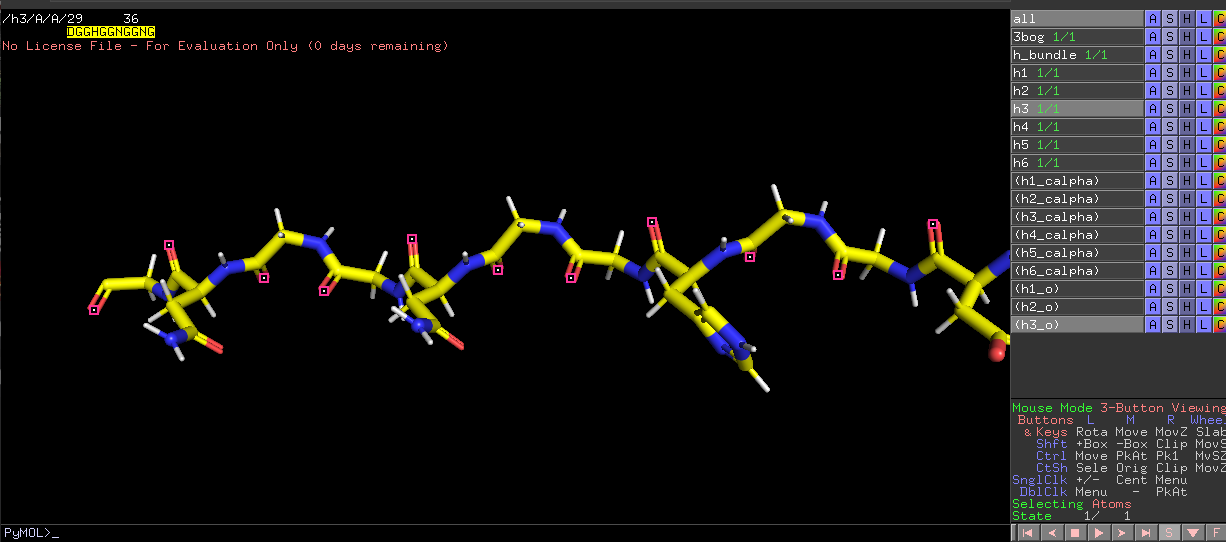
Para este paso se requiere la identificación de los carbonos alfa (cuyos hidrógenos son los donantes de electrones en este enlace) y los oxígenos de carbonilos de la cadena central, que realizan la función de aceptores. Encontrar los es fácil porque ya existe un comando que lo realiza en PyMOL. Tan solo 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

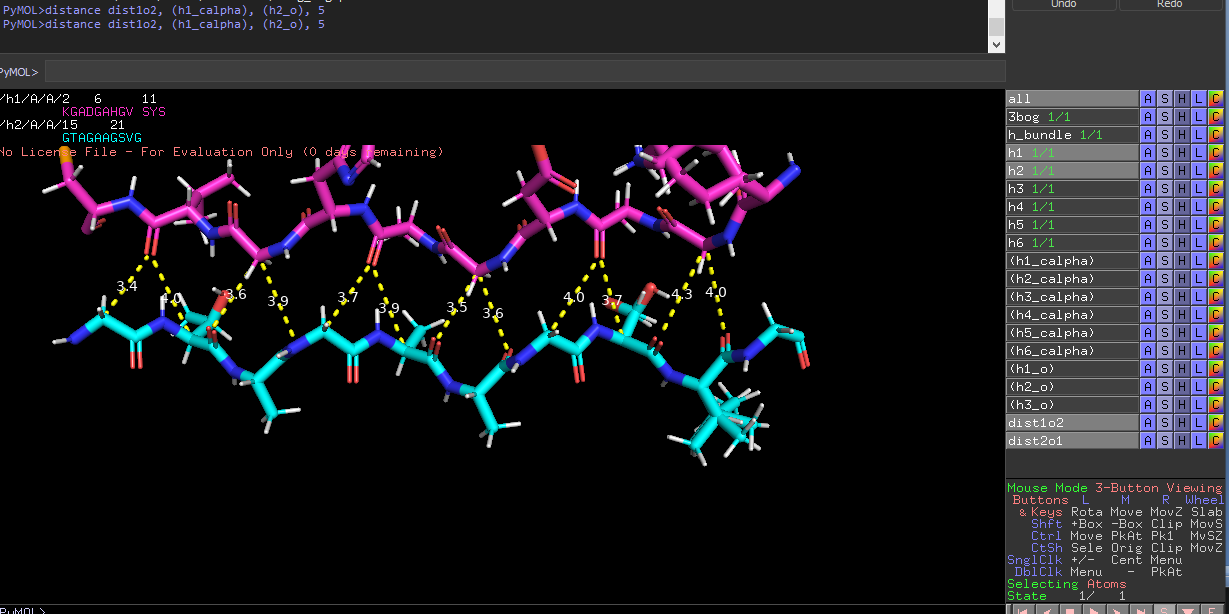
en que distxoy es el nombre que le dará PyMOL al nuevo objeto que almacena las distancias calculadas. hx\_calpha y hy\_o son las selecciones de c-alpha y oxigeno de las dos cadenas, y 5 es el punto de corte de distancia (en angstroms), a partir del cual PyMOL dejará de medir. Se recomienda utilizar 5 **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

