



Facultad de Ingeniería
Escuela de Ingeniería Civil en Bioinformática

Análisis del interactoma de la proteína KCTD5 mediante la plataforma web PPI-MASS

Simón Felipe Zúñiga Almonacid

Matrícula: 2013430850

Profesor Tutor: Dr. Ariela Vergara-Jaque

Profesor Co-tutor: Dr. Óscar Cerda Arancibia

Profesor Co-tutor: Fabio Durán-Verdugo

Profesor Informante: Alejandro Valdés Jiménez

Memoria para optar al título de Ingeniero Civil en Bioinformática

Talca, Chile

Julio, 2021

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su unidad de procesos técnicos certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Talca, 2021

Índice

Resumen	4
Abstract	5
Introducción	7
1. Espectrometría, proteómica e interacciones proteína-proteína	7
2. Interactoma de la proteína KCTD5	9
3. Software para la detección de PPIs a partir de datos de espectrometría de masas	8
4. Perspectivas para el diseño de una plataforma de post-procesamiento para datos de MS	12
5. Fundamentos del diseño de software	13
Motivación y objetivos	16
Metodología	17
1. Pipeline	17
2. Desarrollo del software	20
2.1 Implementación de la plataforma	23
2.2 Software, tecnologías y versiones	25
3. Análisis del interactoma de KCTD5	26
Desarrollo del software	27
1. Requerimientos	27
1.1 Requerimiento general	27
1.2 Requerimientos específicos	27
2. Implementación	28
2.1 Base de datos	29
2.2 Servidor	29
2.3 Pipeline	31
2.4 Arquitectura del proyecto Django	33
3. Plataforma web	34
3.1 Secciones de la plataforma web	34
3.2 Modo de uso	38
Caso de estudio	43
Conclusiones	45
Bibliografía	47

Lista de figuras

- [Figura 1: Análisis cualitativos y cuantitativos basados en MS para el estudio de interacciones proteína-proteína.](#)
- [Figura 2: Pipeline propuesto para el estudio de PPIs a partir de datos de MS.](#)
- [Figura 3: Metodología de desarrollo](#)
- [Figura 4: Patrón de arquitectura general de Django.](#)
- [Figura 5: Red de interacción de KCTD5 obtenida mediante la base de datos STRING.](#)
- [Figura 6: Comunicación entre los diferentes componentes de PPI-MASS.](#)
- [Figura 7: Modelo de la base de datos de PPI-MASS.](#)
- [Figura 8: Diagrama de flujo del análisis de las proteínas de entrada.](#)
- [Figura 9: Diagrama de la arquitectura de la plataforma web en el framework Django.](#)
- [Figura 10: Página de inicio de PPI-MASS.](#)
- [Figura 11: Formulario de envío de trabajos.](#)
- [Figura 12: Búsqueda de resultados mediante el UUID del trabajo.](#)
- [Figura 13: Página de resultados.](#)
- [Figura 14: Formulario de contacto.](#)
- [Figura 15: Manual de uso de la plataforma PPI-MASS.](#)
- [Figura 16: Visualización de resultados en PPI-MASS.](#)
- [Figura 17: Diagrama del uso del software PPI-MASS.](#)
- [Figura 18: Resultado del análisis de la proteína KCTD mediante PPI-MASS.](#)

Lista de tablas

- [Tabla 1: Comparación de softwares actuales para la predicción de PPIs.](#)
- [Tabla 2: Lista resumen de software y versiones que fueron utilizadas para el desarrollo del proyecto.](#)
- [Tabla 3: Lista de requerimientos para el desarrollo del software PPI-MASS](#)

Resumen

Existe una gran cantidad de enfermedades humanas que han sido vinculadas a interacciones proteína-proteína (PPIs). El interactoma humano se compone de más de 600.000 complejos proteicos, los cuales se asocian a una gran diversidad de procesos celulares fundamentales para el correcto funcionamiento y desarrollo del organismo. Un caso de especial interés es el interactoma de la proteína KCTD5, la cual ha mostrado regular canales iónicos y estar asociada con cáncer de mama y la formación de ciertos tumores. Una manera de estudiar proteínas que regulan la función de KCTD5 es mediante ensayos de espectrometría de masas (MS); sin embargo, el post-procesamiento de datos resultantes de MS no es trivial y actualmente existen limitadas herramientas informáticas que brinden al usuario flexibilidad y transparencia para manejar e interpretar los datos. En este sentido, el presente proyecto de tesis describe el desarrollo de una plataforma web para el análisis de datos obtenidos a través de experimentos de MS, integrando información de diferentes bases de datos biológicas y permitiendo al usuario emplear determinados filtros para identificar complejos proteína-proteína con potencial farmacológico. El desempeño de la plataforma PPI-MASS, ha sido inicialmente evaluado usando como parámetros de entrada datos de MS vinculados a la proteína KCTD5. Sin embargo, se prevé su uso para el análisis y detección de cualquier conjunto de PPIs proveniente de experimentos de MS. Cabe señalar, que la detección de PPIs es fundamental para el diseño de fármacos basado en estructura, por lo tanto, se espera que esta plataforma sea un gran aporte en el área biomédica.

Abstract

A large number of human diseases have been linked to protein-protein interactions (PPIs). The human interactome comprises over 600.000 protein complexes, which are associated with a wide variety of cell processes essential for the correct function and development of the human organism. One case of particular interest is the KCTD5 protein interactome. KCTD5 has been shown to regulate ion channels and to be associated with breast cancer and the formation of certain tumors. One way of studying the proteins that regulate the function of KCTD5 is through mass-spectrometry assays (MS); however, the post-processing of the resulting data is not a trivial task. Moreover, there are currently few computational tools that give the user flexibility and transparency to manage and interpret the data. In that regard, this thesis project describes the development of a web platform for the analysis of data obtained through MS assays, incorporating information from different biological databases and allowing the user to employ specific filters to identify protein-protein complexes with pharmaceutical potential. The performance of the PPI-MASS platform has been initially evaluated using MS data associated with the KCTD5 protein as input. However, we expect its use to analyze and detect any set of PPIs from MS assays. It is important to note that the detection of PPIs is essential for structure-based drug design and, because of this, we expect that this platform will be of great assistance in the biomedical field.

Financiamiento

Esta Memoria de Título fue financiada mediante el Proyecto FONDECYT 1200917 de la Agencia Nacional de Investigación y Desarrollo (ANID) otorgado a O.C. y realizada en la Facultad de Ingeniería de la Universidad de Talca y en el Programa de Biología Celular y Molecular del Instituto de Ciencias Biomédicas (ICBM) de la Facultad de Medicina de la Universidad de Chile.

Introducción

1. Espectrometría, proteómica e interacciones proteína-proteína

La espectrometría de masa (MS, por sus siglas en inglés) es una técnica que permite la identificación de estructuras orgánicas en una solución en función de su masa [1].

Los métodos de MS han sido utilizados en diversas aplicaciones a través de su historia. Éstas incluyen la medición de la abundancia de isótopos en una muestra [2], el diagnóstico de patologías a través de imagenología médica [3], el estudio de genomas y proteínas [4]. Su uso en proteómica resulta de gran interés debido a sus aplicaciones para el estudio de enfermedades y el diseño de fármacos, como por ejemplo la detección de modificaciones post traduccionales vinculadas a enfermedades [5].

Cabe destacar, que las proteínas cumplen funciones esenciales para las células y la mayoría de ellas no actúan solas, sino que establecen interacciones con otras moléculas biológicas, las cuales pueden ser estudiadas mediante MS. En particular, las interacciones proteína-proteína (PPIs, por sus siglas en inglés) son de gran interés debido a que juegan un rol fundamental en muchos procesos celulares y rutas metabólicas [6]. Estudiar y entender estas interacciones es crucial para tener una comprensión más acabada de los sistemas biológicos e identificar blancos farmacológicos. Para esto, se han desarrollado diferentes métodos que permiten caracterizar estas interacciones en diversos organismos [7].

Existen dos tipos de análisis generales que pueden ser llevados a cabo para estudiar PPIs mediante MS: cualitativos y cuantitativos [6]. Los análisis cualitativos se han utilizado para estudiar la función de las proteínas, y caracterizar sus redes de interacciones.

Por otro lado, los análisis cuantitativos se han utilizado para medir la abundancia relativa de interacciones específicas en múltiples condiciones biológicas,

permitiendo así estudiar, por ejemplo, las enfermedades relacionadas con estas interacciones [6].

Ambos tipos de análisis se pueden usar en conjunto para tener una mejor comprensión de las PPIs en distintos contextos biológicos, a través de su caracterización y la medición de sus abundancias relativas en diferentes condiciones (Figura 1).

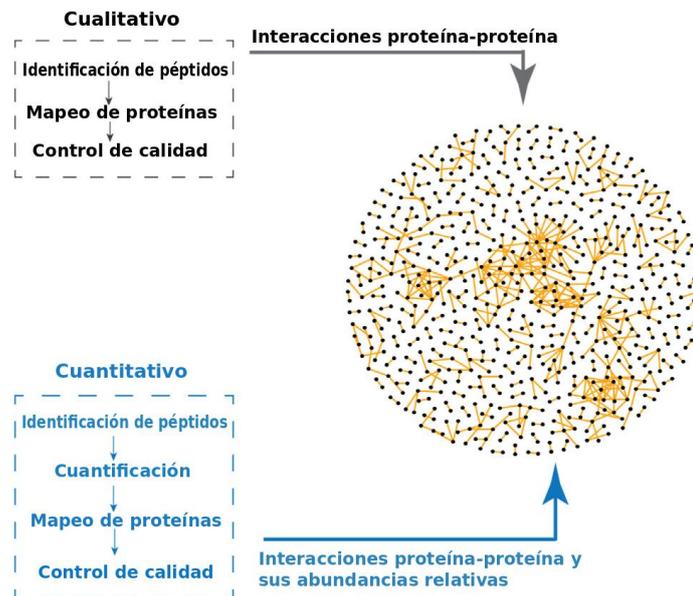


Figura 1: Análisis cualitativos y cuantitativos basados en MS para el estudio de interacciones proteína-proteína. En la parte superior se muestran algunos ejemplos de métodos de análisis cualitativos junto con sus pasos generales y resultados, mientras que en la parte inferior se muestran los métodos de análisis cuantitativos. La combinación de ambos tipos de análisis brinda una imagen completa de las PPIs y sus abundancias relativas. Modificada de Yugandhar, Gupta y Yu, 2019 [6].

La importancia del estudio de las PPIs viene dada por la gran cantidad de enfermedades humanas relacionadas a interacciones anormales. El conjunto de interacciones proteína-proteína se denomina interactoma y en humanos se compone de más de 600.000 complejos hasta ahora identificados [8]. Cabe señalar que interacciones anormales pueden ser causantes directas de distintas enfermedades o producir alteraciones en complejos proteicos interfiriendo en otras interacciones en la red [9]. La formación de complejos anormales está relacionada con una gran cantidad de enfermedades, por lo que surge la necesidad de estudiar las PPIs detalladamente.

Las enfermedades relacionadas con PPIs anormales suelen ser causadas por la formación de interacciones en momentos o localizaciones incorrectas, interacciones con proteínas exógenas o incluso la ausencia de interacciones cruciales [10]. Algunos ejemplos de estas enfermedades son: cáncer relacionado con interacciones anormales del supresor de tumores p53 [11], enfermedades neurodegenerativas causadas por la acumulación de agregados proteicos [12], patologías dermatológicas como la alopecia [13], infecciones virales [14], entre otras.

2. Interactoma de la proteína KCTD5

KCTD5 es una proteína de 25 KDa, miembro de la familia KCTD que se encuentra integrada por 26 miembros [15]. Debe su nombre a “*Potassium Channel Tetramerization Domain*”, ya que es similar al dominio T1 de tetramerización de los canales de potasio dependiente de potencial (K_v). Posee 234 aminoácidos, forma pentámeros, y es el único miembro del cual se ha descrito su estructura tridimensional [15]. En su dominio N-Terminal posee un dominio BTB, necesario para interactuar con Cullin-3 [16] [17] [18]. Recientemente, se ha relacionado esta interacción entre KCTD5 y Cul3, con la mono y poli-ubiquitinación de proteínas y con la degradación de estas por la vía del proteosoma [19] [20] [21]. Esta proteína es altamente expresada en el bazo, testículos, ovarios, próstata, intestino delgado y grueso. En tales órganos, KCTD5 se asocia a diversas funciones como la represión transcripcional, la remodelación citoesquelética y la activación de canales iónicos. Específicamente, KCTD5 ha sido identificado en interacción con el canal iónico TRPM4, regulando su actividad al aumentar su sensibilidad a Ca^{2+} [22]. Además, la expresión de TRPM4 y KCTD5 demostró estar aumentada en células de cáncer de mama y ciertos tumores [22] [23]. Por otra parte, resultados recientes de nuestro laboratorio sugieren que KCTD5 regula las dinámicas de Ca^{2+} inducidas por suero y la respuesta *Store Operated Calcium Entry* (SOCE) [24]. Esto da cuenta de la importancia de estudiar proteínas asociadas a KCTD5, ya que este conocimiento puede ayudar, por ejemplo, en el diseño de fármacos para inhibir interacciones no deseadas.

Sobre la base de esta evidencia, las PPIs constituyen un gran desafío en el diseño de fármacos [25]. Cabe señalar, que la identificación de complejos binarios

proteína-proteína puede conducir luego a caracterizar las áreas interfaciales entre proteínas, las cuales son significativamente mayores y más dinámicas que aquellas asociadas a interacciones proteína-ligando. Por ejemplo, los sitios de unión proteína-proteína están conformados por residuos que no se encuentran necesariamente de manera contigua en la secuencia aminoacídica; además, estas áreas suelen ser planas y no tienen cavidades que permitan que una molécula tipo fármaco pueda penetrar. Sumado a esto, los contactos entre proteínas se producen en regiones de residuos conocidas como *hot spots*, los cuales pueden estar distribuidos a lo largo de toda la superficie de una proteína [25]. En este sentido, es crucial detectar PPIs y conocer la interface de unión de los complejos con el fin de diseñar péptidos o fragmentos que inhiban la unión. Por lo tanto, existe una gran necesidad de diseñar fármacos eficaces para regular PPIs, donde el apoyo de herramientas informáticas es fundamental.