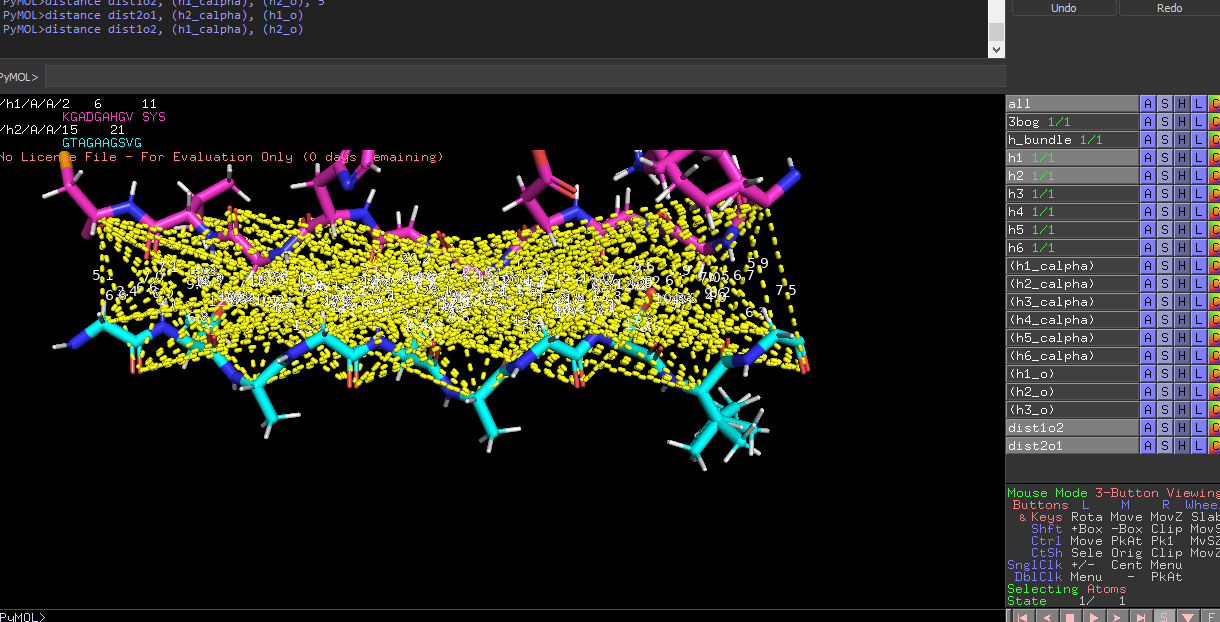
La correcta aplicación de estos comandos genera la figura 26, una representación limpia y clara de las posibles localizaciones de puentes de hidrógeno no canónicos.

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.**

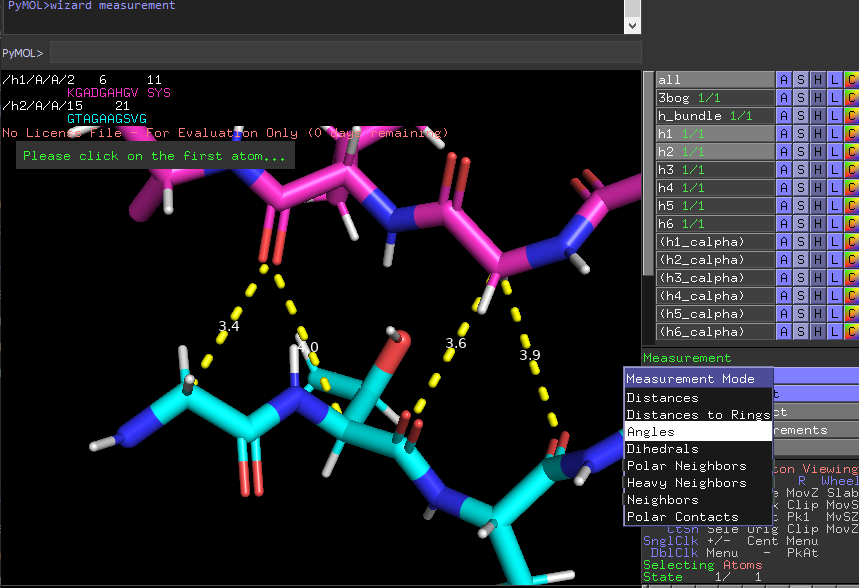
Figura 27. Posible migraña inducida por mala praxis.



Cabe resaltar que, como se ha realizado anteriormente, , 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.

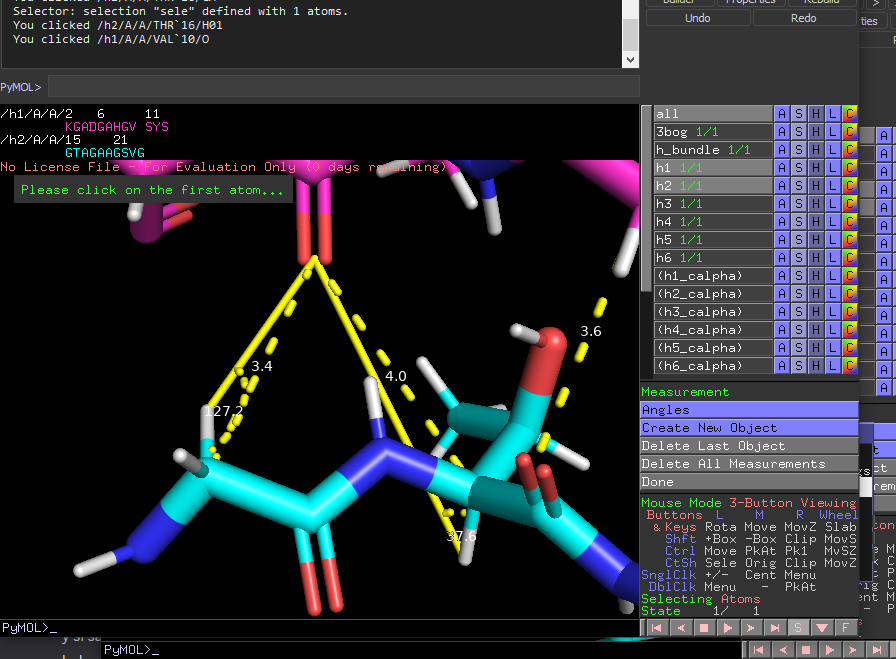


Para medir el ángulo correspondiente a un posible puente de hidrógeno, se debe hacer clic, en orden, primero sobre el **carbono alfa**, después sobre el **hidrógeno**, y por último sobre el **oxígeno**. Alternativamente, también es válido seleccionar primero el **oxígeno**, seguido del **hidrógeno**, 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 más 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, y presentan 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 análisis de hélices PPII.

La siguiente tabla 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) destaca como una solución versátil y robusta para el análisis estructural en biología molecular, siendo la solución más completa. 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 que 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 siguiente tabla 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

## Introducción

PyMOL es la herramienta elegida para este proyecto debido principalmente a su capacidad de integración con Python, lo que permite automatizar y personalizar fácilmente análisis estructurales complejos mediante scripting. 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 el grupo 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 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 inicia la visualización de la proteína en modo **cartoon** mostrando una versión simplificada de su estructura, para posteriormente cambiar al modo **licorice** cuando precisa examinar átomos individuales o realizar mediciones detalladas.

Esto permite cargar de forma sencilla la estructura deseada. Posteriormente, el científico suele visualizar inicialmente la proteína en modo **cartoon** para obtener una representación global clara de su organización estructural, y posteriormente 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, podrían 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. Lo cierto es que esta parte no se hace actualmente.

## 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 4, que reproduce esta misma lógica de forma programada y reproducible.

## 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).

Este procedimiento se ha documentado paso a paso en las figuras indicadas, y ha servido como base para el desarrollo automatizado del módulo descrito en el capítulo siguiente.

# Objetivos y metodología de trabajo

Este proyecto se enmarca en el desarrollo de un módulo que automatice la detección de hélices de PPII en estructuras proteicas, con el objetivo de facilitar su análisis estructural en entornos de investigación científica. A continuación, se detallan los objetivos del trabajo y la metodología seguida para alcanzarlos.

## 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 al estudio de procesos neuronales. 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 los investigadores del Instituto de Química-Física “Blas Cabrera” del CSIC.

## 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 sencilla e intuitiva, que integre de forma eficiente el análisis y la visualización de hélices PPII sobre la estructura proteica cargada.
* **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**, elaborando 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 trabajo se ha desarrollado siguiendo una metodología basada en el ciclo de desarrollo iterativo, combinando fases de análisis, diseño, implementación, pruebas y validación. Como se ha descrito en el apartado 2.2.2 de este trabajo, se ha escogido PyMOL como la herramienta idónea para el desarrollo que nos ocupa.

**Fases del trabajo**

1. **Análisis del problema**: Se estudió la problemática asociada a la identificación manual de hélices PPII y sus limitaciones, revisando tanto los métodos empleados en el laboratorio como los criterios propuestos publicación de referencia de este trabajo (Segura Rodríguez & Laurents, 2024). Se analizaron las características estructurales distintivas de estas hélices, así como los retos asociados a su reconocimiento automático.
2. **Diseño del algoritmo de detección**: Partiendo del análisis previo, se establecieron las condiciones geométricas necesarias para la detección automática de hélices PPII. Esto incluyó la selección de umbrales para los ángulos diédricos (φ y ψ). Además, se incorporó la evaluación de enlaces de hidrógenos no canónicos, utilizando las capacidades de medición de distancias y ángulos que ofrece PyMOL.
3. **Implementación del módulo en Python**: Se desarrolló el código en el entorno de programación Anaconda, utilizando Visual Studio Code como editor principal. El módulo fue integrado como un script ejecutable desde PyMOL, aprovechando sus funciones internas de medición y empleando librerías como tkinter para la interfaz gráfica, os para la gestión de archivos y pymol.cmd como API principal de análisis estructural. La implementación se diseñó con un enfoque modular, lo que facilita futuras ampliaciones o ajustes por parte de otros usuarios.
4. **Integración y validación con datos reales**: El módulo fue integrado en PyMOL y validado empleando estructuras reales procedentes de archivos PDB, seleccionadas a partir del artículo de referencia que fundamenta este proyecto. Estas proteínas presentan dominios organizados en haces de hélices PPII, y sirvieron como base teórica para ajustar y evaluar la lógica del algoritmo desarrollado.

Se compararon los resultados obtenidos mediante el módulo con los análisis previos realizados de forma manual por el equipo de investigación, con el objetivo de comprobar la coherencia y efectividad del sistema automatizado. En particular, la proteína 3BOG fue empleada como principal caso de validación experimental, debido a su estructura bien caracterizada y su frecuente uso como modelo en estudios sobre hélices PPII.  
Las estructuras recogidas en la Tabla 3, descritas por Segura Rodríguez y Laurents (2024), sirvieron como referencia estructural para diseñar el algoritmo de detección propuesto en este trabajo.

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 |

**Fuente**: Adaptado de Segura Rodríguez & Laurents (2024).

1. **Documentación, elaboración de guía de uso y revisión**: Finalmente, se elaboró una guía de usuario con instrucciones detalladas sobre cómo ejecutar el módulo desde PyMOL, así como sobre la interpretación de los resultados generados. Esta documentación incluye capturas de pantalla, ejemplos prácticos y recomendaciones para resolver errores comunes, con el objetivo de facilitar su adopción por parte de la comunidad científica. Además, se realizaron ajustes iterativos sobre el código a medida que se obtenían resultados, mejorando así su estabilidad y funcionalidad final.

Esta metodología combina conocimientos de bioinformática estructural, desarrollo en Python y uso aplicado de herramientas científicas como PyMOL, con el fin de aportar una solución útil, replicable y alineada con las necesidades reales del laboratorio de investigación.

## Contribución de la autora

El desarrollo del módulo presentado en este trabajo ha sido responsabilidad directa de la autora, incluyendo el diseño del algoritmo, la implementación del código en Python, la integración con PyMOL y la elaboración de la interfaz gráfica. Partiendo de los protocolos experimentales utilizados en el laboratorio, se ha realizado un análisis exhaustivo del procedimiento manual para sistematizarlo en forma de una lógica programable. Esta formalización permitió transformar una tarea visual y subjetiva en una herramienta automatizada, reproducible y adaptable.