3. Software para la detección de PPIs a partir de datos de espectrometría de masas

En los últimos años se han desarrollado distintos métodos que utilizan específicamente información de MS para predecir PPIs. Esta información provee un punto de inicio para la caracterización de este tipo de interacciones y el posterior diseño de fármacos. Algunos ejemplos de estos métodos incluyen software como PIPINO¹, MeroX² y Proteo3Dnet³.

PIPINO es un software desarrollado en Java [26], cuya función principal es optimizar el reconocimiento de proteínas que pueden estar vinculadas a una proteína *target*. Este software permite la identificación de potenciales pares de proteínas mediante *scatterplots* interactivos. Las interacciones predichas son validadas con bases de datos de complejos proteicos como BioGRID, MINT, Mentha, DIP, IntAct y HPRD.

MeroX también es un software desarrollado en Java para el reconocimiento de PPIs [27]. Se especializa en el análisis de información de Cross-Linking Mass

¹ <http://www.bioforscher.de/pipino>

² <http://www.stavrox.com/>

³ <https://bioserv.rpbs.univ-paris-diderot.fr/services/Proteo3Dnet/>

Spectrometry (XL-MS) y la predicción de redes de interacciones. Las redes predichas se muestran como grafos, donde cada nodo representa una proteína de la red.

Proteo3Dnet es un servicio web que permite generar redes de PPIs integrando información de MS con información estructural del Protein Data Bank basada en búsquedas de homología [28]. También contempla estructuras provenientes del SWISS-MODEL Repository (SMR). Proteo3Dnet identifica y valida interacciones a través de bases de datos como BioGRID e IntAct. Ofrece un visualizador interactivo de las redes de interacciones mediante Cytoscape.js y lista las proteínas de los complejos reportados en la literatura mediante MolArt; sin embargo, no predice o identifica interacciones desconocidas.

Todos estos software presentan una serie de aspectos en común. En general, el enfoque se centra en la identificación de PPIs y en la predicción de redes de interacción. Todos los programas ofrecen alguna forma de visualización de redes o pares de proteínas predichas mediante gráficos. En el caso de PIPINO y MeroX, ambos son programas que se pueden instalar y ejecutar de manera local. Proteo3Dnet, por otro lado, está disponible como servicio web. Si bien estos software son buenas herramientas para la predicción de PPIs, existe aún mucho potencial que no ha sido del todo explorado en los software actualmente disponibles.

Software	Descargable	Servicio web	Visualización de complejos	Información farmacológica
Pipino	Si	No	No	Si
MeroX	Si	No	No	No
Proteo3Dnet	Si	Si	No*	Si

Tabla 1: Comparación de softwares actuales para la predicción de PPIs. *Proteo3Dnet permite visualizar estructuras proteicas de los complejos reportados en la literatura, pero no predice nuevas interacciones.

4. Perspectivas para el diseño de una plataforma de post-procesamiento para datos de MS

Tomando en consideración los software antes mencionados, éstos presentan importantes limitaciones, por ejemplo, sólo uno de los software está disponible como un servicio web. En el caso de los software locales, el usuario debe instalar un programa y realizar análisis usando recursos computacionales propios, que muchas veces son limitados. Por otro lado, los usuarios se ven obligados a mantener todas las dependencias necesarias para el funcionamiento del software, desde los sistemas operativos hasta las librerías y paquetes que los programas requieren. Así también, el acceso a las bases de datos se vuelve complejo, ya que el uso local implica i) que el programa acceda a las bases de datos a través de la red, lo cual es más lento, o ii) que el usuario descargue las bases de datos de manera local en su propio equipo, lo que representa un gran requerimiento de espacio físico. Todas estas características finalmente limitan la cantidad de usuarios que acceden a tales software.

En el contexto biológico, la mayoría de los programas para el post-procesamiento de datos de espectrometría de masas están diseñados como una “caja negra” donde el usuario no puede manipular parámetros de interés biológico que pudieran ser relevantes para juzgar interacciones entre proteínas. Por ejemplo, es necesario conocer la localización subcelular de un set de proteínas para saber si se localizan en el mismo compartimento celular de la proteína *target* o en un lugar cercano para establecer interacción. También, es importante determinar si existe estructura tridimensional disponible de las proteínas para posteriores análisis bioinformáticos. De esta forma, existen muchos parámetros que el usuario puede manipular para identificar de manera racional interacciones proteína-proteína que no son transparentados por soluciones computacionales actuales.

En base a lo anterior, existen aún muchas características por explorar en las herramientas informáticas dirigidas al análisis de datos de espectrometría de masas, tanto en términos de acceso y usabilidad, como en el tipo de análisis que pueden ser llevados a cabo con estas herramientas. Por ejemplo, Pipino y Proteo3Dnet incorporan información farmacológica, pero sólo para la búsqueda de

interacciones previamente reportadas y no como un posible filtro para identificar PPIs para estudios posteriores. Por otro lado, Proteo3Dnet ofrece una visualización 3D de las proteínas que se sabe interactúan, pero no permite visualizar los contactos físicos que se forman. Finalmente, ninguno de los softwares mencionados permite la predicción de complejos nuevos, por lo tanto, surge la necesidad de desarrollar nuevas plataformas de software, para el post-procesamiento de datos obtenidos a través de experimentos de espectrometría de masas, que permitan al usuario participar en la identificación y predicción de complejos proteína-proteína. Este desafío ha sido abordado mediante el desarrollo de una plataforma web que se detalla en la presente memoria de título.

5. Fundamentos del diseño de software

Es importante considerar los fundamentos del diseño de software y el proceso de desarrollo de una plataforma web desde el punto de vista de la ingeniería de software.

En la gestión de la ingeniería de software, se definen una serie de etapas generales para el desarrollo de un proyecto [29]. Entre ellas encontramos: la iniciación y definición de alcance, planificación del proyecto de software, ejecución, evaluación y cierre. Estas etapas sirven como pauta y son implementadas de diferentes maneras según la metodología de desarrollo que se pretende utilizar, por lo que no necesariamente se siguen de manera lineal y pueden modificarse más de una vez a lo largo del proyecto.

La iniciación y definición de alcance, es la etapa inicial en la que se decide comenzar un proyecto de ingeniería de software. En esta etapa se deben determinar y negociar los requerimientos del software y los objetivos. También, se realiza un análisis de viabilidad, en el que se determina si el proyecto propuesto es viable considerando aspectos como recursos, limitaciones tecnológicas, limitaciones financieras, aspectos sociales y políticos.

Durante la planificación del proyecto de software se define la metodología a seguir y las tecnologías a utilizar. Además, se realiza una evaluación de riesgos, se

acuerdan los entregables de cada etapa del desarrollo, como por ejemplo prototipos, reportes o diagramas de diseño. Sumado a esto, se miden los costos de desarrollo, se distribuyen recursos tanto técnicos como humanos y se establecen plazos generales.

En la ejecución del proyecto de software se adquieren los recursos digitales, de equipamientos o de personal, se establecen métricas de avance y se generan reportes. El proceso de monitoreo es llevado a cabo por el administrador del proyecto para asegurar que se cumplan los plazos establecidos y tomar decisiones en función del avance.

El desarrollo del proyecto debe ser constantemente evaluado. La evaluación puede ser tanto del proyecto en general, como de etapas específicas. Esto está relacionado con el control de calidad del proyecto, ya que es aquí donde se evalúa la satisfacción de los requerimientos definidos y otras métricas, tales como el rendimiento de los prototipos o la satisfacción de los usuarios de prueba.

Finalmente, cuando se cumplen todos los requerimientos y se llevan a cabo todos los planes, para el proyecto completo o para una etapa, se realizan las actividades de cierre. Para ello, se establecen diferentes criterios de cierre, como por ejemplo pruebas de aceptación, testeos automatizados y resolución de problemas y errores.

Como se mencionó anteriormente, estas son etapas que sirven de pauta y no se siguen necesariamente de manera lineal. Esto va a depender de cómo la metodología a utilizar interpreta al ciclo de vida del desarrollo del software (o SDLC, por sus siglas en inglés), que corresponde a la estructura definida para su desarrollo [30]. Algunos ejemplos de modelos incluyen el modelo en cascada, modelo incremental, modelo iterativo y el modelo ágil.

Estos principios y modelos de desarrollo pueden aplicarse a cualquier tipo de software, incluyendo a los servicios web. Si bien estos servicios ofrecen una serie de ventajas, el desarrollar una buena plataforma no es una tarea trivial. A través de los años se han definido diferentes pautas a seguir para asegurar un producto de

calidad. Por ejemplo, la red de desarrolladores de Mozilla (MDN) ha descrito un conjunto de buenas prácticas para el desarrollo web con tecnologías actuales [31].

La MDN señala que entre los aspectos importantes del desarrollo web está la compatibilidad, el nivel de desempeño de la plataforma y la claridad en la interfaz de usuario. En general, las interfaces de usuario deben ser simples, intuitivas y fluidas. Finalmente, la privacidad y seguridad de la plataforma son aspectos muy importantes. Los usuarios deben poder utilizar la plataforma de manera privada sin que se recolecte información no autorizada.

Sobre la base de estos principios, el presente proyecto de tesis describe el desarrollo de una plataforma de software llamada "PPI-MASS". Esta plataforma está orientada al post-procesamiento de datos de espectrometría de masas, con el propósito de filtrar proteínas usando distintos parámetros biológicos e identificar complejos proteína-proteína con potencial para el diseño de fármacos.

Motivación y objetivos

Motivación:

Actualmente, se requieren herramientas informáticas flexibles y accesibles para la identificación de interacciones proteína-proteína (PPIs) a partir de datos de espectrometría de masa (MS). Hasta ahora muchos investigadores analizan los resultados de MS de forma manual, es decir analizando una proteína a la vez, lo cual no es tarea fácil cuando se encuentran cientos o miles de proteínas asociadas a una proteína target. En este sentido, se requiere el desarrollo de una plataforma web para el análisis de datos de MS, que permita una detección racional de interacciones físicas entre proteínas con potencial farmacológico.

Objetivo general:

Diseñar y desarrollar una plataforma de servicio web para el postprocesamiento de datos provenientes de experimentos de MS con el fin de identificar interacciones proteína-proteína, utilizando como caso de estudio datos de MS asociados al interactoma de la proteína KCTD5.

Objetivos específicos:

1. Implementar un pipeline computacional para el post-procesamiento de datos provenientes de experimentos de espectrometría de masas.
2. Automatizar la identificación de interacciones proteína-proteína con potencial farmacológico mediante el uso de parámetros biológicos obtenidos de bases de datos de secuencia, estructura e información clínica.
3. Desarrollar una plataforma web que incorpore el pipeline diseñado y validar su uso a través del análisis del interactoma de la proteína KCTD5.

Metodología

1. Pipeline

Los datos de los experimentos de espectrometría de masas suelen ser almacenados en archivos de texto plano organizados en columnas. Dichos archivos pueden ser visualizados utilizando programas como Excel y contienen en cada columna información respecto a distintas proteínas encontradas en asociación con una proteína *target*. Entre la información incluida en estos archivos están los códigos de acceso de las proteínas a bases de datos como GenBank [32] y UniProt [33]. Estos códigos pueden estar muchas veces repetidos en el archivo de salida o incluso obsoletos, lo cual es importante considerar para el procesamiento de los datos. UniProt contiene información de miles de proteínas en dos secciones: Swiss-Prot para proteínas anotadas manualmente y revisadas (identificadas por el prefijo “sp” en los códigos de acceso); y TrEMBL para proteínas anotadas de manera automatizada (con prefijo “tr”). De manera general, UniProt es un repositorio de secuencias proteicas e información biológica derivada de investigación científica, el cual se ha convertido en una de las fuentes de información de proteínas más importantes a nivel mundial. En esta base de datos se puede encontrar información importante sobre las proteínas como el gen que las codifica, su localización celular, su tamaño, funciones específicas y la disponibilidad de estructura tridimensional. Por lo tanto, esta base de datos es la principal fuente de información para PPI-MASS.

Una vez verificados los códigos de acceso de las proteínas identificadas a partir de espectrometría de masas, éstos pueden ser utilizados para realizar búsquedas a través de referencias cruzadas en otras bases de datos conteniendo información biológica de relevancia para las proteínas en estudio. Considerando que el propósito final de la plataforma desarrollada es permitir al usuario identificar complejos proteicos con potencial farmacológico, se utilizaron además bases de datos que relacionan proteínas con ciertas patologías, fármacos asociados, acciones farmacológicas y que contienen una descripción de funciones específicas. Las bases de datos integradas incluyen DrugBank [34] y Drug Central [35]. Estas bases de datos online son algunas de las más robustas e incluyen información

clínica importante de fármacos y blancos farmacológicos, secuencias, estructuras, rutas metabólicas y acciones farmacológicas asociadas a proteínas. Por otro lado, la base de datos del Protein Data Bank (PDB) [36] provee estructuras tridimensionales para miles de proteínas. Para aquellos casos en que no hay ninguna estructura disponible, el servidor SWISS-MODEL [37] provee modelos de las proteínas obtenidos mediante modelamiento por homología. La información estructural es fundamental para la predicción de PPIs y en el diseño de fármacos, por lo que estos datos resultan de mucha utilidad para los investigadores.