Las decisiones relacionadas con la estructura del código, la segmentación en funciones, la selección de criterios geométricos, la validación de ángulos y distancias, así como la construcción del sistema de interacción con el usuario, han sido llevadas a cabo íntegramente por la autora, aplicando conocimientos adquiridos en el Grado en Ingeniería Informática. Asimismo, se ha desarrollado toda la documentación técnica y de uso, con el objetivo de facilitar la reutilización del módulo por parte de otros investigadores.

## Colaboraciones y contexto experimental

El presente trabajo surge en colaboración con el grupo de investigación del Instituto de Química-Física “Blas Cabrera” (CSIC), que ha aportado la base teórica, los protocolos de análisis manual y las estructuras proteicas empleadas en la validación del módulo. En particular, se ha tomado como referencia el procedimiento descrito en el artículo de Segura Rodríguez y Laurents (2024), el cual establece los criterios geométricos aplicados en este proyecto.

La validación funcional del módulo se ha llevado a cabo sobre proteínas ya estudiadas en el laboratorio, lo que ha permitido comprobar su eficacia frente a resultados previos. No obstante, todas las tareas de análisis, diseño, codificación, pruebas, documentación y redacción del presente trabajo han sido realizadas exclusivamente por la autora, siguiendo las directrices metodológicas marcadas por la tutora académica del TFG.

# Desarrollo del módulo (ligar con los objetivos del punto 4)

## Requisitos funcionales y técnicos del sistema

El desarrollo del módulo para la detección automatizada de hélices de tipo poliprolina II (PPII) parte de un conjunto de requisitos definidos tanto desde el punto de vista funcional, es decir, lo que el módulo debe ser capaz de hacer, como desde una perspectiva técnica, relacionada con su integración, portabilidad y entorno de ejecución.

Desde el punto de vista funcional, el sistema debe:

* Integrarse en el programa PyMOL y desde este entorno, ejecutar el resto de sus funciones.
* Cargar estructuras proteicas en formato PDB.
* Preparar el archivo para su caracterización.
* Analizar las cadenas proteicas presentes e identificar aquellas regiones cuyos residuos presentan ángulos diédricos φ y ψ compatibles con la conformación PPII.
* Permitir establecer márgenes de tolerancia sobre los valores de φ y ψ para adaptarse a la variabilidad de las moléculas reales.
* Detectar posibles enlaces de hidrógeno no canónicos entre hélices colindantes.
* Segregar visualmente en PyMOL las regiones detectadas, marcándolas específicamente.
* Generar un archivo de texto que liste las posibles hélices PPII detectadas con su posición en la secuencia y sus características principales.

A nivel técnico, el módulo debe cumplir con los siguientes requisitos:

* Estar desarrollado en Python 3, empleando el sistema de scripting interno de PyMOL y sus funciones disponibles a través de la API cmd.
* Ser compatible con entornos multiplataforma (Windows, macOS y Linux), ya que PyMOL funciona en todos ellos.
* Poder integrarse en sesiones de trabajo reales del laboratorio, sin requerir herramientas adicionales más allá de las dependencias estándar.
* Ejecutarse directamente desde la interfaz de PyMOL mediante el comando run o cargando un archivo .py.
* Incluir funciones bien documentadas y estructuradas para facilitar su mantenimiento, reutilización y ampliación por parte de otros usuarios.

En función de estos requisitos, se diseñó una solución modular, dividida en bloques funcionales independientes, como el filtrado de residuos, el análisis geométrico, la detección de enlaces y la visualización. Estos bloques pueden utilizarse de forma conjunta o por separado, según las necesidades del usuario. Esta estrategia de diseño busca facilitar la extensión del sistema en futuras versiones sin comprometer su estabilidad ni su facilidad de uso.

## Diseño del algoritmo de detección PPII

El algoritmo desarrollado para la detección de hélices PPII reproduce de forma automatizada los pasos del procedimiento manual descrito anteriormente (ver apartado 2.1.2), incorporando criterios geométricos derivados del protocolo utilizado en el paper de referencia. El objetivo es trasladar a código una secuencia de análisis que, hasta ahora, 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 manera secuencial sobre la estructura proteica cargada en PyMOL. Cada uno de estos bloques responde a un paso del procedimiento manual, lo que garantiza coherencia metodológica y facilita la validación por parte del equipo investigador.

Los pasos principales que componen el diseño del algoritmo son:

1. **Lectura del archivo PDB y selección de la cadena**: el sistema accede a la estructura cargada en PyMOL y permite seleccionar una cadena específica de la proteína, replicando el paso inicial de limpieza y selección realizado manualmente.
2. **Filtrado inicial de residuos**: se recorre la cadena seleccionada y se identifican segmentos continuos de residuos sin interrupciones ni modificaciones, equivalente al aislamiento de regiones candidatas en el análisis manual.
3. **Cálculo de ángulos diedros φ y ψ**: se calculan los ángulos diedros para cada residuo del segmento analizado, comparándolos con los rangos típicos de la conformación PPII. Este paso automatiza la medición manual de ángulos descrita en el procedimiento tradicional.
4. **Identificación de segmentos candidatos**: el algoritmo selecciona tramos con varios residuos consecutivos que presentan valores de φ y ψ compatibles, reproduciendo la detección visual de patrones continuos realizada en el laboratorio.
5. **Evaluación de enlaces de hidrógeno no canónicos** (opcional): se calculan distancias y ángulos entre carbonos alfa y oxígenos de carbonilo de hélices colindantes, para detectar enlaces de hidrógeno no convencionales, tal y como se realiza manualmente mediante comandos de medición en PyMOL.
6. **Visualización y anotación de resultados**: las hélices identificadas se colorean, etiquetan y representan en PyMOL, imitando el proceso de selección, coloreado y nombrado de objetos descrito en el procedimiento experimental.
7. **Parámetros configurables**: todos los umbrales utilizados (valores angulares, longitudes mínimas, distancias máximas) pueden ajustarse, lo que permite adaptar el análisis a diferentes proteínas o criterios del investigador.

Este diseño garantiza que la herramienta automatiza fielmente el proceso manual descrito en el laboratorio, manteniendo su lógica de análisis, pero optimizando el tiempo, reduciendo errores y eliminando la subjetividad del procedimiento visual.

## Estructura del código y funciones principales

El módulo ha sido desarrollado en Python utilizando el sistema de scripting de PyMOL y la biblioteca gráfica estándar Tkinter, lo que permite su integración directa en sesiones de trabajo habituales sin necesidad de herramientas externas. 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 ejecuta desde PyMOL como un archivo .py, que al cargarse abre una interfaz gráfica interactiva 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 estándar. La estructura seleccionada se importa a PyMOL bajo el nombre “proteina” y se representa en estilo *licorice* para facilitar la visualización de átomos individuales.
* **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.
* **eliminar\_solventes()**: elimina las moléculas de solvente presentes en el archivo PDB, como el agua, que podrían interferir en el análisis estructural.
* **ocultar\_side\_chains()**: oculta las cadenas laterales de los aminoácidos, dejando visible solo el esqueleto principal de la proteína (*backbone*), lo cual 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.
* **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.
* **mostrar\_angulos\_phi\_psi()**: similar a la función anterior, pero muestra los resultados en pantalla dentro de una ventana emergente con scroll, facilitando su revisión inmediata sin necesidad de guardar los datos.
* **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 que no están familiarizados con el uso de scripts en la línea de comandos de PyMOL.

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

## Proceso de integración con PyMOL

Una de las ventajas más destacadas de PyMOL es su capacidad para ejecutar scripts en Python directamente desde su entorno gráfico, lo que permite extender su funcionalidad sin necesidad de modificar el código fuente del programa. En este proyecto, se 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 .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 decisión técnica garantiza la compatibilidad multiplataforma y evita dependencias adicionales, lo que facilita su distribución entre distintos usuarios.

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 abre la posibilidad de utilizarlo en análisis automatizados sobre múltiples proteínas mediante scripts externos o bucles de ejecución.

En definitiva, la integración con PyMOL no solo hace que la herramienta sea accesible y funcional para investigadores 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 validación sobre 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 poliprolina II (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, medición de ángulos diedros φ y ψ, y selección de residuos con valores compatibles con la conformación PPII.
* Se han detectado automáticamente los seis segmentos correspondientes a las hélices conocidas, con una concordancia total 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 sobre el modelo molecular, utilizando colores y etiquetas para diferenciar 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 conocidas, 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, no solo sobre 3BOG, sino también sobre otras proteínas con dominios desordenados o regiones enriquecidas en glicina y prolina susceptibles de presentar hélices PPII.

## 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. Este repositorio contiene tanto el archivo principal del script como la documentación técnica necesaria para su instalación y uso.

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. Esta decisión está alineada con los principios de ciencia abierta y desarrollo reproducible que cada vez son más valorados en el ámbito de la bioinformática estructural.

Además del código fuente, el repositorio incluye:

* Una guía de instalación y ejecución 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.

El módulo se distribuye bajo una licencia libre compatible con usos académicos y científicos, garantizando su disponibilidad y evolución futura. Esta apertura contribuye a maximizar el impacto del trabajo y refuerza el valor práctico del proyecto más allá del entorno académico.

## 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.

**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.
* **Pasos**:
  1. Ejecutar PyMOL.
  2. Cargar el módulo desde la interfaz con el comando run.
  3. Utilizar la opción “Seleccionar archivo” para importar el archivo PDB.
  4. Eliminar moléculas de solvente y añadir hidrógenos automáticamente.
* **Resultado esperado**: La estructura aparece cargada, limpia y lista para su análisis estructural en PyMOL.

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

* **Actor**: Usuario
* **Objetivo**: Calcular los ángulos diedros φ y ψ de todos los residuos de una cadena.
* **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 generado con los valores por residuo.
* **Resultado esperado**: Se genera un archivo de texto con los ángulos calculados para cada residuo, permitiendo su posterior análisis o visualización.

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

* **Actor**: Usuario
* **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).
* **Resultado esperado**: Las hélices candidatas se representan en pantalla con colores diferenciados y se genera un resumen exportable con la información asociada.

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

| **Nº** | **Caso de uso** | **Actor** | **Objetivo principal** | **Resultado esperado** |
| --- | --- | --- | --- | --- |
| 1 | Carga y preparación de la estructura PDB | Usuario | Importar una proteína y dejarla lista para análisis estructural | Estructura limpia, con hidrógenos añadidos, lista para análisis en PyMOL |
| 2 | Análisis geométrico de la proteína | Usuario | Calcular y guardar los ángulos φ y ψ de los residuos de una cadena | Archivo con los ángulos diedros por residuo para análisis posterior |
| 3 | Identificación visual de hélices PPII | Usuario | Detectar, representar y revisar hélices PPII en la proteína cargada | Hélices coloreadas y etiquetadas; resumen exportable con datos estructurales clave |

# 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 poliprolina II (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.

Cada hélice detectada fue representada en PyMOL con colores diferenciados 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 fácilmente para su posterior análisis o comparación con otras estructuras.

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.

6.2. 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 un dominio avanzado de las herramientas de visualización como PyMOL. Comparar ambos enfoques permite evaluar de forma objetiva las ventajas que ofrece el módulo desarrollado.

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. Todo este procedimiento es laborioso, propenso a errores y difícil de reproducir de forma consistente entre distintos 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 hace por cálculo directo de distancias y ángulos, sin depender de la inspección visual.