De manera similar, con los códigos de acceso se pueden realizar búsquedas en otras bases de datos referidas al estudio de proteínas como DisGeNET [38] y The Human Protein Atlas [39]. DisGeNET es una base de datos de genes asociadas a enfermedades humanas, conteniendo más de un millón de asociaciones gen-enfermedad. Por otro lado, The Human Protein Atlas es una colección de información de proteínas humanas categorizadas en 6 diferentes atlas: tejido, tipo celular, patologías, proteínas sanguíneas, proteínas cerebrales y localización subcelular. En particular, el atlas de tejidos contiene información de proteínas y sus niveles de expresión en diferentes tejidos.

Para caracterizar las PPIs, es útil tener una idea de qué tan probable es que las proteínas interactúen. Esta estimación fue implementada en nuestra plataforma mediante el software SPRINT [40], el cual permite calcular un *score* en función de la probabilidad de que dos proteínas interactúen. SPRINT se basa en la similitud de secuencias y en evidencia de interacciones ya conocidas para predecir interacciones nuevas. El principio general de SPRINT radica en la idea de que un par de proteínas cuya secuencia es similar a otro par de proteínas que se sabe interactúan, tienen una mayor probabilidad de interactuar. El software identifica regiones similares en las secuencias para realizar la comparación y el cálculo del puntaje, donde un puntaje más alto indica una mayor probabilidad de interacción entre proteínas.

En base a esta información, se desarrolló el siguiente pipeline para el estudio de PPIs a partir de datos de MS:

1. Filtrar códigos de acceso, obteniendo una lista de valores únicos.
2. Utilizar los códigos para consultar UniProt y obtener información de las proteínas incluyendo el gen, nombre, proceso biológico, largo de secuencia, disponibilidad y cobertura de estructura tridimensional, y localización subcelular.
3. Utilizar códigos para consultar bases de datos médicas con el fin de obtener información como tejidos en los que se expresa una proteína, patologías asociadas y fármacos asociados. Las bases de datos consultadas incluyen The Human Protein Atlas para los tejidos, DisGeNET para las patologías, y Drug Central y DrugBank para los fármacos asociados.
4. Utilizar SPRINT para calcular el puntaje de interacción entre las proteínas asociadas a una proteína target, encontradas por experimentos de MS.
5. Generar una lista de proteínas candidatas para establecer interacciones físicas directas con la proteína target, mediante el uso de filtros interactivos en la interfaz web.

La metodología descrita se resume en la siguiente figura:

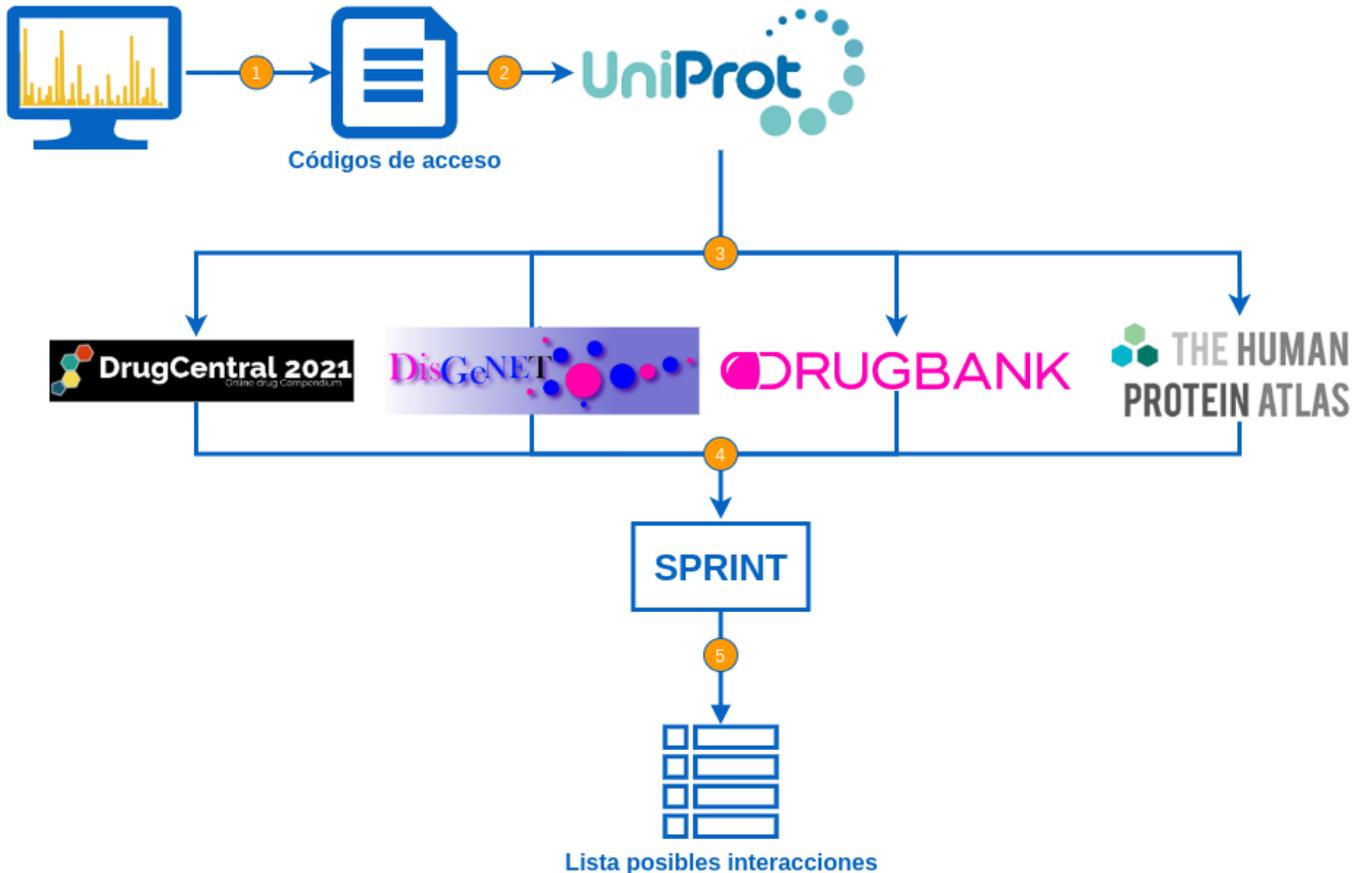


Figura 2: Pipeline propuesto para el estudio de PPIs a partir de datos de MS.

2. Desarrollo del software

De acuerdo con el contexto del proyecto, se consideró su implementación a través de un “modelo ágil”, el cual provee una opción flexible y dinámica respecto a otras metodologías. Este modelo se basa en entregas incrementales de forma progresiva [30], promoviendo la colaboración entre el cliente y los desarrolladores del software al centrarse en ciclos de trabajo cortos y reuniones constantes.

La forma incremental progresiva de esta técnica de ingeniería de software se explica del modo que asume que los requerimientos nunca son abordados totalmente desde un inicio y que por diversos motivos cambian constantemente con el tiempo. Es difícil predecir qué requerimientos de software persistirán y cuáles cambiarán. También es difícil pronosticar cómo cambiarán las prioridades del cliente a medida que avanza el proyecto. Esto otorga la ventaja de un valor organizacional prematuro, ya que requiere de muy poca planificación inicial para dar comienzo al desarrollo del software (codificación) y obtener un primer prototipo que permita establecer nuevos requerimientos futuros hasta alcanzar un producto final. De esta forma, el “modelo ágil” no precisa de un conocimiento detallado de los requerimientos del software *a priori*, sino que una idea general y posibles metas a lograr.

La construcción del producto se lleva a cabo mediante iteraciones autocontenidas, es decir, etapas cortas que contemplan una toma de requerimientos y una planificación específica para la iteración, además de objetivos y condiciones de cierre propios, logrando un entregable (prototipo del producto final usable) rápido. De esta forma, se puede planificar y tomar decisiones para un próximo paso con un elemento de software funcional, tangible y testeable. La duración de cada iteración generalmente es corta, lo que permite un ritmo de desarrollo rápido y fluido. Este modelo fomenta la comunicación continua con los clientes, permitiéndoles participar de manera más activa en el desarrollo y brindar retroalimentación continua, lo que asegura su satisfacción.

Dentro de los distintos enfoques del modelo ágil, se encuentra la filosofía de *Lean Software Development* (o LSD), la cual consiste en una adaptación del sistema de

producción de Toyota y propicia la reducción del trabajo que no aporta valor al resultado final [41]. Para ello, LSD postula los siguientes principios:

1. **Eliminar desperdicios/restos**, como por ejemplo funcionalidades innecesarias o no deseadas.
2. **Amplificar el aprendizaje** para tener una mejor base de conocimiento y un amplio abanico de opciones para desarrollar el software.
3. **Tomar decisiones lo más tarde posible** para minimizar el impacto de los cambios en las condiciones o requerimientos del proyecto.
4. **Entregar resultados rápidamente** y en etapas cortas para controlar mejor el flujo de progreso y asegurar la calidad en cada paso.
5. **Empoderar al equipo** para incentivar la participación de los desarrolladores en la toma de decisiones y el trabajo en equipo, integrando diferentes áreas de conocimiento.
6. **Integración continua** para mantener una arquitectura de proyecto coherente, incluyendo pruebas constantes para evaluar la calidad de cada etapa de avance.
7. **Visualizar todo el conjunto**, analizando las interacciones del software con otros software o entre sus diferentes componentes.

Estos principios incentivan un desarrollo de software limpio, fluido y flexible. En base a esta estrategia, se utilizó una metodología de “modelo ágil” para la planificación y el desarrollo de la plataforma PPI-MASS usando los principios de LSD. Concretamente, se siguió la siguiente metodología de desarrollo:

1. **Toma inicial de requerimientos.**
2. **Diseño de la arquitectura general de la plataforma.**
3. **Ciclos de desarrollo:**
 - a. Toma de requerimientos a implementar en la iteración.
 - b. Planificación y asignación de tareas específicas.
 - c. Codificación.
 - d. Pruebas de funcionamiento y usabilidad.
 - e. Evaluación y retroalimentación.
 - f. Integración al repositorio.

4. Pruebas de uso

5. Publicación de la plataforma.

La toma de requerimientos inicial correspondió al planteamiento de las características generales del software las cuales fueron definidas por el Dr. Oscar Cerda de la Universidad de Chile, quien proporcionó los estudios de espectrometría de masas para implementar una plataforma de análisis. El diseño de la arquitectura general tuvo como objetivo definir y organizar los componentes y módulos del software, como por ejemplo los componentes del pipeline o las diferentes secciones de la plataforma web. Las iteraciones fueron las etapas en las que se llevó a cabo el desarrollo y codificación del software con una duración de entre 1 y 4 semanas. Para cada iteración, se definieron requerimientos específicos a implementar, como lo es el diseño de la base de datos, el desarrollo de scripts para el pipeline o cambios en el aspecto visual de la plataforma. El alojamiento y control de versiones del código del proyecto fue llevado a cabo mediante un repositorio Git [42] proporcionado por GitLab inc. [43]. Cada avance fue testeado mediante pruebas de uso real, esto para asegurar el correcto funcionamiento de la plataforma. Luego, los resultados de cada iteración fueron evaluados para determinar si los requerimientos se cumplieron, en cuyo caso los avances fueron integrados al repositorio y, eventualmente, al servidor de producción. Los detalles del servidor se explican en la sección **Desarrollo**. Una vez terminada y validada cada iteración, el software resultante fue evaluado durante un periodo de prueba, validando su uso mediante el análisis de diferentes sets de datos, incluyendo el caso de uso a través del análisis del interactoma de la proteína KCTD5. La siguiente figura representa visualmente la metodología que se utilizó:

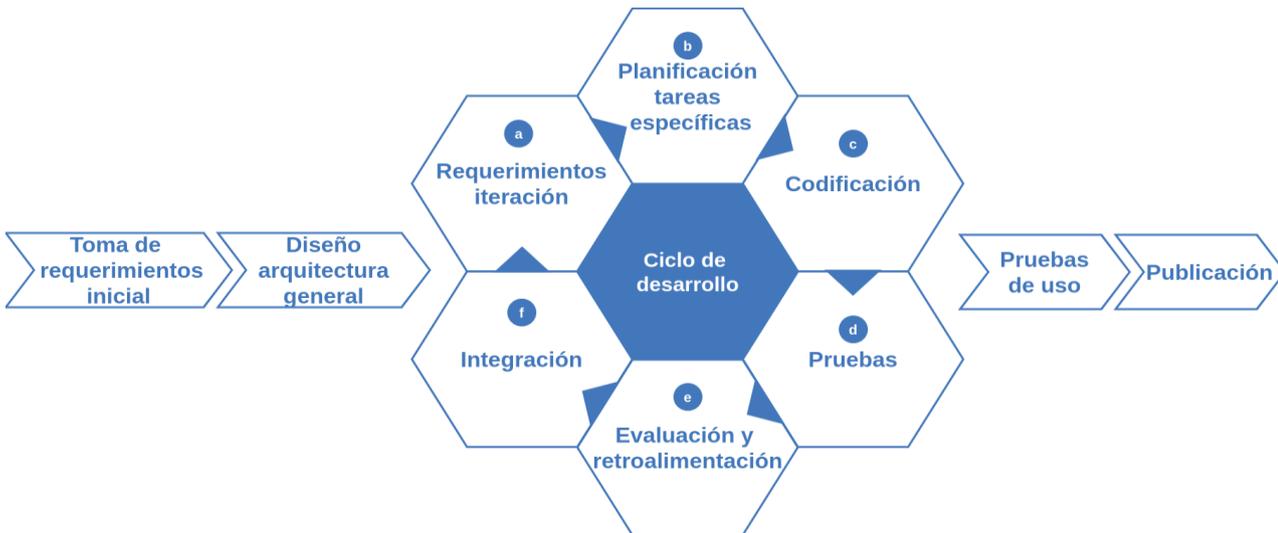


Figura 3: Metodología de desarrollo que se utilizó en el presente proyecto de tesis. Esta metodología se basa en el modelo ágil.