Durante la validación, se comprobó que el módulo identifica las mismas hélices que el análisis manual, con una coincidencia total en los tramos detectados. Sin embargo, lo hace en una fracción del tiempo y con un nivel de detalle mucho mayor, ya que genera salidas estructuradas que pueden archivarse, reutilizarse o compararse con otras ejecuciones. Esta trazabilidad no es posible cuando el análisis depende únicamente de la observación visual del investigador.

En resumen, el sistema automatizado no solo replica con precisión los resultados obtenidos por el método manual, sino que los mejora en términos de rapidez, consistencia y potencial de aplicación en estudios a gran escala. Esta comparación refuerza la utilidad práctica del módulo y justifica su adopción como herramienta de análisis en proyectos que requieren estudiar múltiples estructuras o detectar patrones repetitivos en proteínas poco caracterizadas.

## 6.3. 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 encuentran 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 utilizados 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, dependiendo del número de cadenas y la complejidad de la proteína. Esto supone una mejora sustancial frente al proceso manual, que requiere una intervención prolongada y repetitiva para medir ángulos, evaluar geometría y comprobar interacciones. Esta ganancia en tiempo es especialmente relevante cuando se trabaja con múltiples estructuras o se necesita repetir el análisis con diferentes 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, junto con la automatización del análisis y la generación de salidas exportables, convierte al módulo en una herramienta útil 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 del análisis visual. Esta estandarización es clave en entornos de investigación donde se requiere comparar resultados entre distintos experimentos o equipos.

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

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, también presenta algunas limitaciones que conviene tener en cuenta, especialmente si se desea aplicarlo en contextos más amplios o sobre estructuras menos conocidas.

En primer lugar, la herramienta depende de la calidad del archivo PDB y de la correcta anotación de los átomos implicados. En estructuras incompletas, con residuos no estandarizados o carentes de hidrógenos, algunos cálculos pueden verse afectados o ser inviables. Aunque el módulo incorpora funciones para añadir hidrógenos y limpiar la estructura, estos procesos siguen dependiendo de la correcta interpretación del modelo original.

En segundo lugar, el algoritmo utiliza rangos fijos (aunque configurables) para los valores de φ y ψ característicos de la conformación PPII. Esto implica que conformaciones atípicas, que no se ajustan exactamente a esos márgenes pero que podrían ser funcionalmente equivalentes, pueden quedar fuera de la detección. Por esta razón, el sistema no sustituye la interpretación del investigador, sino que debe entenderse como una herramienta de apoyo al análisis.

Además, el módulo está optimizado para estructuras relativamente simples, como 3BOG. Aunque puede ejecutarse sobre proteínas de mayor tamaño, el tiempo de procesamiento y la claridad de visualización pueden verse afectados si no se adapta previamente la selección de regiones o se optimizan los comandos de representación. El sistema también requiere que el usuario tenga conocimientos básicos de PyMOL y entienda cómo interactuar con la interfaz del visor, aunque se ha intentado reducir al mínimo la complejidad de uso.

Por último, la detección de enlaces de hidrógeno no canónicos sigue siendo parcial. Aunque el módulo permite calcular distancias y ángulos relevantes, la interpretación de qué interacciones son estructuralmente significativas sigue dependiendo del criterio experto. No existe todavía un consenso claro sobre cuántos enlaces son necesarios para validar una hélice PPII, lo que deja abierta cierta ambigüedad que el software no puede resolver por sí solo.

Estas limitaciones no restan utilidad al sistema, pero sí marcan el camino para futuras mejoras, especialmente en lo que respecta a su robustez frente a diferentes tipos de entrada, la interpretación automática de patrones estructurales más sutiles y la incorporación de criterios energéticos o estadísticos en la evaluación.

6.5. Aplicabilidad en otros modelos estructurales

Aunque la validación del módulo se ha centrado en la proteína modelo 3BOG, 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 es capaz de adaptarse a proteínas de distinto tamaño, complejidad y nivel de anotación estructural.

Esta versatilidad resulta especialmente útil en el contexto de estudios que exploran regiones desordenadas, dominios ricos en glicina y prolina o secuencias peptídicas no estructuradas que podrían adoptar conformaciones PPII bajo ciertas condiciones. En estos casos, donde las herramientas convencionales tienden a clasificar estas regiones como “coil” o desorden, el módulo proporciona un enfoque alternativo basado en criterios geométricos objetivos.

Además, el sistema permite realizar análisis masivos de bases de datos estructurales, ya que puede integrarse en scripts externos para el procesamiento automatizado de múltiples archivos. 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 de interés en estudios evolutivos o comparativos, donde se analicen familias de proteínas con variaciones sutiles en su plegamiento. El uso del módulo en diferentes estructuras permite detectar patrones comunes de conformación PPII o identificar mutaciones que alteran dicha conformación.

Por último, dado que el sistema no depende de una base de datos concreta ni de herramientas externas, puede utilizarse con estructuras obtenidas por cristalografía, RMN o predicción mediante inteligencia artificial, siempre que estén disponibles en el formato adecuado. Esto lo convierte en una herramienta flexible y de amplio alcance dentro del análisis estructural.

En conjunto, el módulo no solo es válido como apoyo al estudio de 3BOG, sino que tiene potencial para integrarse en flujos de trabajo más amplios en el ámbito de la biología estructural, la bioinformática y el diseño de biomoléculas.

# Conclusiones y trabajo futuro

Este capítulo recoge las conclusiones obtenidas tras el desarrollo del proyecto, evaluando el grado de cumplimiento de los objetivos planteados inicialmente. Asimismo, se proponen posibles líneas de trabajo futuro que podrían mejorar, ampliar o profundizar en los resultados alcanzados.

## 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 poliprolina II (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 llevó a cabo un análisis exhaustivo de la problemática asociada a la identificación manual de hélices PPII, recogiendo 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.

Posteriormente, se diseñó un algoritmo capaz de identificar de forma automática segmentos compatibles con la conformación PPII, mediante el análisis de los ángulos diedros φ y ψ, así como la evaluación de posibles enlaces de hidrógeno no canónicos entre hélices colindantes. El algoritmo fue implementado en Python y adaptado al entorno PyMOL mediante una interfaz gráfica sencilla que permite su uso sin conocimientos avanzados de programación.

La validación experimental, realizada sobre la proteína modelo 3BOG, confirmó que el sistema desarrollado 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, lo que refuerza su utilidad como herramienta de apoyo a la investigación estructural.

En conjunto, puede concluirse que el trabajo ha alcanzado satisfactoriamente sus objetivos, aportando una solución técnica que automatiza un proceso complejo y facilitando su integración en flujos de trabajo reales en entornos científicos. Además, sienta las bases para futuras ampliaciones orientadas al análisis masivo de estructuras y al desarrollo de herramientas más avanzadas en el campo de la bioinformática estructural.

## 7.2. Aportaciones del módulo

El módulo desarrollado en este trabajo constituye una aportación técnica y funcional relevante en el ámbito del análisis estructural de proteínas, al automatizar la detección de hélices de tipo poliprolina II (PPII) mediante criterios geométricos. Su implementación permite ejecutar un proceso hasta ahora manual de forma reproducible, rápida y objetiva, 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 la integración del sistema en un entorno ya familiar para investigadores en bioinformática estructural, como es PyMOL. Esto permite su adopción inmediata en flujos de trabajo reales, sin necesidad de herramientas externas ni conocimientos avanzados en programación. La interfaz gráfica desarrollada facilita su uso incluso por parte de usuarios no expertos, lo que amplía el alcance 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.

## 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, convendría mejorar la precisión del algoritmo mediante la incorporación de modelos estadísticos o técnicas de aprendizaje automático, que permitan clasificar automáticamente las hélices con mayor fiabilidad, incluso en estructuras no canónicas. Asimismo, sería pertinente extender el análisis a interacciones entre hélices PPII y otras regiones estructurales no-PPII, que no han sido consideradas en el alcance del presente proyecto.

Desde el punto de vista de la usabilidad, podría desarrollarse una interfaz gráfica integrada directamente en PyMOL, que facilite la ejecución del módulo por parte de usuarios sin experiencia en scripting. También se contempla la posibilidad de integrarlo en otros visores o entornos bioinformáticos complementarios, lo que ampliaría su aplicabilidad en distintos contextos de análisis estructural.

Finalmente, la creación de una base de datos con estructuras confirmadas de hélices PPII permitiría validar el módulo a mayor escala y reforzar su valor como herramienta de investigación. Este recurso facilitaría la evaluación sistemática de su rendimiento y fomentaría su uso en áreas como el diseño racional de proteínas o el estudio de interacciones moleculares complejas.

Este capítulo presenta el desarrollo técnico del módulo creado para la detección semiautomática de hélices de poliprolina II (PPII) en estructuras proteicas, detallando las decisiones de diseño, las funcionalidades implementadas y el proceso de automatización llevado a cabo.

El desarrollo del módulo ha sido concebido como una respuesta directa a la necesidad, identificada en el laboratorio, de automatizar un procedimiento que hasta el momento se realiza de forma manual mediante herramientas visuales como PyMOL. Si bien este análisis manual ha sido útil para el estudio de determinadas proteínas, resulta limitado en términos de escalabilidad, reproducibilidad y objetividad. El presente trabajo introduce una solución computacional que incorpora los criterios utilizados en el laboratorio, formalizándolos en un algoritmo programable y aplicable de forma sistemática.

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

* **Automatización completa del proceso de identificación de hélices PPII**, mediante el diseño de un sistema que reduce la intervención manual y permite obtener resultados consistentes, reproducibles y escalables a 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 de uso dentro de 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**, utilizando como prueba proteínas reales analizadas previamente, lo que permitió contrastar los resultados obtenidos automáticamente con los resultados manuales de referencia.
* **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.

Aunque el diseño conceptual del algoritmo parte de una necesidad identificada conjuntamente con el equipo investigador, el desarrollo técnico, las decisiones de implementación y la programación del módulo han sido responsabilidad exclusiva de la autora. El trabajo ha sido supervisado por la dirección académica del TFG, así como con el apoyo ocasional del equipo del laboratorio en cuestiones puntuales de interpretación estructural.

Este capítulo define, por tanto, el marco técnico del trabajo y detalla la aportación desarrollada, distinguiendo claramente entre los recursos existentes que han sido utilizados (como PyMOL o estructuras PDB públicas) y las soluciones originales creadas específicamente para este proyecto.

### Tabla resumen de tareas

| **Tarea principal** | **Responsable** |
| --- | --- |
| Análisis del procedimiento manual y sistematización | Silvia (autora) |
| Diseño e implementación del algoritmo en Python | Silvia (autora) |
| Automatización del proceso de detección en PyMOL | Silvia (autora) |
| Integración con la interfaz de PyMOL | Silvia (autora) |
| Pruebas y validación con proteínas reales | Silvia (autora) |
| Redacción de la guía de uso y documentación técnica | Silvia (autora) |

# Conclusiones y trabajo futuro

Recuerda escribir siempre al menos un párrafo de introducción al comienzo de cada capítulo. En este capítulo final debes explicar las conclusiones de tu trabajo e indicar algunas posibles líneas de trabajo futuro.

Las **conclusiones** evalúan en qué medida consideras alcanzados tus objetivos iniciales. Se trata de hacer una valoración global de tus resultados, pero siempre puestos en relación con los objetivos que inicialmente te habías planteado.

En las **líneas de trabajo futuro** puedes explicar cómo se podría mejorar o continuar el trabajo que has presentado y finalizado. Puedes aprovechar aquí para explicar cómo se podría mejorar algún objetivo que consideras no haber podido alcanzar del todo. las líneas de trabajo futuro serían posibles objetivos de un nuevo trabajo que vendría como continuación de este.