2.1 Implementación de la plataforma

Una plataforma web se puede dividir en dos conceptos: el *front-end* o el lado del cliente y el *back-end* o el lado del servidor. El *front-end* es la presentación visual de la plataforma con la que el usuario interactúa en su dispositivo. Se construye en base al lenguaje HTML, pudiendo personalizar el estilo visual con CSS y dar dinamismo a la interfaz con lenguajes de programación como Javascript. Por otro lado, el *back-end* es donde se realizan todas las tareas lógicas de programación e integración de procesos que debe llevar a cabo el servidor, como el procesamiento de datos o la generación del contenido que verá el usuario.

Normalmente los desarrolladores tienen que implementar por sí mismos diversos componentes, como conexiones con bases de datos o sistemas de autenticación; sin embargo, el desarrollo web moderno requiere de tiempos de trabajo cortos, por lo que el uso de *frameworks* se ha vuelto más predominante.

Un *framework* de desarrollo es un conjunto de herramientas que asisten en el desarrollo de software, definiendo metodologías y estructuras de proyectos específicas. Los *frameworks* ofrecen facilidades tales como la gestión de bases de datos, usuarios y archivos. Un ejemplo de *framework* de desarrollo web es Django [44], el cual está basado en el lenguaje de programación Python [45]. Django implementa una variante del patrón de arquitectura Modelo-Vista-Controlador denominada Modelo-Template-Vista [46]. El modelo es la capa de acceso a la base

de datos, donde se implementan métodos para ingresar, eliminar y actualizar registros de la base de datos. La vista representa la lógica principal del software, incluyendo el procesamiento de inputs y outputs. Finalmente, los *templates* constituyen la capa de presentación del software, es decir, los aspectos visuales y de interacción con el usuario. Por otro lado, la comunicación entre diferentes capas se implementa mediante *Middlewares*, los cuales son capas de software intermedias que proveen servicios comunes para que los desarrolladores puedan incorporar aplicaciones o desarrollos propios al *framework*, conservando la arquitectura, seguridad y desempeño propio del *framework* [47]. De esta forma, los *Middlewares* se encargan por ejemplo del procesamiento de solicitudes (*requests*) y respuestas (*responses*) del servidor. Django está pensado para acelerar el desarrollo de software, para lo cual cuenta con un sistema para el diseño y gestión de bases de datos basado en objetos (ORM), un simplificado sistema de *templates* y vistas para gestionar la presentación del contenido. Además, contiene módulos de gestión de usuarios y sesiones. Por otro lado, Django ofrece una capa intermedia de seguridad para gestionar la autenticación de usuarios, la encriptación de contraseñas y la prevención de posibles ataques de filtrado de datos no autorizado como la popular inyección de SQL [48]. Finalmente, Django define una arquitectura flexible, altamente escalable que separa el proyecto en módulos o *apps* con propósitos específicos. La modularidad permite mantener el código organizado y fácil de administrar, además de facilitar el crecimiento del software.

En base a lo expuesto anteriormente, en el presente proyecto de tesis se utilizó el *framework* de desarrollo web Django. Para el diseño estético (o *front-end*), se utilizó Bootstrap, el cual es un *framework* de diseño basado en CSS que permite diseñar sitios web responsivos de manera rápida junto a jQuery [49] para la generación de elementos dinámicos en la interfaz. jQuery es una librería de Javascript que permite manipular fácilmente el contenido del sitio web. Para la gestión de bases de datos, se utilizó PostgreSQL [50] que provee una serie de ventajas, como ofrecer un software *open source*, facilidad de instalación y configuración, soporte de JSON para almacenar datos con mayor flexibilidad y escalabilidad para el crecimiento de los sistemas. Concretamente, la plataforma web PPI-MASS permite al usuario enviar trabajos de análisis, los cuales son registrados en la base de datos y ejecutados en el servidor. Al completarse los trabajos, el usuario puede visualizar

los resultados mediante la plataforma. La figura 4 resume el patrón de arquitectura de un software de Django típico en función de las características del presente proyecto de tesis.

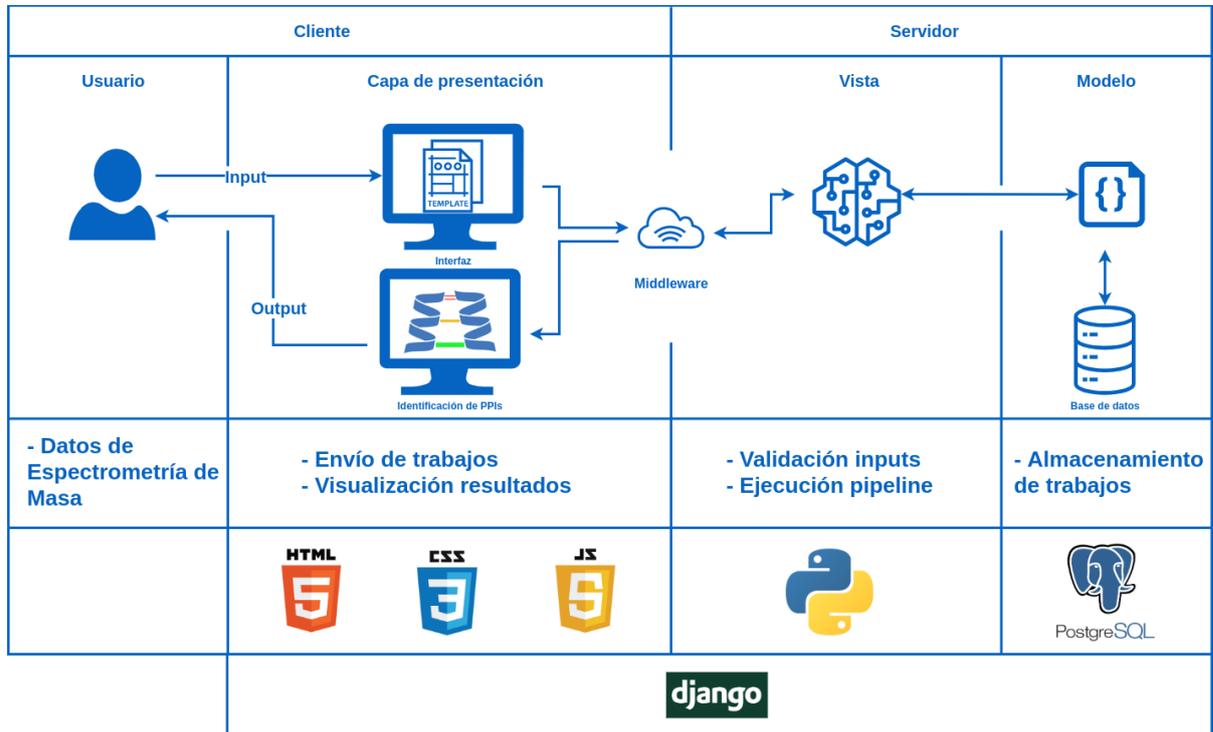


Figura 4: Patrón de arquitectura general de Django. La figura muestra cada capa de la arquitectura típica de una aplicación de Django. Además, se detalla la función de cada capa en el contexto del presente proyecto y las tecnologías asociadas en cada caso.

2.2 Software, tecnologías y versiones

La Tabla 2 lista los software y tecnologías que se utilizaron en el presente proyecto de tesis. Se especifican las versiones correspondientes a cada software y su caducidad, que corresponde a la fecha en la que cada versión dejará de recibir soporte y actualizaciones.

Software	Versión	Caducidad
Python3	3.8	Octubre 2024
Django	3.1	Diciembre 2021
PostgreSQL	10	Noviembre 2022

Git	2.27	-----
GitLab	14	Junio 2021
Bootstrap	5.0	-----
jQuery	3.5	-----

Tabla 2: Lista resumen de software y versiones que fueron utilizadas para el desarrollo del proyecto.

3. Análisis del interactoma de KCTD5

El interactoma de KCTD5, analizado a través de experimentos de espectrometría de masa en el laboratorio del Dr. Oscar Cerda, fue usado como caso de estudio para probar la plataforma PPI-MASS. Esta proteína ha sido asociada con cáncer de mama y la formación de ciertos tumores, particularmente cuando sus niveles de expresión son altos e interactúa con el canal iónico TRPM4 [22]. KCTD5 ha sido vinculada además con otras proteínas (Figura 6) y actualmente el Dr. Oscar Cerda cuenta con datos de espectrometría de masa para detectar nuevos candidatos estableciendo asociación física con esta proteína.

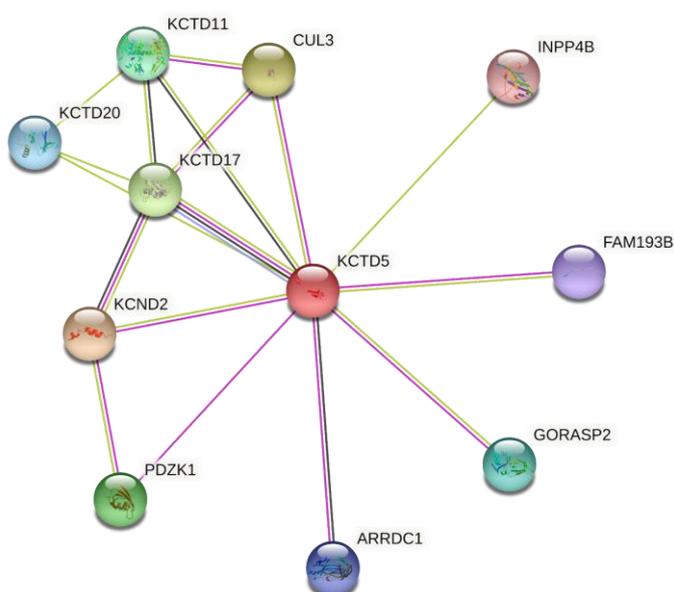


Figura 5: Red de interacción de KCTD5 obtenida mediante la base de datos STRING. La red incluye las proteínas que han sido hasta ahora reportadas en interacción con KCTD5.

A partir de experimentos llevados a cabo en nuestro laboratorio, se han identificado alrededor de 165 proteínas asociadas con KCTD5. Hasta ahora, se desconoce cuántas de estas proteínas establecen contacto físico directo con KCTD5, pudiendo actuar como blancos farmacológicos. La plataforma PPI-MASS fue utilizada para identificar las proteínas de este conjunto que presenten características afines para unirse a KCTD5 y que cuenten con modelos estructurales para realizar futuros análisis de bioinformática y simulación molecular. De particular interés fueron aquellos complejos asociados a patologías neurológicas, cardíacas y cáncer.

Desarrollo del software

1. Requerimientos

1.1 Requerimiento general

Dada la dificultad del post procesamiento de datos provenientes de experimentos de espectrometría de masas y la escasez de herramientas de software libre que permitan realizar este procedimiento de manera sencilla, flexible y transparente, se requirió el desarrollo de una herramienta informática para realizar el análisis de estos datos y ayudar a la predicción de interacciones proteína-proteína.

1.2 Requerimientos específicos

A continuación se presentan los requerimientos del proyecto. Esto representa las características necesarias para la ejecución de la plataforma PPI-MASS.

Requerimiento	Descripción
R1	Procesar una lista de proteínas proveniente de experimentos de espectrometría de masas.
R2	Identificar la proteína target del experimento.
R3	Recopilar información sobre las proteínas de entrada a partir de las bases de datos UniProt, PDB/SWISS-MODEL, The Human Protein Atlas (tissues), DisGeNET, DrugBank y Drug Central.

R4	Calcular el puntaje de interacción de SPRINT para cada proteína de entrada con respecto a la proteína target del experimento.
R5	Permitir al usuario acceder a los resultados de sus trabajos en cualquier momento.
R6	Permitir al usuario contactar y obtener soporte a través de la plataforma.
R7	Presentar resultados al usuario de manera sencilla y ordenada.
R8	Permitir al usuario filtrar proteínas de acuerdo a la información obtenida de las bases de datos.
R9	Permitir al usuario exportar resultados con los filtros aplicados.
R10	Implementar un manual de usuario en la plataforma.