## Conclusiones del trabajo

Tras una breve introducción de esta sección, dedica al menos un párrafo a analizar cómo has intentado alcanzar cada uno de los objetivos que te habías planteado, y en qué medida lo consideras cumplido. Se trata de recordar cómo los diferentes contenidos y resultados de los capítulos anteriores materializan tu nivel de cumplimiento de estos objetivos.

## Líneas de trabajo futuro

Explica aquí posibles objetivos o líneas de acción que podrían ser continuación del trabajo que has presentado.

Referencias bibliográficas

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>

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>

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>

PyMOLWiki. (s.f.). Distance. Retrieved 22 de marzo de 2025, from <https://pymolwiki.org/index.php/Distance>

Jmol Development Team. (2025). *Jmol: An open-source Java viewer for chemical structures in 3D*. http://www.jmol.org/

Open Babel. (2025). *Avogadro*. https://avogadro.cc/

Sayle, R. A. and Milner-White, E. J. (2025). *RasMol: Molecular Graphics Program*. http://www.rasmol.org

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

University of California, San Francisco. (2025a). *ChimeraX*. https://www.cgl.ucsf.edu/chimerax/

University of California, San Francisco. (2025b). *UCSF Chimera*. https://www.cgl.ucsf.edu/chimera/

\_\_

Chemical Computing Group. (2025). *MOE: Molecular Operating Environment*. https://www.chemcomp.com/MOE-Molecular\_Operating\_Environment.htm

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

Jmol Development Team. (2025). *Jmol: An open-source Java viewer for chemical structures in 3D*. <http://www.jmol.org/>

Open Babel. (2025). *Avogadro*. <https://avogadro.cc/>

PyMOLWiki. (s.f.). *PyMOL*. Recuperado el 22 de marzo de 2025, de https://pymolwiki.org/index.php/Main\_Page

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

Schrödinger, LLC. (2025). *Maestro*. https://www.schrodinger.com/maestro

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

University of California, San Francisco. (2025a). *UCSF ChimeraX*. https://www.cgl.ucsf.edu/chimerax/

University of California, San Francisco. (2025b). *UCSF Chimera*. https://www.cgl.ucsf.edu/chimera/

\_\_

Aquí tienes la referencia de PyMOL en formato APA 7ª edición, actualizada y lista para usar en tu gestor (como Zotero):

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

También puedes, si prefieres enfatizar el carácter de código abierto, usar esta versión citando el desarrollador original:

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

Ambas son aceptadas; si tu enfoque está más ligado al desarrollo con código abierto y al uso académico, la segunda (DeLano) es más adecuada. Si estás usando la versión moderna mantenida por Schrödinger, usa la primera.

Las de la tabla del artículo son

 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

 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

 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

 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

 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

 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

 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

 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 módulp

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.

**Función: seleccionar\_archivo()**

def seleccionar\_archivo():

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

if archivo:

cmd.load(archivo, "proteina")

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

**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")

**Función: eliminar\_solventes()**

def eliminar\_solventes():

cmd.remove("solvent")

**Función: anadir\_hidrogenos()**

def anadir\_hidrogenos():

cmd.h\_add()

El código completo, junto con la guía de uso y el archivo ejecutable, se encuentra disponible en el repositorio público del proyecto en GitHub: [https://github.com/silviaenma/ppii-detector-pymol]

Índice de acrónimos

|  |  |
| --- | --- |
| **Acrónimo** | **Significado** |
| CSIC | Consejo Superior de Investigaciones Científicas |
| DSSP | Define Secondary Structure of Proteins |
| PDB | Protein Data Bank |
| PPII | Hélice de poliprolina tipo II |
| PyMOL | Python Molecular Graphics Tool |
| TFG | Trabajo de Fin de Grado |
| UCSF | University of California, San Francisco |
| VMD | Visual Molecular Dynamics |
| VS Code | Visual Studio Code |

. Antes de describir en detalle las fases, resulta fundamental presentar PyMOL, la herramienta principal sobre la cual se ha desarrollado la solución propuesta.

#### **Uso de PyMOL y sus funcionalidades**

PyMOL es un programa de visualización molecular de código abierto ampliamente utilizado en el ámbito de la bioinformática y la biología estructural. Permite representar, analizar y manipular estructuras tridimensionales de macromoléculas, como proteínas, ácidos nucleicos o complejos moleculares, a partir de archivos estructurales en formato PDB (Protein Data Bank).

Entre las funcionalidades de PyMOL que resultan relevantes para este proyecto destacan:

* **Carga y visualización de estructuras**: PyMOL permite importar archivos PDB y representar las moléculas de múltiples formas (bastón, cinta, superficie, esferas, etc.), facilitando el análisis visual de su organización tridimensional.
* **Medición de ángulos y distancias**: A través de sus herramientas de medición, PyMOL puede calcular ángulos diédricos (phi y psi) entre átomos consecutivos de la cadena polipeptídica, información clave para la identificación de tipos específicos de estructuras secundarias, como las hélices de poliprolina II.
* **Identificación y análisis de enlaces de hidrógeno**: El programa ofrece funcionalidades automáticas y manuales para detectar enlaces de hidrógeno, ayudando a distinguir entre diferentes tipos de estructuras secundarias basándose en su red de interacciones.
* **Scripting y automatización**: Una de las características más potentes de PyMOL es la posibilidad de extender sus capacidades mediante scripts en Python, permitiendo la automatización de tareas repetitivas, la implementación de algoritmos personalizados y el análisis masivo de datos estructurales.
* **Representación y anotación personalizada**: PyMOL permite modificar colores, etiquetas y estilos de representación de acuerdo a las necesidades específicas del análisis, facilitando la interpretación y la comunicación de resultados.

Estas funcionalidades convierten a PyMOL en la herramienta ideal para el desarrollo de un módulo que automatice la detección de hélices PPII, ya que combina la potencia gráfica con la flexibilidad del lenguaje Python.