Tabla 3: Lista de requerimientos para el desarrollo del software PPI-MASS

2. Implementación

La implementación y la relación de cada componente del proyecto se puede resumir como se muestra en la siguiente figura:

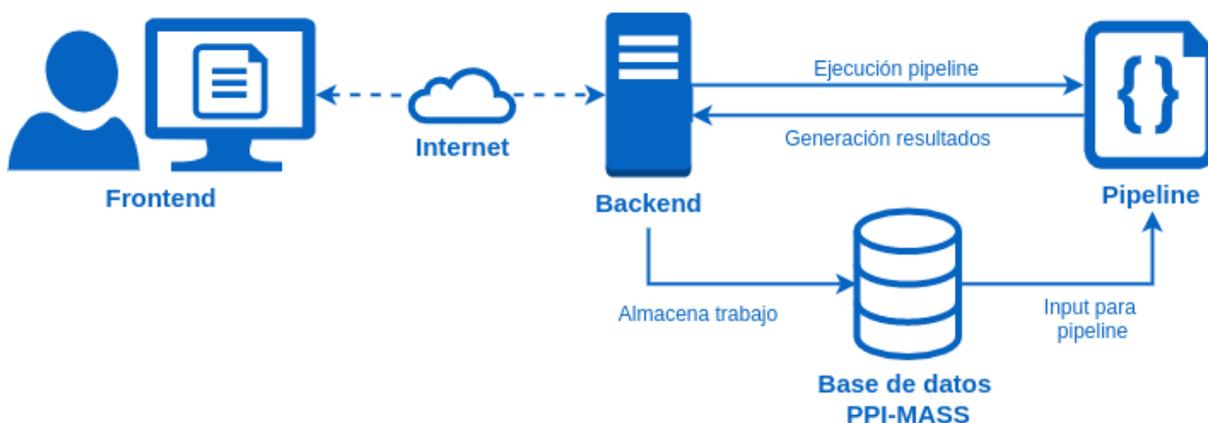


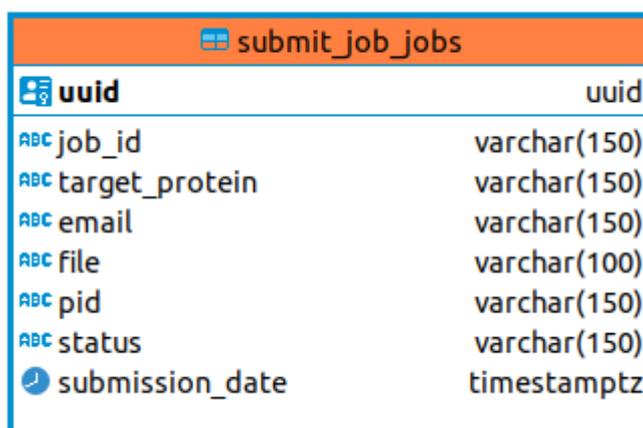
Figura 6: Comunicación entre los diferentes componentes de PPI-MASS. 1) El usuario envía un trabajo de análisis al servidor mediante la interfaz del *frontend*. 2) El *backend* de PPI-MASS registra el trabajo en la base de datos. 3) El servidor ejecuta el pipeline de análisis, el cual utiliza como input los datos almacenados en la base de datos. 4) El *backend* finalmente obtiene los resultados generados por el pipeline, los cuales pueden ser visualizados por el usuario en la interfaz.

2.1 Base de datos

La base de datos principal de PPI-MASS fue implementada en PostgreSQL, el cual es un sistema de gestión de bases de datos relacionales. Dado que sólo se requiere almacenar los trabajos enviados por los usuarios, la base de datos consta de una sola tabla. Las columnas de esta tabla son:

- **uuid:** Identificador único universal que se utiliza para identificar trabajos específicos.
- **job_id:** Nombre que el usuario provee para el trabajo.
- **target_protein:** Proteína target del experimento de MS.
- **email:** Dirección de correo electrónico provista por el usuario para el envío de los resultados.
- **file:** Archivo que contiene la lista de proteínas que se obtienen como resultado del experimento de MS.
- **pid:** Código que identifica el proceso de ejecución del trabajo.
- **status:** Estado de ejecución del trabajo. Por ejemplo: Pendiente, En ejecución, Listo o Fallido.
- **submission_date:** Hora y fecha de envío del trabajo.

La tabla de la base de datos se resume en la siguiente figura:



submit_job_jobs	
uuid	uuid
job_id	varchar(150)
target_protein	varchar(150)
email	varchar(150)
file	varchar(100)
pid	varchar(150)
status	varchar(150)
submission_date	timestampz

Figura 7: Modelo de la base de datos de PPI-MASS.

2.2 Servidor

PPI-MASS se alojó en un servidor linux con un procesador Intel Xeon 2.4 GHz, 6 GB de memoria RAM y un procesador de 4 núcleos. Para la ejecución de los

trabajos almacenados en la base de datos de PPI-MASS, se configuró el servidor para buscar y ejecutar los trabajos pendientes cada un minuto.

PPI-MASS requiere de implementaciones locales de las bases de datos UniProt, The Human Protein Atlas (tissues), DrugBank, Drug Central y DisGeNET para su funcionamiento. En el caso de UniProt y DrugBank, PPI-MASS utiliza versiones en JSON de estas bases de datos, las cuales fueron generadas con Python a partir de los archivos XML que pueden ser descargados desde sus respectivos sitios web. The Human Protein Atlas provee archivos TSV (*tab separated values*), de los cuales se descargó el archivo de *normal tissues*, que es donde PPI-MASS busca la información. Drug Central tiene una implementación en PostgreSQL, la cual fue descargada desde su sitio web y configurada localmente en el servidor. Finalmente, DisGeNET tiene una implementación en SQLite, la cual fue descargada en el servidor desde su sitio web.

PPI-MASS reporta la disponibilidad de estructuras tridimensionales para cada proteína en la lista de análisis, además de su porcentaje de cobertura. Para ello, se descargaron todas las estructuras de proteínas humanas desde Protein Data Bank (PDB) y los modelos desde SWISS-MODEL. Para cada estructura, se contó la cantidad de residuos y se calculó el porcentaje de cobertura con respecto al largo de secuencia obtenido desde la información de UniProt. Finalmente, los códigos de las estructuras de cada proteína y sus respectivas coberturas fueron almacenados en un archivo JSON, el cual es consultado por PPI-MASS para entregar estos datos en los resultados.

El software también utiliza SPRINT para calcular el puntaje de interacción de las proteínas de entrada con respecto a la proteína target. Para ello se descargó y compiló manualmente el código fuente de SPRINT en el servidor. Este programa requiere como entrada las secuencias FASTA de las proteínas. Para ello, se descargaron las secuencias de todas las proteínas humanas y se almacenaron en un archivo multi-FASTA, a partir del cual se obtienen las secuencias necesarias para cada trabajo. Por otro lado, SPRINT requiere un set de entrenamiento que corresponde a las interacciones ya conocidas. Dicho set de entrenamiento se obtuvo a partir de las interacciones depositadas en la base de datos BioGRID [51].

BioGRID es una de las bases de datos de interacciones proteína-proteína más robustas y completas, por lo cual es una excelente fuente de información para este propósito. Pese a ello, se contempla incorporar más bases de datos de interacciones en el futuro.

2.3 Pipeline

PPI-MASS.py

Descripción: Script principal del software PPI-MASS. Se encarga del procesamiento de las proteínas de entrada, la búsqueda de información en las bases de datos y la predicción de interacciones con SPRINT. Este script recibe como parámetros de entrada la información del trabajo desde la base de datos.

En primera instancia, la lista de códigos UniProt es validada. Un código podría ser considerado inválido si es un código obsoleto, incorrecto o correspondiente a otro organismo que no sea humano. Luego, se generan estadísticas que son reportadas al usuario en la plataforma durante la ejecución del trabajo, como la cantidad de proteínas válidas encontradas y la lista de códigos inválidos en caso de que el usuario desee revisarlos manualmente.

Luego, se comienza a recopilar la información desde las bases de datos para cada proteína. Esta información es almacenada en un archivo de salida, cuyo contenido es el que se muestra al usuario en la página de resultados a través de una tabla HTML interactiva.

Finalmente, se obtiene la secuencia FASTA de cada proteína de entrada y se ejecuta SPRINT para calcular el puntaje de interacción de cada proteína con respecto a la proteína target.

El procedimiento se resume en el siguiente diagrama de flujo:

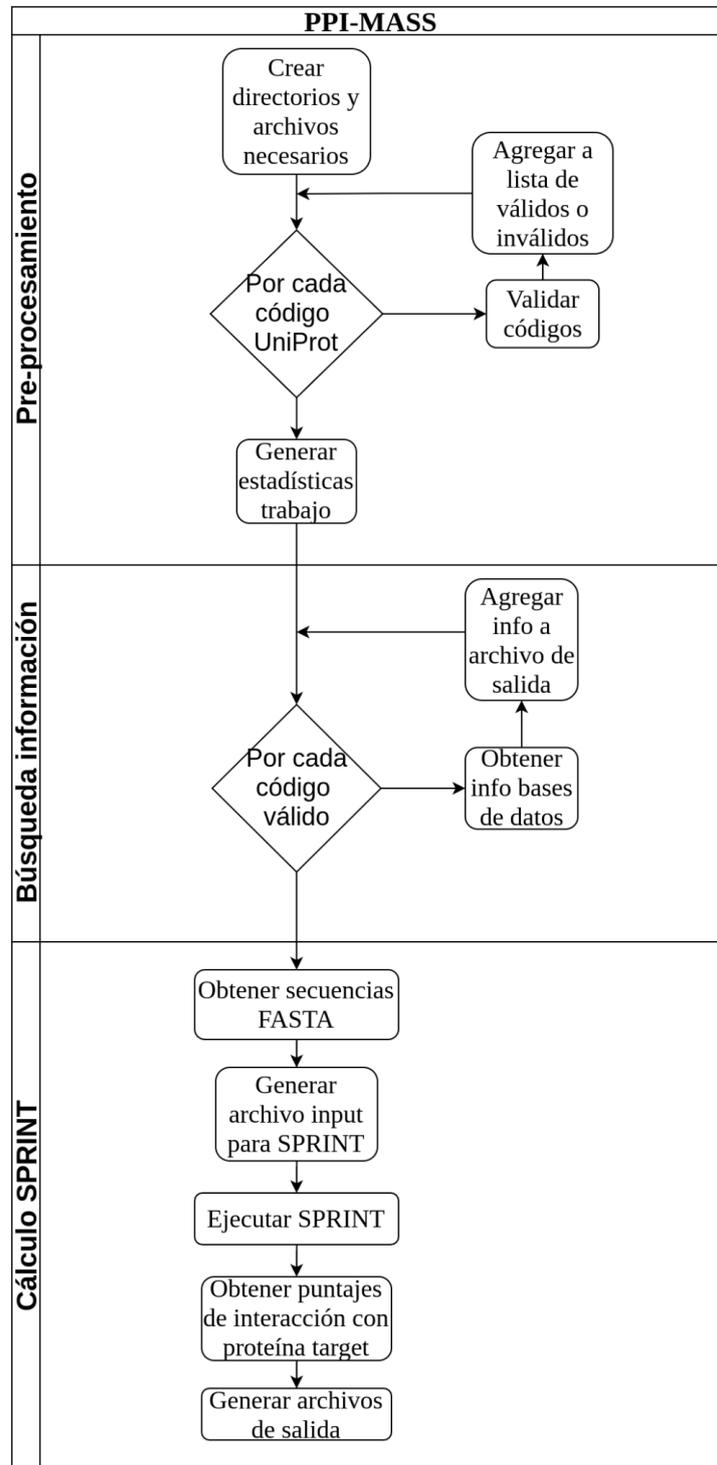


Figura 8: Diagrama de flujo del análisis de las proteínas de entrada. El procedimiento se divide en tres partes principales: Pre-procesamiento, búsqueda de información y cálculo del puntaje de interacción con SPRINT.

conexion.py

Descripción: Script en Python que busca los trabajos pendientes en la base de datos y llama al script principal para ejecutarlos, creando hilos para cada uno. Este script es ejecutado en el servidor cada un minuto mediante un *cron*.

config.py

Descripción: Genera las conexiones para las bases de datos de Postgres, tanto para la base de datos de PPI-MASS, como para la base de datos de Drug Central.

2.4 Arquitectura del proyecto Django

A continuación se describe la implementación de la plataforma web y sus distintos componentes.

Un proyecto típico de Django organiza el código en *apps*, las cuales son módulos con funcionalidades específicas. Cada *app* sigue el patrón de Modelo-Vista-Template, tal como se indica en la sección de **Metodología**. La plataforma web de PPI-MASS es sencilla y se compone de cinco *apps*: **main**, **submit_job**, **results**, **contact** y **user_guide**.

La *app* **main** es el punto de ingreso a la plataforma PPI-MASS. Tiene como función dar la presentación al sitio y entregar acceso a información rápida al usuario sobre la plataforma.

Submit_job es la *app* que se encarga del envío y registro de los trabajos para la plataforma. Permite el envío de trabajos a través de un formulario que es registrado en una base de datos para los trabajos.

Results es la *app* más grande de PPI-MASS. Permite la búsqueda de los resultados de cada trabajo, el monitoreo del progreso, la presentación y exportación de los resultados

Contact y **user_guide** son las *apps* más simples. **Contact** es un formulario de contacto de los usuarios hacia el desarrollador y **user_guide** un acceso directo al manual de uso de la plataforma.

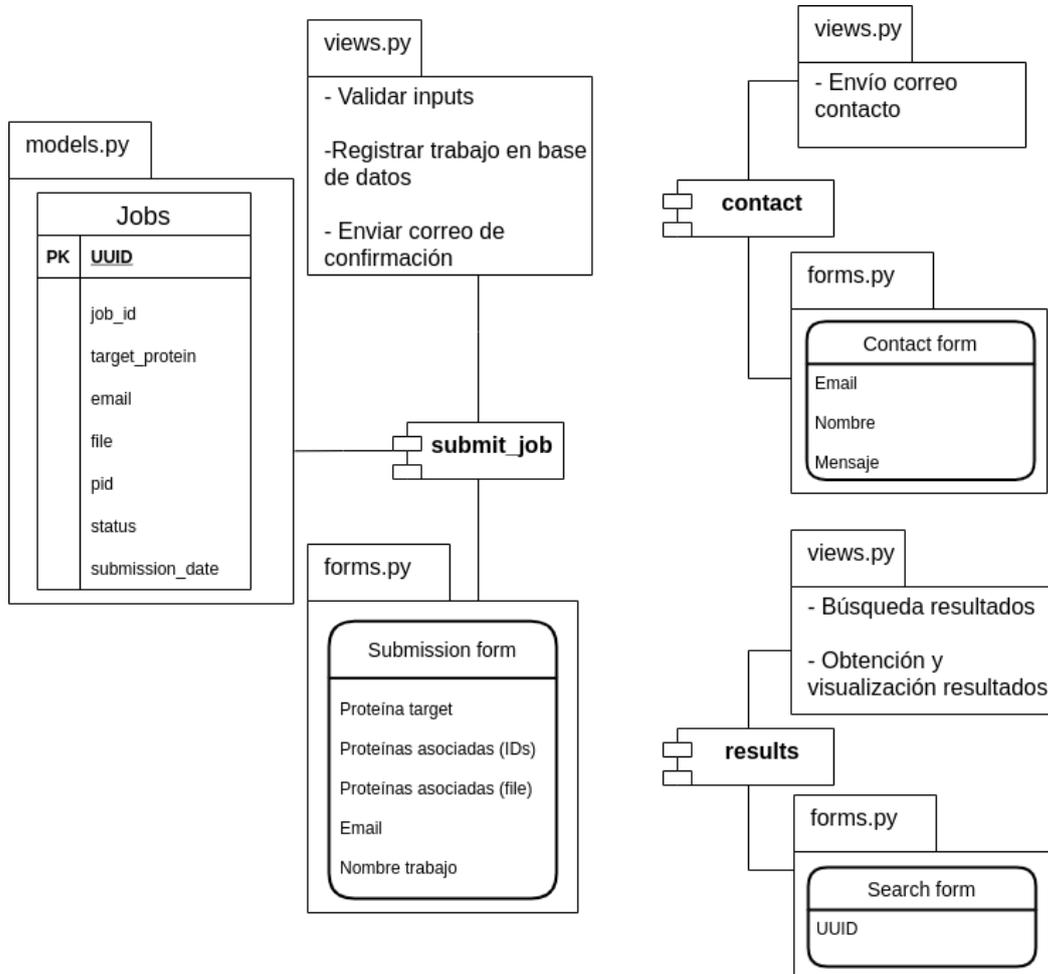


Figura 9: Diagrama de la arquitectura de la plataforma web en el *framework* Django. El diagrama describe las *apps* y sus componentes principales, mostrando sus funciones específicas. Las *apps* **main** y **user_guide** sólo contienen los *templates* HTML, por lo que se omitieron en este diagrama.

3. Plataforma web

3.1 Secciones de la plataforma web

A continuación se describen las secciones y componentes principales de la plataforma web PPI-MASS. El link <https://minicad.appsbio.utalca.cl/ppi-mass/> ha sido configurado para acceder a la plataforma.

Página de inicio

La página de inicio es el punto de entrada a la plataforma web. Tiene la función de informar rápidamente al usuario sobre el software y de presentar un *call to action* para invitar al usuario a enviar un trabajo. Además, presenta información sobre el grupo MINICAD y el financiamiento del proyecto.

The screenshot shows the homepage of the PPI-MASS website. At the top, there is a dark blue header with the PPI-MASS logo and tagline: "Protein-protein interactions from mass spectrometry-based proteomics data". Navigation links for Home, Submit Job, Results, Contact, and User guide are visible in the top right corner.

The main content area features two columns of information:

- Mass Spectrometry:** Accompanied by a monitor icon showing a mass spectrum. Text: "Mass spectrometry-based proteomics represents an accurate and sensitive strategy for identifying protein-protein interactions."
- Analysis:** Accompanied by a magnifying glass icon. Text: "PPI-MASS offers a web platform to guide users in filtering a protein set obtained from MS data and finding protein-protein interactions (PPIs) with pharmacological potential."

A "Submit a job!" button is centered below these two columns.

Below this is an "About us" section with a horizontal line. It contains two main blocks:

- MINICAD:** Text: "The Millennium Nucleus of Ion Channel-Associated Diseases (MINICAD) constitutes an original multistrategic initiative addressed to develop and design novel and innovative therapeutic tools for ion channel-related diseases. Our research philosophy relies on the premise that ion channels proteins constitute non-canonical, but essential signaling molecules part of the complexity of the cellular and systemic physiology. Thus, our main philosophy is to study those mechanisms not only in a biophysical stand-off-view, but in a cellular and physiological context." A "Visit website" link with a right-pointing arrow is provided.
- MiNICAD:** Logo for the Millennium Nucleus of Ion Channel Associated Diseases.

Below the "About us" section is another horizontal line, followed by two more blocks:

- Logos:** Logos for the Agencia Nacional de Investigación y Desarrollo (ANID) and the Milenio Iniciativa Científica.
- Iniciativa Milenio:** Text: "The Millennium Science Initiative is a program of the National Agency of Research and Development (ANID, www.anid.cl) at the Chilean Ministry of Science, Technology, Knowledge and Innovation." "The Millennium Science Initiative funds the creation and development of research centers of excellence, in the areas of social sciences and natural and exact sciences, which are awarded through public competitions, considering their scientific merits. These centers are divided into Millennium Institutes and Millennium Nuclei, which differ mainly, by the number of researchers, financing periods and financing amount." A "Visit website" link with a right-pointing arrow is provided.

The footer of the page is a dark blue bar containing the logos of ANID, Milenio, and MiNICAD.

Figura 10: Página de inicio de PPI-MASS.

Envío de trabajos

Esta página tiene la función de permitir al usuario enviar trabajos para su ejecución en el servidor. Tiene el formulario de envío de trabajos, el cual se muestra en la siguiente figura. Los inputs de este formulario se describen en la sección de **Modo de uso**.

Post-processing of MS data to identify protein-protein interactions Load example data

*: Mandatory field

Target protein*

UniProt ID (e.g. Q8TD43)

Associated proteins

UniProt IDs (e.g. protein set identified by mass spectrometry-based proteomics data) separated by commas or newlines

-OR- Click here to upload a file (flat text) with the UniProt IDs separated by newlines
 Examinar... Ningún archivo seleccionado.

Email address*

Email to which the job URL will be sent

Job name*

e.g. project name

Run PPI-MASS Reset

Figura 11: Formulario de envío de trabajos.

Resultados

Esta sección se compone de dos páginas. La primera es la búsqueda de trabajos. Esta página permite al usuario buscar sus trabajos en base al UUID y acceder a los resultados o ver el progreso de su trabajo.

Search job results

Job UUID*

Search!

Figura 12: Búsqueda de resultados mediante el UUID del trabajo.

La siguiente página muestra los resultados como tal. En esta página el usuario podrá visualizar e interactuar con los datos para cada proteína. El componente principal es la tabla interactiva que presenta la información recolectada para cada proteína que el usuario ingresó. Permite filtrar los resultados en base a la información recopilada y exportarlos en formatos Excel (XLSX), PDF y CSV. En la sección **Modo de uso** se explica su funcionamiento en mayor detalle.

Job: **Example_data** Download data:   

Status: Finished
 UUID: 909d93ac-facb-4b2d-b936-a3d64ebbe559
 Proteins found: 19

[Clear filters](#)

Show entries
 Showing 1 to 10 of 19 entries Previous [1](#) [2](#) Next

UniProtKB entry	Gene	Protein name	Biological process	Sequence length	Structure availability	Subcellular location	Tissue specificity	Associated pathologies	Associated drugs	SPRINT score
UniProtKB entry	Gene	Protein name	Biological process		PDB S-M N/A	Subcellular location	Tissue specificity	Associated path	Associated d	
P01111	NRAS	GTPase NRas	YES <input type="checkbox"/>	189	PDB	<input type="checkbox"/> Golgi apparatus membrane Cell membrane	YES <input type="checkbox"/> →	YES <input type="checkbox"/> →	N/A	547.941
P01112	HRAS	GTPase HRas	YES <input type="checkbox"/>	189	PDB	<input type="checkbox"/> Golgi apparatus Cytoplasm Cell membrane Perinuclear region Nucleus Golgi apparatus membrane	YES <input type="checkbox"/> →	YES <input type="checkbox"/> →	YES <input type="checkbox"/>	546.717
P01116	KRAS	GTPase KRas	YES <input type="checkbox"/>	189	PDB	<input type="checkbox"/> Cytosol Cytoplasm Cell membrane	YES <input type="checkbox"/> →	YES <input type="checkbox"/> →	N/A	542.617
P35637	FUS	RNA-binding protein FUS	YES <input type="checkbox"/>	526	PDB	<input type="checkbox"/> Nucleus	YES <input type="checkbox"/> →	YES <input type="checkbox"/> →	N/A	441.101

Figura 13: Página de resultados.

Formulario de contacto

Esta página contiene un formulario de contacto. Esto permitirá al usuario contactar al equipo a cargo del software mediante un correo electrónico. De esta forma, el usuario podrá resolver dudas o recibir soporte.

Contact us!

Your Email*

Your name

Your message*

Figura 14: Formulario de contacto.

Manual de usuario

Esta página contiene información detallada respecto al contenido y uso de PPI-MASS. El manual permite al usuario conocer sobre la plataforma, aprender cómo enviar trabajos, además de cómo visualizar y filtrar resultados.

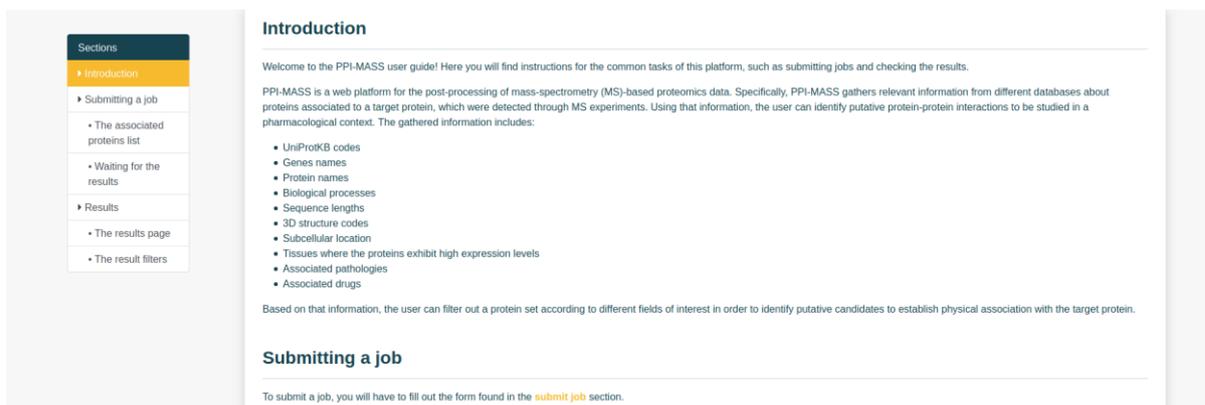


Figura 15: Manual de uso de la plataforma PPI-MASS.

3.2 Modo de uso

Envío de trabajos

Para el envío de trabajos se requieren los siguientes datos como entrada:

- 1. El código UniProt de la proteína target**
- 2. Una lista de proteínas asociadas**
- 3. Una dirección de correo electrónico**
- 4. Un nombre para el trabajo**

El código de la proteína target corresponde al código UniProt de la proteína usada como base del experimento de MS y es utilizado para calcular el puntaje de interacción con las demás proteínas utilizando SPRINT.

La lista de proteínas asociadas corresponde a la lista de proteínas obtenida a partir del experimento de MS, de las cuales se pretende descubrir si establecen interacciones físicas con la proteína target. Esta lista se compone de los códigos UniProt de cada proteína. El usuario tiene dos opciones para proveer esta lista, i) ingresarlas directamente al campo del formulario, separadas por comas, espacios o en diferentes líneas, o ii) subir un archivo de texto plano con los códigos separados en diferentes líneas. Si el usuario selecciona un archivo para subir a la plataforma, el campo del formulario para ingresar los códigos directamente será desactivado, por lo que el archivo tomará prioridad. Una manera sencilla de generar este archivo (en caso de que los resultados del experimento de MS estén en un

archivo Excel) es copiando y pegando las celdas directamente a un archivo de texto plano.

Al enviar el trabajo, la plataforma generará un link para que el usuario pueda acceder a los resultados en cualquier momento. La dirección de correo electrónico debe ser provista para enviar este link.

Por último, el usuario debe proporcionar un nombre para su trabajo. Esto es principalmente para que el usuario pueda diferenciar múltiples trabajos de manera fácil.

Visualización de resultados

Una vez enviado el trabajo, la plataforma redirigirá al usuario a la página de resultados. Cuando el trabajo aún no está listo, esta página mostrará el estado y progreso del trabajo. La página se refrescará automáticamente cada 30 segundos para actualizar esta información hasta que el trabajo esté listo. Una vez listo, se presentarán los resultados en una tabla interactiva. Dicha tabla contiene las siguientes columnas:

1. **Código UniProt:** El código de acceso de UniProt para la proteína.
2. **Gen:** El nombre del gen que codifica para la proteína.
3. **Nombre de la Proteína:** Nombre común de la proteína.
4. **Proceso Biológico:** Proceso biológico en el que se ve involucrada la proteína.
5. **Largo de secuencia:** Número de aminoácidos de la proteína.
6. **Disponibilidad de estructura:** Las estructuras tridimensionales (PDB) o modelos (SWISS-MODEL) disponibles para la proteína. Además, se muestra el porcentaje de cobertura en cada caso.
7. **Localización subcelular:** Localización donde se puede encontrar la proteína en la célula.
8. **Especificidad de tejidos:** Tejidos humanos en donde la proteína tiene un nivel de expresión alto, según se especifica en The Human Protein Atlas.
9. **Patologías asociadas:** Patologías asociadas al gen de la proteína, según se especifica en DisGeNET.

10. Fármacos asociados: Fármacos que tienen como target a la proteína, obtenidos a partir de DrugBank y Drug Central.

11. Puntaje SPRINT: Puntaje de interacción calculado utilizando SPRINT. Mientras más alto, más probable es que la proteína establezca contacto físico con la proteína target.

Algunos campos suelen contener demasiada información como para mostrarla de manera óptima en la tabla, en cuyo caso se muestra un botón (con un ícono de una *i*) que permite presentar esta información en una ventana emergente. Para estos campos, los filtros actúan sobre el texto en estas ventanas emergentes. Por otro lado, los botones con el ícono de flecha tienen un hipervínculo hacia la fuente desde la cual se obtuvo la información correspondiente.

UniProtKB entry	Gene	Protein name	Biological process	Sequence length	Structure availability	Subcellular location	Tissue specificity	Associated pathologies	Associated drugs	SPRINT score
P01111	NRAS	GTPase NRas	YES <i>i</i>	189	PDB	<i>i</i> Golgi apparatus membrane Cell membrane	YES <i>i</i> →	YES <i>i</i> →	N/A	547.941
P01112	HRAS	GTPase Hras	YES <i>i</i>	189	PDB	<i>i</i> Golgi apparatus Cytoplasm Cell membrane Perinuclear region Nucleus Golgi apparatus membrane	YES <i>i</i> →	YES <i>i</i> →	YES <i>i</i>	546.717
P01116	KRAS	GTPase Kras	YES <i>i</i>	189	PDB	<i>i</i> Cytosol Cytoplasm Cell membrane	YES <i>i</i> →	YES <i>i</i> →	N/A	542.617
P35637	FUS	RNA-binding protein FUS	YES <i>i</i>	526	PDB	<i>i</i> Nucleus	YES <i>i</i> →	YES <i>i</i> →	N/A	441.101

Figura 16: Visualización de resultados en PPI-MASS. 1) Información sobre el trabajo, incluyendo su estado, identificador (UUID), proteínas encontradas y códigos de proteínas inválidos. 2) Botones para exportar resultados. 3) Tabla interactiva y filtros asociados a los campos principales. 4) Información de cada proteína analizada. Los botones con un ícono de una *i* despliegan la información del respectivo campo en ventanas emergentes. Los botones con un ícono de flecha redirigen hacia la fuente desde la cual se obtuvo la información en ese campo.

Las proteínas pueden ser filtradas según esta información. Al ingresar texto en los campos, la tabla automáticamente se actualizará para mostrar las proteínas cuya información coincida con el texto ingresado. Las búsquedas pueden ser de texto

parcial e incluso se pueden utilizar expresiones regulares para búsquedas más avanzadas.

En la esquina superior izquierda se muestra información sobre el trabajo. Concretamente, se muestra el estado, el UUID del trabajo, la cantidad de proteínas válidas encontradas y la lista de códigos inválidos. Por otra parte, en la esquina superior derecha se encuentran botones para exportar los resultados en formatos XLXS, CSV y PDF. Al hacer click en alguno de estos botones, el usuario podrá descargar un archivo conteniendo los resultados del trabajo con cualquier filtro que haya aplicado. Finalmente, el botón *clear filters* permite al usuario reiniciar la tabla a su estado inicial, borrando todos los filtros aplicados.

Así, el uso del software PPI-MASS se puede resumir como se muestra en la figura siguiente:

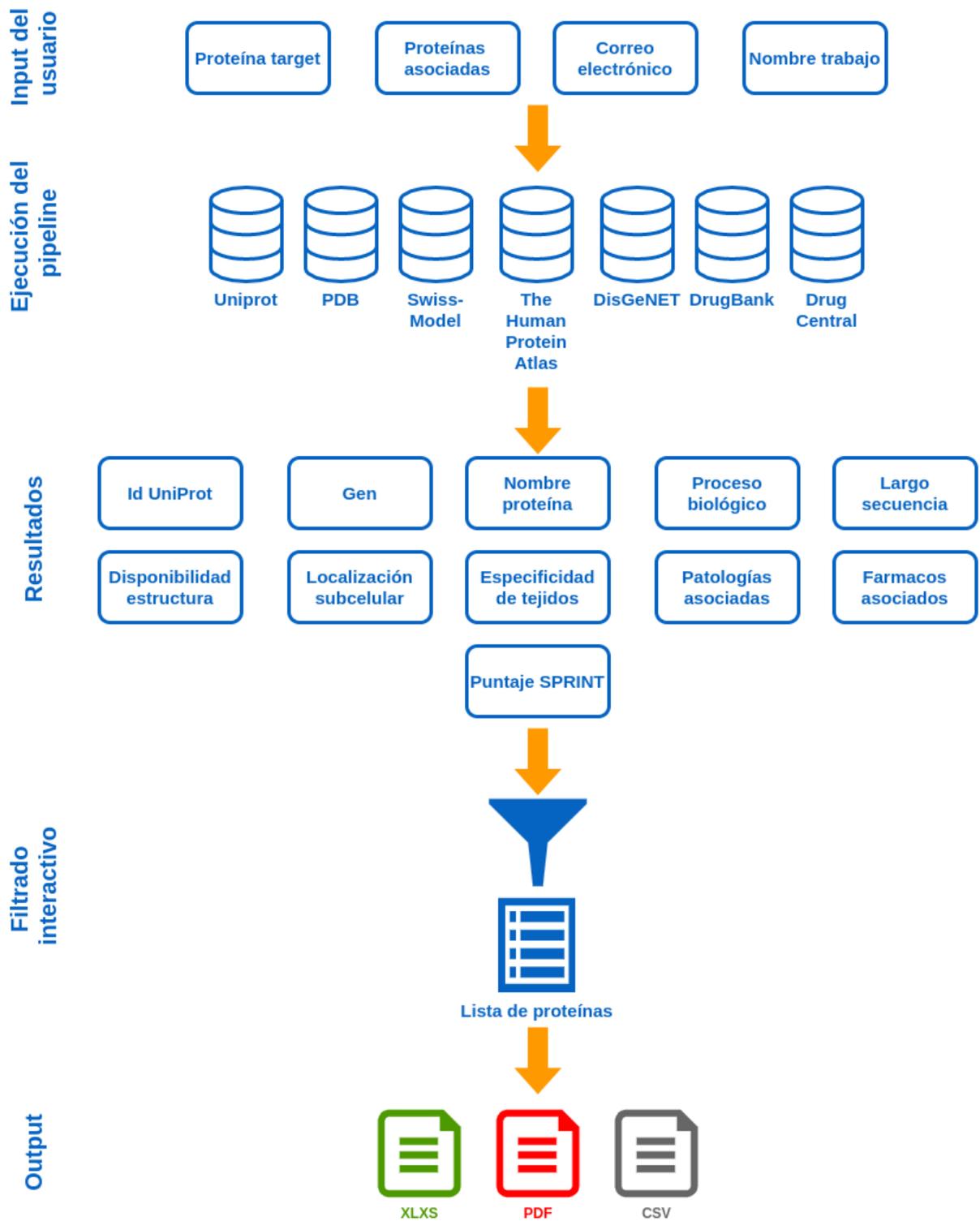


Figura 17: Diagrama del uso del software PPI-MASS.

Caso de estudio

En el presente proyecto de tesis se desarrolló una plataforma web, PPI-MASS, para la implementación y ejecución de un pipeline de análisis de datos provenientes de experimentos de MS. Se utilizó el set de datos de proteínas asociadas a KCTD5⁴, provenientes de experimentos de MS, para validar el funcionamiento del software desarrollado. Dicho set de datos constaba de 930 proteínas, de las cuales 11 correspondían a proteínas de organismos diferentes al humano, por lo que fueron ignoradas por el software. El análisis de estos datos tomó alrededor de 25 minutos. Entre las 919 proteínas válidas del set de datos, se encontraron varias proteínas cuyas interacciones con KCTD5 ya habían sido reportadas previamente en la literatura. En particular, la proteína con el puntaje de SPRINT más alto fue la proteína Cullin 3, la cual ya había sido reportada en interacción con KCTD5 por Balasco et al. en 2014 [52] y Cho et al. en 2020 [53], validando el potencial predictivo del software.

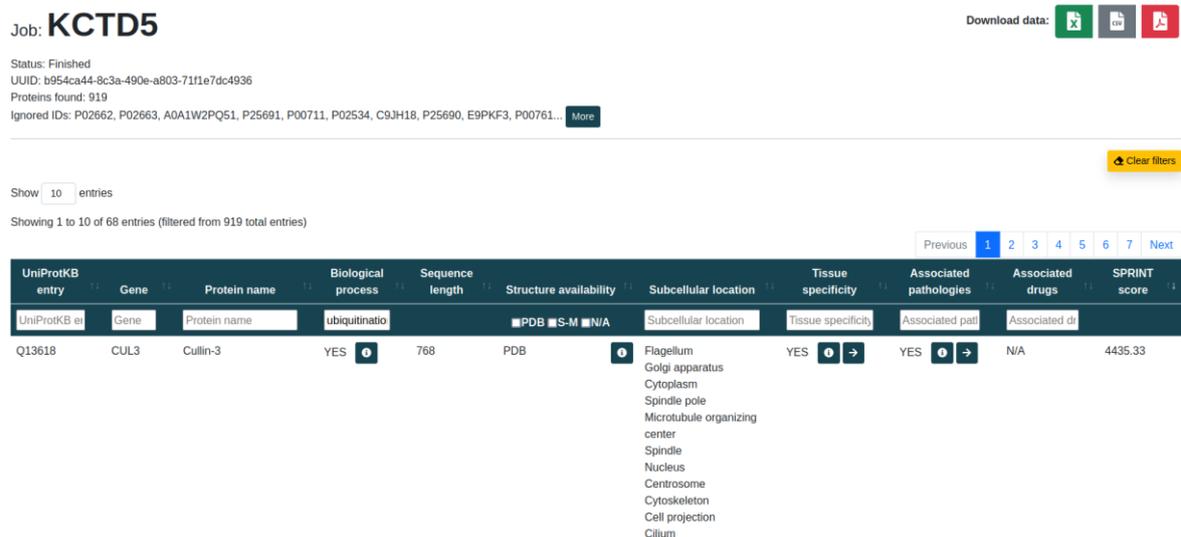


Figura 18: Resultado del análisis de la proteína KCTD mediante PPI-MASS. Se filtró por el término “ubiquitination” en el filtro de *Biological process*, lo que redujo el set de proteínas de 919 a sólo 68. Entre las proteínas resultantes se destaca la proteína Cullin 3.

Durante el desarrollo del proyecto, se utilizó también un set de datos de MS del canal iónico TRPM4⁵, el cual constaba de 96 proteínas. De entre ellas, las que

⁴ <https://minicad.appsbio.usalca.cl/ppi-mass/results/01be82e8-47bc-47e4-bead-76335766a150>
⁵ <https://minicad.appsbio.usalca.cl/ppi-mass/results/c6246a2f-2fd2-406b-841d-06843b1fa86b>

presentaron mayores puntajes de SPRINT fueron las proteínas de la familia 14-3-3. Particularmente, la proteína 14-3-3 gamma ha sido reportada en interacción con el canal TRPM4 por Cho et al. en el año 2014 [54], donde se observó que su expresión en la superficie celular es mediada por la unión con 14-3-3 gamma.

Estos resultados dan cuenta de la utilidad de PPI-MASS para identificar PPIs y del potencial de esta plataforma para el estudio de datos de MS de manera sencilla y práctica. A partir del trabajo realizado en este proyecto de tesis, se publicó un artículo en el *journal* *Frontiers in Molecular Biosciences* [55].

Conclusiones

La identificación de PPIs es un paso fundamental para el estudio de enfermedades y para el diseño de fármacos. PPI-MASS representa una herramienta con un gran potencial para ayudar en el proceso de analizar datos de MS para el descubrimiento de interacciones proteína-proteína. A través de los puntajes de SPRINT se lograron encontrar interacciones ya reportadas en la literatura científica para las proteínas KCTD5 y TRPM4, lo que indica que otras proteínas con puntajes altos también podrían potencialmente establecer interacciones físicas. Los filtros implementados en PPI-MASS son una característica de gran utilidad para la búsqueda de estas interacciones, permitiendo utilizar información clave para encontrar proteínas de interés. Por ejemplo, proteínas que participan en el mismo proceso biológico, que se encuentran en la misma localización subcelular o se expresan en los mismos tejidos, podrían presentar una mayor probabilidad de establecer contactos físicos. Además, el filtro de patologías o fármacos asociados permite a los investigadores centrarse en proteínas con aplicaciones médicas o que estén directamente relacionadas con un proceso celular de interés farmacológico.

Por otro lado, la metodología utilizada, la cual se basó en el modelo *agile*, fue un factor importante para el desarrollo del proyecto. Las iteraciones cortas de la metodología permitieron llevar un ritmo de desarrollo constante pero fluido, permitiendo adaptar los requerimientos del cliente a medida que se desarrollaba el proyecto. Un punto clave fue planificar parcialmente las funcionalidades necesarias del software e integrar avances de manera rápida. Esto permitió tener un *feedback* constante respecto a los avances y aseguró el cumplimiento a cabalidad de los requerimientos iniciales. La filosofía de desarrollo *Lean Software Development* (LSD), la cual orientó el desarrollo del software, también tuvo una influencia positiva en el resultado final del proyecto al priorizar sólo las funcionalidades necesarias. Esto permitió utilizar el tiempo de trabajo de manera eficiente, lo que tuvo como resultado final un software completo, simple y eficaz en un tiempo de desarrollo relativamente corto.

PPI-MASS representa una herramienta con mucho potencial, por lo que hay bastantes proyecciones respecto a su futuro. Por ejemplo, se contempla incorporar

nuevas bases de datos biológicas para proveer más información a los investigadores. También se planea incorporar más bases de datos de interacciones, además de BioGRID, para realizar una estimación del puntaje de interacción de SPRINT más robusta. Finalmente, se contempla normalizar el puntaje de SPRINT para permitir al usuario juzgar de mejor manera la diferencia en la probabilidad de interacción entre las diferentes proteínas de entrada. Todo esto se desarrollará en futuras versiones de PPI-MASS.

Bibliografía

- [1] S. Maher, F. P. M. Jjunju, y S. Taylor, «100 years of mass spectrometry: Perspectives and future trends», *Rev. Mod. Phys.*, vol. 87, n.º 1, pp. 113-135, ene. 2015, doi: 10.1103/RevModPhys.87.113.
- [2] C. Szakal, J. Kozole, M. F. Russo, B. J. Garrison, y N. Winograd, «Surface Sensitivity in Cluster-Ion-Induced Sputtering», *Phys. Rev. Lett.*, vol. 96, n.º 21, p. 216104, jun. 2006, doi: 10.1103/PhysRevLett.96.216104.
- [3] A. L. Dill, L. S. Eberlin, D. R. Iba, y R. G. Cooks, «Perspectives in imaging using mass spectrometry», *Chem. Commun.*, vol. 47, n.º 10, pp. 2741-2746, 2011, doi: 10.1039/C0CC03518A.
- [4] J. R. Yates, C. I. Ruse, y A. Nakorchevsky, «Proteomics by Mass Spectrometry: Approaches, Advances, and Applications», *Annu. Rev. Biomed. Eng.*, vol. 11, n.º 1, pp. 49-79, ago. 2009, doi: 10.1146/annurev-bioeng-061008-124934.
- [5] J. Zhang *et al.*, «Top-Down Quantitative Proteomics Identified Phosphorylation of Cardiac Troponin I as a Candidate Biomarker for Chronic Heart Failure», *J. Proteome Res.*, vol. 10, n.º 9, pp. 4054-4065, sep. 2011, doi: 10.1021/pr200258m.
- [6] K. Yugandhar, S. Gupta, y H. Yu, «Inferring Protein-Protein Interaction Networks From Mass Spectrometry-Based Proteomic Approaches: A Mini-Review», *Computational and Structural Biotechnology Journal*, vol. 17, pp. 805-811, 2019, doi: 10.1016/j.csbj.2019.05.007.
- [7] H. Yu *et al.*, «High-Quality Binary Protein Interaction Map of the Yeast Interactome Network», *Science*, vol. 322, n.º 5898, pp. 104-110, oct. 2008, doi: 10.1126/science.1158684.
- [8] L. Bonetta, «Protein-protein interactions: Interactome under construction», *Nature*, vol. 468, n.º 7325, pp. 851-852, dic. 2010, doi: 10.1038/468851a.
- [9] U. Kuzmanov y A. Emili, «Protein-protein interaction networks: probing disease mechanisms using model systems», *Genome Med*, vol. 5, n.º 4, p. 37, 2013, doi: 10.1186/gm441.
- [10] D. Ryan y J. Matthews, «Protein-protein interactions in human disease», *Current Opinion in Structural Biology*, vol. 15, n.º 4, pp. 441-446, ago. 2005, doi: 10.1016/j.sbi.2005.06.001.

- [11] A. C. Joerger y A. R. Fersht, «Structure–function–rescue: the diverse nature of common p53 cancer mutants», *Oncogene*, vol. 26, n.º 15, pp. 2226-2242, abr. 2007, doi: 10.1038/sj.onc.1210291.
- [12] M. S. Forman, J. Q. Trojanowski, y V. M.-Y. Lee, «Neurodegenerative diseases: a decade of discoveries paves the way for therapeutic breakthroughs», *Nat Med*, vol. 10, n.º 10, pp. 1055-1063, oct. 2004, doi: 10.1038/nm1113.
- [13] M. A. Brooke, D. Nitoiu, y D. P. Kelsell, «Cell-cell connectivity: desmosomes and disease», *J. Pathol.*, vol. 226, n.º 2, pp. 158-171, ene. 2012, doi: 10.1002/path.3027.
- [14] Y. Ma-Lauer, J. Lei, R. Hilgenfeld, y A. von Brunn, «Virus–host interactomes — antiviral drug discovery», *Current Opinion in Virology*, vol. 2, n.º 5, pp. 614-621, oct. 2012, doi: 10.1016/j.coviro.2012.09.003.
- [15] I. S. Dementieva *et al.*, «Pentameric Assembly of Potassium Channel Tetramerization Domain-Containing Protein 5», *Journal of Molecular Biology*, vol. 387, n.º 1, pp. 175-191, mar. 2009, doi: 10.1016/j.jmb.2009.01.030.
- [16] Z. Liu, Y. Xiang, y G. Sun, «The KCTD family of proteins: structure, function, disease relevance», *Cell Biosci*, vol. 3, n.º 1, p. 45, 2013, doi: 10.1186/2045-3701-3-45.
- [17] L. Pirone *et al.*, «Thermal and Chemical Stability of Two Homologous POZ/BTB Domains of KCTD Proteins Characterized by a Different Oligomeric Organization», *BioMed Research International*, vol. 2013, pp. 1-8, 2013, doi: 10.1155/2013/162674.
- [18] M. Skoblov *et al.*, «Protein partners of KCTD proteins provide insights about their functional roles in cell differentiation and vertebrate development: Insights & Perspectives», *BioEssays*, vol. 35, n.º 7, pp. 586-596, jul. 2013, doi: 10.1002/bies.201300002.
- [19] Y. Bayón *et al.*, «KCTD5, a putative substrate adaptor for cullin3 ubiquitin ligases: KCTD5, a new substrate-specific adaptor for Cul3», *FEBS Journal*, vol. 275, n.º 15, pp. 3900-3910, ago. 2008, doi: 10.1111/j.1742-4658.2008.06537.x.
- [20] N. Rutz, R. Heilbronn, y S. Weger, «Interactions of cullin3/KCTD5 complexes with both cytoplasmic and nuclear proteins: Evidence for a role in protein stabilization», *Biochemical and Biophysical Research Communications*, vol. 464, n.º 3, pp. 922-928, ago. 2015, doi: 10.1016/j.bbrc.2015.07.069.

- [21] H. He, Y. Peng, S. Fan, Y. Chen, X. Zheng, y C. Li, «Cullin3/KCTD5 induces monoubiquitination of Δ Np63 α and impairs its activity», *FEBS Lett*, vol. 592, n.º 13, pp. 2334-2340, jul. 2018, doi: 10.1002/1873-3468.13104.
- [22] J. Rivas *et al.*, «KCTD5, a novel TRPM4-regulatory protein required for cell migration as a new predictor for breast cancer prognosis», *FASEB j.*, vol. 34, n.º 6, pp. 7847-7865, jun. 2020, doi: 10.1096/fj.201901195RRR.
- [23] G. Narayan *et al.*, «Gene dosage alterations revealed by cDNA microarray analysis in cervical cancer: Identification of candidate amplified and overexpressed genes», *Genes Chromosom. Cancer*, vol. 46, n.º 4, pp. 373-384, abr. 2007, doi: 10.1002/gcc.20418.
- [24] J. Canales, P. Cruz, N. Díaz, D. Riquelme, E. Leiva-Salcedo, y O. Cerda, «K+ Channel Tetramerization Domain 5 (KCTD5) Protein Regulates Cell Migration, Focal Adhesion Dynamics and Spreading through Modulation of Ca²⁺ Signaling and Rac1 Activity», *Cells*, vol. 9, n.º 10, p. 2273, oct. 2020, doi: 10.3390/cells9102273.
- [25] D. E. Scott, A. R. Bayly, C. Abell, y J. Skidmore, «Small molecules, big targets: drug discovery faces the protein–protein interaction challenge», *Nat Rev Drug Discov*, vol. 15, n.º 8, pp. 533-550, ago. 2016, doi: 10.1038/nrd.2016.29.
- [26] S. Kalkhof, S. Schildbach, C. Blumert, F. Horn, M. von Bergen, y D. Labudde, «PIPINO: A Software Package to Facilitate the Identification of Protein-Protein Interactions from Affinity Purification Mass Spectrometry Data», *BioMed Research International*, vol. 2016, pp. 1-13, 2016, doi: 10.1155/2016/2891918.
- [27] M. Götze, C. Iacobucci, C. H. Ihling, y A. Sinz, «A Simple Cross-Linking/Mass Spectrometry Workflow for Studying System-wide Protein Interactions», *Anal. Chem.*, vol. 91, n.º 15, pp. 10236-10244, ago. 2019, doi: 10.1021/acs.analchem.9b02372.
- [28] G. Postic *et al.*, «Probing Protein Interaction Networks by Combining MS-Based Proteomics and Structural Data Integration», *J. Proteome Res.*, vol. 19, n.º 7, pp. 2807-2820, jul. 2020, doi: 10.1021/acs.jproteome.0c00066.
- [29] P. Bourque, R. E. Fairley, y IEEE Computer Society, *Guide to the software engineering body of knowledge*. 2014.
- [30] I. H. Sarker, F. Faruque, U. Hossen, y A. Rahman, «A Survey of Software Development Process Models in Software Engineering», *IJSEIA*, vol. 9, n.º 11, pp. 55-70, nov. 2015, doi: 10.14257/ijseia.2015.9.11.05.

- [31] Mozilla Developer Network, «The web and web standards», *MDN Web Docs*. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards (accedido nov. 07, 2020).
- [32] D. A. Benson *et al.*, «GenBank», *Nucleic Acids Research*, vol. 41, n.º D1, pp. D36-D42, nov. 2012, doi: 10.1093/nar/gks1195.
- [33] The UniProt Consortium, «UniProt: a worldwide hub of protein knowledge», *Nucleic Acids Research*, vol. 47, n.º D1, pp. D506-D515, ene. 2019, doi: 10.1093/nar/gky1049.
- [34] D. S. Wishart, «DrugBank: a comprehensive resource for in silico drug discovery and exploration», *Nucleic Acids Research*, vol. 34, n.º 90001, pp. D668-D672, ene. 2006, doi: 10.1093/nar/gkj067.
- [35] O. Ursu *et al.*, «DrugCentral: online drug compendium», *Nucleic Acids Res*, vol. 45, n.º D1, pp. D932-D939, ene. 2017, doi: 10.1093/nar/gkw993.
- [36] H. M. Berman, «The Protein Data Bank», *Nucleic Acids Research*, vol. 28, n.º 1, pp. 235-242, ene. 2000, doi: 10.1093/nar/28.1.235.
- [37] A. Waterhouse *et al.*, «SWISS-MODEL: homology modelling of protein structures and complexes», *Nucleic Acids Research*, vol. 46, n.º W1, pp. W296-W303, jul. 2018, doi: 10.1093/nar/gky427.
- [38] J. Piñero *et al.*, «The DisGeNET knowledge platform for disease genomics: 2019 update», *Nucleic Acids Research*, p. gkz1021, nov. 2019, doi: 10.1093/nar/gkz1021.
- [39] M. Uhlen *et al.*, «A pathology atlas of the human cancer transcriptome», *Science*, vol. 357, n.º 6352, p. eaan2507, ago. 2017, doi: 10.1126/science.aan2507.
- [40] Y. Li y L. Ilie, «SPRINT: ultrafast protein-protein interaction prediction of the entire human interactome», *BMC Bioinformatics*, vol. 18, n.º 1, p. 485, dic. 2017, doi: 10.1186/s12859-017-1871-x.
- [41] M. Poppendieck y M. A. Cusumano, «Lean Software Development: A Tutorial», *IEEE Softw.*, vol. 29, n.º 5, pp. 26-32, sep. 2012, doi: 10.1109/MS.2012.107.
- [42] L. Torvalds, *Git*. 2005. [En línea]. Disponible en: <https://git-scm.com/>
- [43] GitLab Inc., *GitLab*. GitLab Inc., 2014. [En línea]. Disponible en: <https://about.gitlab.com/>

- [44] Django Software Foundation, *Django*. Django Software Foundation, 2020. [En línea]. Disponible en: <https://djangoproject.com>
- [45] G. van Rossum, F. L. Drake, y G. Van Rossum, *Python 3 Reference Manual*, Release 3.0.1 [Repr.]. Hampton, NH: Python Software Foundation, 2010.
- [46] Django Software Foundation, «Django is a “MTV” framework», *Django FAQ: General*. <https://docs.djangoproject.com/en/3.1/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>
- [47] L. Etzkorn, *Introduction to middleware: web services, object components, and cloud computing*. 2017.
- [48] OWASP, «Top 10 Web Application Security Risks», *OWASP Top Ten*. <https://owasp.org/www-project-top-ten/>
- [49] J. Chaffer y K. Swedberg, *jQuery reference guide: a comprehensive exploration of the popular JavaScript library*. Birmingham: Packt Publ, 2007.
- [50] K. Douglas y S. Douglas, *PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases*, 1st ed. Indianapolis, Ind: Sams, 2003.
- [51] C. Stark, «BioGRID: a general repository for interaction datasets», *Nucleic Acids Research*, vol. 34, n.º 90001, pp. D535-D539, ene. 2006, doi: 10.1093/nar/gkj109.
- [52] N. Balasco *et al.*, «Molecular recognition of Cullin3 by KCTDs: Insights from experimental and computational investigations», *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, vol. 1844, n.º 7, pp. 1289-1298, jul. 2014, doi: 10.1016/j.bbapap.2014.04.006.
- [53] H. J. Cho *et al.*, «Cullin 3/KCTD5 Promotes the Ubiquitination of Rho Guanine Nucleotide Dissociation Inhibitor 1 and Regulates Its Stability», *J. Microbiol. Biotechnol.*, vol. 30, n.º 10, pp. 1488-1494, oct. 2020, doi: 10.4014/jmb.2007.07033.
- [54] C.-H. Cho *et al.*, «Depletion of 14-3-3 γ reduces the surface expression of Transient Receptor Potential Melastatin 4b (TRPM4b) Channels and attenuates TRPM4b-mediated glutamate-induced neuronal cell death», *Mol Brain*, vol. 7, n.º 1, p. 52, dic. 2014, doi: 10.1186/s13041-014-0052-3.
- [55] M. González-Avenidaño *et al.*, «PPI-MASS: An Interactive Web Server to Identify Protein-Protein Interactions From Mass Spectrometry-Based

Proteomics Data», *Front. Mol. Biosci.*, vol. 8, p. 701477, jul. 2021, doi:
10.3389/fmolb.2021.701477.